ISSUE 1 | VOL. 1 | MARCH 2024

EXPLORE

PYTHON, TOÁN VÀ ỨNG DỤNG KINH TẾ

10 DAYS
WHAT DREAMS
ARE MADE OF ?



MATHEMATICS
AND
PROGRAMMING

WRITER: DR HAPPY
AI ENGINEER: NGUYÊN VŨ

nghеai.edu.vn

GỬI TẶNG BẠN

“ ”

Be happy

TRI ÂN



Các bạn mến

Đầu tiên admin gửi lời cảm ơn đến những người thân Cha Mẹ, Anh Xã và con trai Minh Đức đã luôn ủng hộ khoản thời gian qua về mặt tinh thần và cả hỗ trợ công việc hàng ngày để mình có thể tập trung hoàn thành cuốn sách này.

Cảm ơn em Vũ Hoàng Nguyên, luôn bên cạnh ủng hộ chương trình, em hỗ trợ rất nhiều về mặt design code và proof reading để có được cuốn sách hoàn chỉnh, nếu không có em chắc sản phẩm sẽ không hoàn thiện tốt đẹp.

Có lẽ nguồn động lực nhiều nhất để admin có thể giữ vững đó là những người bạn đã ủng hộ hết mình, với bề dày lượng kiến thức các bạn đã có thì nội dung kiến thức sách này khá nhẹ, không đóng vai trò gì quan trọng để đưa các bạn lên mức cao hơn, nhưng các bạn đã ủng hộ chương trình, admin biết vì chỉ muốn ủng hộ admin và cùng đồng hành để admin có thêm nhiều động lực hơn mà thôi.

Cảm ơn những người bạn mới quen cùng đồng hành 10 days yêu thương, các bạn cũng là nguồn động lực để mình viết chỉnh chu hơn, vì chỉ mong rằng sự chỉnh chu của mình một phần nào đó tôn trọng và cảm ơn vì mình đã có cơ hội được gặp các bạn, được đồng hành và cùng nhau làm những việc tử tế và rèn hơn cho chính mình. Cảm ơn các bạn đã chia sẻ những công việc các bạn làm, mỗi lần đọc mình cảm thấy mình yêu đời và thấy cuộc sống quanh chúng ta thật hạnh phúc.

Mình cũng không thể quên ơn trên, đấng tạo hóa, người cũng đã soi sáng cho mình cần biết những gì mình cần làm.

Một lần nữa xin chân thành cảm ơn

Dr happy

Ninh Thuận,
24/03/2024

What dreams are made of ?

MATHEMATICS AND PROGRAMMING

March 2024



Mến chào các bạn, mình là Aiko. Có lẽ một số bạn đã biết mình qua những câu chuyện và bài viết trên page Toán và Lập Trình mà mình thường xuyên chia sẻ. Hôm nay, mình rất hân hạnh được bắt đầu hành trình đồng hành cùng các bạn trong khoảng thời gian 10 ngày tới. Đây sẽ là một hành trình đầy thú vị và ý nghĩa, nơi chúng ta cùng nhau khám phá và đắm chìm trong thế giới kỳ diệu của Toán học và Trí tuệ Nhân tạo (AI).

Mục tiêu của chúng ta trong 10 ngày này không chỉ là học tập mà còn là để cảm nhận và yêu mến hơn những khía cạnh tuyệt vời của Toán học và AI. Toán học không chỉ là những con số khô khan, mà còn là ngôn ngữ của vũ trụ, giúp chúng ta hiểu và giải quyết vô số vấn đề trong cuộc sống. Còn AI, với sự kết hợp hoàn hảo giữa Toán học và công nghệ, đang mở ra những chân trời mới, giúp định hình tương lai của nhân loại.

Nhưng điều làm hành trình này thực sự đặc biệt, là cách chúng ta áp dụng những kiến thức đó vào lĩnh vực kinh tế. Trong thế giới ngày nay, việc hiểu biết và sử dụng Toán học và AI trong kinh tế không chỉ giúp tối ưu hóa quyết định kinh doanh, mà còn mở ra những cơ hội mới cho sự phát triển bền vững và công bằng. Từ việc phân tích xu hướng thị trường, dự báo tình hình tài chính, đến việc tối ưu hóa các chiến lược đầu tư và quản lý rủi ro, Toán học và AI trở thành công cụ mạnh mẽ, giúp các nhà kinh tế học và doanh nghiệp đạt được hiệu quả cao nhất trong quyết định của mình.

Trong 10 ngày này, chúng ta sẽ cùng nhau tìm hiểu về những ứng dụng thú vị này, và thậm chí, thử sức với việc giải quyết một số bài toán kinh tế thực tế thông qua Toán học và AI. Điều này không chỉ giúp chúng ta hiểu rõ hơn về giá trị thực sự mà hai lĩnh vực này mang lại, mà còn cảm nhận được niềm vui khi áp dụng kiến thức vào thực tiễn, giải quyết những vấn đề cụ thể mà xã hội đang đối mặt.

TOÁN VÀ LẬP TRÌNH

love love love

Vậy bạn có sẵn lòng đồng hành cùng mình trong 10 ngày này không? Mình tin rằng, với sự cởi mở, tò mò và quyết tâm, chúng ta sẽ cùng nhau tạo nên một hành trình đáng nhớ, một hành trình mà sau này khi nhìn lại, chúng ta sẽ thấy tự hào về bản thân mình vì đã dám thử sức, dám khám phá và dám yêu thích những điều mới mẻ.Hãy cùng mình bước vào thế giới kỳ diệu của Toán học và AI, khám phá những bí ẩn, giải quyết những thách thức và tận hưởng niềm vui của việc học hỏi và sáng tạo. Mình chắc chắn rằng, sau 10 ngày này, chúng ta sẽ có những kỷ niệm đẹp và kiến thức quý giá để mang theo trên con đường phía trước.

Thân ái, Aiko (Dr Happy)

TABLE OF CONTENTS

1 Day 1: Python - ROI

Page : 07

2 Day 2: Biến - Kiểu dữ liệu

Page: 16

3 Day 3: Quyết định thông minh

Page: 32

4 Day 4: Numpy và phân tích kinh tế

Page: 50

5 Day 5: Pandas - Khám phá dữ liệu

Page: 72

TABLE OF CONTENTS

6 Day 6: Bí mật của xác suất
Page: 98

7 Day 7: Phân phối và rủi ro
Page : 107

8 Day 8: Số liệu nói lên điều gì?
Page: 119

9 Day 9: Nghệ thuật trực quan hóa
Page: 141

10 Day 10: Dự án và tình huống
Page: 152

Day 1: Python - ROI

TOÁN VÀ LẬP TRÌNH

Phần 1: Python và cách cài đặt

- "Hello World"?

Trong không gian yên tĩnh của thư viện, nơi những tia nắng ban mai len lỏi qua những kẽ lá, mình lại tìm thấy mình đang lạc bước giữa những hàng sách cao ngất. Đây là nơi chúng mình, những người bạn mọt sách, thường dành hàng giờ để lắng nghe những câu chuyện từ khắp mọi nơi, từ những cuốn sách phiêu lưu hấp dẫn đến những tập truyện kể về lịch sử hào hùng của toán học.
Nhưng ngày hôm nay, không giống như bình thường, một cuốn sách mới đã lọt vào tầm mắt mình, với tựa đề "AI và Tương Lai". Sự tò mò bỗng chốc dâng trào khi lật từng trang sách, tiếp thu những kiến thức mới mẻ về thế giới kỳ diệu của Trí tuệ Nhân tạo. Trong suốt quá trình đọc, mình không ngừng ghi chép lại những ý tưởng và suy nghĩ, tự hỏi liệu có một ngôn ngữ lập trình nào vừa mạnh, hiện đại, có tính ứng dụng cao mà lại dễ học?

Mình mở máy tính và bắt đầu tìm kiếm trên các trang web uy tín, hy vọng tìm thấy câu trả lời cho những thắc mắc của mình. Và rồi, như một dấu hiệu của số phận, bảng xếp hạng các ngôn ngữ lập trình được sử dụng nhiều nhất hiện ra trước mắt. Vị trí đầu tiên, không ngoài dự đoán, là Python - ngôn ngữ lập trình được yêu thích bởi sự đơn giản nhưng mang lại nhiều hỗ trợ như thế.

Trái tim mình bùng cháy với niềm đam mê mới, quyết định bắt đầu hành trình học Python. Và không còn gì tốt hơn là bắt đầu với "Hello World" - câu chào mà bất kỳ ai bước vào thế giới lập trình nào cũng phải nói. Mình tự hỏi, tại sao lại chính là "Hello World"? Câu trả lời dường như rất đơn giản, nhưng cũng vô cùng sâu sắc: nó là biểu tượng của sự khởi đầu, một lời chào mừng đến với một thế giới mới mà tôi sắp khám phá.

Trong mê cung của thông tin và công cụ lập trình, làm sao để mình tìm thấy chìa khóa mở cánh cửa đầu tiên, và phần mềm nào sẽ là người hướng dẫn đáng tin cậy trên con đường in "Hello World" lên màn hình, khắc dấu bước chân đầu tiên của mình vào thế giới lập trình? Với tâm hồn của một nhà thám hiểm, mình lướt qua biển cả thông tin, thu thập từng mảnh ghép quý giá.

Trong cuộc hành trình tìm kiếm công cụ lý tưởng để bước chân vào thế giới Python, mình đã lướt qua một loạt các ngọn hải đăng dẫn lối: IDLE với vẻ ngoài giản dị nhưng ấm áp, nơi bắt đầu hoàn hảo cho những người mới mở cánh cửa vào lập trình; PyCharm, chiếc cầu nối giữa sự dễ dàng và chuyên nghiệp, nơi ẩn chứa bí mật của những ưu điểm vượt trội; và Visual Studio Code, bản giao hưởng của sự linh hoạt, nơi mỗi tiện ích mở rộng như một nốt nhạc riêng biệt, cùng hòa quyện vào bản hòa ca đa ngôn ngữ, trong đó có Python.

Be happy

Tuy nhiên, dưới bầu trời rộng lớn của lựa chọn, mình tìm thấy một ngôi sao sáng giữa đêm đen - Google Colab. Đây không chỉ là một phần mềm, mà còn là một cánh cổng mở ra không gian học tập mà không giới hạn, nơi không cần phải lo lắng về những rào cản kỹ thuật hay sự phức tạp trong việc cài đặt. Google Colab, với sự đơn giản và trực quan của mình, dường như đã vẽ nên một bức tranh mới, nơi mọi dòng code Python không chỉ là những câu chuyện được kể, mà còn là những giấc mơ được viết lên trên những đám mây. Ở đây, việc viết, chạy code và lưu trữ tập tin trở nên dễ dàng và thuận tiện như chưa từng thấy, mở ra một không gian học tập mà ở đó, mỗi người mới đều có thể tìm thấy tiếng nói của mình trong thế giới lập trình.

Trong dải ngân hà rực rỡ của công cụ lập trình, tôi chọn Google Colab làm ngôi sao dẫn đường trên hành trình khám phá Python của mình. Như một nhà thám hiểm trong rừng sâu của kiến thức, tôi đã ghi lại những bí kíp vàng để chinh phục những thách thức đầu tiên trên Google Colab, bước đầu tiên của mình vào thế giới mã hóa diệu kỳ: Bạn hãy cùng mình làm theo các bước như sau nhé:

🌟 Bước 1: Làm Quen và Đăng Nhập
Mở trình duyệt yêu thích của bạn, gõ "Google Colab" và ấn Enter. Đừng quên đăng nhập bằng tài khoản Google mà bạn yêu thích nhé!

📓 Bước 2: Tạo Một Cuốn Sổ Mới
Khi đã vào được Google Colab, tìm đến "File" trên menu, chọn "New notebook". Ta-da! Một cuốn sổ mới mở ra trước mắt, sẵn sàng để bạn chinh phục!

✏️ Bước 3: Thảo Luận Cùng Python
Trong cuốn sổ mới của bạn, bạn sẽ thấy một không gian trắng toát đợi bạn khai phá. Hãy gõ: print("Hello, World!")

Và thế là bạn đã bắt đầu cuộc trò chuyện đầu tiên với Python rồi đấy!

🏃‍♂️ Bước 4: Đưa Code Lên Mây
Để code bay lên mây, chỉ cần nhấn vào nút "Run" hình tam giác nhỏ, hoặc phím tắt "Shift + Enter". Và chờ xem điều kỳ diệu xảy ra!

🔍 Bước 5: Khám Phá và Chinh Phục
Google Colab không chỉ giúp bạn viết code mà còn là nơi bạn có thể mời các thư viện nổi tiếng như NumPy, Pandas để chơi cùng. Và đừng quên bạn còn có thể lưu trữ và chia sẻ dữ liệu với Google Drive nữa đấy!

💾 Bước 6: Lưu Giữ Kỷ Niệm
Cuốn sổ của bạn sẽ tự động được lưu trong Google Drive. Bạn cũng có thể lưu lại mọi lúc bằng cách chọn "File" > "Save", hoặc mở lòng mình chia sẻ cuốn sổ này với thế giới qua "File" > "Share".

Chinh phục được 6 bước này, bạn đã nhẹ nhàng bước chân vào thế giới Python rồi đấy! Bây giờ, thay vì dừng lại ở "Hello World", sao chúng ta không tạo ra những dấu ấn cá nhân hơn nhỉ? Bắt đầu từ mình nào! Hãy gửi điều gì đó thật ý nghĩa, như một lời tri ân chân thành tới những trái tim luôn sẵn lòng giúp đỡ xung quanh:

print("Cảm ơn tấm lòng tử tế của bạn dành cho mọi người! Bạn thật sự làm cho thế giới này tốt đẹp hơn.")

Be happy

ERROR



Chúng ta vừa lướt qua những bước như chơi một bản nhạc mà không hề hấn gì, thật là một hành trình suôn sẻ và. Nhưng đường lập trình, dù chỉ là việc in ra dòng chữ "Hello, World!" cũng có thể gặp phải những "ổ gà" nhỏ. Dưới đây là một số lỗi mà bạn có thể va phải, để bạn biết và chuẩn bị sẵn sàng!

>>> 1. LỖI CÚ PHÁP (SYNTAX ERROR)

Thiếu Dấu Ngoặc Đơn: Nếu bạn quên một trong hai dấu ngoặc đơn (print("Hello, World! hoặc print"Hello, World!")), Python sẽ không hiểu và báo lỗi cú pháp.

Dùng Sai Dấu Ngoặc: Việc sử dụng ngoặc kép thay vì ngoặc đơn hoặc ngược lại (print('Hello, World!") hoặc print("Hello, World!')) cũng sẽ gây ra lỗi.

2. LỖI ĐÁNH MÁY (TYPO ERROR) <<<

Sai Chính Tả trong print: Gõ nhầm prnt, pritn thay vì print cũng sẽ làm cho chương trình của bạn bị lỗi.





>>> 3. LỖI KÝ TỰ ĐẶC BIỆT (CHARACTER ERROR)

Ký Tự Đặc Biệt Trong Chuỗi: Nếu trong chuỗi của bạn có chứa ký tự như dấu ngoặc kép (") và bạn không sử dụng escape character (\"), điều này có thể gây lỗi.

Dr happy

PHẦN 3: ROI



Chỉ việc "nói chuyện" với máy tính qua những dòng code thôi thì chưa đủ để khám phá hết quỹ đạo rộng lớn của lập trình đâu! Ngoài tình yêu với thế giới code mênh mông, mình còn đắm chìm trong những công thức và ứng dụng toán học, đặc biệt là khi nó vũ điệu cùng kinh tế. Có một bí mật nhỏ mình muốn chia sẻ cùng bạn: mình cũng mới chỉ bắt đầu lắng nghe nhịp điệu của kinh tế thôi. Vậy nên, chúng mình sẽ cùng nhau bắt đầu từ những bước đi nhẹ nhàng, khám phá những công thức cơ bản và cách ứng dụng chúng trong lĩnh vực kinh tế. Và một trong số đó, là hiểu và áp dụng ROI - Return on Investment.

Nhưng mà, chợt nhận ra, ROI là cái gì nhỉ? Sao mình không cùng nhau làm một chuyến phiêu lưu nhỏ để tìm hiểu về ROI trước khi mình cùng nhau trải nghiệm và tạo ra điều gì đó thú vị với những công cụ mà mình đang có nhỉ? Hãy cùng nhau bắt đầu thôi nào! ROI, hay Return on Investment, không chỉ là một công thức toán học khô khan; nó còn là ngọn hải đăng cho những người dấn thân vào biển cả mênh mông của kinh doanh và đầu tư. Với ROI, chúng ta có thể đo lường sự thành công của một khoản đầu tư, không chỉ qua con số lợi nhuận thu về mà còn qua cái nhìn sâu sắc về giá trị thực sự mà khoản đầu tư đó mang lại so với nguồn lực đã bỏ ra.

Hãy để mình đưa bạn vào một chuyến phiêu lưu ngắn qua thế giới của ROI bằng một câu chuyện mà từ đó, bạn sẽ thấy được ý nghĩa và giá trị thực sự của nó:

Câu Chuyện về Minh và Giấc Mơ Cà Phê.

Minh, một chàng trai trẻ đầy nhiệt huyết và đam mê với cà phê, mơ ước mở một quán cà phê nhỏ trong khu phố. Với số vốn tiết kiệm được, 20 triệu đồng, Minh quyết tâm biến giấc mơ thành hiện thực. Anh bắt đầu từ việc chọn lựa không gian, trang trí quán và nhập những loại cà phê chất lượng.

Sau một năm miệt mài, quán cà phê của Minh không chỉ là nơi tụ họp của những tâm hồn yêu cà phê mà còn trở thành ngôi nhà thứ hai của nhiều người. Khi tổng kết lại, Minh phát hiện quán cà phê của mình đã thu về 30 triệu đồng sau khi trừ đi mọi chi phí.

Đứng trước kết quả này, Minh nhớ lại công thức ROI mà một người bạn đã chia sẻ:

$$\text{ROI} = \frac{\text{Lợi nhuận thu được - Chi phí đầu tư}}{\text{Chi phí đầu tư}} \times 100\%$$

Minh bắt tay vào tính toán: từ 30 triệu thu về trừ đi 20 triệu vốn ban đầu, lợi nhuận thu được là 10 triệu. Chia số này cho 20 triệu vốn đầu tư và nhân với 100, Minh có ROI là 50%.

Kết quả này không chỉ là minh chứng cho sự thành công vượt bậc của quán cà phê mà còn là nguồn cảm hứng và động lực cho Minh và những ai đang ấp ủ giấc mơ của mình. ROI không chỉ là con số, mà còn là câu chuyện về sự kiên trì, đam mê và khao khát chinh phục.

Trong mỗi khoản đầu tư, dù lớn hay nhỏ, ROI giúp chúng ta hiểu rằng mỗi quyết định, mỗi nỗ lực đều có giá trị của nó. Đó là lý do tại sao việc áp dụng và hiểu sâu về ROI không chỉ là kiến thức cần thiết trong kinh doanh mà còn là bài học quý báu trong cuộc sống, khích lệ chúng ta không ngừng mơ ước, dấn thân và vươn xa.

Minh Coffee Shop
123 Anywhere St., Any City, ST 12345







PHẦN 4



DÙNG PYTHON CÀI ĐẶT ROI

Bây giờ, hãy bắt tay vào công việc thú vị của ngày hôm nay: sáng tạo ra một cây cầu nối giữa thế giới lập trình và khái niệm ROI. Để bắt đầu hành trình này, chúng ta cần trang bị những công cụ gì? Dưới đây là danh sách những bước chuẩn bị cần thiết:

🌈 Bước 1: Lên Đường Tới Google Colab

Mở trình duyệt yêu thích của bạn và ghé thăm Google Colab. Đăng nhập bằng tài khoản Google của bạn để bắt đầu cuộc phiêu lưu.

📚 Bước 2: Tạo Cuốn Sổ Mới

Trong không gian Google Colab, hãy tìm nút + New Notebook để mở một cuốn sổ tập mới. Đây sẽ là nơi bạn viết nên câu chuyện của mình với Python.

✨ Bước 3: Bắt Đầu Với Python

Trong cuốn sổ mới này, bạn sẽ thấy một ô code chờ bạn chinh phục. Hãy bắt đầu với những dòng code đơn giản như print(“Hello, Python World!“) để mở màn.

🚀 Bước 4: Khiến Code Sống Động

Sử dụng nút Run hoặc phím tắt Shift + Enter để thực thi code của bạn. Ngắm nhìn kết quả hiện ra, cảm nhận niềm vui từ những bước đi đầu tiên.

📈 Bước 5: Định Nghĩa ROI

Đến lúc để định nghĩa công thức ROI trong một hàm Python. Hãy nhập code sau vào một ô mới: Bạn copy code sau

```
def calculate_ROI(gain, cost):
    roi = (gain - cost) / cost * 100
    return roi
```

🧪 Bước 6: Thử Nghiệm với Dữ Liệu

Hãy thử hàm của bạn bằng cách nhập một ví dụ thực tế, như việc đầu tư vào cửa hàng nhỏ trong câu chuyện của chúng ta. Gõ code sau và nhấn Run:

```
print(calculate_ROI(12000000, 10000000))
```

💾 Bước 7: Lưu và Chia Sẻ Thành Quả

Google Colab tự động lưu cuốn sổ của bạn vào Google Drive. Bạn cũng có thể lưu một bản sao hoặc chia sẻ cuốn sổ với bạn bè hoặc đồng nghiệp để cùng nhau học hỏi và phát triển.

Be happy



Bạn đã thử nghiệm với những bước hướng dẫn trên và đạt được kết quả chưa?

print(calculate_ROI(12000000, 10000000))?

def?

Bạn đã thử nghiệm với những bước hướng dẫn trên và đạt được kết quả chưa? Cảm giác của bạn thế nào, có gặp chút thách thức nào không? Có lẽ trong quá trình lập trình, bạn có thể hơi băn khoăn về bước tạo hàm với def hay không biết làm sao để sử dụng hàm đó qua câu lệnh print(calculate_ROI(12000000, 10000000))?

Đừng lo lắng, hãy giữ bình tĩnh, mình sẽ giải cùng bạn!

Hãy tưởng tượng def trong Python giống như việc bạn đang tạo ra một công thức nấu ăn. Công thức này sẽ hướng dẫn bạn làm ra một món ăn cụ thể từ những nguyên liệu bạn có. Trong lập trình, def giúp bạn định nghĩa một "công thức" để giải quyết một vấn đề cụ thể, mà chúng ta gọi là hàm.

Ví dụ Thực Tế:

Giả sử bạn muốn tính tổng tiền tiết kiệm trong hai tuần. Bạn có thể tạo một hàm để làm điều đó:

```
def tinh_tong_tiet_kiem(tuan_1, tuan_2):
    return tuan_1 + tuan_2
```

Ở đây:

def bắt đầu công thức của bạn.
tinh_tong_tiet_kiem là tên bạn đặt cho công thức này, giống như bạn đặt tên cho một món ăn.
(tuan_1, tuan_2) chính là nguyên liệu cần thiết cho công thức, ở đây là số tiền bạn tiết kiệm được mỗi tuần.

return tuan_1 + tuan_2 cho biết công thức sẽ làm gì với nguyên liệu: cộng chúng lại. Để áp dụng công thức này vào thực tế, bạn chỉ cần gọi nó và cho biết bạn đã tiết kiệm được bao nhiêu mỗi tuần:

```
so_tien_tiet_kiem = tinh_tong_tiet_kiem(500000, 600000)
print(so_tien_tiet_kiem) # In ra: 1100000
```

Khi đó, tinh_tong_tiet_kiem(500000, 600000) chính là việc bạn áp dụng công thức nấu ăn với 500.000 đồng bạn tiết kiệm được tuần đầu và 600.000 đồng tuần thứ hai. Hàm sẽ "nấu" và "trả về" cho bạn tổng số tiền là 1.100.000 đồng.

Như vậy, def giúp bạn tạo ra những "công thức" linh hoạt, giúp giải quyết các vấn đề cụ thể một cách gọn gàng và hiệu quả.

Day 1: Note



🌟 Cuốn Sổ Tay Phiêu Lưu Cùng Aiko

Ahoy! Bạn đã sẵn sàng chưa? Hãy cùng mình, Aiko, điểm qua cuộc hành trình hôm nay qua những ghi chú thú vị trong cuốn sổ tay của bạn. 📖✨

🚀 Khởi Hành!
Bước 1: Ta đã mở ra cánh cửa thế giới Python, tạo nên một chương trình đầu tiên và thả mình vào dòng chảy của mã lập trình.

🐞 Những Ngã Rẽ và Cạm Bẫy:
Bước 2: Trên đường đi, ta gặp phải vài con bọ nhỏ (lỗi lập trình ý mà), nhưng mình đã nhớ cách nhìn ra chúng và sửa chữa khi dùng hàm print.

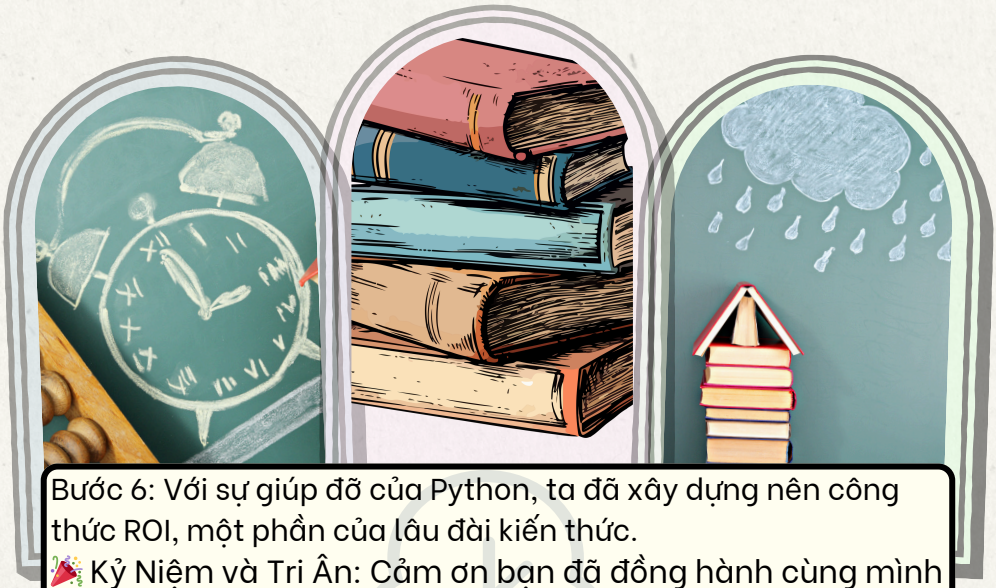
💡 Bí Mật Của Kho Báu:
Bước 3: Ta đã khám phá ra ý nghĩa thực sự của báu vật ROI, công thức ma thuật cho những khoản đầu tư.

📚 Bài Học Từ Thuật Sĩ:
Bước 4: Ta đã được thuật sĩ giải thích về def - cách tạo ra những phép thuật (hàm) trong Python.

🌈 Cách Dẫn Dắt Bằng Phép Thuật:
Bước 5: Ta đã học cách gọi lại những phép thuật (hàm) mà mình hoặc người khác đã tạo, sử dụng chúng để làm nên điều kỳ diệu.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH



Bước 6: Với sự giúp đỡ của Python, ta đã xây dựng nên công thức ROI, một phần của lâu đài kiến thức.

🎉 Kỷ Niệm và Tri Ân: Cảm ơn bạn đã đồng hành cùng mình trong chuyến phiêu lưu này! Mình hy vọng bạn đã tìm thấy niềm vui và sự hào hứng trong mỗi bước chân, mỗi dòng code.

Hẹn Gặp Lại: Đừng quên, mỗi ngày là một phiêu lưu mới! Mình sẽ chờ bạn ở chương tiếp theo của cuộc hành trình lập trình, đầy ắp những bí mật và niềm vui mới. Chúc bạn một ngày tràn đầy năng lượng, hạnh phúc và thành công!





TODAY

Biến

Kiểu dữ liệu

Dự báo lợi nhuận đầu tư

VÀI DÒNG LƯU Ý

BY AIKO

Xin nhấn mạnh: Những mô tả và câu chuyện về đầu tư trong văn bản này thuần túy là sản phẩm của trí tưởng tượng, dựng lên nhằm mục đích làm cho những khái niệm đầu tư trở nên thú vị và gần gũi hơn với đời sống hàng ngày. Với kiến thức hiện có về biến và kiểu dữ liệu, Aiko mong muốn chia sẻ những nội dung phù hợp và gần gũi nhất với mọi người. Tuy nhiên, để phát triển một ứng dụng chuyên nghiệp, cần có sự am hiểu sâu rộng và kỹ lưỡng hơn nhiều về lập trình, cũng như việc áp dụng những nguyên tắc logic và ứng dụng thực tế trong môi trường chuyên nghiệp.

BẠN ĐÃ SẴN SÀNG CHO NGÀY HÔM NAY?

Hãy cùng nhau lên chuyến tàu khám phá thế giới lập trình với Python, nơi chúng ta sẽ mở cửa vào vùng đất kỳ diệu của biến và kiểu dữ liệu. Dưới bầu trời rộng lớn của kiến thức, mỗi biến giống như một ngôi sao lấp lánh, mỗi kiểu dữ liệu là một hành tinh độc đáo. Hôm nay, chúng ta bắt đầu cuộc hành trình từ căn phòng thí nghiệm ma thuật, nơi mỗi bình thuỷ tinh chứa đựng không chỉ những giọt nước mắt tiên cá mà còn là ánh sáng từ những ngôi sao xa xôi.



Be happy

PHẦN 1: MA THUẬT CỦA BIẾN VÀ KIỂU DỮ LIỆU

BY AIKO

1: PYTHON (MA THUẬT CỦA BIẾN VÀ KIỂU DỮ LIỆU)

Trong thế giới Python, mỗi biến là một chiếc lọ kỳ diệu. Hãy tưởng tượng bạn có một chiếc lọ tên là giac_mo, bên trong đựng 100 giấc mơ to lớn, và chiếc lọ khác tên là anh_sang_sao, ánh sáng từ ngôi sao xa xôi chiếu rọi. Khi bạn gọi tên chúng, ma thuật hiện hình, giống như cách chúng ta khám phá và sử dụng dữ liệu trong lập trình.

2: VƯƠNG QUỐC KINH DOANH BẤT ĐỘNG SẢN

Cuộc phiêu lưu tiếp theo đưa chúng ta tới lãnh địa của Kinh Doanh Bất Động Sản, nơi biến và kiểu dữ liệu không chỉ là những khái niệm mà là những công cụ đắc lực. Hãy tưởng tượng bạn đang ngồi tại bàn làm việc, trước mắt là bản đồ về những dự án bất động sản và cơ hội chứng khoán. Mỗi quyết định đầu tư, từ dauTu_BatDongSan đến dauTu_ChungKhoan, đều là một bước đi trên con đường dẫn tới thành công và thịnh vượng.

3: MỞ RỘNG TẦM NHÌN QUA CÁC KỊCH BẢN ĐẦU TƯ

Aiko tiếp tục cuộc hành trình, khám phá một loạt kịch bản đầu tư thông qua Python:

1. Vườn Ươm: Nơi đầu tư ổn định giống như việc chăm sóc một vườn cây, đợi chờ hạt giống trưởng thành thành cây lớn.
2. Chinh Phục Mặt Trăng: Đầu tư mạo hiểm vào công nghệ mới hoặc thị trường mới nổi, giống như chuyến du hành tới mặt trăng đầy rủi ro nhưng tiềm năng tăng trưởng lớn.
3. Bức Tranh: Đa dạng hóa đầu tư như hoàn thiện một bức tranh từ nhiều mảnh ghép, cân đối giữa rủi ro và cơ hội.
4. Phượng Hoàng: Đầu tư vào thị trường sau sự sụt giảm lớn, tin tưởng vào khả năng hồi phục và thăng hoa trở lại.
5. Suối Nguồn: Tạo ra dòng thu nhập thụ động từ bất động sản cho thuê hay cổ tức, như suối nước nhỏ không ngừng chảy.

4: HÀNH TRÌNH KIẾN THỨC VỀ ĐẦU TƯ

Chúng ta sẽ cùng nhau du ngoạn qua một loạt các kịch bản đầu tư đa dạng, khám phá và vượt qua từng thách thức với những bài học và câu chuyện ly kỳ. Hãy chuẩn bị tinh thần để cùng nhau chinh phục mỗi trang kiến thức, gặt hái những bài học quý giá trên hành trình này.

Do happy

I.PYTHON (MA THUẬT CỦA BIẾN VÀ KIỂU DỮ LIỆU)



BIẾN, GIẤC MƠ, ÁNH SAO SÁNG,BÍ MẬT, NGÔI SAO, LỌ THỦY TINH

Khi bình minh ban mai từ từ nhô lên, một ngày mới lại đến, bạn hào hứng đón chờ điều gì? Cùng nhau tiếp tục cuộc phiêu lưu mào hiểm vào thế giới lập trình, hôm nay chúng ta sẽ khám phá vùng đất mới của biến và kiểu dữ liệu. Hãy cùng nhau dạo một vòng qua câu chuyện thực tế, nơi biến và kiểu dữ liệu không còn xa lạ mà thân thiện như những người bạn hàng ngày.

Hãy tưởng tượng bạn đang bước vào phòng thí nghiệm đầy ma thuật của một phù thủy, nơi chứa đầy những bình thuỷ tinh lấp lánh, mỗi cái chứa đựng một phép màu đặc biệt. Trong một bình, có thể có những giọt nước mắt của nàng tiên cá, biểu tượng cho những ước mơ và mong ước sâu kín. Trong bình khác, có thể chứa ánh sáng huyền ảo từ một ngôi sao xa xôi, chứa đựng những bí mật của vũ trụ.

Trong vương quốc Python, “biến“ là những bình thuỷ tinh kỳ diệu ấy. Hãy thử tưởng tượng, bạn có một bình thuỷ tinh tên là giac_mo, và bạn cho vào đó giá trị 100, đại diện cho hàng trăm giấc mơ bạn ấp ủ:

```
giac_mo = 100
```

Và một bình khác, bạn đặt tên là anh_sang_sao, lưu giữ bên trong là ánh sáng kì diệu từ ngôi sao xa xôi, với giá trị là ’Ánh sáng huyền bí’: anh_sang_sao = “Ánh sáng huyền bí“

Tên của mỗi biến, như giac_mo và anh_sang_sao, chính là phép thuật giúp bạn dễ dàng nhận diện và sử dụng năng lượng phép thuật trong mỗi “bình thuỷ tinh“ mà không lẫn lộn. Bạn có thể “đọc“ phép màu này bằng cách sử dụng phép gọi tên của chúng:

```
print(giac_mo) # Sẽ in ra: 100
print(anh_sang_sao) # Sẽ in ra: Ánh sáng huyền bí
```

Qua việc đặt tên cho biến và gán giá trị, chúng ta không chỉ đơn giản là lưu trữ thông tin mà còn là việc tạo ra và sử dụng phép màu, giống như cách mà một phù thủy sáng tạo ra những điều kỳ diệu từ những bình thuỷ tinh trong phòng thí nghiệm của mình.

Be happy



2. VƯƠNG QUỐC KINH DOANH BẤT ĐỘNG SẢN

Sau khi dạo một vòng qua lãnh địa của giấc mơ và ma thuật, chúng ta bước vào một chương mới đầy hứa hẹn: cuộc phiêu lưu trong vùng đất của Kinh Doanh Bất Động Sản. Hãy cùng nhau tham gia vào bộ phim của thực tế, nơi biến và kiểu dữ liệu không chỉ là những khái niệm khô khan nữa, mà là những công cụ sống động giúp chúng ta khám phá và dự đoán những cơ hội kinh doanh tiềm năng.

Tưởng tượng bạn đang ngồi trong văn phòng của một công ty đầu tư lớn, với bản đồ các dự án bất động sản và chứng khoán trải rộng trên bàn. Mỗi dự án là một cơ hội, và công cụ bạn cần để đánh giá chúng chính là Python, với "biến" và "kiểu dữ liệu" là những chìa khóa.

Ví dụ ứng dụng:

1. Những Khoản Đầu Tư

Bạn có một dự án bất động sản ở trung tâm thành phố và một cơ hội đầu tư vào thị trường chứng khoán đang lên. Hãy đặt tên cho chúng:

```
dauTu_BatDongSan = 5000000 # 5 triệu cho dự án bất động sản
dauTu_ChungKhoan = 2000000 # 2 triệu cho cơ hội chứng khoán
```

Biến ở đây giống như các khoản đầu tư bạn sẵn sàng rót vào mỗi dự án, mỗi con số đều chứa đựng một câu chuyện về hy vọng và rủi ro. Chúng ta đang sử dụng hai biến để đại diện cho hai khoản đầu tư khác nhau: dauTu_BatDongSan và dauTu_ChungKhoan.

Mỗi biến này đều lưu trữ một giá trị số, và kiểu dữ liệu của chúng là số nguyên (int).
Ví dụ:

dauTu_BatDongSan = 5000000 đây là biến được sử dụng để lưu trữ số tiền 5 triệu đồng, đại diện cho khoản đầu tư vào bất động sản.

dauTu_ChungKhoan = 2000000 đây là biến lưu trữ số tiền 2 triệu đồng, đại diện cho khoản đầu tư vào thị trường chứng khoán.

Giải thích Kiểu Dữ liệu:

Kiểu Dữ liệu (Data Type): Kiểu dữ liệu xác định loại giá trị mà một biến có thể lưu trữ, ví dụ như số, văn bản, danh sách, v.v. Trong Python, kiểu dữ liệu cơ bản bao gồm int (số nguyên), float (số thực), str (chuỗi ký tự), và boolean (giá trị đúng/sai). Trong trường hợp của chúng ta:

Giá trị 5000000 và 2000000

được lưu trữ trong dauTu_BatDongSan và dauTu_ChungKhoan lần lượt là các số nguyên, do đó, kiểu dữ liệu của cả hai biến này là int. Như vậy, biến và kiểu dữ liệu trong ví dụ này giúp chúng ta hiểu rõ và quản lý các khoản đầu tư cụ thể trong kế hoạch kinh doanh của mình, đồng thời cung cấp một cách rõ ràng và logic để thể hiện số tiền dự kiến rót vào mỗi dự án, qua đó phản ánh mức độ hy vọng và rủi ro mà chúng ta sẵn lòng chấp nhận.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH

2. VƯƠNG QUỐC KINH DOANH BẤT ĐỘNG SẢN



2. Dự Báo Tương Lai:

Tỷ lệ lợi nhuận không phải là ước mơ, mà là dự báo dựa trên phân tích thị trường. Đặt tên cho những dự đoán này:

```
tyLeLoiNhuan_BatDongSan = 0.1 # Mong đợi 10% lợi nhuận từ bất động sản

tyLeLoiNhuan_ChungKhoan = 0.15 # Mong đợi 15% lợi nhuận từ chứng khoán
```

Giống như việc đoán xem mây sẽ tạo hình gì trên bầu trời, những biến này giúp chúng ta hình dung về lợi nhuận tiềm năng.

Trong ví dụ này, tyLeLoiNhuan_BatDongSan và tyLeLoiNhuan_ChungKhoan đều là biến trong Python, được sử dụng để lưu trữ tỷ lệ lợi nhuận dự kiến từ hai loại hình đầu tư khác nhau: bất động sản và chứng khoán. Giá trị của biến này được thể hiện dưới dạng số thực (floating-point numbers), với 0.1 tương đương với 10% và 0.15 tương đương với 15%.

Tỷ lệ lợi nhuận không phải lấy từ không trung, mà là kết quả của việc phân tích kỹ lưỡng thị trường, các báo cáo tài chính, và dự báo kinh tế. Việc đặt tên cho những dự đoán tỷ lệ lợi nhuận này trong dạng biến cho phép chúng ta dễ dàng tham chiếu và sử dụng chúng trong các phép tính và so sánh sau này.

```
tyLeLoiNhuan_BatDongSan = 0.1
```



Biến này lưu giá trị tỷ lệ lợi nhuận dự kiến từ đầu tư vào bất động sản là 10%. Giá trị này thể hiện sự kỳ vọng về việc mỗi đồng vốn đầu tư vào bất động sản sẽ mang lại 10% lợi nhuận.

```
tyLeLoiNhuan_ChungKhoan = 0.15
```

Tương tự, biến này lưu giá trị tỷ lệ lợi nhuận dự kiến từ đầu tư vào chứng khoán là 15%. Điều này cho thấy sự kỳ vọng rằng đầu tư vào chứng khoán có thể sẽ sinh lợi nhiều hơn so với bất động sản, với mức lợi nhuận là 15% cho mỗi đồng vốn đầu tư.

Việc sử dụng những biến này trong các phép tính sau sẽ giúp định lượng và so sánh cụ thể giữa hai hình thức đầu tư, giúp nhà đầu tư đưa ra quyết định dựa trên bản chất dự báo và tính toán logic hơn là cảm xúc hay ước mơ không cơ sở.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH



2. VƯƠNG QUỐC KINH DOANH BẤT ĐỘNG SẢN

3. Tính Toán - Vẽ Bức Tranh Lợi Nhuận:

Khi đã có bản đồ và la bàn, hãy bắt đầu hành trình tìm kiếm kho báu:

```
loiNhuan_BatDongSan = dauTu_BatDongSan * tyLeLoiNhuan_BatDongSan

loiNhuan_ChungKhoan = dauTu_ChungKhoan * tyLeLoiNhuan_ChungKhoan

print("Lợi nhuận từ Bất động sản:", loiNhuan_BatDongSan)

print("Lợi nhuận từ Chứng khoán:", loiNhuan_ChungKhoan)
```

Giống như việc vẽ bức tranh tương lai bằng màu sắc của lợi nhuận và hy vọng, phép tính này giúp bạn thấy được hình ảnh rõ nét của thành công.Đoạn code này thực hiện tính toán lợi nhuận dự kiến từ hai loại hình đầu tư: bất động sản và chứng khoán, dựa trên số tiền đầu tư ban đầu và tỷ lệ lợi nhuận dự kiến cho mỗi loại.

Giải thích từng dòng code:

```
loiNhuan_BatDongSan = dauTu_BatDongSan * tyLeLoiNhuan_BatDongSan
```

Dòng này tính toán lợi nhuận dự kiến từ đầu tư bất động sản. dauTu_BatDongSan là số tiền bạn đầu tư vào bất động sản, và tyLeLoiNhuan_BatDongSan là tỷ lệ phần trăm lợi nhuận bạn mong đợi từ khoản đầu tư này.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH LUYỆN

2. VƯƠNG QUỐC KINH DOANH BẤT ĐỘNG SẢN



```
loiNhuan_ChungKhoan = dauTu_ChungKhoan * tyLeLoiNhuan_ChungKhoan
```

Tương tự như trên, dòng này tính toán lợi nhuận dự kiến từ đầu tư chứng khoán. dauTu_ChungKhoan là số tiền bạn đầu tư vào thị trường chứng khoán, và tyLeLoiNhuan_ChungKhoan là tỷ lệ phần trăm lợi nhuận bạn mong đợi. Kết quả là số tiền lợi nhuận dự kiến bạn có thể thu được từ thị trường chứng khoán.

```
print("Lợi nhuận từ Bất động sản:", loiNhuan_BatDongSan)
```

Dòng này in ra lợi nhuận dự kiến từ bất động sản, giúp bạn dễ dàng nhìn thấy kết quả tính toán.

```
print("Lợi nhuận từ Chứng khoán:", loiNhuan_ChungKhoan)
```

Dòng này in ra lợi nhuận dự kiến từ bất động sản, giúp bạn dễ dàng nhìn thấy kết quả tính toán.

```
print("Lợi nhuận từ Chứng khoán:", loiNhuan_ChungKhoan)
```

Dòng này in ra lợi nhuận dự kiến từ chứng khoán, tương tự như dòng trên nhưng áp dụng cho thị trường chứng khoán.

Ý Nghĩa: Qua đoạn code này, bạn không chỉ thực hiện được các phép tính toán đơn giản trong Python mà còn có thể áp dụng kiến thức này vào việc đánh giá các cơ hội đầu tư trong thực tế. Việc tính toán này giống như việc vẽ nên bức tranh tương lai với những gam màu lợi nhuận và hy vọng, giúp bạn hình dung được kết quả mà mỗi quyết định đầu tư có thể mang lại, từ đó đưa ra lựa chọn đầu tư sáng suốt và hiệu quả.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH







PHẦN 3

Mở Rộng Tầm Nhìn Qua Các Kịch Bản Đầu Tư

Trong thế giới đầu tư đa màu sắc, mỗi quyết định mang lại hơi thở của một câu chuyện riêng biệt. Aiko đã khám phá những câu chuyện này, thông qua lăng kính của biến và kiểu dữ liệu trong Python, để hiểu rõ hơn về các kịch bản đầu tư thường gặp:

🌳 Kịch Bản "Vườn Ươm": Tăng Trưởng Ổn Định

Câu Chuyện: Như một vườn cây cần thời gian để lớn lên, đầu tư vào trái phiếu chính phủ hay cổ phiếu công ty lớn giống như việc chăm sóc những cây non trưởng thành, dựa vào sự ổn định và dự đoán được của chúng.

Nhân Vật: khoanDauTu, tyLeLoiNhuan với kiểu dữ liệu là int cho số tiền và float cho tỷ lệ lợi nhuận mong đợi.

🚀 Kịch Bản "Chinh Phục Mặt Trăng": Tăng Trưởng Nhanh

Câu Chuyện: Đầu tư vào công nghệ mới hoặc thị trường mới nổi giống như một chuyến du hành tới mặt trăng, đầy rủi ro nhưng tiềm năng tăng trưởng vô cùng lớn.

Nhân Vật: khoanDauTu, tyLeLoiNhuan, mucRuiRo với float cho tỷ lệ lợi nhuận và rủi ro, int cho số tiền.

🧩 Kịch Bản "Bức Tranh": Đa Dạng Hóa

Câu Chuyện: Đầu tư vào đa dạng tài sản giống như việc hoàn thiện một bức tranh từ nhiều mảnh ghép, giảm thiểu rủi ro tổng thể và tạo nên một tác phẩm đầu tư hài hòa.

Nhân Vật: dauTu_TaiSanX, tyLe_TaiSanX với float cho tỷ lệ từng tài sản và int cho tổng số tiền đầu tư.

🏞️ Kịch Bản "Phượng Hoàng": Hồi Phục

Câu Chuyện: Đầu tư vào thị trường sau sự sụt giảm lớn như chứng kiến sự tái sinh của phượng hoàng, tin vào khả năng hồi phục và thăng hoa trở lại.

Nhân Vật: giaTriHienTai, giaTriHoiPhuc với float cho giá trị tài sản, int cho số lượng.

💰 Kịch Bản "Suối Nguồn": Thu Nhập Thụ Động

Câu Chuyện: Đầu tư vào bất động sản cho thuê hay cổ tức giống như việc tạo ra một suối nguồn nhỏ, đều đặn mang lại dòng chảy thu nhập thụ động cho cuộc sống của bạn.

Nhân Vật: dauTuBanDau, thuNhapThuDong với int cho số tiền đầu tư và float cho dòng thu nhập thụ động mong đợi.



CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH

HÀNH TRÌNH: ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

NEW

KỊCH BẢN "VƯỜN ƯƠM": TĂNG TRƯỞNG ỔN ĐỊNH



Hãy bước vào khu vườn của những ước mơ đầu tư, nơi mỗi khoản đầu tư giống như một hạt giống đầy hứa hẹn. Chúng ta sẽ cùng nhau trải qua một hành trình chăm sóc và nuôi dưỡng chúng, với sự hỗ trợ của Python, như một người bạn đồng hành đáng tin cậy.



4 BƯỚC THỰC HIỆN

Bước 1: Chuẩn Bị Mảnh Đất Mơ Ước

Gieo Hạt Giống: Bắt đầu với việc xác định số vốn ban đầu mà bạn muốn đầu tư vào "vườn ươm" của mình. Hãy nghĩ về vốn_đầu_tư như là mảnh đất màu mỡ, sẵn sàng cho những hạt giống tương lai.

```
vốn_đầu_tư = 10000 # 10.000 đô la là mảnh đất của bạn
```



Bước 2: Chăm Sóc với Nước và Phân Bón

Nuôi Dưỡng: Xác định tỷ lệ lợi nhuận dự kiến từ việc đầu tư của bạn. Đây giống như việc tưới nước và bón phân cho mảnh đất, giúp hạt giống phát triển tốt.

```
tỷ_lệ_lợi_nhuận_dự_kiến = 0.05 # 5% mỗi năm, nuôi dưỡng cho hạt giống của bạn
```

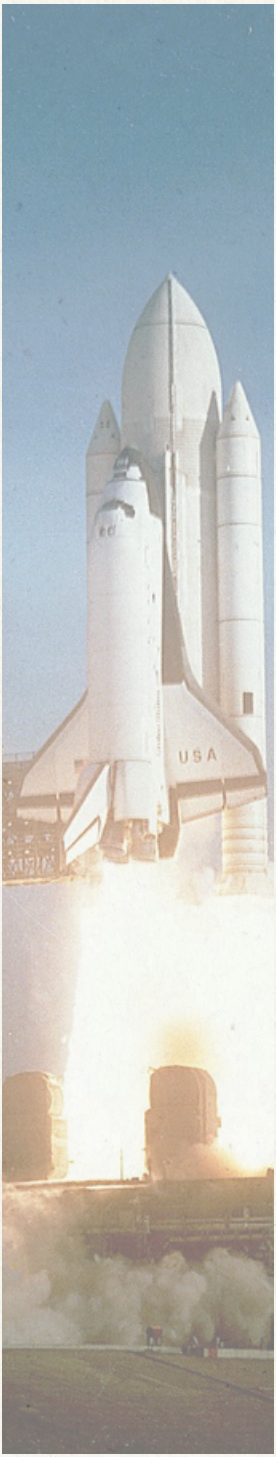
MỞ GOOGLE COLAB VÀ COPY CODE [BƯỚC 1 -2]





Dr happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH



HÀNH TRÌNH: ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

Hãy cùng nhau bước lên tàu vũ trụ của những nhà đầu tư táo bạo, chuẩn bị cho chuyến "Chinh Phục Mặt Trăng" đầy kỳ thú. Đây không phải là cuộc phiêu lưu bình thường, mà là hành trình đầu tư vào những lĩnh vực mới mẻ như công nghệ tiên tiến hoặc thị trường mới nổi. Có thể có những rủi ro không lường trước được nhưng cũng chứa đựng cơ hội tăng trưởng phi thường.

KỊCH BẢN "CHINH PHỤC MẶT TRĂNG": TĂNG TRƯỞNG NHANH

Bước 1: Chuẩn Bị Tàu Vũ Trụ

Lên Kế Hoạch: Xác định khoanDauTu của bạn - số tiền bạn sẵn lòng rót vào chuyến phiêu lưu này. Mỗi đồng vốn giống như nhiên liệu cho tàu vũ trụ, quyết định quãng đường bạn có thể đi.

khoanDauTu = 10000 # 10.000 đô la là nhiên liệu cho tàu vũ trụ

Bước 2: Định Vị Hành Tinh Mục Tiêu

Phân Tích Điểm Đến: Định rõ tyLeLoiNhuan dự kiến - tỷ lệ lợi nhuận mà bạn hy vọng sẽ đạt được từ lĩnh vực công nghệ hoặc thị trường mới này. Đồng thời, đừng quên đánh giá mucRuiRo - những thách thức và bất trắc có thể gặp phải trên hành trình.

Python code
tyLeLoiNhuan = 0.2 # Kỳ vọng lợi nhuận 20%
mucRuiRo = 0.1 # Rủi ro 10%

Bước 3: Phóng Tàu và Theo Dõi Hành Trình

Dự Đoán Hành Trình: Sử dụng Python để mô phỏng chuyến bay, kết hợp khoanDauTu, tyLeLoiNhuan, và mucRuiRo để tính toán kết quả dự kiến của chuyến đầu tư này.

Python code:
loiNhuanDuKien = khoanDauTu * tyLeLoiNhuan
ruiRoDuKien = khoanDauTu * mucRuiRo

Be happy



HÀNH TRÌNH: ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

Hãy cùng nhau bước lên tàu vũ trụ của những nhà đầu tư táo bạo, chuẩn bị cho chuyến "Chinh Phục Mặt Trăng" đầy kỳ thú. Đây không phải là cuộc phiêu lưu bình thường, mà là hành trình đầu tư vào những lĩnh vực mới mẻ như công nghệ tiên tiến hoặc thị trường mới nổi. Có thể có những rủi ro không lường trước được nhưng cũng chứa đựng cơ hội tăng trưởng phi thường.

KỊCH BẢN "CHINH PHỤC MẶT TRĂNG": TĂNG TRƯỞNG NHANH

🌟 Bước 4: Đánh Giá Kết Quả Hành Trình

Phân Tích Kết Quả: Đánh giá kết quả dự kiến từ chuyến "Chinh Phục Mặt Trăng" thông qua loiNhuanDuKien và ruiRoDuKien, giúp bạn hiểu được cả những mặt sáng lẫn bóng tối của cuộc phiêu lưu.

Python Copy code

```
print("Lợi nhuận dự kiến từ chuyến du hành:", loiNhuanDuKien, "đô la.")
print("Rủi ro có thể gặp phải:", ruiRoDuKien, "đô la.")
```

Chuyến "Chinh Phục Mặt Trăng" không chỉ là một cuộc phiêu lưu đầu tư mà còn là hành trình khám phá bản thân, mở rộng tầm nhìn và dũng cảm đối diện với những không gian mới. Với Python như người bạn đồng hành, bạn không chỉ có công cụ để tính toán mà còn có ánh sáng dẫn đường trong bóng tối của vũ trụ bao la, giúp bạn đi từ hy vọng tới hiện thực.

Do happy



HÀNH TRÌNH: ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

Hãy cùng nhau bước lên tàu vũ trụ của những nhà đầu tư táo bạo, chuẩn bị cho chuyến "Chinh Phục Mặt Trăng" đầy kỳ thú. Đây không phải là cuộc phiêu lưu bình thường, mà là hành trình đầu tư vào những lĩnh vực mới mẻ như công nghệ tiên tiến hoặc thị trường mới nổi. Có thể có những rủi ro không lường trước được nhưng cũng chứa đựng cơ hội tăng trưởng phi thường.

KỊCH BẢN "CHINH PHỤC MẶT TRĂNG": TĂNG TRƯỞNG NHÁNH

Full Code:

```
# 🌟 Bước 1: Chuẩn Bị Tàu Vũ Trụ
# Lên kế hoạch: Xác định số tiền đầu tư cho chuyến phiêu lưu này
khoanDauTu = 10000 # Đặt số tiền đầu tư là 10.000 đô la

# 🌟 Bước 2: Định Vị Hành Tinh Mục Tiêu
# Phân tích điểm đến: Xác định tỷ lệ lợi nhuận dự kiến và mức rủi ro tiềm ẩn
tyLeLoiNhuan = 0.2 # Kỳ vọng lợi nhuận 20%
mucRuiRo = 0.1 # Rủi ro tiềm ẩn 10%

# 🌟 Bước 3: Phóng Tàu và Theo Dõi Hành Trình
# Dự đoán hành trình: Tính toán lợi nhuận và rủi ro dự kiến từ chuyến đầu tư
loiNhuanDuKien = khoanDauTu * tyLeLoiNhuan # Tính lợi nhuận dự kiến
ruiRoDuKien = khoanDauTu * mucRuiRo # Tính rủi ro tiềm ẩn

# 🌟 Bước 4: Đánh Giá Kết Quả Hành Trình
# Phân tích kết quả: Đánh giá và hiểu biết kết quả dự kiến từ chuyến phiêu lưu
print(f"Lợi nhuận dự kiến từ chuyến du hành: {loiNhuanDuKien} đô la.")
print(f"Rủi ro có thể gặp phải: {ruiRoDuKien} đô la.")
```

Qua từng bước, từ việc lên kế hoạch đầu tư cho đến đánh giá kết quả dự kiến, Python giúp chúng ta minh họa rõ ràng mỗi phần của cuộc phiêu lưu "Chinh Phục Mặt Trăng". Đoạn mã này không chỉ giúp bạn tính toán lợi nhuận và rủi ro tiềm ẩn, mà còn mở ra cánh cửa cho sự mạo hiểm đầy thông minh và có tính toán.

Be happy

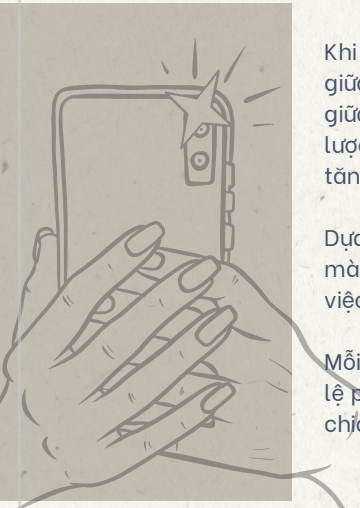
HÀNH TRÌNH: ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

Bạn đang đứng trước một tờ giấy trắng lớn, trong tay là bảng màu của những cơ hội đầu tư. Mỗi lựa chọn của bạn, từ bất động sản, cổ phiếu, đến trái phiếu, giống như việc chọn một màu sắc khác nhau để phác họa lên tờ giấy ấy. Hãy hình dung bạn là một họa sĩ tài ba, đang đứng trước một bức tranh còn trống rỗng, sẵn sàng để khắc họa những ước mơ và tham vọng của mình. Trong tay bạn là bảng màu đầy ắp những lựa chọn đầu tư: dauTu_TaiSanA, dauTu_TaiSanB, dauTu_TaiSanC... Mỗi khoản đầu tư giống như một màu sơn đặc trưng, từ màu xanh lá của sự an toàn, bền vững, đến màu đỏ rực rỡ của sự đam mê, mạo hiểm, và cơ hội lớn.

dauTu_TaiSanA có thể là một bức tranh về bất động sản, vững chắc và kiên cố, mang màu xanh của sự an nhàn và ổn định. dauTu_TaiSanB lại như những cổ phiếu công nghệ mới nổi, đầy màu sắc và sự biến đổi, chói lọi với màu vàng của sự sáng tạo và đổi mới. Trong khi đó, dauTu_TaiSanC có thể là trái phiếu chính phủ, với màu xám của sự đáng tin cậy và an toàn.

Kịch Bản "Bức Tranh": Đa Dạng Hóa

Mỗi lựa chọn của bạn, từ bất động sản, cổ phiếu, đến trái phiếu, giống như việc chọn một màu sắc khác nhau để phác họa lên tờ giấy ấy



Khi bạn kết hợp những mảnh ghép này, không chỉ đơn thuần là lựa chọn giữa màu sắc, mà là việc tạo ra một bức tranh đầu tư đa dạng, cân đối giữa rủi ro và cơ hội. Mỗi nét vẽ, mỗi lựa chọn, đều là một phần của chiến lược đa dạng hóa, nhằm giảm thiểu rủi ro tổng thể và tối đa hóa cơ hội tăng trưởng.

Dựa vào câu chuyện về việc đa dạng hóa đầu tư như một bức tranh đầy màu sắc, chúng ta có thể mô phỏng qua một đoạn mã Python để thể hiện việc phân bổ vốn vào các loại tài sản khác nhau.

Mỗi loại tài sản sẽ được đại diện bằng một biến, và chúng ta sẽ sử dụng tỷ lệ phần trăm của tổng số vốn đầu tư cho mỗi loại để thể hiện việc phân chia đầu tư.

Do happy



HÀNH TRÌNH:
ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

Kịch Bản "Bức Tranh": Đa Dạng Hóa

```
# Tổng số tiền đầu tư
tongVonDauTu = 10000 # Giả sử bạn có 10.000 đô la để đầu tư

# Định nghĩa tỷ lệ phần trăm đầu tư cho mỗi loại tài sản
tyLe_TaiSanA = 0.4 # 40% cho Tài sản A - có thể là bất động sản
tyLe_TaiSanB = 0.3 # 30% cho Tài sản B - có thể là cổ phiếu công nghệ
tyLe_TaiSanC = 0.3 # 30% cho Tài sản C - có thể là trái phiếu hoặc quỹ đầu tư an toàn

# Tính toán số tiền đầu tư cho mỗi loại tài sản
dauTu_TaiSanA = tongVonDauTu * tyLe_TaiSanA
dauTu_TaiSanB = tongVonDauTu * tyLe_TaiSanB
dauTu_TaiSanC = tongVonDauTu * tyLe_TaiSanC

# In ra số tiền đầu tư cho mỗi loại tài sản
print(f"Số tiền đầu tư vào Tài sản A: {dauTu_TaiSanA} đô la")
print(f"Số tiền đầu tư vào Tài sản B: {dauTu_TaiSanB} đô la")
print(f"Số tiền đầu tư vào Tài sản C: {dauTu_TaiSanC} đô la")

# In ra tổng quan về bức tranh đầu tư
print("\nBức tranh đầu tư của bạn bao gồm:")
print(f"- Màu xanh của sự an nhàn từ Tài sản A với {tyLe_TaiSanA*100}% tổng vốn.")
print(f"- Sắc đỏ của cơ hội và mạo hiểm từ Tài sản B với {tyLe_TaiSanB*100}% tổng vốn.")
print(f"- Màu vàng của sự cân bằng từ Tài sản C với {tyLe_TaiSanC*100}% tổng vốn.")
```

Dr happy

HÀNH TRÌNH ÁP DỤNG KIẾN THỨC VÀO ĐẦU TƯ

KỊCH BẢN "PHƯỢNG HOÀNG": HỒI PHỤC

Hãy bắt đầu hành trình tái sinh kỳ diệu từ tro tàn của thị trường, giống như sự hồi sinh oanh liệt của phượng hoàng trong kịch bản "Phượng Hoàng". Cùng nhau, bước qua từng giai đoạn với niềm tin và hy vọng.

BƯỚC 1: NHÌN NHẬN HIỆN TRẠNG

Trước hết, hãy đánh giá giaTriHienTai của tài sản sau cơn bão thị trường. Đây là thời điểm khi tài sản của bạn ở mức thấp nhất, giống như phượng hoàng rơi vào đống tro tàn.

giaTriHienTai = 5000.0 # Tài sản hiện đang ở mức 5.000 đô la

BƯỚC 2: VẼ NÊN TƯƠNG LAI

Tiếp theo, hãy dùng trí tưởng tượng của mình để dự đoán giaTriHoiPhuc. Đây là ước mơ về ngày tài sản không chỉ trở lại giá trị ban đầu mà còn tăng trưởng mạnh mẽ.

giaTriHoiPhuc = 8000.0 # Dự đoán tài sản sẽ hồi phục và tăng lên 8.000 đô la

BƯỚC 3: TÍNH TOÁN HÀNH TRÌNH PHỤC HỒI

Hãy sử dụng công cụ của mình để tính toán tangGia, đo lường quãng đường phục hồi và tăng trưởng mà bạn mong đợi từ tài sản.

tangGia = giaTriHoiPhuc - giaTriHienTai # Tính sự tăng trưởng từ hiện tại đến tương lai

BƯỚC 4: MỪNG SỰ HỒI SINH

Cuối cùng, hãy nhìn lại quãng đường đã đi và mừng sự hồi sinh kỳ diệu của tài sản, như phượng hoàng lột xác từ đống tro tàn.

print(f"Tài sản đã hồi sinh từ {giaTriHienTai} đô la lên {giaTriHoiPhuc} đô la, tăng {tangGia} đô la.")

print("\nHãy tin vào sức mạnh hồi phục của thị trường. Giống như Phượng Hoàng, mỗi cơ hội đều bắt đầu từ những thách thức và sự kiên nhẫn đợi chờ.")

Trong kịch bản "Phượng Hoàng", bạn như một nhà tiên tri đầy tầm nhìn, đặt niềm tin vào sự hồi sinh và thăng hoa của thị trường sau một giai đoạn sụt giảm nặng nề. Dưới đây là đoạn code Python thể hiện câu chuyện này:

```
# Định nghĩa giá trị hiện tại của tài sản, sau sự sụt giảm
giaTriHienTai = 5000.0 # Giả sử giá trị hiện tại của tài sản là 5.000 đô la sau sự sụt giảm

# Dự đoán giá trị hồi phục của tài sản, tin tưởng vào sự tái sinh như phượng hoàng
giaTriHoiPhuc = 8000.0 # Dự đoán giá trị tài sản sẽ hồi phục lên 8.000 đô la

# Tính toán sự tăng giá của tài sản sau khi hồi phục
tangGia = giaTriHoiPhuc - giaTriHienTai

# In ra sự tái sinh kỳ diệu của tài sản từ tro tàn
print(f"Tài sản đã hồi sinh từ {giaTriHienTai} đô la lên {giaTriHoiPhuc} đô la, tăng {tangGia} đô la, thật giống như sự tái sinh của Phượng Hoàng từ đống tro tàn.")

# In ra thông điệp về sự hồi phục và thăng hoa
print("\nĐầu tư vào thị trường sau sự sụt giảm chính là niềm tin vào khả năng hồi phục và thăng hoa trở lại. Giống như Phượng Hoàng, mỗi sự sụp đổ đều là cơ hội cho một sự khởi đầu mới mẻ và rực rỡ hơn.")
```

Be happy

Hành Trình Áp Dụng Kiến Thức Vào Đầu Tư



Hãy hình dung bạn đang khám phá một khu vườn xanh mướt, nơi giữa lòng đất mẹ ẩn chứa một suối nguồn trong veo, nhỏ nhưng không ngừng chảy, mang lại sự sống cho mọi sinh vật xung quanh. Kịch bản "Suối Nguồn" trong thế giới đầu tư của chúng ta cũng tương tự như vậy, nơi bạn đặt niềm tin vào các khoản đầu tư như bất động sản cho thuê hay cổ tức, từ đó tạo ra một dòng thu nhập thụ động đều đặn cho cuộc sống của mình.

Bước 1: Chọn Mảnh Đất Màu Mỡ
Định Vị Suối Nguồn: Quyết định dauTuBanDau của bạn, số vốn bạn sẵn sàng gieo trồng vào khu vườn đầu tư của mình để nuôi dưỡng suối nguồn thu nhập.

dauTuBanDau = 50000 # 50.000 đô la là số tiền bạn đầu tư ban đầu

Bước 2: Nuôi Dưỡng Suối Nguồn
Lên Kế Hoạch Dòng Chảy: Xác định thuNhapThuDong dự kiến từ suối nguồn đầu tư của bạn, tỷ lệ phần trăm dòng thu nhập thụ động hàng năm mà bạn mong đợi.

thuNhapThuDong = 0.05 # Kỳ vọng 5% thu nhập thụ động hàng năm từ tổng số vốn đầu tư

Bước 3: Tận Hưởng Dòng Chảy Thu Nhập
Thu Hoạch Nước Từ Suối: Tính toán số tiền thu nhập thụ động hàng năm mà suối nguồn đầu tư của bạn mang lại.

thuNhapHangNam = dauTuBanDau * thuNhapThuDong
print(f"Mỗi năm, suối nguồn đầu tư của bạn sẽ mang lại {thuNhapHangNam:.2f} đô la dưới dạng thu nhập thụ động.")

Bước 4: Đánh Giá và Mở Rộng Suối Nguồn
Phản Ánh và Phát Triển: Đánh giá dòng chảy thu nhập thụ động và tìm cách nuôi dưỡng, mở rộng suối nguồn của bạn để nó mang lại nhiều giá trị hơn cho cuộc sống.

Full Code:
Dưới đây là đoạn code Python hoàn chỉnh cho kịch bản "Suối Nguồn", thể hiện việc tạo ra và tận hưởng thu nhập thụ động từ việc đầu tư:

```
# Bước 1: Chọn Mảnh Đất Màu Mỡ
# Định Vị Suối Nguồn: Xác định số tiền đầu tư ban đầu
dauTuBanDau = 50000 # 50.000 đô la là số tiền bạn đầu tư ban đầu

# Bước 2: Nuôi Dưỡng Suối Nguồn
# Lên Kế Hoạch Dòng Chảy: Xác định tỷ lệ thu nhập thụ động mong đợi từ đầu tư
thuNhapThuDong = 0.05 # Kỳ vọng 5% thu nhập thụ động hàng năm từ tổng số vốn đầu tư

# Bước 3: Tận Hưởng Dòng Chảy Thu Nhập
# Tính toán số tiền thu nhập thụ động hàng năm
thuNhapHangNam = dauTuBanDau * thuNhapThuDong
print(f"Mỗi năm, suối nguồn đầu tư của bạn sẽ mang lại {thuNhapHangNam:.2f} đô la dưới dạng thu nhập thụ động.")

# Bước 4: Đánh Giá và Mở Rộng Suối Nguồn
# Xem xét cơ hội đầu tư mới hoặc cải thiện suối nguồn hiện tại để tăng thu nhập thụ động
print("Hãy xem xét cơ hội đầu tư mới hoặc tìm cách cải thiện và mở rộng suối nguồn thu nhập thụ động của bạn để nó mang lại nhiều giá trị hơn cho cuộc sống của bạn.")
```

Be happy

DAY 3: QUYẾT ĐỊNH THÔNG MINH



IF - ELSE

Chúng ta bắt đầu hành trình khám phá if-else, không chỉ như một công cụ trong lập trình mà còn như một biểu tượng của sự lựa chọn và quyết định trong cuộc sống. Dưới đây là những lý do sâu sắc mà tại sao if-else lại quan trọng và thú vị, mỗi lý do không chỉ là một bài học mà còn là nguồn cảm hứng để bạn đam mê hơn trong hành trình lập trình:

- Sự Lựa Chọn Mỗi Ngày
- Phát Triển Tư Duy Logic
- Nâng Cao Kỹ Năng Lập Kế Hoạch
- Ứng Dụng Rộng Rãi Trong Đời Sống
- Cảm Hứng Từ Cuộc Sống

"Bí Mật Được Che Giấu Của If-Else". Khi đứng trước ngã tư đầy ẩn số, if-else như người hướng dẫn đáng tin cậy, chỉ lối cho chúng ta trên mỗi bước đi

PHÉP THUẬT CHIA LẤY DƯ



KHÁM PHÁ ĐẢO LOGIC, BÍ ẨN KHU RỪNG SỐ HỌC

Nhưng, if-else giống như chiếc chìa khóa mở cánh cửa đến với quyết định logic, nhưng không phải lúc nào nó cũng là phép màu tốt nhất cho mọi bài toán trong hành trình sáng tạo của mình, việc lựa chọn công cụ phù hợp là chìa khóa. If-else là một phần của kho báu lập trình

IF - ELSE [LIFE NOT EASY ?]

why if -else?



Chúng ta bắt đầu hành trình khám phá if-else, không chỉ như một công cụ trong lập trình mà còn như một biểu tượng của sự lựa chọn và quyết định trong cuộc sống. Dưới đây là những lý do sâu sắc mà tại sao if-else lại quan trọng và thú vị, mỗi lý do không chỉ là một bài học mà còn là nguồn cảm hứng để bạn đam mê hơn trong hành trình lập trình:

Sự Lựa Chọn Mỗi Ngày: Cuộc sống của chúng ta được vẽ nên từ những lựa chọn, từ những quyết định nhỏ nhặt như chọn món ăn cho bữa sáng đến những quyết định lớn như chọn hướng đi cho sự nghiệp. If-else là ngôn ngữ của những lựa chọn ấy, giúp chúng ta mô phỏng quyết định trong chương trình, cho phép máy tính "cảm nhận" và "phản ứng" linh hoạt với từng tình huống.



Phát Triển Tư Duy Logic: Học if-else giống như chơi một trò chơi giải đố, nơi mỗi câu đố cần được giải quyết bằng lối suy nghĩ logic và sắc bén. Qua đó, bạn không chỉ học cách viết mã mà còn rèn luyện khả năng nhận diện, phân tích vấn đề và tìm ra lời giải đúng đắn nhất từ những tình huống phức tạp.



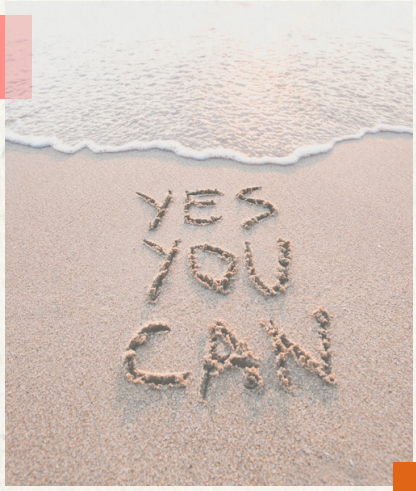
Nâng Cao Kỹ Năng Lập Kế Hoạch: If-else giúp bạn trở thành người lập kế hoạch tài ba, bởi lẽ nó yêu cầu bạn phải dự đoán trước và chuẩn bị cho mọi khả năng có thể xảy ra. Từ đó, bạn học cách đưa ra kế hoạch chi tiết và dự bị cho những biến số không ngờ tới, giúp bạn luôn đi một bước trước và đạt được mục tiêu một cách chủ động.

Be happy

Ứng Dụng Rộng Rãi Trong Đời Sống: If-else không giới hạn trong bốn bức tường của lập trình mà lan rộng ra khắp mọi lĩnh vực cuộc sống, từ tự động hóa các nhiệm vụ hàng ngày, đưa ra quyết định tài chính, đến việc giải quyết các vấn đề phức tạp trong công việc và cuộc sống cá nhân. Nó là chìa khóa mở ra cánh cửa của sự sáng tạo và giải pháp thông minh, giúp cuộc sống trở nên đơn giản và thú vị hơn.

IF - ELSE [LIFE NOT EASY ?]

why if -else? (continute



Cảm Hứng Từ Cuộc Sống: Tự Động Hóa Cuộc Sống Hàng Ngày: If-else giúp bạn tưởng tượng ra một ngày mà mọi quyết định, từ việc chọn trang phục đến việc lựa chọn thực đơn, đều được tự động hóa, giải phóng bạn khỏi những gánh nặng nhỏ nhặt hàng ngày và cho phép bạn tập trung vào những điều quan trọng hơn.

Quản Lý Tài Chính Cá Nhân: If-else mở ra cánh cửa cho sự thông minh trong quản lý tài chính, giúp bạn tự đặt ra các ngưỡng và điều kiện để chi tiêu, tiết kiệm, và đầu tư một cách hiệu quả, từ đó xây dựng một tương lai tài chính vững chắc và bền vững.



Định Hình Nghề Nghiệp: Trong sự nghiệp, if-else như một ngọn đèn dẫn đường, giúp bạn so sánh, đánh giá và chọn lựa con đường đi phù hợp nhất với mình, dựa trên những điều kiện và ưu tiên cá nhân.



Kỹ thuật if-else không chỉ là một phần của lập trình mà còn là một phần của cuộc sống, mang lại cho bạn cách nhìn và phương pháp mới để đối diện và giải quyết những thách thức hàng ngày, mở ra những cơ hội và khả năng mới trong một thế giới đầy màu sắc và sự đổi mới.

"If-else" chất đến vậy, bạn đã sẵn sàng tham gia vào cuộc phiêu lưu lập trình cùng mình?

Chào mừng bạn đến với cuộc hành trình khám phá if-else, nơi mỗi bước chân chúng ta đặt ra không chỉ là một dấu ấn trên con đường kiến thức mà còn là cơ hội để sáng tạo và vẽ nên những câu chuyện mới lạ.

Be happy

BẢN ĐỒ GIÚP CHÚNG TA KHÔNG BỊ LẠC GIỮA BIỂN KIẾN THỨC MÊNH MÔNG

Chào mừng bạn đến với cuộc hành trình khám phá if-else, nơi mỗi bước chân chúng ta đặt ra không chỉ là một dấu ấn trên con đường kiến thức mà còn là cơ hội để sáng tạo và vẽ nên những câu chuyện mới lạ. Hãy tưởng tượng chúng ta là những nhà thám hiểm, cùng nhau du ngoạn qua những con đường chưa từng biết, từ những hiểu biết cơ bản ban đầu cho đến việc áp dụng những bài học đó vào các dự án tràn đầy sức sống.

Đầu tiên việc lắng nghe câu chuyện của Logic nơi chúng ta làm quen với những nhân vật chính: giá trị "True" và "False", những người bạn "True" và "False", cùng với đội quân toán tử điều kiện. Đây chính là những nhân vật mở đầu cho cuộc hành trình kỳ diệu của chúng ta, giúp chúng ta làm quen với thế giới Logic huyền bí.

Tiếp theo, chúng ta sẽ dấn thân vào "Bí Mật Được Che Giấu Của If-Else". Khi đứng trước ngã tư đầy ẩn số, if-else như người hướng dẫn đáng tin cậy, chỉ lối cho chúng ta trên mỗi bước đi. Học cách sử dụng if-else, chính là việc khám phá chìa khóa mở ra muôn vàn cánh cửa mới trên con đường khám phá.

Khi kiến thức đã nằm lòng, "Cuộc Chơi Của Những Thử Thách" chính là bài test tiếp theo. Mỗi thử thách là một câu đố mở, yêu cầu chúng ta áp dụng sự hiểu biết về if-else để giải quyết, giúp kiến thức trở nên sâu sắc và bền vững hơn trong tâm trí.

Và cuối cùng, không thể thiếu "Dấu Ấn Cá Nhân Trong Dự Án". Đây là lúc chúng ta mang những gì đã học vào thực tế, thông qua một dự án nhỏ. Việc này không chỉ giúp kiểm chứng kiến thức mà còn cho phép chúng ta sáng tạo, biến ý tưởng thành hiện thực, để lại dấu ấn độc đáo của mình trong thế giới lập trình.

Khám Phá Đảo Logic:

Cư Dân Chính:

- True: Đại diện cho sự đúng đắn, xác thực. Trong if-else, khi một điều kiện được đánh giá là "True", khối lệnh tương ứng sẽ được thực thi.
- False: Tượng trưng cho sự sai lầm, không xác thực. Nếu điều kiện là "False", khối lệnh trong phần else (nếu có) sẽ được thực thi.
- Toán Tử Điều Kiện - Bạn Đồng Hành Cần Thiết:
- Trong thế giới Logic, có một số toán tử điều kiện bạn cần quan tâm để xác định mối quan hệ giữa các biến và giá trị:
- Toán tử bằng (==): Kiểm tra xem hai giá trị có bằng nhau không.
- Toán tử không bằng (!=): Kiểm tra xem hai giá trị có khác nhau không.
- Toán tử lớn hơn (>), nhỏ hơn (<): So sánh giữa hai giá trị.
- Toán tử lớn hơn hoặc bằng (>=), nhỏ hơn hoặc bằng (<=): So sánh giữa hai giá trị với điều kiện bằng.



Dr happy





Nào chúng mình cùng nhau test thử một ví dụ nho nhỏ: Bí mật của Số Chẵn và Số Lẻ

🌟 Bí Ẩn Của Khu Rừng Số Học

Trong khu rừng này, có một con đường mòn dẫn đến hai ngôi làng bí ẩn: Làng Số Chẵn và Làng Số Lẻ. Để biết được con đường nào dẫn đến làng nào, chúng ta cần phải giải một bí ẩn: Làm sao để biết một số là số chẵn hay số lẻ?



NUMBER 4

Hãy thử nghiệm phép thuật này với số 4, một viên đá quý mà chúng ta tìm thấy trên con đường:

python
# Viên đá quý mà chúng ta tìm thấy là số 4
viên_đá_quý = 4

🔍 Phép Thuật Chia Lấy Dư

Chúng ta có một phép thuật trong túi: Toán tử chia lấy dư (%). Phép thuật này sẽ giúp chúng ta xác định một số bí mật, nằm ẩn sau mỗi con số.

Hãy thử nghiệm phép thuật này với số 4, một viên đá quý mà chúng ta tìm thấy trên con đường:



PYTHON

# Sử dụng phép thuật chia lấy dư để khám phá bí mật
if viên_đá_quý % 2 == 0:
print("Hỡi kẻ phiêu lưu, con đường này dẫn đến Làng Số Chẵn.")
else:
print("Cẩn thận! Con đường này đưa chúng ta đến Làng Số Lẻ.")
Khi áp dụng phép thuật, nếu điều kiện cho "True" (tức là không có số dư khi chia cho 2), bí ẩn được hé lộ: con đường dẫn đến Làng Số Chẵn. Ngược lại, nếu phép thuật cho "False" (có số dư), chúng ta sẽ bước vào Làng Số Lẻ.

Be happy

Câu Chuyện Về Cánh Cổng Bí Ẩn

Chúng ta đang đứng trước một cánh cổng bí ẩn trong khu vườn của phù thủy. Cánh cổng này chỉ mở ra khi chúng ta nói đúng mật khẩu. Phù thủy đã đưa ra một điều kiện: "Nếu bạn nói ra một số lớn hơn 10, cánh cổng sẽ mở ra cho bạn."



NUMBER 12

Chúng ta hãy sử dụng kiến thức về "True" và "False" để xác định xem chúng ta có thể qua cánh cổng hay không.

💡 Ví Dụ Minh Họa:
Giả sử chúng ta chọn số 12 làm mật khẩu:

```
python code
# Mật khẩu mà chúng ta chọn
mật_khẩu = 12
# Điều kiện để qua cánh cổng
if mật_khẩu > 10:
 print("Cánh cổng mở ra, chào mừng bạn đến với khu vườn bí ẩn!")
else:
 print("Rất tiếc, cánh cổng không mở. Hãy thử mật khẩu khác.")
```

Trong ví dụ này, mật_khẩu > 10 là điều kiện được đặt ra.

PYTHON

Khi điều kiện mật_khẩu > 10 trả về "True" (đúng vì 12 lớn hơn 10), câu lệnh print("Cánh cổng mở ra, chào mừng bạn đến với khu vườn bí ẩn!") sẽ được thực thi. Điều này tượng trưng cho việc chúng ta đã nói đúng mật khẩu và được phép qua cánh cổng.

Nếu chúng ta chọn một số nhỏ hơn hoặc bằng 10, điều kiện sẽ trả về "False" và câu lệnh trong phần else sẽ được thực thi, tượng trưng cho việc chúng ta không qua được cánh cổng.

Câu chuyện này cho thấy sức mạnh của việc sử dụng "True" và "False" trong việc kiểm soát dòng chảy của chương trình. Chúng ta có thể áp dụng cùng một nguyên tắc để kiểm tra nhiều điều kiện khác nhau, mở ra cánh cổng đến những khu vườn bí ẩn và những cuộc phiêu lưu mới trong thế giới lập trình.

Be happy

BẠN HÃY THỬ NHỮNG KHÁI NIỆM LOGIC KHÁC CHO NHỮNG BÀI TOÁN MÌNH GỢI Ý:



Thời tiết, xem phim, coffee?

1. Quyết định mặc gì dựa trên thời tiết và lịch trình (and)

Giả sử bạn cần quyết định xem nên mặc gì trước khi ra khỏi nhà. Bạn sẽ quyết định mặc áo khoác nếu trời lạnh và bạn có kế hoạch ra ngoài:

Thời tiết lạnh (True) và bạn có kế hoạch ra ngoài (True): Mặc áo khoác.
Thời tiết lạnh (True) nhưng bạn không có kế hoạch ra ngoài (False): Có thể không cần mặc áo khoác.

2. Xem phim hoặc đi dạo (or)

Bạn muốn thư giãn vào buổi tối và đang phân vân giữa việc xem một bộ phim hoặc đi dạo. Bạn sẽ quyết định xem phim nếu trời mưa hoặc phim bạn muốn xem đã có trên dịch vụ streaming:

Trời mưa (True) hoặc phim đã có (True): Xem phim.
Trời không mưa (False) và phim chưa có (False): Có thể bạn sẽ chọn đi dạo.

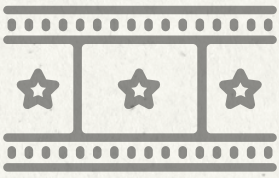
3. Không uống cà phê sau 6 giờ tối (not)

Bạn quyết định không uống cà phê sau 6 giờ tối để không ảnh hưởng đến giấc ngủ:

Nếu bây giờ là 7 giờ tối (True): Sử dụng not để nhắc nhở bản thân không uống cà phê.
~~Nếu bây giờ là 5 giờ chiều (False): Bạn vẫn có thể uống cà phê nếu muốn.~~







Do happy



Brainstorm idea: WEATHER .....

Ứng Dụng Quản Lý Trang Phục Theo Thời Tiết và Lịch Trình

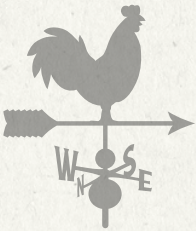
Bài Toán: Phát triển một ứng dụng di động hoặc web giúp người dùng lựa chọn trang phục phù hợp với thời tiết và các kế hoạch trong ngày. Ứng dụng sẽ đưa ra gợi ý về việc nên mặc gì dựa trên dự báo thời tiết và lịch trình cụ thể của người dùng.

Cách Tiếp Cận:

Lấy dữ liệu thời tiết từ API thời tiết và thông tin lịch trình từ lịch cá nhân của người dùng.
Xác định điều kiện thời tiết (nhiệt độ, mưa, nắng, v.v.) và kiểm tra lịch trình người dùng (có sự kiện ngoài trời hay không).
Áp dụng logic and để kết hợp các điều kiện và đưa ra gợi ý trang phục.

WEATHER
Quản Lý Trang Phục Theo Thời Tiết và Lịch Trình









Brainstorm idea: RELAX, SLEEEP .....

Hệ Thống Gợi Ý Hoạt Động Thư Giãn Buổi Tối
Bài Toán: Tạo một hệ thống hoặc ứng dụng gợi ý hoạt động thư giãn cho buổi tối dựa trên điều kiện thời tiết và sở thích cá nhân.

Cách Tiếp Cận:

Người dùng nhập sở thích cá nhân (xem phim, đi dạo, đọc sách, v.v.) và cài đặt sở thích thời tiết cho mỗi hoạt động (ví dụ: thích đi dạo khi nắng, xem phim khi mưa).
Hệ thống sử dụng dự báo thời tiết từ API và áp dụng logic or để kết hợp các sở thích và điều kiện thời tiết, đưa ra gợi ý hoạt động thư giãn phù hợp.

Ứng Dụng Nhắc Nhở Thói Quen Tốt Cho Giấc Ngủ
Bài Toán: Phát triển một ứng dụng nhắc nhở người dùng tránh những thói quen có thể ảnh hưởng đến giấc ngủ, như uống cà phê sau một thời gian nhất định trong ngày.

Cách Tiếp Cận:

Người dùng cài đặt “giờ cấm cà phê“ (ví dụ: không uống cà phê sau 6 giờ tối).
Ứng dụng theo dõi thời gian thực và sử dụng logic not để kiểm tra xem hiện tại có phải là thời gian “cấm cà phê“ không. Nếu có, ứng dụng sẽ gửi nhắc nhở hoặc gợi ý thay thế cà phê bằng thức uống khác không chứa caffeine.

RELAX	HỆ THỐNG GỢI Ý HOẠT ĐỘNG THƯ GIÃN BUỔI TỐI
SLEEP	NHẮC NHỞ THÓI QUEN TỐT CHO GIẤC NGỦ

Do happy



Chúng mình thử code bài tập số 2.

Hệ Thống Gợi Ý Hoạt Động Thư Giãn Buổi Tối

```
# Giả định rằng chúng ta đã có dữ liệu thời tiết và sở thích cá nhân của người dùng

# Dữ liệu mẫu
thoi_tiet_hien_tai = “mưa“ # Các tùy chọn có thể là “nắng“, “mưa“, “lạnh“
so_thich_ca_nhan = [“xem phim“, “đi dạo“, “đọc sách“]

# Hàm gợi ý hoạt động dựa trên thời tiết và sở thích
def goi_y_hoat_dong(thoi_tiet, so_thich):
 hoat_dong_goi_y = []

 if “xem phim“ in so_thich and thoi_tiet == “mưa“:
  hoat_dong_goi_y.append(“xem phim tại nhà“)

 if “đi dạo“ in so_thich and thoi_tiet == “nắng“:
  hoat_dong_goi_y.append(“đi dạo trong công viên“)

 if “đọc sách“ in so_thich and (thoi_tiet == “nắng“ or thoi_tiet == “mưa“):
  hoat_dong_goi_y.append(“thư giãn với một cuốn sách“)

 # Nếu không có hoạt động phù hợp, gợi ý một hoạt động thư giãn khác
 if not hoat_dong_goi_y:
  hoat_dong_goi_y.append(“thử một sở thích mới tại nhà“)

 return hoat_dong_goi_y

# Gọi hàm và in ra gợi ý
hoat_dong_duoc_goi_y = goi_y_hoat_dong(thoi_tiet_hien_tai, so_thich_ca_nhan)
print(“Hoạt động gợi ý cho buổi tối của bạn:“)
for hoat_dong in hoat_dong_duoc_goi_y:
 print(“-“, hoat_dong)
```

Rõ ràng dựa trên thời tiết hiện tại là “mưa“ và sở thích cá nhân bao gồm “xem phim“, “đi dạo“, và “đọc sách“, hệ thống đã đưa ra gợi ý hoạt động thư giãn cho buổi tối:

Xem phim tại nhà: Vì bạn thích xem phim và thời tiết hiện tại là mưa, việc xem một bộ phim mới hoặc bộ phim yêu thích tại nhà sẽ là lựa chọn thú vị. Thư giãn với một cuốn sách: Với thời tiết mưa, việc đọc sách tại một góc yêu thích trong nhà có thể mang lại cảm giác thư giãn và dễ chịu.

Chúng ta thấy rằng hệ thống sử dụng logic or để kết hợp các sở thích và điều kiện thời tiết, từ đó đưa ra gợi ý phù hợp cho hoạt động thư giãn buổi tối.



Welcome cấu trúc if - else!

"Life is an if-else statement; sometimes you choose one path, sometimes another. The key is to make the best decision with the information you have at the moment."

Cấu trúc cơ bản

```
if điều_kiện:
# Khối lệnh thực hiện nếu điều kiện là True
else:
# Khối lệnh thực hiện nếu điều kiện là False
```

Trong đó:

if điều_kiện: Kiểm tra một điều kiện. Nếu điều kiện đúng (True), khối lệnh ngay sau if sẽ được thực thi.
else: Khối lệnh trong phần else sẽ được thực thi nếu điều kiện trong if không đúng (False).

cấu trúc if -else if -else (mở rộng)

```
if điều_kiện_1:
# Thực hiện khi điều_kiện_1 đúng
elif điều_kiện_2:
# Thực hiện khi điều_kiện_1 sai và điều_kiện_2 đúng
else:
# Thực hiện khi cả điều_kiện_1 và điều_kiện_2 đều sai
```

Câu chuyện tiếp cận:

Giả sử bạn muốn làm một món bánh và quyết định loại bánh sẽ phụ thuộc vào các nguyên liệu bạn có sẵn. Cụ thể, bạn sẽ quyết định làm bánh quy, bánh gato, hoặc bánh mì tùy thuộc vào nguyên liệu chính bạn có.

Điều kiện 1: Nếu bạn có bơ và đường (điều_kiện_1 đúng)
Khi bạn có đủ bơ và đường, bạn quyết định làm bánh quy. Bạn sẽ lấy bơ, đường, trộn chúng với bột mì, một chút muối và vanilla. Bạn nhào nặn hỗn hợp thành các viên nhỏ và nướng chúng. Kết quả là những chiếc bánh quy giòn tan, thơm mùi bơ và vanilla.

Điều kiện 2: Nếu không có đủ bơ và đường nhưng có trứng và sữa (điều_kiện_1 sai, nhưng điều_kiện_2 đúng)
Trong trường hợp bạn không có đủ bơ và đường nhưng lại có trứng và sữa, bạn quyết định làm bánh gato. Bạn đánh bông trứng với đường, sau đó từ từ thêm bột mì, sữa và một ít dầu ăn vào hỗn hợp. Bạn đổ hỗn hợp vào khuôn và nướng. Bánh gato sau khi nướng xong sẽ mềm mịn, xốp nhẹ.
Trường hợp khác: Nếu không có đủ các nguyên liệu trên (cả hai điều kiện trên đều không đúng)

Nếu bạn không có đủ nguyên liệu để làm bánh quy hoặc bánh gato, bạn quyết định sẽ làm bánh mì. Bạn chỉ cần bột mì, nước, men và một chút muối. Bạn trộn các nguyên liệu lại với nhau, nhào nặn và để bột nở. Sau đó, bạn đem nướng và sẽ có những ổ bánh mì thơm nức, vỏ giòn và ruột mềm.

Qua ví dụ này, bạn có thể thấy cách mà các cấu trúc điều khiển if, elif (tương đương với if else), và else giúp bạn quyết định loại bánh để làm dựa trên nguyên liệu bạn có. Điều này giúp bạn linh hoạt trong việc lựa chọn và sử dụng hiệu quả nguyên liệu mà không lãng phí.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH

Welcome cấu trúc if - else!

"Life is an if-else statement; sometimes you choose one path, sometimes another. The key is to make the best decision with the information you have at the moment."

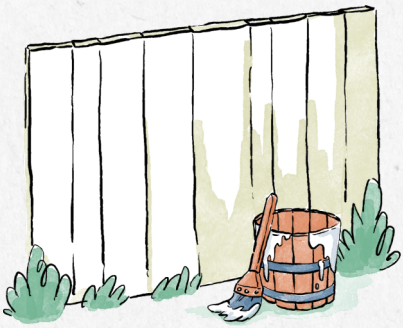
cấu trúc if -else if -else (mở rộng)

```
# Định nghĩa các nguyên liệu sẵn có
co_bo_va_duong = True  # Thay đổi giá trị này tùy vào nguyên liệu bạn có
co_trung_va_sua = False  # Thay đổi giá trị này tùy vào nguyên liệu bạn có

# Kiểm tra nguyên liệu và quyết định loại bánh để làm
if co_bo_va_duong:
    print("Bạn có đủ bơ và đường.")
    print("Hãy làm bánh quy! Dùng bơ, đường, bột mì, muối và vanilla để tạo ra những chiếc bánh quy thơm ngon.")
elif co_trung_va_sua:
    print("Bạn không có đủ bơ và đường nhưng có trứng và sữa.")
    print("Hãy làm bánh gato! Dùng trứng, đường, bột mì, sữa và dầu ăn để tạo ra chiếc bánh gato mềm mịn, xốp nhẹ.")
else:
    print("Bạn không có đủ nguyên liệu để làm bánh quy hoặc bánh gato.")
    print("Hãy làm bánh mì! Chỉ cần bột mì, nước, men và muối để tạo ra những ổ bánh mì thơm nức, vỏ giòn và ruột mềm.")
```

Đoạn mã này bắt đầu bằng cách định nghĩa hai biến co_bo_va_duong và co_trung_va_sua để biểu thị cho việc bạn có bơ và đường hay trứng và sữa. Bạn có thể thay đổi giá trị của hai biến này để phản ánh nguyên liệu bạn đang có.

Sau đó, đoạn mã sử dụng cấu trúc điều khiển if, elif và else để quyết định loại bánh nào bạn nên làm dựa trên nguyên liệu sẵn có. Mỗi phần của cấu trúc điều khiển này tương ứng với một lựa chọn khác nhau: làm bánh quy, bánh gato, hoặc bánh mì, và in ra hướng dẫn cơ bản cho loại bánh đó.







TÍNH ĐIỂM HÒA VỐN CHO SẢN PHẨM MỚI

Sau khi khám phá thế giới if-else, chúng ta sẽ dấn thân vào một cuộc phiêu lưu mới với một thách thức kinh tế thú vị: điểm hòa vốn. Hãy cùng nhau tìm hiểu xem điểm hòa vốn là gì và tại sao nó lại quan trọng trong việc ra quyết định cho các dự án kinh doanh nhé!

Khi bắt đầu một dự án hay phát triển sản phẩm mới, việc tính toán điểm hòa vốn (break-even point) là một bước quan trọng giúp bạn xác định số lượng sản phẩm cần bán để bù đắp chi phí và bắt đầu sinh lời.

Dưới đây là cách tiếp cận để tính điểm hòa vốn cho sản phẩm mới của bạn:



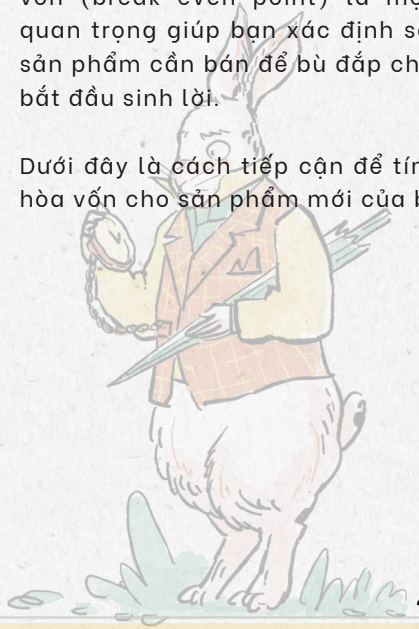
BƯỚC 1: XÁC ĐỊNH CHI PHÍ CỐ ĐỊNH (FIXED COSTS)

Chi phí cố định là những chi phí không thay đổi dù số lượng sản phẩm sản xuất có thay đổi. Ví dụ: tiền thuê mặt bằng, lương nhân viên, chi phí quảng cáo.

BƯỚC 2: XÁC ĐỊNH GIÁ BÁN SẢN PHẨM VÀ CHI PHÍ BIẾN ĐỔI (VARIABLE COSTS)

Giá Bán Sản Phẩm: Đây là giá mà bạn dự định bán sản phẩm cho khách hàng.

Chi Phí Biến Đổi: Chi phí này thay đổi dựa trên số lượng sản phẩm sản xuất. Ví dụ: nguyên liệu, tiền công trực tiếp.



Do happy



BƯỚC 3: TÍNH ĐÓNG GÓP BIÊN LỢI NHUẬN (CONTRIBUTION MARGIN

Đóng góp biên lợi nhuận được tính bằng cách trừ chi phí biến đổi khỏi giá bán sản phẩm.

Công thức:
Đóng Góp Biên Lợi Nhuận = Giá Bán Sản Phẩm – Chi Phí Biến Đổi

BƯỚC 4: TÍNH ĐIỂM HÒA VỐN (BREAK-EVEN POINT)

Cuối cùng, bạn sẽ tính điểm hòa vốn bằng cách chia chi phí cố định cho đóng góp biên lợi nhuận. Công thức:

Điểm Hòa Vốn = Chi Phí Cố Định / Đóng Góp Biên Lợi Nhuận





Ví Dụ Minh Họa

Giả sử chi phí cố định cho sản phẩm mới là 100,000 USD, giá bán dự kiến là 50 USD/sản phẩm, và chi phí biến đổi là 20 USD/sản phẩm.

Đóng Góp Biên Lợi Nhuận: 50-20=30
50-20=30 USD/sản phẩm

Điểm Hòa Vốn: 100,000÷30=3333.33
100,000÷30=3333.33 sản phẩm
Vậy bạn cần bán ít nhất khoảng 3334 sản phẩm để hòa vốn.

Be happy

Tính điểm hòa vốn không chỉ giúp chúng ta hiểu rõ hơn về chiến lược giá và mục tiêu doanh số, mà còn là cơ sở để đánh giá tính khả thi của sản phẩm và dự án nữa đó.

elegant

HOME DECOR STORY



ALEX HOME DECOR

Chúng mình sẽ cùng nhau đọc qua câu chuyện của bạn Alex để chúng mình có thể hình dung hơn cách mà Alex sẽ làm kinh doanh



Trong tâm trí của Alex, ý tưởng về việc khởi nghiệp đã lâu nay bắt đầu hình thành một kế hoạch cụ thể, như một câu chuyện logic mà mỗi chương đều tiếp nối và dẫn dắt lẫn nhau một cách mạch lạc.

Ngày đầu tiên, Alex ngồi trong văn phòng, trước một tờ giấy trắng và một ly cà phê nóng. Đây không chỉ là sự khởi đầu cho một ngày làm việc, mà còn là khởi đầu cho hành trình của một dự án kinh doanh. Alex biết rằng mọi thứ đều cần có lý do, một logic rõ ràng - giống như trong lập trình, mỗi quyết định đều cần dựa trên điều kiện và kết quả cụ thể.

Alex bắt đầu với việc liệt kê chi phí cố định. Như một công thức if-else, Alex tự hỏi: “Nếu tôi thuê mặt bằng này, chi phí sẽ là bao nhiêu? Nếu tôi tuyển thêm nhân viên, tôi cần trả bao nhiêu tiền lương?“. Mỗi câu hỏi đều dẫn đến một quyết định, và cuối cùng, tổng chi phí cố định mà Alex tính toán được là 100,000 USD.

Cuối cùng, Alex sử dụng logic “if-then-else“ để tính điểm hòa vốn: “Nếu mỗi sản phẩm mang lại 30 USD lợi nhuận, thì tôi cần bán bao nhiêu sản phẩm để bù đắp 100,000 USD chi phí cố định?“. Câu trả lời là khoảng 3334 sản phẩm.

Be happy



elegant HOME DECOR STORY

CÀI PYTHON CHO BÀI TOÁN HÒA VỐN CỦA ALEX

ALEX HOME DECOR

Với bài toán kinh doanh của Alex, vậy chúng mình sẽ cài đặt như thế nào?

Để tính điểm hòa vốn sử dụng if-else, chúng ta có thể thêm một số logic vào mã lệnh để xử lý các trường hợp khác nhau, chẳng hạn như khi đóng góp biên lợi nhuận bằng 0 (tránh chia cho 0) hoặc khi chi phí cố định bằng 0 (sản phẩm không cần bán bất kỳ đơn vị nào để hòa vốn). Dưới đây là ví dụ:

```
# Định nghĩa các biến theo ví dụ minh họa
chi_phi_co_dinh = 100000 # USD
gia_ban_san_pham = 50 # USD/sản phẩm
chi_phi_bien_doi = 20 # USD/sản phẩm

# Tính Đóng Góp Biên Lợi Nhuận
dong_gop_bien_loi_nhuan = gia_ban_san_pham - chi_phi_bien_doi

# Sử dụng if-else để xử lý các trường hợp đặc biệt
if dong_gop_bien_loi_nhuan <= 0:
 print("Lỗi: Đóng góp biên lợi nhuận không hợp lệ (<=0). Không thể tính điểm hòa vốn.")
elif chi_phi_co_dinh <= 0:
   print("Lưu ý: Không có chi phí cố định. Sản phẩm không cần bán bất kỳ đơn vị nào để hòa vốn.")
else:
  # Tính Điểm Hòa Vốn
  diem_hoa_von = chi_phi_co_dinh / dong_gop_bien_loi_nhuan
  print("Đóng Góp Biên Lợi Nhuận:", dong_gop_bien_loi_nhuan, "USD/sản phẩm")
  print("Điểm Hòa Vốn:", round(diem_hoa_von), "sản phẩm")
```

Trong mã này, if-else được sử dụng để kiểm tra các điều kiện trước khi thực hiện tính toán. Điều này giúp đảm bảo mã của bạn phản hồi một cách thích hợp đối với các tình huống không mong muốn, như khi không thể tính toán điểm hòa vốn do đóng góp biên lợi nhuận không hợp lệ hoặc không có chi phí cố định.

Chúng mình đã cùng đi qua if-else và cách sử dụng if-else sao cho hiệu quả và ứng dụng, trong hành trình học tập này, mình thấy một số điểm cần lưu ý của if-else:

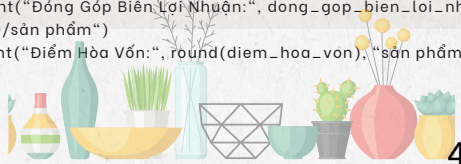
📝 Note Nhanh: Lưu Ý Khi Dùng If-Else trong Python

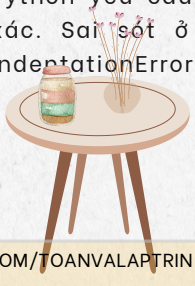
🎯 Điều Kiện Phải Rõ Ràng

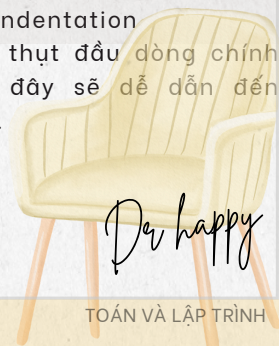
Đảm bảo rằng mỗi if đều dựa trên một điều kiện cụ thể và không gây nhầm lẫn.

🛠️ Cẩn Thận với Indentation

Python yêu cầu thụt đầu dòng chính xác. Sai sót ở đây sẽ dễ dẫn đến IndentationError.



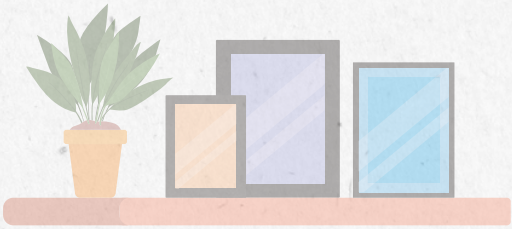




elegant

HOME DECOR STORY

CÀI PYTHON CHO BÀI TOÁN HÒA VỐN CỦA ALEX



🧐 Giảm Bớt If-Else Lồng Nhau
Đừng để code trở nên rối beng với quá nhiều if-else lồng nhau. Sử dụng elif hoặc chia nhỏ thành hàm.
🧐 Cẩn Thận với None
Kiểm tra giá trị None bằng is None hoặc is not None.
🔍 Dùng Ternary Operators Cho Đơn Giản
Cho phép viết if-else ngắn gọn: x = 5 if condition else 0.
📊 Kiểm Tra Hiệu Quả
Dùng if x in [a, b, c]: để kiểm tra nhiều giá trị một cách gọn lẹ.
Ghi chú này sẽ giúp bạn nắm vững cách sử dụng if-else trong Python một cách hiệu quả và tránh những lỗi phổ biến. Hãy coi đây như một cẩm nang nhỏ bên mình khi lập trình nhé!

Trong thế giới lập trình đa sắc màu, if-else giống như chiếc chìa khóa mở cánh cửa đến với quyết định logic, nhưng không phải lúc nào nó cũng là phép màu tốt nhất cho mọi bài toán. Hãy cùng khám phá những con đường thú vị khác, nơi if-else có thể nhường chỗ cho những phép thuật mới!

1. Điều Kỳ Diệu Của Dictionaries:
Tại sao phải mệt mỏi với mê cung if-else khi bạn có thể sử dụng dictionaries như một bản đồ kho báu, nơi từng từ khóa dẫn bạn đến giá trị mong muốn một cách tức thì?

2. Sự Thanh Lịch Của Ternary Operator:
Đối mặt với những quyết định đơn giản, hãy để ternary operator làm điều đó thay bạn, biến những dòng code dài ngoằng thành một câu chuyện ngắn gọn và thanh lịch.

3. Những Chuyến Phiêu Lưu Với Elif và Hàm:
Khi bản đồ if-else trở nên quá phức tạp, có lẽ đã đến lúc tách chúng ra thành những chuyến phiêu lưu nhỏ với elif hoặc những hàm riêng biệt, giúp câu chuyện của bạn dễ theo dõi hơn.

4. Khám Phá Kho Báu Của Thư Viện:
Trước khi đắm mình vào biển if-else, đừng quên khám phá kho báu của thư viện Python và các thư viện bên ngoài, nơi có thể có một hàm tiện ích hoàn hảo chờ bạn khám phá.

Dr happy



CÀI PYTHON CHO BÀI TOÁN HÒA VỐN CỦA ALEX

5. Bản Giao Hưởng Của OOP:
Trong thế giới lập trình hướng đối tượng, đa hình và kế thừa như bản giao hưởng mở ra một không gian mới, nơi if-else nhường chỗ cho sự diễn dịch tinh tế giữa các lớp và đối tượng.

6. Bản Đồ Đến Giải Pháp Mở Rộng:
Khi bài toán trở nên cồng kềnh, hãy mở bản đồ Strategy Pattern hoặc State Pattern, tìm ra lộ trình mở rộng và linh hoạt, tránh bẫy của if-else rối rắm.

Trong hành trình sáng tạo của mình, việc lựa chọn công cụ phù hợp là chìa khóa. If-else là một phần của kho báu lập trình, nhưng đừng ngần ngại khám phá và sử dụng những phép thuật mới để câu chuyện code của bạn thêm phần hấp dẫn và diệu kỳ!

Vậy chúng mình cũng đã hoàn thành xong ngày thứ 3 cùng nhau rồi, Aiko cảm thấy rất hạnh phúc vì đã cùng mình trải qua một ngày thứ 3 đầy thú vị và ý nghĩa. Mình hẹn gặp lại các bạn và học tập cùng nhau ở ngày tiếp theo nhé.



Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

Numpy



Numpy, hay "Numerical Python", là một câu chuyện về sự tiến hóa không ngừng trong lĩnh vực khoa học dữ liệu. Khởi nguồn từ những năm đầu của thế kỷ 21, Numpy ra đời như một phần của nỗ lực tạo ra một thư viện toán học mạnh mẽ cho Python, giúp ngôn ngữ này có thể cạnh tranh với các ngôn ngữ lập trình toán học khác như MATLAB.



3 dự án phân tích về kinh tế

1. Tên Dự Án 1: "Phân Tích Tác Động Kinh Tế của Dịch Bệnh COVID-19"
2. Dự Án 2: "Dự Báo Xu Hướng Thị Trường Chứng Khoán Sử Dụng Dữ Liệu Lịch Sử"
3. Dự Án 3: "Phân Tích Ảnh Hưởng Của Chính Sách Tiền Tệ Đến Lạm Phát"

Hôm nay đánh dấu một bước ngoặt đầy hứa hẹn trong hành trình học tập của chúng ta, khi chúng ta chính thức mở ra chương mới, đầy ắp những phát hiện và khám phá mới mẻ. Chúng ta không chỉ đơn thuần bước vào một phần học mới; chúng ta đang tiến vào một không gian học thuật đầy sắc màu, nơi thế giới số liệu và dữ liệu được khám phá và hiểu biết qua cái nhìn của Numpy - một công cụ mạnh mẽ trong lĩnh vực khoa học dữ liệu.

Khi chúng ta dấn thân vào việc tìm hiểu Numpy, ta không chỉ làm quen với những thao tác cơ bản về mảng và ma trận, mà còn bắt đầu mở rộng tầm nhìn của mình vào một lĩnh vực rộng lớn hơn: phân tích kinh tế. Numpy không chỉ giúp chúng ta xử lý dữ liệu một cách hiệu quả; nó còn giúp chúng ta 'đọc' dữ liệu, hiểu được những câu chuyện ẩn giấu đằng sau từng con số, từng chuỗi dữ liệu. Mỗi dãy số, mỗi bảng số liệu không chỉ là những thông tin khô khan, mà chúng còn là những mảnh ghép của một bức tranh kinh tế lớn hơn, nơi mỗi mảnh ghép đều có câu chuyện riêng để kể.

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Trong quá trình tìm hiểu này, chúng ta sẽ học cách sử dụng Numpy để khám phá, phân tích, và cuối cùng là 'kể' những câu chuyện kinh tế qua các chỉ số quan trọng như GDP, CPI. Chúng ta sẽ không chỉ xem xét những số liệu này qua các năm một cách đơn thuần, mà còn học cách phân tích chúng, tìm ra những xu hướng, những điểm nổi bật, và những hiểu biết sâu sắc hơn về cách thế giới kinh tế vận động và phát triển.

Vì vậy, hãy cùng nhau bước vào chương này với lòng hào hứng và khát khao khám phá, sẵn sàng đối mặt với những thách thức và cơ hội học tập mới. Chúng ta không chỉ học để 'biết', mà còn học để 'hiểu' và 'kể lại' - để mỗi số liệu, mỗi dãy số không chỉ là thông tin, mà còn trở thành những câu chuyện ý nghĩa, đầy cảm hứng.

Hãy cùng nhau bắt đầu một hành trình khám phá Numpy đầy thú vị với 7 bước mà bạn sẽ không thể quên!

Bước 1: Khai màn với Numpy

Khám phá Kỳ quan Numpy: Hãy lắng nghe những câu chuyện hấp dẫn về lịch sử của Numpy, khám phá những ứng dụng kỳ diệu và hiểu rõ tại sao Numpy lại trở thành người hùng không thể thiếu trong thế giới khoa học dữ liệu.

Lý do chọn Numpy: Đắm chìm trong thế giới lợi ích mà Numpy mang lại, từ tốc độ đến linh hoạt, khiến mọi nhà phân tích dữ liệu đều mê mẩn.

Bước 2: Đặt bệ phóng cho Numpy
Cài đặt không gian làm việc: Hãy cùng nhau thiết lập sân chơi cho Numpy trên máy tính của bạn, sử dụng pip hoặc conda, để chuẩn bị cho cuộc phiêu lưu kỳ thú sắp tới. Nếu bạn dùng Colab thì có lẽ và dễ dàng hơn nữa.

Bước 3: Những bước đầu tiên với Mảng NumpyTạo dựng thế giới mảng:
Học cách tạo ra những mảng Numpy từ danh sách Python đơn giản và khám phá những bí mật ẩn giấu sau những thuộc tính của chúng.
Ma thuật thao tác mảng: Thực hiện những phép thuật với lập chỉ mục, cắt mảng và biến hóa hình dạng mảng, mở ra cánh cửa tới những khám phá mới.

Do happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH.VN TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Trong quá trình tìm hiểu này, chúng ta sẽ học cách sử dụng Numpy để khám phá, phân tích, và cuối cùng là 'kể' những câu chuyện kinh tế qua các chỉ số quan trọng như GDP, CPI. Chúng ta sẽ không chỉ xem xét những số liệu này qua các năm một cách đơn thuần, mà còn học cách phân tích chúng, tìm ra những xu hướng, những điểm nổi bật, và những hiểu biết sâu sắc hơn về cách thế giới kinh tế vận động và phát triển.

Vì vậy, hãy cùng nhau bước vào chương này với lòng hào hứng và khát khao khám phá, sẵn sàng đối mặt với những thách thức và cơ hội học tập mới. Chúng ta không chỉ học để 'biết', mà còn học để 'hiểu' và 'kể lại' - để mỗi số liệu, mỗi dãy số không chỉ là thông tin, mà còn trở thành những câu chuyện ý nghĩa, đầy cảm hứng.

Hãy cùng nhau bắt đầu một hành trình khám phá Numpy đầy thú vị với 7 bước mà bạn sẽ không thể quên!

TỔNG QUAN 7 BƯỚC

Bước 1: Khai màn với Numpy

Khám phá Kỳ quan Numpy: Hãy lắng nghe những câu chuyện hấp dẫn về lịch sử của Numpy, khám phá những ứng dụng kỳ diệu và hiểu rõ tại sao Numpy lại trở thành người hùng không thể thiếu trong thế giới khoa học dữ liệu.

Lý do chọn Numpy: Đắm chìm trong thế giới lợi ích mà Numpy mang lại, từ tốc độ đến linh hoạt, khiến mọi nhà phân tích dữ liệu đều mê mẩn.

Bước 2: Đặt bệ phóng cho Numpy
Cài đặt không gian làm việc: Hãy cùng nhau thiết lập sân chơi cho Numpy trên máy tính của bạn, sử dụng pip hoặc conda, để chuẩn bị cho cuộc phiêu lưu kỳ thú sắp tới. Nếu bạn dùng Colab thì có lẽ và dễ dàng hơn nữa.

Bước 3: Những bước đầu tiên với Mảng NumpyTạo dựng thế giới mảng:
Học cách tạo ra những mảng Numpy từ danh sách Python đơn giản và khám phá những bí mật ẩn giấu sau những thuộc tính của chúng.
Ma thuật thao tác mảng: Thực hiện những phép thuật với lập chỉ mục, cắt mảng và biến hóa hình dạng mảng, mở ra cánh cửa tới những khám phá mới.

Do happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Bước 4: Numpy và Bản giao hưởng Kinh Tế
Hòa nhạc GDP và CPI: Hãy bước vào buổi hòa nhạc của các chỉ số kinh tế, nơi GDP và CPI là những nghệ sĩ chính, và tìm hiểu về sức mạnh của họ trong bản giao hưởng kinh tế.
Nhảy múa cùng Dữ liệu: Sử dụng Numpy như là đôi giày nhảy múa lý tưởng, giúp bạn dễ dàng tải, xử lý và phân tích dữ liệu GDP và CPI, khám phá những điệu nhảy qua các năm.

Bước 5: Nghệ thuật Phân Tích Dữ Liệu

Chế ngự Dữ liệu: Trở thành nhà giả kim của dữ liệu, học cách biến dữ liệu thô thành vàng mười bằng cách làm sạch và chuẩn bị chúng cho quá trình phân tích.
Phân tích là Bức tranh: Vẽ nên những bức tranh phân tích dữ liệu sử dụng các hàm tính toán của Numpy, từ đó tìm ra những xu hướng và bí mật ẩn giấu trong dữ liệu.

Bước 6: Phiêu lưu Tự Học Numpy
Khám phá Bí mật Numpy: Dấn thân vào cuộc phiêu lưu tự học, khám phá những chức năng và kỹ thuật nâng cao trong Numpy, mở ra những cánh cửa mới của kiến thức.

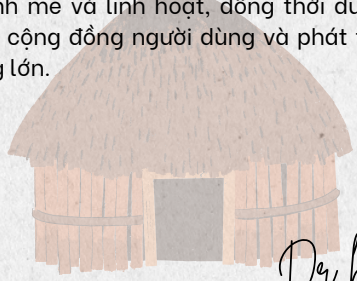
Bước 7: Dự Án Dấu Ấn Cá Nhân
Cùng tạo dự án của riêng bạn: Áp dụng tất cả những gì bạn đã học vào một dự án cá nhân, sử dụng dữ liệu kinh tế thực tế để tạo ra những phân tích có ý nghĩa.

Đi sâu từng bước:

Bạn đã chuẩn bị sẵn sàng cho Bước 1 chưa? Hãy hít một hơi thật sâu, nạp đầy năng lượng, để chúng ta cùng nhau khởi đầu một ngày mới.

BƯỚC 1: Khai màn với Numpy
Numpy, hay "Numerical Python", là một câu chuyện về sự tiến hóa không ngừng trong lĩnh vực khoa học dữ liệu. Khởi nguồn từ những năm đầu của thế kỷ 21, Numpy ra đời như một phần của nỗ lực tạo ra một thư viện toán học mạnh mẽ cho Python, giúp ngôn ngữ này có thể cạnh tranh với các ngôn ngữ lập trình toán học khác như MATLAB.

Phát triển từ thư viện Numeric, tiền thân đầu tiên của Numpy, và sau đó là Numarray - một phiên bản cải tiến nhưng không tương thích với Numeric, Numpy được Travis Oliphant tạo ra vào năm 2005 như một sự hợp nhất của hai thư viện này, mang lại sự tương thích và hiệu suất tối ưu. Numpy kế thừa và tinh chỉnh các ưu điểm của cả hai, đặt nền móng cho một thư viện xử lý số liệu đa chiều mạnh mẽ và linh hoạt, đồng thời duy trì một cộng đồng người dùng và phát triển rộng lớn.



Do happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Lịch sử phát triển của Numpy không chỉ là câu chuyện về sự tiến bộ kỹ thuật, mà còn là minh chứng cho sự hợp tác và chia sẻ kiến thức trong cộng đồng mã nguồn mở. Các đóng góp từ hàng nghìn nhà phát triển trên khắp thế giới đã làm cho Numpy không ngừng phát triển, trở thành công cụ không thể thiếu trong toolbox của bất kỳ nhà khoa học dữ liệu nào.

Qua từng giai đoạn phát triển, từ Numeric đến Numpy, từ một công cụ xử lý mảng đơn giản đến một hệ sinh thái toàn diện cho khoa học dữ liệu, Numpy đã trở thành chứng nhân và người đồng hành cho hàng loạt bước tiến vượt bậc trong lĩnh vực này. Nó không chỉ là một công cụ, mà còn là một phần của lịch sử phát triển của khoa học dữ liệu, khắc sâu dấu ấn của mình qua mỗi ứng dụng, mỗi dự án mà nó góp phần tạo nên.

Numpy, với khả năng xử lý mảng và ma trận cực kỳ mạnh mẽ, đã trở thành một trong những công cụ không thể thiếu trong vũ trụ khoa học dữ liệu. Sức mạnh của Numpy không chỉ nằm ở tốc độ xử lý dữ liệu nhanh chóng nhờ vào việc tối ưu hóa các thao tác toán học phức tạp, mà còn ở tính linh hoạt và khả năng tương thích với nhiều thư viện khác trong hệ sinh thái Python.

Đi sâu từng bước:

Numpy cung cấp một hệ thống đa chiều mảng số liệu, cho phép các nhà khoa học dữ liệu thực hiện các phép tính toán học và thống kê một cách chính xác và hiệu quả trên dữ liệu lớn, từ đó rút ra các insight quan trọng. Đặc biệt, khả năng tương thích cao của Numpy với các thư viện máy học và trí tuệ nhân tạo như TensorFlow và PyTorch đã làm tăng thêm giá trị của nó, khiến Numpy trở thành một phần không thể tách rời trong bất kỳ dự án khoa học dữ liệu nào, từ phân tích dữ liệu đến xây dựng mô hình máy học.

Ngoài ra, Numpy còn hỗ trợ các tính năng tiên tiến như broadcasting và vectorization, giúp giảm thiểu việc sử dụng vòng lặp trong Python, qua đó tối ưu hóa hiệu suất và tốc độ xử lý dữ liệu. Điều này không chỉ giúp cải thiện đáng kể hiệu suất tính toán mà còn làm cho code trở nên gọn gàng, dễ đọc hơn.

Vì vậy, Numpy không chỉ là công cụ mạnh giúp đơn giản hóa và tăng tốc độ phân tích dữ liệu, mà còn là một nguồn cảm hứng cho các nhà khoa học dữ liệu, giúp họ mở rộng khả năng sáng tạo và khám phá những phương pháp mới trong việc giải quyết các vấn đề phức tạp. Trong thế giới khoa học dữ liệu ngày càng phát triển, Numpy chính là người bạn đồng hành đáng tin cậy, góp phần tạo nên nền tảng vững chắc cho những khám phá và đột phá tiếp theo.

Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Dưới đây là dòng thời gian phác họa quá trình phát triển của Numpy, từ những ngày đầu tiên đến vị thế nổi bật hiện tại trong cộng đồng khoa học dữ liệu:

1995: Numeric, thư viện đầu tiên cho Python có khả năng xử lý mảng đa chiều, được giới thiệu bởi Jim Hugunin. Đây là bước khởi đầu cho việc tính toán khoa học trong Python.

2001: Một thư viện mới, Numarray, được phát triển bởi Perry Greenfield nhằm cải thiện và mở rộng các khả năng của Numeric, đặc biệt là cho các mảng lớn và tính toán đa chiều.

2005: Travis Oliphant, một nhà phát triển quan trọng của cả Numeric và Numarray, quyết định hợp nhất hai thư viện này để tạo ra Numpy. Mục tiêu là kết hợp sức mạnh và khả năng của cả hai thư viện trong một gói duy nhất, giúp giảm bớt sự phân mảnh và tạo ra một nền tảng thống nhất cho tính toán khoa học trong Python.

2006: Phiên bản chính thức đầu tiên của Numpy, Numpy 1.0, được phát hành. Điều này đánh dấu một bước ngoặt quan trọng, thiết lập Numpy như một công cụ cốt lõi trong hệ sinh thái Python cho tính toán khoa học.

Từ 2006 đến nay: Numpy tiếp tục phát triển với sự đóng góp của một cộng đồng lớn các nhà khoa học, kỹ sư và nhà phát triển. Các phiên bản mới được phát hành đều đặn, mỗi lần thêm vào những cải tiến về hiệu suất, tính năng mới và sự tương thích với các thư viện khác.

2010s: Numpy trở thành nền tảng cho nhiều thư viện khác trong hệ sinh thái khoa học dữ liệu của Python, bao gồm pandas, matplotlib, SciPy, scikit-learn và các thư viện máy học/deep learning như TensorFlow và PyTorch.

2020: Tính đến năm 2020 và những năm tiếp theo, Numpy vẫn tiếp tục là một phần không thể thiếu trong bất kỳ dự án khoa học dữ liệu nào, được công nhận rộng rãi về sự linh hoạt, hiệu suất và tính ổn định.

Quá trình phát triển của Numpy không chỉ phản ánh sự tiến triển trong lĩnh vực khoa học dữ liệu và tính toán khoa học, mà còn thể hiện tinh thần cộng tác và chia sẻ kiến thức trong cộng đồng mã nguồn mở.

Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

Bước 2: Đặt bệ phóng cho Numpy

Để cài đặt và sử dụng Numpy trên Google Colab, bạn thực ra không cần phải thực hiện bất kỳ bước cài đặt nào vì Numpy đã được cài đặt sẵn trong môi trường Google Colab. Dưới đây là các bước để bạn bắt đầu làm việc với Numpy trên Google Colab:

Truy cập Google Colab: Mở trình duyệt web và truy cập vào Google Colab.

Tạo mới hoặc mở sổ ghi chép Colab: Nếu bạn muốn bắt đầu với một sổ ghi chép mới, chỉ cần chọn "New Notebook". Bạn cũng có thể mở một sổ ghi chép đã tồn tại từ Google Drive của mình.

Nhập Numpy: Để bắt đầu sử dụng Numpy, bạn chỉ cần nhập thư viện vào sổ ghi chép của mình. Điều này có thể được thực hiện bằng cách sử dụng câu lệnh import trong một ô code như sau:

```
import numpy as np
```

Điều này sẽ nhập Numpy với bí danh np, giúp bạn có thể truy cập dễ dàng các chức năng của Numpy qua tiền tố np.

Bắt đầu sử dụng Numpy: Bây giờ bạn có thể bắt đầu sử dụng Numpy để tạo mảng, thực hiện các phép toán và xử lý dữ liệu. Ví dụ, để tạo một mảng Numpy, bạn có thể sử dụng:

```
arr = np.array([1, 2, 3, 4, 5])
print(arr)
```

Lưu và chia sẻ sổ ghi chép: Khi bạn đã hoàn thành công việc của mình, bạn có thể lưu sổ ghi chép và chia sẻ với người khác. Google Colab tự động lưu sổ ghi chép của bạn vào Google Drive.

Như vậy, bạn đã sẵn sàng để làm việc với Numpy trên Google Colab mà không cần phải lo lắng về việc cài đặt hay cấu hình môi trường.

Bước 3: Những bước đầu tiên với Mảng Numpy: Tạo dựng thế giới mảng:

Numpy không chỉ là một thư viện trong Python, nó là một hộp công cụ đầy ắp phép màu cho bất kỳ ai đam mê khoa học dữ liệu. Hãy cùng nhau khám phá thế giới diệu kỳ này:

1. Mảng Ma Thuật (Numpy Arrays)
Phép Tạo Mảng: Như một pháp sư, bạn có thể biến danh sách Python thành mảng Numpy chỉ bằng một lời thần chú np.array(). Mảng Numpy có thể là dải ngân hà một chiều, mặt phẳng hai chiều hoặc thậm chí là không gian đa chiều.
Bí Mật Của Mảng: Mỗi mảng Numpy ẩn chứa những bí mật như shape (hình dạng vũ trụ của nó), size (tổng số hành tinh), dtype (chất liệu cấu thành), và ndim (số lượng chiều không gian).Võ Thuật Mảng: Numpy dạy bạn cách để vũ đạo với mảng, biến hóa chúng qua reshape, kết hợp những vũ trụ qua concatenate, hoặc chia cắt chúng bằng split.

Dr happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

```
import numpy as np

# Phép Tạo Mảng
# Dải ngân hà một chiều
dai_ngan_ha = np.array([2, 4, 6, 8, 10])

# Mặt phẳng hai chiều
mat_phang = np.array([[1, 3, 5], [7, 9, 11]])

# Không gian đa chiều
khong_gian = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])

# Bí Mật Của Mảng
print("Hình dạng dải ngân hà:", dai_ngan_ha.shape)
print("Tổng số hành tinh trong dải ngân hà:", dai_ngan_ha.size)
print("Chất liệu cấu thành dải ngân hà:", dai_ngan_ha.dtype)
print("Chiều không gian của dải ngân hà:", dai_ngan_ha.ndim)

# Võ Thuật Mảng
# Biến hóa dải ngân hà qua reshape
dai_ngan_ha_reshaped = dai_ngan_ha.reshape(5, 1)
print("Dải ngân hà sau khi biến hình:", dai_ngan_ha_reshaped)

# Kết hợp những vũ trụ qua concatenate
vu_tru_moi = np.concatenate((dai_ngan_ha, dai_ngan_ha))
print("Vũ trụ mới sau khi kết hợp:", vu_tru_moi)

# Chia cắt vũ trụ bằng split
phan_chia_vu_tru, _ = np.split(vu_tru_moi, [5])
print("Vũ trụ sau khi phân chia:", phan_chia_vu_tru)
```

2. Khám Phá Vũ Trụ (Truy Cập và Slicing)
Dấu Chân Của Nhà Thám Hiểm: Dùng chỉ số để lần theo dấu chân trên bề mặt mảng, mỗi bước chân là một khám phá.
Cánh Cửa Thời Gian: Slicing cho phép bạn mở cánh cửa, bước vào một phần của vũ trụ mảng, và khám phá những điều kỳ diệu ẩn sau.

```
# Nhập khẩu Numpy
import numpy as np

# Tạo mảng Numpy đa chiều
vung_dat_ky_bí = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

# Dấu Chân Của Nhà Thám Hiểm: Dùng chỉ số để khám phá
# Chọn một phần tử bí ẩn - vị trí của viên ngọc thứ 6
vien_ngoc_bi_an = vung_dat_ky_bí[1, 1] # Hàng thứ 2, cột thứ 2
print("Viên ngọc bí ẩn:", vien_ngoc_bi_an)

# Cánh Cửa Thời Gian: Slicing để khám phá
# Khám phá một phần của vùng đất kỳ bí - hàng thứ 2 và 3, cột thứ 1 và 2
kho_bau_bi_mat = vung_dat_ky_bí[1:3, 0:2]
print("Kho báu bí mật:\n", kho_bau_bi_mat)

# Bonus: Khám phá hàng đầu tiên sử dụng slicing
hang_dau_tien = vung_dat_ky_bí[0, :]
# Tất cả các cột của hàng thứ nhất
print("Hàng đầu tiên:", hang_dau_tien)

# Bonus: Khám phá cột thứ ba sử dụng slicing
cot_thu_ba = vung_dat_ky_bí[:, 2] #
```


CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

3. Bản Giao Hưởng Đại Số (Đại Số Tuyến Tính)

Nhạc Cụ Ma Trận:

Numpy là nhạc trưởng, dùng ma trận như những nhạc cụ, tạo nên bản giao hưởng của phép toán cộng, trừ, nhân và chia.
Hòa Âm Ánh Sáng: Numpy thể hiện nghệ thuật qua định thức, hàng số, ma trận nghịch đảo và giải điệu của các hệ phương trình tuyến tính.

Hãy biến Numpy thành nhạc trưởng của chúng ta, điều khiển các nhạc cụ ma trận tạo nên bản giao hưởng tuyệt vời của toán học. Dưới đây là ví dụ minh họa cho "Nhạc Cụ Ma Trận" và "Hòa Âm Ánh Sáng":

```
import numpy as np

# Tạo hai ma trận để thực hiện bản giao hưởng toán học
ma_tran_A = np.array([[1, 2], [3, 4]])
ma_tran_B = np.array([[5, 6], [7, 8]])

# Nhạc Cụ Ma Trận
# Cộng ma trận - tiếng vang của sự gặp gỡ
giao_huong_cong = ma_tran_A + ma_tran_B
print("Giao hưởng cộng:\n", giao_huong_cong)

# Trừ ma trận - vẻ đẹp của sự chia ly
giao_huong_tru = ma_tran_A - ma_tran_B
print("Giao hưởng trừ:\n", giao_huong_tru)

# Nhân ma trận - sự kết hợp hoàn hảo
giao_huong_nhan = np.dot(ma_tran_A, ma_tran_B)
print("Giao hưởng nhân:\n", giao_huong_nhan)

# Chia ma trận - sự phân chia hài hòa
# Lưu ý: Trong đại số tuyến tính, không có phép "chia" ma trận trực tiếp, nhưng có thể nhân với ma trận nghịch đảo
ma_tran_B_inv = np.linalg.inv(ma_tran_B)
giao_huong_chia = np.dot(ma_tran_A, ma_tran_B_inv)
print("Giao hưởng 'chia':\n", giao_huong_chia)

# Hòa Âm Ánh Sáng
# Định thức - bí mật của vũ trụ
dinh_thuc_A = np.linalg.det(ma_tran_A)
print("Định thức của A:", dinh_thuc_A)

# Hàng số - sức mạnh tiềm ẩn
hang_so_A = np.linalg.eigvals(ma_tran_A)
print("Hàng số của A:", hang_so_A)

# Ma trận nghịch đảo - gương mặt phản chiếu
nghich_dao_A = np.linalg.inv(ma_tran_A)
print("Ma trận nghịch đảo của A:\n", nghich_dao_A)

# Giải điệu của hệ phương trình tuyến tính
# Giả sử ma_tran_A * x = ma_tran_B
# Ta cần tìm x
x = np.linalg.solve(ma_tran_A, ma_tran_B[:,0])# Chỉ lấy cột đầu tiên của B để giải
print("Giải của hệ phương trình A*x = B:\n", x)
```

Trong ví dụ trên, chúng ta đã sử dụng Numpy để thực hiện các phép toán cộng, trừ và nhân giữa hai ma trận ma_tran_A và ma_tran_B. Chúng ta cũng khám phá cách tính định thức, tìm các giá trị riêng (hàng số), tạo ma trận nghịch đảo, và giải một hệ phương trình tuyến tính sử dụng ma trận.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH

Numpy | GDP | CPI

4. Thí Nghiệm Khoa Học (Tính Toán Khoa Học)
Hộp Phép Thuật: Mỗi hàm toán học trong Numpy giống như một dụng cụ thần kỳ trong tay nhà phù thủy, từ sum đến sqrt, từ log đến exp.
Thư Viện Bí Mật: Các hàm thống kê mở ra cánh cửa thư viện bí mật, nơi giấu kín bí mật về trung bình, trung vị, độ lệch chuẩn và phương sai.

Chúng ta sẽ biến Numpy thành một hộp phép thuật, khám phá sức mạnh của các hàm toán học và thống kê. Dưới đây là ví dụ về "Hộp Phép Thuật" và "Thư Viện Bí Mật":

```
import numpy as np

# Tạo một mảng Numpy đầy ma thuật
mang_ma_thuat = np.array([1, 2, 3, 4, 5])

# Hộp Phép Thuật
# Tổng hợp - phép thuật cơ bản nhất
tong_hop = np.sum(mang_ma_thuat)
print("Tổng hợp:", tong_hop)

# Căn bậc hai - tiết lộ sức mạnh ẩn giấu
can_bac_hai = np.sqrt(mang_ma_thuat)
print("Căn bậc hai:", can_bac_hai)

# Logarit - khám phá chiều sâu của thời gian
logarit = np.log(mang_ma_thuat)
print("Logarit:", logarit)

# Hàm mũ - tiếp cận cánh cửa của vũ trụ
ham_mu = np.exp(mang_ma_thuat)
print("Hàm mũ:", ham_mu)

# Thư Viện Bí Mật
# Trung vị - cân bằng hoàn hảo giữa ánh sáng và bóng tối
trung_vi = np.median(mang_ma_thuat)
print("Trung vị:", trung_vi)

# Độ lệch chuẩn - đo lường sự hỗn loạn
do_lech_chuan = np.std(mang_ma_thuat)
print("Độ lệch chuẩn:", do_lech_chuan)

# Phương sai - bí mật của sự đa dạng
phuong_sai = np.var(mang_ma_thuat)
print("Phương sai:", phuong_sai)
```

Trong ví dụ này, mang_ma_thuat chứa một chuỗi số từ 1 đến 5. Chúng ta sử dụng "Hộp Phép Thuật" để thực hiện các phép toán toán học như tính tổng, căn bậc hai, logarit và hàm mũ trên mảng này. Mỗi hàm toán học tiết lộ một khía cạnh khác nhau của dữ liệu, như là mở cánh cửa đến những khám phá mới.

Tiếp theo, "Thư Viện Bí Mật" của chúng ta tiết lộ những sự thật ẩn sâu trong dữ liệu thông qua các hàm thống kê như tính trung bình, trung vị, độ lệch chuẩn và phương sai. Những hàm thống kê này giúp chúng ta hiểu rõ hơn về tính chất và xu hướng trung tâm của dữ liệu, cũng như mức độ phân tán của dữ liệu xung quanh giá trị trung bình của nó.

Qua "Hộp Phép Thuật" và "Thư Viện Bí Mật", Numpy không chỉ là công cụ mạnh mẽ trong việc xử lý và phân tích dữ liệu, mà còn mở ra một thế giới mới với vô số khả năng khám phá và hiểu biết sâu sắc về dữ liệu.

Dr happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

```
import numpy as np

# Tạo một mảng một chiều như một dải ngân hà
dai_ngan_ha = np.array([1, 2, 3])

# Tạo một số vũ trụ màu sắc - một mảng một chiều khác
vu_tru_mau_sac = np.array([4])

# Khiêu vũ giữa các vì sao: Broadcasting
# Numpy tự động "mở rộng" vu_tru_mau_sac để nó có thể phù hợp với kích thước của dai_ngan_ha
vien_ngoc = dai_ngan_ha + vu_tru_mau_sac
print("Viên ngọc sau khiêu vũ:", vien_ngoc)

# Một ví dụ phức tạp hơn với mảng 2D
mat_phang = np.array([[1, 2, 3], [4, 5, 6]])

# Broadcasting một mảng một chiều với mảng hai chiều
kho_bau_bi_an = mat_phang + dai_ngan_ha
print("Kho báu bí mật sau khiêu vũ:\n", kho_bau_bi_an)
```

Trong ví dụ đầu tiên, dai_ngan_ha là một mảng một chiều và vu_tru_mau_sac là một mảng chứa chỉ một phần tử. Khi chúng "khiêu vũ" cùng nhau thông qua phép toán cộng, Numpy tự động "mở rộng" vu_tru_mau_sac để phép toán có thể thực hiện trên mỗi phần tử của dai_ngan_ha.

Trong ví dụ thứ hai, mat_phang là một mảng hai chiều, và dai_ngan_ha là một mảng một chiều. Khi thực hiện phép cộng, Numpy "mở rộng" dai_ngan_ha theo chiều dọc để nó phù hợp với mỗi

6. Bản Đồ Kho Báu (Boolean Indexing và Fancy Indexing)

Xác Định Vị Trí Kho Báu: Boolean Indexing giống như việc bạn có bản đồ kho báu, chỉ rõ những kho báu ẩn giấu trong mảng. Fancy Indexing giúp bạn nhanh chóng chọn ra những viên ngọc quý giá từ vũ trụ mảng của mình.

Boolean Indexing và Fancy Indexing trong Numpy giúp bạn trở thành một nhà thám hiểm tài ba, khám phá và chọn lọc kho báu từ vũ trụ mảng của mình. Dưới đây là cách bạn có thể sử dụng hai kỹ thuật này để xác định vị trí và thu thập kho báu:

Xác Định Vị Trí Kho Báu với Boolean Indexing

```
import numpy as np

# Tạo một mảng chứa kho báu ẩn giấu
kho_bau = np.array([10, 23, 45, 33, 67, 89, 42, 35, 21])

# Sử dụng bản đồ kho báu (Boolean Indexing) để tìm những kho báu lớn hơn 40
ban_do_kho_bau = kho_bau > 40 # Bản đồ này sẽ chỉ ra True tại những vị trí có kho báu lớn hơn 40
kho_bau_an_giau = kho_bau[ban_do_kho_bau]
print("Kho báu ẩn giấu:", kho_bau_an_giau)
```

Trong ví dụ này, kho_bau là một mảng chứa các giá trị khác nhau, tượng trưng cho những kho báu. Boolean Indexing được sử dụng để tạo ra một "bản đồ kho báu" (ban_do_kho_bau), chỉ rõ những kho báu lớn hơn 40. Khi áp dụng bản đồ này lên mảng kho_bau, bạn sẽ thu được kho_bau_an_giau chỉ chứa những kho báu mà bạn đang tìm kiếm.

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

Chọn Những Viên Ngọc Quý Giá với Fancy Indexing

python Copy code

```
# Tạo một mảng mới chứa những viên ngọc
vien_ngoc = np.array([101, 102, 103, 104, 105, 106, 107, 108, 109])

# Sử dụng Fancy Indexing để chọn những viên ngọc quý giá (vị trí 2, 4, 6)
vi_tri_ngoc_quy_gia = [2, 4, 6]
ngoc_quy_gia = vien_ngoc[vi_tri_ngoc_quy_gia]
print("Những viên ngọc quý giá:", ngoc_quy_gia)
```

Ở đây, vien_ngoc là một mảng chứa các giá trị, mỗi giá trị tượng trưng cho một viên ngọc. Fancy Indexing cho phép bạn trực tiếp chọn ra những viên ngọc dựa trên vị trí cụ thể của chúng (vi_tri_ngoc_quy_gia). Kết quả ngoc_quy_gia chứa chính xác những viên ngọc bạn muốn thu thập từ vũ trụ mảng.

Cả Boolean Indexing và Fancy Indexing đều là công cụ vô giá trong việc thăm dò và trích xuất những thông tin quý giá từ dữ liệu của bạn, giúp bạn nắm bắt được những kho báu ẩn giấu với sự chính xác và hiệu quả.

Bước 4: Numpy và Bản giao hưởng Kinh Tế

Đối với mục "Numpy và Bản giao hưởng Kinh Tế", bạn sẽ thực hiện các bước sau để khám phá và phân tích dữ liệu kinh tế, đặc biệt là GDP và CPI, sử dụng Numpy:

1. Thu Thập Dữ Liệu

Bước đầu tiên là thu thập dữ liệu về GDP (Tổng Sản Phẩm Nội Địa) và CPI (Chỉ Số Giá Tiêu Dùng) từ các nguồn đáng tin cậy như trang web của các tổ chức quốc tế (ví dụ: Ngân hàng Thế giới, IMF, hoặc các cơ quan thống kê quốc gia).

2. Chuẩn Bị Dữ Liệu

Sau khi thu thập, bạn cần làm sạch và chuẩn bị dữ liệu, chẳng hạn như loại bỏ giá trị NaN, chuyển đổi dữ liệu sang định dạng thích hợp, và cấu trúc lại dữ liệu để phù hợp cho việc phân tích.

3. Phân Tích Dữ Liệu

Sử dụng Numpy để thực hiện phân tích dữ liệu. Bạn có thể bắt đầu với việc tính các chỉ số thống kê cơ bản (trung bình, trung vị, độ lệch chuẩn, v.v.) cho dữ liệu GDP và CPI, sau đó tiến hành các phân tích phức tạp hơn như so sánh tốc độ tăng trưởng GDP qua các năm, hoặc phân tích sự liên quan giữa GDP và CPI.

4. Trực Quan Hóa Dữ Liệu

Mặc dù Numpy không phải là thư viện dành cho trực quan hóa, nhưng bạn có thể sử dụng Numpy để chuẩn bị dữ liệu trước khi trực quan hóa nó với các thư viện như Matplotlib hoặc Seaborn. Việc này giúp hiển thị dữ liệu một cách trực quan thông qua các biểu đồ như biểu đồ đường, biểu đồ cột, để dễ dàng nhận biết các xu hướng và mối liên hệ.

Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Dưới đây là một ví dụ mã giả về cách bạn có thể sử dụng Numpy để phân tích dữ liệu GDP và CPI:

Dưới đây là mã giả tổng quan hóa cách thực hiện 4 bước "Numpy và Bản giao hưởng Kinh Tế" bằng cách giả định rằng dữ liệu đã được thu thập và chuẩn bị:

```
import numpy as np
import matplotlib.pyplot as plt
```

```
# Bước 1 & 2: Thu Thập và Chuẩn Bị Dữ Liệu
# Giả định rằng chúng ta đã có dữ liệu GDP và CPI đã được thu thập và làm sạch
# Dữ liệu GDP qua 5 năm (giả định)
gdp = np.array([45000, 47000, 49000, 51000, 53000])# Dữ liệu GDP
# Dữ liệu CPI qua 5 năm (giả định)
cpi = np.array([100, 103, 106, 109, 112])
# Dữ liệu CPI
years = np.array([2016, 2017, 2018, 2019, 2020])# Các năm tương ứng
```

```
# Bước 3: Phân Tích Dữ Liệu
# Tính tốc độ tăng trưởng GDP hàng năm
gdp_growth_rate = np.diff(gdp) / gdp[:-1] * 100
# Tính tỷ lệ lạm phát hàng năm dựa trên CPI
cpi_inflation_rate = np.diff(cpi) / cpi[:-1] * 100

# Bước 4: Trực Quan Hóa Dữ Liệu
# Trực quan hóa dữ liệu GDP
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1) # 1 hàng, 2 cột, vị trí thứ nhất
plt.plot(years, gdp, marker='o', linestyle='-', color='b')
plt.title('GDP qua các năm')
plt.xlabel('Năm')
plt.ylabel('GDP')

# Trực quan hóa dữ liệu CPI
plt.subplot(1, 2, 2) # 1 hàng, 2 cột, vị trí thứ hai
plt.plot(years, cpi, marker='s', linestyle='--', color='r')
plt.title('CPI qua các năm')
plt.xlabel('Năm')
plt.ylabel('CPI')

plt.tight_layout() # Điều chỉnh layout để không bị chồng chéo
plt.show()

# Trực quan hóa tốc độ tăng trưởng GDP và tỷ lệ lạm phát CPI
plt.figure(figsize=(10, 5))
plt.plot(years[1:], gdp_growth_rate, marker='o', label='Tăng trưởng GDP (%)', color='green')
plt.plot(years[1:], cpi_inflation_rate, marker='s', label='Lạm phát CPI (%)', linestyle='--', color='purple')
plt.title('Tốc độ tăng trưởng GDP và tỷ lệ lạm phát CPI qua các năm')
plt.xlabel('Năm')
plt.legend()
plt.show()
```

Bước 5: Nghệ thuật Phân Tích Dữ Liệu

Để thực hiện Bước 5 "Nghệ thuật Phân Tích Dữ Liệu", bạn cần đi sâu vào quá trình chế ngự dữ liệu và biến nó thành thông tin hữu ích thông qua quá trình làm sạch, chuẩn bị, và phân tích. Dưới đây là cách bạn có thể áp dụng Numpy để đạt được mục tiêu này:

Chế Ngự Dữ Liệu
Đầu tiên, bạn cần "chế ngự" dữ liệu thô của mình, loại bỏ những phần không cần thiết hoặc sai lệch, và chuẩn bị chúng cho quá trình phân tích:

CONTACT: 084 6120 2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

```
import numpy as np

# Giả sử dữ liệu thô của bạn chứa cả giá trị hợp lệ và giá trị NaN (Not a Number)
du_lieu_tho = np.array([45, 67, np.nan, 90, np.nan, 33, 56])

# Lọc ra những giá trị hợp lệ (không phải NaN)
du_lieu_sach = du_lieu_tho[~np.isnan(du_lieu_tho)]
print("Dữ liệu sau khi làm sạch:", du_lieu_sach)
```

Trong ví dụ này, chúng ta sử dụng một mảng Numpy chứa cả giá trị hợp lệ và giá trị NaN. Chúng ta lọc ra những giá trị NaN để chỉ giữ lại dữ liệu hợp lệ, chuẩn bị cho phân tích tiếp theo.

Phân Tích Là Bức Tranh

Sau khi dữ liệu đã được làm sạch, bước tiếp theo là phân tích dữ liệu để khám phá các xu hướng và thông tin ẩn giấu:

python Copy code

```
# Tính các chỉ số thống kê cơ bản để phân tích dữ liệu
trung_binh = np.mean(du_lieu_sach)
trung_vi = np.median(du_lieu_sach)
do_lech_chuan = np.std(du_lieu_sach)

print("Trung bình:", trung_binh)
print("Trung vị:", trung_vi)
print("Độ lệch chuẩn:", do_lech_chuan)

# Giả sử bạn muốn phân tích sự biến đổi của dữ liệu theo thời gian
# Dữ liệu giả định cho giá trị qua 5 năm
gia_tri_nam = np.array([45, 47, 55, 67, 76])

# Tính tốc độ tăng trưởng hàng năm
tang_truong_hang_nam = np.diff(gia_tri_nam) / gia_tri_nam[:-1] * 100
print("Tốc độ tăng trưởng hàng năm (%):", tang_truong_hang_nam)
```

Trong phần này, chúng ta tính các chỉ số thống kê cơ bản như trung bình, trung vị, và độ lệch chuẩn để có cái nhìn tổng quan về dữ liệu. Sau đó, chúng ta phân tích sự biến đổi của dữ liệu qua các năm bằng cách tính tốc độ tăng trưởng hàng năm, giúp khám phá xu hướng phát triển qua thời gian.

Thông qua việc chế ngự và phân tích dữ liệu như một nghệ sĩ, bạn không chỉ biến dữ liệu thô thành thông tin có giá trị mà còn vẽ nên bức tranh đầy màu sắc về hiện trạng và xu hướng phát triển của đối tượng nghiên cứu. Numpy, với các hàm tính toán mạnh mẽ và linh hoạt của mình, là công cụ không thể thiếu trong quá trình biến dữ liệu thô thành "vàng" của nhà phân tích dữ liệu.

Bước 6: Phiêu lưu Tự Học Numpy

Để tiến xa hơn trên hành trình tự học Numpy ở cấp độ cao hơn, bạn có thể tập trung vào những kỹ thuật và chủ đề nâng cao sau đây:

1. Khám Phá Universal Functions (ufuncs)

Numpy cung cấp các universal functions (ufuncs) cho phép thực hiện các phép toán trên mảng một cách nhanh chóng và hiệu quả. Hãy tìm hiểu về cách sử dụng và tạo ra các ufuncs riêng để tối ưu hóa các tính toán của bạn.

Be happy

```
# Ví dụ: Sử dụng ufunc np.add để thực hiện phép cộng trên mỗi phần tử
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
```

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

```
import numpy as np

# Giả sử dữ liệu thô của bạn chứa cả giá trị hợp lệ và giá trị NaN (Not a Number)
du_lieu_tho = np.array([45, 67, np.nan, 90, np.nan, 33, 56])

# Lọc ra những giá trị hợp lệ (không phải NaN)
du_lieu_sach = du_lieu_tho[~np.isnan(du_lieu_tho)]
print("Dữ liệu sau khi làm sạch:", du_lieu_sach)
```

Trong ví dụ này, chúng ta sử dụng một mảng Numpy chứa cả giá trị hợp lệ và giá trị NaN. Chúng ta lọc ra những giá trị NaN để chỉ giữ lại dữ liệu hợp lệ, chuẩn bị cho phân tích tiếp theo.

Phân Tích Là Bức Tranh

Sau khi dữ liệu đã được làm sạch, bước tiếp theo là phân tích dữ liệu để khám phá các xu hướng và thông tin ẩn giấu:

```
python copy code
# Tính các chỉ số thống kê cơ bản để phân tích dữ liệu
trung_binh = np.mean(du_lieu_sach)
trung_vi = np.median(du_lieu_sach)
do_lech_chuan = np.std(du_lieu_sach)

print("Trung bình:", trung_binh)
print("Trung vị:", trung_vi)
print("Độ lệch chuẩn:", do_lech_chuan)

# Giả sử bạn muốn phân tích sự biến đổi của dữ liệu theo thời gian
# Dữ liệu giả định cho giá trị qua 5 năm
gia_tri_nam = np.array([45, 47, 55, 67, 76])

# Tính tốc độ tăng trưởng hàng năm
tang_truong_hang_nam = np.diff(gia_tri_nam) / gia_tri_nam[:-1] * 100
print("Tốc độ tăng trưởng hàng năm (%):", tang_truong_hang_nam)
```

Trong phần này, chúng ta tính các chỉ số thống kê cơ bản như trung bình, trung vị, và độ lệch chuẩn để có cái nhìn tổng quan về dữ liệu. Sau đó, chúng ta phân tích sự biến đổi của dữ liệu qua các năm bằng cách tính tốc độ tăng trưởng hàng năm, giúp khám phá xu hướng phát triển qua thời gian.

Thông qua việc chế ngự và phân tích dữ liệu như một nghệ sĩ, bạn không chỉ biến dữ liệu thô thành thông tin có giá trị mà còn vẽ nên bức tranh đầy màu sắc về hiện trạng và xu hướng phát triển của đối tượng nghiên cứu. Numpy, với các hàm tính toán mạnh mẽ và linh hoạt của mình, là công cụ không thể thiếu trong quá trình biến dữ liệu thô thành "vàng" của nhà phân tích dữ liệu.

Bước 6: Phiêu lưu Tự Học Numpy

Để tiến xa hơn trên hành trình tự học Numpy ở cấp độ cao hơn, bạn có thể tập trung vào những kỹ thuật và chủ đề nâng cao sau đây:

1. Khám Phá Universal Functions (ufuncs)

Numpy cung cấp các universal functions (ufuncs) cho phép thực hiện các phép toán trên mảng một cách nhanh chóng và hiệu quả. Hãy tìm hiểu về cách sử dụng và tạo ra các ufuncs riêng để tối ưu hóa các tính toán của bạn.

Do happy

```
# Ví dụ: Sử dụng ufunc np.add để thực hiện phép cộng trên mỗi phần tử
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
c = np.add(a, b)
```

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

2. Tối Ưu Hóa Bộ Nhớ và Hiệu Suất với Numpy
Học cách sử dụng Numpy một cách hiệu quả hơn về mặt bộ nhớ và hiệu suất. Điều này bao gồm việc hiểu về cách Numpy lưu trữ và xử lý dữ liệu, cũng như cách tận dụng các kỹ thuật như memory views và copy-on-write.

Để sử dụng Numpy hiệu quả hơn về mặt bộ nhớ và hiệu suất, quan trọng là phải hiểu cách thức Numpy lưu trữ và xử lý dữ liệu. Dưới đây là một ví dụ về việc sử dụng memory views (các dạng xem bộ nhớ) trong Numpy, điều này giúp bạn thao tác dữ liệu mà không cần phải tạo ra bản sao dữ liệu, từ đó tiết kiệm bộ nhớ:

```
import numpy as np

# Tạo một mảng Numpy lớn
mang_lon = np.arange(1000000) # Một mảng từ 0 đến 999999

# Thay vì tạo một bản sao, ta sử dụng "view" để truy cập một phần của mảng mà không tốn thêm bộ nhớ
phan_mang = mang_lon[500000:500010] # Lấy ra một phần nhỏ của mảng mà không tạo bản sao

# Thực hiện một phép toán trên "view"
phan_mang *= 10 # Nhân mỗi phần tử trong phần mảng này với 10

# Kiểm tra xem thay đổi trong "view" có ảnh hưởng đến mảng gốc không
print("Phần tử gốc sau khi thay đổi trong 'view':", mang_lon[500000:500010])
```



Ví dụ này minh họa cách phan_mang được sử dụng như một view của mang_lon, nghĩa là khi bạn thay đổi dữ liệu trong phan_mang, dữ liệu trong mang_lon cũng thay đổi tương ứng, mà không cần tạo ra một bản sao của mảng gốc. Điều này giúp giảm bớt lượng bộ nhớ cần thiết và tăng hiệu suất, đặc biệt khi làm việc với dữ liệu lớn.

Lưu ý rằng sử dụng views cần phải cẩn thận, vì thay đổi dữ liệu trong view cũng sẽ thay đổi dữ liệu trong mảng gốc. Điều này có thể gây ra các hậu quả không mong muốn nếu bạn không muốn thay đổi mảng gốc.

3. Tương Tác giữa Numpy và các Thư Viện Khác
Numpy tích hợp tốt với nhiều thư viện khác trong hệ sinh thái Python. Hãy khám phá cách sử dụng Numpy cùng với các thư viện như Pandas, Matplotlib, SciPy, và thư viện máy học như Scikit-learn hoặc TensorFlow để mở rộng khả năng phân tích và mô hình hóa dữ liệu của bạn.

Numpy là một thư viện cốt lõi cho các tính toán khoa học trong Python và có khả năng tích hợp cao với nhiều thư viện khác. Dưới đây là một ví dụ minh họa cách Numpy có thể làm việc cùng với Pandas, Matplotlib và Scikit-learn để thực hiện các tác vụ từ việc xử lý dữ liệu đến trực quan hóa và xây dựng mô hình máy học.

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH.VN TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

```
import numpy as np
import pandas as pd

# Tạo một mảng Numpy
data = np.array([[1, 2], [3, 4], [5, 6]])

# Chuyển mảng Numpy thành DataFrame Pandas
df = pd.DataFrame(data, columns=["Column1", "Column2"])
print("DataFrame từ mảng Numpy:\n", df)
```

Sử Dụng Numpy với Matplotlib

```
python
Copy code
import matplotlib.pyplot as plt

# Tạo dữ liệu sử dụng Numpy
x = np.linspace(0, 10, 100) # Tạo một mảng 100 điểm từ 0 đến 10
y = np.sin(x) # Tính giá trị sin của mỗi điểm

# Trực quan hóa dữ liệu sử dụng Matplotlib
plt.plot(x, y)
plt.title("Đồ thị Sin")
plt.xlabel("x")
plt.ylabel("sin(x)")
plt.show()
```

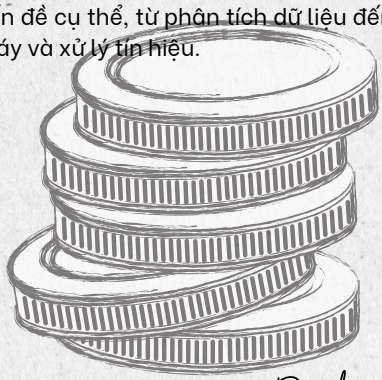
Sử Dụng Numpy với Scikit-learn

```
python
Copy code
from sklearn.linear_model import LinearRegression

# Tạo dữ liệu cho mô hình hồi quy
X = np.array([[1], [2], [3], [4]])
y = np.dot(X, np.array([2])) + 3 # y = 2x + 3

# Sử dụng Scikit-learn để tạo và huấn luyện mô hình hồi quy tuyến tính
model = LinearRegression()
model.fit(X, y)

# Sử dụng mô hình để dự đoán
predictions = model.predict(X)
print("Dự đoán:", predictions)
```

Trong các ví dụ trên, chúng ta đã sử dụng Numpy để tạo dữ liệu và sau đó sử dụng dữ liệu đó trong các thư viện khác như Pandas để xử lý và quản lý dữ liệu, Matplotlib để trực quan hóa dữ liệu và Scikit-learn để xây dựng mô hình hồi quy tuyến tính. Sự kết hợp giữa Numpy và các thư viện khác mở rộng khả năng phân tích và mô hình hóa dữ liệu, giúp cho việc xử lý dữ liệu khoa học trở nên mạnh mẽ và linh hoạt hơn.

4. Nghiên Cứu Case Studies và Dự Án Thực Tế

Áp dụng kiến thức Numpy vào việc giải quyết các case study và dự án thực tế. Điều này giúp bạn hiểu rõ cách áp dụng các kỹ thuật Numpy để giải quyết những vấn đề cụ thể, từ phân tích dữ liệu đến học máy và xử lý tín hiệu.



Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

5. Đóng Góp cho Cộng Đồng Numpy
Một cách học tập sâu sắc khác là tham gia vào cộng đồng Numpy, đóng góp mã nguồn, tài liệu hoặc tham gia vào các cuộc thảo luận. Điều này giúp bạn tiếp cận với những vấn đề thực tế, cũng như học hỏi từ kinh nghiệm của người khác.

Tham gia vào cộng đồng Numpy có thể mở ra nhiều cơ hội học hỏi và phát triển cho bạn. Dưới đây là một số cách bạn có thể bắt đầu:

1. Tham Gia Các Diễn Đàn và Nhóm Thảo Luận
Tham gia vào các diễn đàn trực tuyến, như Stack Overflow, Reddit (ví dụ: r/Python hoặc r/learnpython), hoặc nhóm Google của Numpy, nơi bạn có thể đặt câu hỏi, chia sẻ kiến thức và giải đáp thắc mắc cho người khác.

2. Đóng Góp cho Dự Án Mã Nguồn Mở Numpy
Bạn có thể tham gia đóng góp cho Numpy bằng cách sửa lỗi, thêm tính năng mới, hoặc cải thiện tài liệu. Hãy bắt đầu bằng việc xem qua trang GitHub của Numpy để tìm hiểu các vấn đề đang cần sự giúp đỡ.

3. Tạo và Chia Sẻ Tài Liệu, Hướng Dẫn
Nếu bạn có kỹ năng viết hoặc tạo video, hãy tạo tài liệu hoặc hướng dẫn về cách sử dụng Numpy và chia sẻ chúng trên blog cá nhân, YouTube hoặc các nền tảng chia sẻ kiến thức khác như Medium.

4. Tham Gia và Tổ Chức Các Buổi Workshop, Hội Thảo
Tham gia hoặc tổ chức các buổi workshop và hội thảo về Numpy hoặc khoa học dữ liệu. Đây là cơ hội tốt để học hỏi từ trải nghiệm thực tiễn và xây dựng

5. Tham Gia Phát Triển Các Dự Án Cộng Tác
Tìm kiếm cơ hội tham gia vào các dự án cộng tác, nơi bạn có thể sử dụng và áp dụng kiến thức Numpy của mình trong các tình huống thực tế, đồng thời học hỏi từ những người tham gia khác.

6. Góp Ý và Phản Hồi cho Cộng Đồng
Góp ý và phản hồi về các tính năng mới, báo cáo lỗi hoặc cải thiện tài liệu. Sự đóng góp của bạn có thể giúp làm cho Numpy trở nên tốt hơn cho tất cả mọi người.

7. Mạng Xã Hội và Blogging
Sử dụng mạng xã hội hoặc blog cá nhân để chia sẻ kiến thức, kinh nghiệm và dự án liên quan đến Numpy. Điều này không chỉ giúp bạn củng cố kiến thức của mình mà còn giúp truyền cảm hứng cho người khác.

Tham gia và đóng góp cho cộng đồng không chỉ giúp bạn phát triển kỹ năng và kiến thức cá nhân, mà còn mở rộng mạng lưới chuyên nghiệp và tạo ra sự ảnh hưởng tích cực đến cộng đồng. Đây là một phần quan trọng của quá trình học tập và phát triển trong bất kỳ lĩnh vực nào.



Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

6. Thử Thách với Các Bài Toán Tính Toán Khoa Học Phức Tạp
Để thách thức bản thân với các bài toán tính toán khoa học phức tạp sử dụng Numpy, bạn có thể bắt đầu với một số ví dụ cụ thể như sau:

1. Mô Phỏng Vật Lý: Mô Phỏng Quỹ Đạo Của Một Hành Tinh
Bạn có thể sử dụng Numpy để tính toán quỹ đạo của một hành tinh dựa trên các định luật vật lý cơ bản.

```
import numpy as np
import matplotlib.pyplot as plt

G = 6.67428e-11 # Hằng số hấp dẫn
M = 5.972e24 # Khối lượng Trái Đất

# Điều kiện ban đầu
x0 = 0 # Vị trí ban đầu
y0 = 6.371e6 + 400e3 # Vị trí ban đầu cách mặt đất 400km
vx0 = 7660 # Vận tốc ban đầu (m/s)
vy0 = 0

# Thời gian và bước thời gian
t = 0
dt = 1

# Lưu trữ vị trí
x, y = [x0], [y0]

# Trực quan hóa quỹ đạo
plt.plot(x, y)
plt.xlabel('X')
plt.ylabel('Y')
plt.axis('equal')
plt.show()
```

2. Tối Ưu Hóa: Tìm Giá Trị Tối Ưu Cho Một Hàm Số
Sử dụng Numpy để thực hiện các phép tính liên quan đến việc tìm giá trị tối ưu (tối đa hoặc tối thiểu) cho một hàm số cụ thể.

Python Copy code

```
def ham_so(x):
return x**2 + 10*np.sin(x)

# Sử dụng phương pháp đạo hàm để tìm điểm cực tiểu của hàm số
x = np.arange(-10, 10, 0.1)
y = ham_so(x)

dx = 0.1
x_min = x[np.argmin(y)]

# Vẽ đồ thị hàm số và điểm cực tiểu
plt.plot(x, y)
plt.plot(x_min, ham_so(x_min), 'ro')
plt.show()
```

3. Xử Lý Ảnh: Lọc Nhiễu Trong Ảnh
Numpy có thể được sử dụng để xử lý ảnh, chẳng hạn như áp dụng một bộ lọc trung bình để loại bỏ nhiễu từ ảnh.

Python Copy code

```
import matplotlib.pyplot as plt
from scipy.ndimage import convolve

# Giả sử img là một ma trận Numpy đại diện cho ảnh
# img = ...

# Tạo một bộ lọc trung bình
kernel = np.ones((3, 3)) / 9

# Áp dụng bộ lọc lên ảnh
img_filtered = convolve(img, kernel)
```

Be happy

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy | GDP | CPI

```
# Hiển thị ảnh gốc và ảnh đã lọc
plt.subplot(1, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Ảnh Gốc')

plt.subplot(1, 2, 2)
plt.imshow(img_filtered, cmap='gray')
plt.title('Ảnh Đã Lọc')
plt.show()
```

Những ví dụ này chỉ là điểm khởi đầu cho bạn trong việc áp dụng Numpy vào giải quyết các bài toán khoa học phức tạp

Bước 7: Dự Án Dấu Ấn Cá Nhân

Tên Dự Án 1: "Phân Tích Tác Động Kinh Tế của Dịch Bệnh COVID-19"

Mô Tả Dự Án:

Trong thời đại hiện nay, dịch bệnh COVID-19 đã và đang tác động mạnh mẽ đến nền kinh tế toàn cầu, từ việc làm, sản xuất, thương mại đến tiêu dùng và đầu tư. "Phân Tích Tác Động Kinh Tế của Dịch Bệnh COVID-19" là một dự án cá nhân nhằm sử dụng dữ liệu kinh tế thực tế để phân tích và hiểu rõ hơn về những tác động này. Bằng cách sử dụng Numpy cùng với các thư viện phân tích dữ liệu khác như Pandas và Matplotlib trong Python, dự án sẽ:

Thu Thập Dữ Liệu: Tập hợp dữ liệu từ các nguồn chính thống như Ngân hàng Thế giới, IMF, các cơ quan thống kê quốc gia và tổ chức y tế về GDP, tỉ lệ thất nghiệp, chỉ số giá tiêu dùng (CPI), số ca nhiễm và tử vong do COVID-19, v.v.

Chuẩn Bị và Làm Sạch Dữ Liệu: Sử dụng Numpy và Pandas để xử lý và làm sạch dữ liệu, loại bỏ các giá trị thiếu sót hoặc không phù hợp, chuẩn hóa dữ liệu để dễ dàng phân tích.

Phân Tích Dữ Liệu: Áp dụng các kỹ thuật phân tích dữ liệu và thống kê để phân tích tác động của dịch bệnh đến các chỉ số kinh tế chính, như GDP, tỉ lệ thất nghiệp, CPI, v.v., và so sánh chúng trước và sau đại dịch.

Trực Quan Hóa Kết Quả: Sử dụng Matplotlib và Seaborn để tạo các biểu đồ trực quan hóa dữ liệu, giúp mô tả rõ ràng các xu hướng và tác động của COVID-19 đến nền kinh tế.

Kết Luận và Đề Xuất: Dựa trên kết quả phân tích, đưa ra các kết luận về tác động của dịch bệnh và đề xuất các biện pháp kinh tế có thể giúp giảm nhẹ những tác động tiêu cực.

Tầm Quan Trọng:

Dự án này không chỉ giúp hiểu rõ hơn về tác động kinh tế của dịch bệnh COVID-19 mà còn cung cấp cái nhìn tổng quan về cách dữ liệu được sử dụng để đưa ra các quyết định chính sách. Việc phân tích này có thể giúp chính phủ, doanh nghiệp, và các tổ chức quốc tế xây dựng các chiến lược ứng phó hiệu quả hơn với các tình huống tương tự trong tương lai, nhằm giảm thiểu tác động đến nền kinh tế và xã hội.

Dự Án 2: "Dự Báo Xu Hướng Thị Trường Chứng Khoán Sử Dụng Dữ Liệu Lịch Sử"

Mô Tả Dự Án:

Thị trường chứng khoán là một biểu hiện quan trọng của nền kinh tế, phản ánh không chỉ tình hình kinh doanh của các công ty mà còn cả tâm lý của nhà đầu tư và các yếu tố vĩ mô khác. Dự án này tập trung vào việc sử dụng dữ liệu lịch sử của thị trường chứng khoán để phân tích và dự báo xu hướng tương lai. Sử dụng Numpy cùng với Pandas và các thư viện máy học

CONTACT: (+84) 9110 2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH H.VN TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Thu Thập Dữ Liệu: Tập hợp dữ liệu giá cổ phiếu lịch sử từ các sàn giao dịch lớn, cũng như các yếu tố vĩ mô có thể ảnh hưởng đến giá cổ phiếu.

Tiền Xử Lý và Phân Tích Dữ Liệu: Làm sạch và chuẩn hóa dữ liệu, sử dụng Numpy và Pandas để tìm hiểu mối quan hệ giữa các biến và tác động của chúng đến giá cổ phiếu.

Xây Dựng Mô Hình Dự Báo: Áp dụng các thuật toán học máy để xây dựng mô hình dự báo xu hướng giá cổ phiếu, sử dụng Scikit-learn hoặc TensorFlow.

Đánh Giá và Tối Ưu Mô Hình: Kiểm tra hiệu suất mô hình thông qua các phương pháp đánh giá và tối ưu hóa nó để đạt được kết quả tốt nhất.

Trực Quan Hóa và Báo Cáo: Sử dụng Matplotlib và Seaborn để trực quan hóa kết quả dự báo và so sánh chúng với dữ liệu thực tế, từ đó rút ra những kết luận và khuyến nghị cho nhà đầu tư.

Tầm Quan Trọng:
Dự án này giúp nhà đầu tư có cái nhìn sâu sắc về thị trường chứng khoán, cải thiện khả năng dự báo và quyết định đầu tư dựa trên dữ liệu và phân tích khoa học, giảm thiểu rủi ro và tối ưu hóa lợi nhuận.

Dự Án 3: "Phân Tích Ảnh Hưởng Của Chính Sách Tiền Tệ Đến Lạm Phát"

Mô Tả Dự Án:
Chính sách tiền tệ của ngân hàng trung ương có ảnh hưởng đáng kể đến lạm phát, tăng trưởng kinh tế và thậm chí là tỷ giá hối đoái. Dự án này nhằm phân tích ảnh hưởng của chính sách tiền tệ đến lạm phát, sử dụng dữ liệu từ các quốc gia khác nhau qua các giai đoạn khác nhau. Sử dụng Numpy, Pandas và các kỹ thuật phân tích thống kê, dự án sẽ:

Thu Thập Dữ Liệu: Thu thập dữ liệu về lãi suất cơ bản, lạm phát, và các chỉ số kinh tế khác từ ngân hàng trung ương và các tổ chức tài chính quốc tế.

Chuẩn Bị và Phân Tích Dữ Liệu: Phân tích mối quan hệ giữa lãi suất và lạm phát sử dụng phân tích hồi quy và các phương pháp thống kê khác.

Trực Quan Hóa Kết Quả: Trực quan hóa mối quan hệ và xu hướng qua thời gian giữa chính sách tiền tệ và lạm phát, giúp hiểu rõ cách chúng tương tác với nhau.

Đánh Giá Tác Động: Sử dụng các mô hình thống kê để đánh giá tác động của chính sách tiền tệ đến lạm phát, cân nhắc các yếu tố khác như tăng trưởng kinh tế và tỷ giá hối đoái.

So Sánh Quốc Gia: So sánh ảnh hưởng của chính sách tiền tệ đến lạm phát giữa các quốc gia khác nhau, tìm ra những bài học và mô hình có thể áp dụng.

Tầm Quan Trọng:
Dự án này cung cấp cái nhìn toàn diện về cách chính sách tiền tệ ảnh hưởng đến lạm phát và kinh tế vĩ mô, giúp các nhà hoạch định chính sách, nhà kinh tế và nhà đầu tư hiểu rõ hơn về các điều kiện kinh tế và cách chúng có thể thay đổi dưới ảnh hưởng của

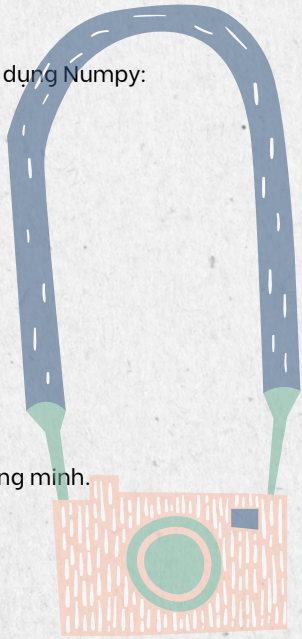
CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH.VN TOÁN VÀ LẬP TRÌNH

DAY 4: NUMPY VÀ PHÂN TÍCH KINH TẾ

Numpy	GDP	CPI

Phần 3: Những điểm cần lưu ý của Numpy:

Dưới đây là một bảng note về những điểm cần lưu ý khi sử dụng Numpy:



🌟 Numpy & Hiệu Suất:
✨ Vectorization là chìa khóa - không cần for loops!
✨ Mảng đa chiều (ndarray) là trái tim của Numpy.

🚀 Broadcasting:
✨ Cùng nhảy múa với các mảng kích thước khác nhau.

📐 Kiểu Dữ Liệu:
✨ Mảng homogenous - tất cả cùng một kiểu dữ liệu, nhé!

🛠️ Hiệu Quả Sử Dụng:
✨ Sử dụng các hàm và kỹ thuật của Numpy một cách thông minh.

🔍 Memory Layout:
✨ Hiểu cách dữ liệu được sắp xếp trong bộ nhớ.

👀 "Views" và "Copies":
✨ Biết sự khác biệt giữa chúng giúp quản lý bộ nhớ tốt hơn.

🔗 Tích Hợp Thư Viện:
✨ Numpy chơi thân với Pandas, Matplotlib, SciPy...

🌐 Cộng Đồng Numpy:
✨ Không ngừng học hỏi từ tài liệu, hướng dẫn, và diễn đàn.

📚 Học Hỏi Liên Tục:
✨ Cập nhật với những tính năng mới của Numpy.

Thế là chúng mình cũng đã xong ngày thứ 4 học tập rồi, Cảm ơn các bạn đã dành thời gian cùng chúng mình học tập ở ngày thứ 4 này. Mong rằng ngày thứ 4 này mang đến cho bạn kết quả và bức tranh thú vị hơn về Numpy

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Hôm nay, team của chúng ta sẽ cùng nhau bước vào một hành trình mới mẻ và hấp dẫn trong lĩnh vực khoa học dữ liệu - một chương mới với Pandas và khám phá dữ liệu. Đây không chỉ là những kiến thức quan trọng mở ra cánh cửa thế giới dữ liệu đầy hứa hẹn, mà còn là chìa khóa giúp chúng ta khai phá những câu chuyện thú vị đằng sau từng hàng số, từng bảng biểu.

Bắt đầu với Pandas, thư viện Python thông minh này sẽ là người bạn đồng hành đắc lực trong quá trình xử lý và phân tích dữ liệu. Đối mặt với những tập hồ sơ lớn, dường như chất chứa cả thế giới số liệu, Pandas sẽ giúp chúng ta biến "mớ hỗn độn" ấy thành một cuốn sách rõ ràng, dễ hiểu, mở ra cánh cửa hiểu biết về những câu chuyện mà dữ liệu ấy muốn kể.

Tại Sao Chúng Ta Chọn Pandas?

Biến Dữ Liệu Thành Câu Chuyện: Pandas mang lại khả năng "đọc" dữ liệu như đọc một cuốn sách hấp dẫn, giúp chúng ta nhanh chóng phát hiện ra những xu hướng, những điểm đặc biệt hay thậm chí là những bất thường trong biển dữ liệu. Mỗi con số, mỗi dữ liệu, giờ đây không chỉ là một phần của báo cáo mà còn là những mảnh ghép của câu chuyện lớn hơn.

Làm Chủ Dữ Liệu, Không Làm Nô Lệ Cho Nó: Với Pandas, không cần thiết phải là một chuyên gia công nghệ thông tin hay một nhà toán học. Thư viện này mang đến khả năng phân tích dữ liệu một cách thân thiện và dễ tiếp cận. Chúng ta sẽ cảm thấy mình như đang điều khiển dữ liệu, sắp xếp và phân tích chúng theo ý muốn mà không cần phải bỏ ra hàng giờ trời chỉ để "giải mã".

Phát Hiện và Kể Lại Câu Chuyện Đằng Sau Số Liệu: Pandas không chỉ giúp chúng ta xem và sắp xếp dữ liệu một cách có hệ thống, mà còn mở ra khả năng khám phá những câu chuyện thú vị đằng sau chúng. Việc trực quan hóa dữ liệu trở nên đơn giản, giúp chúng ta dễ dàng chia sẻ và trình bày dữ liệu một cách sinh động, hấp dẫn hơn.

Từ Dữ Liệu Đến Quyết Định: Trong thời đại ngày nay, dữ liệu quả thực là vàng. Nhưng giá trị thực sự của dữ liệu chỉ được hiện hữu khi chúng ta biết cách phân tích và sử dụng chúng một cách khôn ngoan. Pandas giúp chúng ta biến những dữ liệu thô thành thông tin có giá trị, là nền tảng vững chắc cho những quyết định thông minh,

CONTACT: (+84) 9 110 - 219 - HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Bắt Đầu Hành Trình Khám Phá

Hãy cùng nhau bắt đầu hành trình khám phá này, nơi Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

Chúng ta sẽ cùng nhau đi qua từng bước, từ việc thiết lập môi trường làm việc, khám phá và thao tác cơ bản với dữ liệu, đến việc đắm chìm trong những phân tích nâng cao và cuối cùng là áp dụng những kiến thức đã học vào việc giải quyết các vấn đề thực tế, như phân tích xu hướng tiêu dùng từ dữ liệu bán hàng.

Vậy thì, bạn đã sẵn sàng cho chuyến phiêu lưu này chưa? Hãy để chúng ta cùng nhau đắm chìm trong thế giới của Pandas và khám phá ra những điều thú vị nằm ẩn giấu sau hàng ngàn, hàng triệu dòng dữ liệu. Hãy bắt đầu hành trình khoa học dữ liệu này, không chỉ có kiến thức mà còn là niềm vui mỗi khi khám phá ra điều mới.

Phần 1: Cài đặt, làm quen với pandas

Pandas và một số thư viện hữu ích khác như NumPy thường đã được cài đặt sẵn. Bạn chỉ cần nhập một dòng lệnh đơn giản để chào đón Pandas vào thế giới dữ liệu của mình: Còn chờ gì nữa mà chúng ta không bật google colab lên nào.

Hãy thử nhập code dưới đây và xem điều kỳ diệu mà Pandas mang lại:





PANDAS - KHÁM PHÁ DỮ LIỆU

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

```
import pandas as pd
# Tạo một DataFrame
dulieu = {
'Tên': ['An', 'Bình', 'Hoa', 'Lan'],
'Tuổi': [20, 25, 22, 19],
'Thành Phố': ['Hà Nội', 'TP HCM', 'Đà Nẵng', 'Hải Phòng']
}
df = pd.DataFrame(dulieu)
# Hiển thị DataFrame
print(df)

# In ra tên của những người trên 20 tuổi
print(df[df['Tuổi'] > 20]['Tên'])

# In ra tên của những người ở Hà Nội
print(df[df['Thành Phố'] == 'Hà Nội']['Tên'])

# In ra tên và tuổi của những người ở TP HCM
print(df[df['Thành Phố'] == 'TP HCM'][['Tên', 'Tuổi']])
```

Sau khi chạy đoạn code trên, bạn sẽ nhận thấy cách Pandas giúp chúng ta dễ dàng truy cập và xử lý dữ liệu. Hãy chia sẻ với mình những kết quả bạn nhận được, mình rất muốn nghe về trải nghiệm của bạn!

Với đoạn code trên chúng ta sẽ có những view nhìn như thế nào? Phân tích dữ liệu từ DataFrame trên có thể được tiếp cận từ nhiều khía cạnh khác nhau để khám phá và rút ra thông tin hữu ích. Dưới đây là một số cách bạn có thể phân tích dữ liệu này sử dụng Pandas trong Python:

1. Phân Tích Độ Tuổi

Phân tích phổ tuổi: Tính toán các chỉ số thống kê cơ bản như tuổi trung bình, tuổi nhỏ nhất và lớn nhất. Điều này giúp hiểu biết về phân bố độ tuổi trong nhóm.

Xác định nhóm tuổi: Phân loại dữ liệu dựa trên các nhóm tuổi như thanh niên, trưởng thành để hiểu rõ về đặc điểm độ tuổi của nhóm.

2. Phân Tích Theo Địa Lý

Tổng hợp dữ liệu theo thành phố: Tính toán số người đến từ mỗi thành phố, giúp xác định thành phố nào có số lượng người nhiều nhất trong bộ dữ liệu.

Phân tích dựa trên khu vực địa lý: Phân loại dữ liệu theo vùng miền (Bắc, Trung, Nam) để khám phá sự phân bố địa lý của nhóm.

3. Kết Hợp Độ Tuổi và Địa Lý

Phân tích đa biến: Kết hợp thông tin độ tuổi và địa lý để khám phá mối quan hệ giữa độ tuổi và vị trí địa lý của những người trong danh sách.

Xác định xu hướng: Tìm kiếm bất kỳ xu hướng nào, ví dụ như thành phố nào có dân số trẻ trung nhất hoặc có sự tập trung đặc biệt nào của độ tuổi tại một thành phố cụ thể không.

4. Phân Tích Dữ Liệu Cụ Thể

Phân tích người trên 20 tuổi: Xác định và phân tích đặc điểm của những người trên 20 tuổi, bao gồm cả địa điểm sinh sống và phân bố tuổi tác.

Khám phá dữ liệu theo thành phố: Phân tích đặc điểm của những người đến từ mỗi thành phố, như độ tuổi trung bình, số lượng người, v.v.



Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

5. Trực quan Hóa Dữ Liệu

Biểu đồ tuổi: Sử dụng biểu đồ (ví dụ: histogram, box plot) để trực quan hóa phân phối độ tuổi trong nhóm.

Biểu đồ địa lý: Trực quan hóa số lượng người theo từng thành phố bằng cách sử dụng biểu đồ cột hoặc biểu đồ pie.Khi bạn tiến hành phân tích, hãy chú ý đến việc rút ra những hiểu biết có ý nghĩa từ dữ liệu, và nếu có thể, áp dụng những hiểu biết đó vào các vấn đề thực tế hoặc quyết định cụ thể.

Sau khi DataFrame được tạo ra, chúng ta sẽ thực hiện một số thao tác để xuất thông tin cụ thể từ dữ liệu này. Đầu tiên, hãy in ra toàn bộ DataFrame để có cái nhìn tổng quan:

```
print(df)
```

Tiếp theo, để hiểu rõ hơn về phân bố độ tuổi trong nhóm, chúng ta sẽ xuất ra tên của những người có tuổi lớn hơn 20:

```
print(df[df['Tuổi'] > 20]['Tên'])
```

Nếu muốn biết ai đang sống ở Hà Nội, một dòng lệnh đơn giản sẽ giúp chúng ta lọc thông tin này:

```
print(df[df['Thành Phố'] == 'Hà Nội']['Tên'])
```

Và cuối cùng, chúng ta có thể quan tâm đến việc tìm hiểu thông tin cụ thể hơn về những người sống ở TP HCM, bao gồm cả tên và tuổi của họ:

```
print(df[df['Thành Phố'] == 'TP HCM'][['Tên', 'Tuổi']])
```

Khái niệm về DataFrame và Series

Chắc hẳn bạn đã nghe nhiều về Pandas, đặc biệt là hai người bạn thân thiết của nó: DataFrame và Series. Hãy cùng mình khám phá về họ qua một cái nhìn gần gũi và ấm áp hơn nhé!

DataFrame - Bạn có thể hình dung DataFrame như một quyển album kỷ niệm gia đình, nơi mà mỗi trang giữ những khoảnh khắc, sự kiện quan trọng. Trong thế giới dữ liệu của Pandas, DataFrame chính là một bảng dữ liệu mạch lạc, nơi mỗi hàng là một bản ghi của dữ liệu (như mỗi bức ảnh trong album), và mỗi cột là một loại thông tin cụ thể (tương tự như các chủ đề của album như ngày tháng, sự kiện, nhân vật). DataFrame giúp tổ chức dữ liệu một cách khoa học, giống như cách bạn sắp xếp từng trang album một cách cẩn thận và đầy ý nghĩa.

Series - Giờ đây, nếu bạn lấy một trang trong album đó và chỉ chú ý đến một dòng chú thích dưới mỗi bức ảnh, đó chính là Series. Nói một cách đơn giản, Series là một dãy dữ liệu một chiều, giống như một dãy các chú thích dưới mỗi bức ảnh trong album của bạn, mỗi dòng chú thích mô tả một bức ảnh cụ thể. Trong Pandas, mỗi Series đều có một chỉ mục riêng, giúp bạn dễ dàng tìm kiếm và tham chiếu đến thông tin bạn cần, giống như cách bạn tìm một chú thích trong album để biết câu chuyện

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Khi bạn bắt đầu làm việc với Pandas, việc hiểu rõ về DataFrame và Series sẽ giúp bạn tự tin hơn trong việc khám phá và phân tích dữ liệu. Chúng giống như những người bạn đồng hành, giúp bạn ghi lại, sắp xếp và kể lại những câu chuyện dữ liệu một cách có hệ thống

Để hiểu rõ hơn mình sẽ mô tả theo một khía cạnh khác rõ ràng hơn về DataFrame và Series nhé. Hai cấu trúc dữ liệu chính mà Pandas cung cấp, DataFrame và Series, giống như là hai công cụ đa năng trong bộ dụng cụ nấu ăn của bạn, giúp bạn chế biến và thưởng thức dữ liệu như thế giới ẩm thực vô cùng phong phú.

DataFrame: Bảng Dữ Liệu Linh Hoạt
Hãy tưởng tượng DataFrame như là một chiếc bàn ăn lớn, nơi bạn có thể sắp xếp một bữa tiệc với đủ loại món ăn. Trong Python, DataFrame giúp tổ chức dữ liệu trong một bảng có hàng và cột, giống như một bảng tính Excel. Mỗi hàng có thể tượng trưng cho một món ăn, trong khi mỗi cột lại đại diện cho các thành phần như tên món, nguyên liệu, lượng calo, v.v.

Ví dụ, bạn đang lên kế hoạch cho một bữa tiệc và muốn tổ chức các món ăn cùng thông tin liên quan:

```
python Copy code
import pandas as pd

# Dữ liệu cho bữa tiệc
duLieuBuaTiec = {
'Tên Món': ['Salad', 'Pasta', 'Pizza'],
'Nguyên Liệu Chính': ['Rau', 'Mì', 'Bột'],
'Lượng Calo': [120, 300, 270]
}

# Tạo DataFrame
buaTiec = pd.DataFrame(duLieuBuaTiec)

# Xuất bảng dữ liệu
print(buaTiec)
```

Series: Dòng Dữ Liệu Đơn Giản
Giờ hãy nghĩ về Series như là một danh sách mua sắm. Khi bạn cần mua nguyên liệu cho món Salad, bạn sẽ liệt kê tất cả mọi thứ cần mua như rau xanh, cà chua, dầu giấm, v.v. Trong Pandas, Series chính là một dãy dữ liệu một chiều, giống như một cột trong DataFrame hoặc một danh sách mua sắm vậy.

Một ví dụ đơn giản về Series khi bạn lập danh sách nguyên liệu cho món Salad:

```
python
Copy code
# Danh sách nguyên liệu cho Salad
nguyenLieuSalad = pd.Series(['Rau xanh', 'Cà chua', 'Dầu giấm', 'Hành tây'])

# Xuất danh sách
print(nguyenLieuSalad)
```

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Ứng Dụng Trong Cuộc Sống

Hãy tưởng tượng bạn đang lập kế hoạch mua sắm cho cả tuần. Bạn sử dụng DataFrame để tổ chức danh sách mua sắm với các cột như "Ngày", "Món Ăn", và "Nguyên Liệu Cần Mua". Mỗi hàng trong DataFrame đại diện cho một ngày, giúp bạn dễ dàng theo dõi và sắp xếp kế hoạch mua sắm của mình.

Khi đến siêu thị, bạn dùng Series để liệt kê những gì cần mua cho món ăn đó, giống như cầm theo một danh sách nguyên liệu đã được sắp xếp gọn gàng.

Qua những ví dụ này, bạn có thể thấy Pandas không chỉ là công cụ hữu ích trong công việc mà còn gần gũi và dễ áp dụng trong cuộc sống hàng ngày, từ việc lên kế hoạch bữa ăn cho đến việc sắp xếp danh sách mua sắm. Pandas giúp chúng ta 'nấu chín' dữ liệu và 'thưởng thức' những thông tin hữu ích một cách dễ dàng và ngon lành.

Phần 2: Đọc và Xử Lý Dữ Liệu với Pandas

Trong thế giới dữ liệu ngày nay, việc thu thập và xử lý dữ liệu trở thành một bước không thể thiếu trong bất kỳ dự án phân tích dữ liệu nào. Pandas, với sức mạnh và tính linh hoạt của mình, trở thành công cụ lý tưởng để đối mặt với những thách thức này. Trong phần này, chúng ta sẽ khám phá cách đọc, xem xét, và làm sạch dữ liệu sử dụng Pandas, chuẩn bị cho những bước phân tích sâu hơn.

Đọc Dữ Liệu

Pandas hỗ trợ đọc dữ liệu từ nhiều nguồn khác nhau như CSV, Excel, SQL, JSON và nhiều định dạng khác. Điều này giúp việc thu thập dữ liệu trở nên dễ dàng và linh hoạt.

CSV: pd.read_csv('duong_dan_file.csv')
Excel: pd.read_excel('duong_dan_file.xlsx')
SQL: pd.read_sql(query, connection)





Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Dưới đây là một số ví dụ về cách sử dụng Pandas để đọc dữ liệu từ các nguồn phổ biến:

Đọc dữ liệu từ CSV:
Để đọc một tập tin CSV, bạn có thể sử dụng hàm read_csv và chỉ định đường dẫn tới tập tin đó:

Python Copy code

```
import pandas as pd

# Đọc dữ liệu từ tập tin CSV
df_csv = pd.read_csv('duong_dan_file.csv')
```

Đọc dữ liệu từ Excel:
Pandas cũng cho phép bạn dễ dàng đọc dữ liệu từ các tập tin Excel sử dụng hàm read_excel:

```
# Đọc dữ liệu từ tập tin Excel
df_excel = pd.read_excel('duong_dan_file.xlsx')
```

Đọc dữ liệu từ SQL:
Nếu bạn muốn truy vấn dữ liệu trực tiếp từ một cơ sở dữ liệu SQL, Pandas hỗ trợ điều này thông qua hàm read_sql. Bạn sẽ cần cung cấp câu truy vấn SQL và đối tượng kết nối tới cơ sở dữ liệu:

```
# Đọc dữ liệu từ một cơ sở dữ liệu SQL
df_sql = pd.read_sql(query, connection)
```

Sử dụng các hàm này, bạn có thể dễ dàng tải dữ liệu vào một DataFrame của Pandas, từ đó thực hiện các thao tác phân tích và xử lý dữ liệu tiếp theo. Pandas giúp việc làm việc với dữ liệu trở nên trực quan và tiện lợi, từ việc đọc dữ liệu đầu vào cho đến khám phá và phân tích dữ liệu phức tạp.

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Xem Xét Dữ Liệu Đầu Vào
Sau khi đã đọc dữ liệu vào DataFrame, việc đầu tiên cần làm là xem xét sơ bộ dữ liệu:

.head() và .tail(): Lần lượt hiển thị n hàng đầu tiên và n hàng cuối cùng của DataFrame, giúp bạn có cái nhìn sơ bộ về dữ liệu.
.describe(): Cung cấp thống kê tóm tắt về dữ liệu số học trong DataFrame, bao gồm trung bình, độ lệch chuẩn, giá trị nhỏ nhất, giá trị lớn nhất, và các phần trăm.
Xử Lý Dữ Liệu Thiếu
Dữ liệu thiếu là một thách thức phổ biến trong bất kỳ dự án phân tích nào. Pandas cung cấp các công cụ để xử lý dữ liệu thiếu:

Loại bỏ: Sử dụng .dropna() để loại bỏ bất kỳ hàng nào chứa giá trị NaN.
Điền giá trị: .fillna(value) giúp điền giá trị cụ thể vào chỗ các giá trị NaN.

Giả sử chúng ta có một DataFrame đơn giản lưu trữ thông tin về một nhóm người bao gồm tên, tuổi và thành phố:

```
import pandas as pd
import numpy as np

# Khởi tạo DataFrame
data = {
'Tên': ['An', 'Bình', 'Hoa', 'Lan', 'Khoa'],
'Tuổi': [20, np.nan, 22, 19, 25], # Giả sử chúng ta không biết tuổi của Bình
'Thành Phố': ['Hà Nội', 'TP HCM', 'Đà Nẵng', 'Hải Phòng', np.nan] # Giả sử chúng ta không biết thành phố của Khoa
}
df = pd.DataFrame(data)
```

Be happy



Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

```
# Hiển thị DataFrame
print("DataFrame ban đầu:")
print(df)
.head() và .tail()
```

Để có cái nhìn sơ bộ về DataFrame, bạn có thể sử dụng .head() để xem các hàng đầu tiên và .tail() để xem các hàng cuối cùng.

```
# Hiển thị 3 hàng đầu tiên
print("\n3 hàng đầu tiên:")
print(df.head(3))

# Hiển thị 2 hàng cuối cùng
print("\n2 hàng cuối cùng:")
print(df.tail(2))
.describe()
```

Phương thức .describe() cung cấp thống kê tóm tắt về dữ liệu số học trong DataFrame như trung bình, độ lệch chuẩn, giá trị nhỏ nhất, giá trị lớn nhất và các phần trăm.

```
# Thống kê tóm tắt
print("\nThống kê tóm tắt về dữ liệu số học:")
print(df.describe())
```

Xử Lý Dữ Liệu Thiếu

Dữ liệu thiếu là một vấn đề phổ biến mà chúng ta thường gặp phải. Pandas cung cấp các phương pháp .dropna() và .fillna() để giúp chúng ta xử lý những giá trị thiếu này.

```
# Loại bỏ hàng chứa giá trị NaN
df_cleaned = df.dropna()
print("\nDataFrame sau khi loại bỏ hàng chứa giá trị NaN:")
print(df_cleaned)
```

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

```
# Điền giá trị thiếu bằng giá trị cụ thể
df_filled = df.fillna({'Tuổi': 21, 'Thành Phố': 'Không rõ'})
print("\nDataFrame sau khi điền giá trị thiếu:")
print(df_filled)
```

Thông qua ví dụ này, bạn có thể thấy cách sử dụng các phương pháp cơ bản của Pandas để hiểu và làm sạch dữ liệu, là những bước đầu tiên quan trọng trong quá trình phân tích dữ liệu.

Làm Sạch và Chuẩn Hóa Dữ Liệu

Làm sạch dữ liệu là một trong những công việc quan trọng nhất trong quy trình phân tích dữ liệu, giúp đảm bảo dữ liệu đầu vào chính xác và đồng nhất. Dưới đây là cách thực hiện một số thao tác làm sạch dữ liệu cơ bản sử dụng Pandas:

Loại Bỏ Dữ Liệu Nhiễu

Trong quá trình thu thập dữ liệu, thường sẽ có những cột hoặc hàng không cần thiết, chứa thông tin nhiễu làm ảnh hưởng đến quá trình phân tích. Bạn có thể loại bỏ chúng như sau:

```
Python Copy code
# Giả sử df là DataFrame của bạn
# Loại bỏ một cột không cần thiết
df_cleaned = df.drop('CộtKhôngCầnThiết', axis=1)
```

```
# Loại bỏ một hàng dựa trên chỉ số
df_cleaned = df.drop(3, axis=0)
```

Trong đó, axis=1 chỉ định loại bỏ cột, và axis=0 chỉ định loại bỏ hàng.

Chuẩn Hóa Định Dạng Dữ Liệu

Đảm bảo tất cả dữ liệu trong một cột tuân thủ một định dạng thống nhất là rất quan trọng, đặc biệt là đối với dữ liệu ngày tháng, để tránh những hiểu nhầm và sai lệch trong quá trình phân tích.

Dr happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Python Copy code
Chuẩn hóa định dạng ngày tháng
df['NgàyTháng'] = pd.to_datetime(df['NgàyTháng'], format='%d/%m/%Y')
Việc sử dụng hàm pd.to_datetime giúp chuyển đổi cột ngày tháng từ định dạng chuỗi sang định dạng datetime của Pandas, dựa trên định dạng đã chỉ định.

Thay Đổi Kiểu Dữ Liệu Của Cột
Trong một số trường hợp, kiểu dữ liệu của cột không phản ánh chính xác loại dữ liệu mà bạn muốn làm việc. Ví dụ, một cột chứa số nhưng lại được nhận diện dưới dạng chuỗi (string). Bạn có thể thay đổi kiểu dữ liệu của cột như sau:

Python Copy code
Thay đổi kiểu dữ liệu của cột từ string sang integer
df['Tuổi'] = df['Tuổi'].astype(int)
Sử dụng .astype(int) để thay đổi kiểu dữ liệu của cột Tuổi từ chuỗi sang số nguyên.

Qua việc loại bỏ dữ liệu nhiễu, chuẩn hóa định dạng dữ liệu, và thay đổi kiểu dữ liệu của cột, bạn sẽ có một bộ dữ liệu sạch và đồng nhất, sẵn sàng cho các bước phân tích tiếp theo.

Tinh Chỉnh DataFrame
Lọc Dữ Liệu
Giả sử bạn có một bảng dữ liệu về một nhóm người, bao gồm tên, tuổi và thành phố họ đang sống. Bạn muốn tìm ra những người có tuổi trên 20. Đây là cách bạn có thể lọc dữ liệu đó:

Dr happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Python Copy code
# Lọc những hàng có 'Tuổi' trên 20
df_filtered = df[df['Tuổi'] > 20]
Lệnh này sẽ trả về một DataFrame mới chỉ chứa thông tin của những người có tuổi trên 20.

Chọn Lọc Cột và Hàng
Pandas cung cấp .loc[] và .iloc[] để chọn lọc dữ liệu một cách chính xác, dựa trên nhãn hàng và cột hoặc vị trí số nguyên.

Sử dụng .loc[] để chọn dựa trên nhãn:

Python Copy code
# Chọn hàng có nhãn 'An' và các cột 'Tuổi' và 'Thành Phố'
an_info = df.loc['An', ['Tuổi', 'Thành Phố']]
Sử dụng .iloc[] để chọn dựa trên vị trí số nguyên:

Python Copy code
# Chọn hàng thứ hai và cột thứ nhất và thứ ba (đếm từ 0)
row_col_selected = df.iloc[1, [0, 2]]
Sắp Xếp Dữ Liệu
Khi bạn muốn sắp xếp dữ liệu dựa trên một hoặc nhiều cột, sử dụng .sort_values():

Python Copy code
# Sắp xếp dữ liệu tăng dần dựa trên cột 'Tuổi'
df_sorted = df.sort_values(by='Tuổi')

# Sắp xếp giảm dần dựa trên cột 'Tuổi'
df_sorted_desc = df.sort_values(by='Tuổi', ascending=False)

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA

- Đọc
- Xử Lý Dữ Liệu
- Phân Tích



Dự Án Thực Hành: Một Hành Trình Phân Tích Xu Hướng Tiêu Dùng

Bạn đang ngồi tại một quán cà phê yêu thích, với tách cà phê nóng hổi bên cạnh và laptop mở ra trước mặt. Hôm nay, chúng ta sẽ cùng nhau bắt tay vào một dự án thú vị: khám phá những xu hướng tiêu dùng ẩn giấu trong dữ liệu bán hàng bằng cách sử dụng Pandas, một công cụ mà chúng ta đã dành thời gian tìm hiểu.

Mục Tiêu Dự Án

Mục tiêu của chúng ta không chỉ là áp dụng những kỹ thuật đã học để "đào sâu" vào dữ liệu bán hàng, mà còn là để hiểu được những câu chuyện thú vị mà dữ liệu muốn kể cho chúng ta. Chúng ta sẽ tìm hiểu về những sản phẩm bán chạy, những thời điểm mua sắm ưa thích của khách hàng, và từ đó, phát hiện ra những xu hướng tiêu dùng hiện tại.

Các Bước Thực Hiện

Thu Thập Dữ Liệu: Mọi hành trình đều bắt đầu bằng bước đầu tiên. Chúng ta sẽ bắt đầu từ việc thu thập một bộ dữ liệu bán hàng, có thể là dữ liệu thực từ doanh nghiệp hoặc một bộ dữ liệu mô phỏng.

Tiền Xử Lý và Làm Sạch Dữ Liệu: Như bước chuẩn bị cho một bữa tiệc, chúng ta sẽ "dọn dẹp" dữ liệu của mình, loại bỏ những phần không cần thiết, điền vào những khoảng trống và đảm bảo mọi thứ đều sẵn sàng cho bữa tiệc phân tích dữ liệu.

Phân Tích Đơn Giản: Bây giờ là lúc để "nếm thử" dữ liệu của chúng ta bằng cách áp dụng một số phân tích đơn giản. Chúng ta sẽ xem xét doanh số bán hàng, xác định những sản phẩm bán chạy và những thời điểm mà khách hàng thích mua sắm nhất.

Phân Tích Xu Hướng: Với một ly cà phê bên cạnh, chúng ta sẽ cùng nhau "đi sâu" vào dữ liệu, tìm kiếm những xu hướng tiêu dùng theo thời gian, sản phẩm, và nhóm khách hàng. Đây là phần thú vị nhất, nơi chúng ta có thể phát hiện ra những câu chuyện đằng sau những con số.





Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Trực Quan Hóa Kết Quả: Và cuối cùng, chúng ta sẽ "vẽ" những câu chuyện của mình lên những biểu đồ sinh động sử dụng Matplotlib hoặc Seaborn, giúp chúng ta và người khác dễ dàng nhận ra những mô hình và xu hướng trong dữ liệu.

Kết Quả và Đánh Giá
Sau khi đã "làm chủ" dữ liệu, chúng ta sẽ ngồi lại và nhìn nhận lại những gì chúng ta đã khám phá. Chúng ta sẽ đánh giá những xu hướng tiêu dùng mà chúng ta đã phát hiện, sử dụng những hình ảnh trực quan mà chúng ta đã tạo ra. Đồng thời, chúng ta cũng sẽ thảo luận về những kết quả này, đặt ra những giả thuyết về nguyên nhân của những xu hướng đó và suy nghĩ về những bước tiếp theo có thể thực hiện.

Mô tả đoạn code demo

Bước 1: Thu Thập Dữ Liệu
Giả sử chúng ta có một bộ dữ liệu mẫu về bán hàng được lưu trong một tệp CSV có tên là sales_data.csv.

Bạn có thể sao chép đoạn mã sau và lưu vào một tệp có tên là sales_data.csv, sau đó sử dụng Pandas để đọc và thực hành các thao tác đã được mô tả:

plaintext Copy code

```
Product_ID,Product_Type,Quantity_Sold,Price_Per_Unit,Sale_Date
P001,Electronics,34,450,2021-01-15
P002,Clothing,20,35,2021-02-01
P003,Home Decor,15,60,2021-02-18
P004,Books,45,25,2021-03-05
P005,Electronics,24,870,2021-03-15
```

Sau khi lưu dữ liệu trên vào tệp sales_data.csv, bạn có thể sử dụng đoạn mã Pandas sau để đọc và xem xét dữ liệu:

Dr happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Bước 2: Tiền Xử Lý và Làm Sạch Dữ Liệu
Trước tiên, hãy nhập Pandas và đọc dữ liệu từ tệp CSV:

```
Python Copy code
import pandas as pd

# Đọc dữ liệu bán hàng
df = pd.read_csv('sales_data.csv')

# Xem 5 hàng đầu tiên để hiểu dữ liệu
print(df.head())
```

Sau đó, tiến hành làm sạch dữ liệu:

```
Python Copy code
# Loại bỏ các hàng có dữ liệu thiếu
df_cleaned = df.dropna()

# Điền giá trị 0 vào các ô trống, nếu cần
df_filled = df.fillna(0)

# Chuẩn hóa định dạng dữ liệu, ví dụ định dạng ngày tháng
df_cleaned['Sale_Date'] = pd.to_datetime(df_cleaned['Sale_Date'], format='%Y-%m-%d')
```

Bước 3: Phân Tích Đơn Giản
Tiếp theo, hãy tiến hành một số phân tích đơn giản để có cái nhìn tổng quan về dữ liệu:

```
Python Copy code
# Đếm số lượng sản phẩm bán được theo loại
product_counts = df_cleaned['Product_Type'].value_counts()
print(product_counts)

# Xác định doanh số bán hàng theo thời gian
sales_over_time = df_cleaned.groupby('Sale_Date')['Quantity_Sold'].sum()
print(sales_over_time)
```

Bước 4: Phân Tích Xu Hướng
Để phân tích xu hướng tiêu dùng:

```
Python Copy code
# Phân tích xu hướng tiêu dùng theo sản phẩm
trend_by_product = df_cleaned.groupby('Product_Type')['Quantity_Sold'].sum()

# Phân tích xu hướng tiêu dùng theo tháng
df_cleaned['Month'] = df_cleaned['Sale_Date'].dt.month
trend_by_month = df_cleaned.groupby('Month')['Quantity_Sold'].sum()
```

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Bước 5: Trực Quan Hóa Kết Quả
Sử dụng Matplotlib hoặc Seaborn để trực quan hóa dữ liệu:

```
Python Copy code
import matplotlib.pyplot as plt
import seaborn as sns

# Trực quan hóa số lượng sản phẩm bán được theo loại
plt.figure(figsize=(10, 6))
sns.barplot(x=product_counts.index, y=product_counts.values)
plt.title('Số Lượng Sản Phẩm Bán Được Theo Loại')
plt.xlabel('Loại Sản Phẩm')
plt.ylabel('Số Lượng')
plt.xticks(rotation=45)
plt.show()
```

Phần 3: Đi Sâu Hơn
Khi đã thành thạo với các thao tác cơ bản, chúng ta sẽ tiến xa hơn với việc thao tác dữ liệu nâng cao, phân tích dữ liệu thông qua các phương pháp thống kê cơ bản và trực quan hóa dữ liệu. Bạn sẽ được hướng dẫn sử dụng các thư viện như Matplotlib và Seaborn để "vẽ" lên những câu chuyện từ dữ liệu.

Chúng ta cùng bắt đầu với một số bước cơ bản nhất.

Bước 1: Cài đặt và Nhập Matplotlib

Đầu tiên, bạn cần đảm bảo Matplotlib đã được cài đặt trong môi trường Python của bạn. Mình dùng google colab nên sẽ dễ dàng và nhanh hơn.

Bước 1:

```
import matplotlib.pyplot as plt
```

Bước 2: Vẽ Biểu Đồ Đơn Giản
Giả sử bạn có một số liệu bán hàng theo tháng và muốn vẽ biểu đồ để thể hiện doanh số:

```
# Dữ liệu
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May']
sales = [200, 300, 400, 500, 600]

# Vẽ biểu đồ
plt.plot(months, sales)

# Thêm tiêu đề và nhãn
plt.title('Doanh Số Bán Hàng Theo Tháng')
plt.xlabel('Tháng')
plt.ylabel('Doanh số')
```

Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

```
# Hiển thị biểu đồ
plt.show()
```

Bước 3: Tùy Chỉnh Biểu Đồ

Matplotlib cho phép bạn tùy chỉnh biểu đồ của mình một cách linh hoạt, từ màu sắc đến kiểu dáng:

```
# Vẽ biểu đồ với màu sắc và định dạng dạng dấu chấm
plt.plot(months, sales, color='red', marker='o', linestyle='--')

# Thêm chú giải
plt.legend(['Doanh số'])

# Thêm lưới
plt.grid(True)

# Hiển thị biểu đồ
plt.show()
```

Bước 4: Vẽ Nhiều Dữ Liệu Trên Cùng Một Biểu Đồ

Bạn cũng có thể so sánh nhiều tập dữ liệu cùng một lúc:

```
# Dữ liệu bổ sung
expenses = [100, 200, 150, 300, 400]

# Vẽ biểu đồ doanh số
plt.plot(months, sales, marker='o', label='Doanh số')

# Vẽ biểu đồ chi phí
plt.plot(months, expenses, marker='s', label='Chi phí')

# Thêm chú giải
plt.legend()

# Hiển thị biểu đồ
plt.show()
```

Bắt đầu với Matplotlib có thể dễ dàng như vài dòng code. Điều quan trọng nhất là thử nghiệm và vui vẻ với việc trực quan hóa dữ liệu. Bạn có thể biến dữ liệu khô khan trở nên sống động và dễ hiểu, giúp bạn và người khác nắm bắt thông tin quan trọng một cách nhanh chóng.





Be happy

Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Phần 4: Khám Phá Nâng Cao

Xử lý dữ liệu thời gian là một kỹ năng quan trọng trong phân tích dữ liệu, nhất là khi bạn làm việc với các tập dữ liệu phức tạp chứa thông tin về thời gian. Pandas cung cấp các công cụ giúp bạn dễ dàng xử lý và phân tích dữ liệu thời gian. Hãy cùng khám phá cách thức làm việc với dữ liệu thời gian trong Pandas.

Bước 1: Chuyển Đổi Dữ Liệu Thời Gian
Khi bạn đọc dữ liệu từ một tệp CSV hoặc Excel, cột thời gian thường được nhận diện như một chuỗi (string). Bước đầu tiên là chuyển đổi cột này sang định dạng thời gian của Pandas để có thể xử lý một cách dễ dàng:

```
import pandas as pd

# Giả sử 'df' là DataFrame của bạn và 'date_column' là cột chứa dữ liệu thời gian
df['date_column'] = pd.to_datetime(df['date_column'], format='%Y-%m-%d')
```

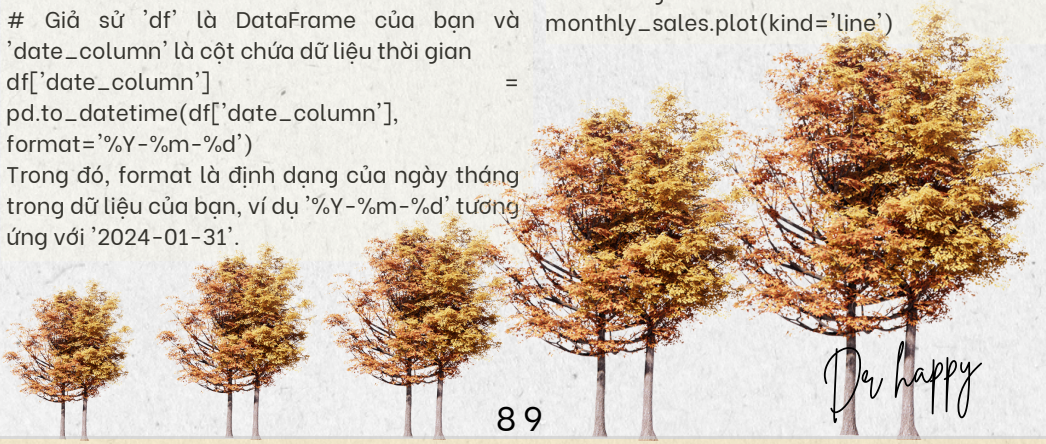
Trong đó, format là định dạng của ngày tháng trong dữ liệu của bạn, ví dụ '%Y-%m-%d' tương ứng với '2024-01-31'.

Bước 2: Thao Tác với Dữ Liệu Thời Gian
Sau khi đã chuyển đổi dữ liệu thời gian, bạn có thể dễ dàng trích xuất thông tin như năm, tháng, ngày, giờ từ cột thời gian:

```
df['year'] = df['date_column'].dt.year
df['month'] = df['date_column'].dt.month
df['day'] = df['date_column'].dt.day
```

Bước 3: Thực Hiện Phân Tích Chuỗi Thời Gian
Dữ liệu thời gian mở ra cơ hội cho nhiều loại phân tích, từ xu hướng đến mùa vụ:

```
# Tính tổng doanh số bán hàng theo tháng
monthly_sales = df.groupby(df['date_column'].dt.month)['sales_column'].sum()

# Vẽ biểu đồ xu hướng doanh số bán hàng theo tháng
monthly_sales.plot(kind='line')
```



Pandas

Pandas không chỉ là công cụ mà còn là người bạn đồng hành, giúp chúng ta mở ra những câu chuyện, những bài học quý báu từ dữ liệu. Qua mỗi bước đi, từ việc giới thiệu cơ bản, cài đặt, đến việc thực hành và áp dụng vào dự án thực tế, chắc chắn chúng ta sẽ cùng nhau học hỏi, khám phá và phát triển, không chỉ là những nhà khoa học dữ liệu mà còn là những nhà kể chuyện, những nhà thám hiểm trong thế giới rộng lớn của dữ liệu.

DAY 5

AGENDA
- Đọc
- Xử Lý Dữ Liệu
- Phân Tích

Bước 4: Xử Lý Dữ Liệu Phức Tạp

Trong trường hợp dữ liệu thời gian của bạn rất phức tạp, chẳng hạn như dữ liệu chuỗi thời gian với nhiều mẫu dữ liệu trong một ngày, bạn có thể cần thực hiện các phép toán như resampling hoặc windowing:

```
# Resampling dữ liệu theo tuần và tính giá trị trung bình
weekly_data = df.resample('W', on='date_column')['sales_column'].mean()

# Sử dụng rolling window để tính giá trị trung bình di động 7 ngày
rolling_data = df.set_index('date_column')['sales_column'].rolling(window=7).mean()
```

Chúng mình đã cùng nhau đi qua khá nhiều về pandas giờ chúng mình takenote lại ghi chú nhỏ này.

📚 Định Dạng Dữ Liệu Đúng

🎯 Kiểm tra và đảm bảo kiểu dữ liệu của mỗi cột phù hợp với nội dung.

🔧 Sử dụng .astype() để thay đổi kiểu dữ liệu khi cần.

💧 Xử Lý Dữ Liệu Thiếu

🗑️ Dùng .dropna() để loại bỏ hàng/cột có giá trị thiếu.

✏️ Dùng .fillna() để điền giá trị thiếu với một giá trị cụ thể.

📋 Làm Việc An Toàn

🛡️ Thao tác trên bản sao DataFrame bằng .copy() để tránh thay đổi không mong muốn.

🚀 Hiệu Suất

⚡ Tránh sử dụng vòng lặp trực tiếp; hãy tận dụng các phương thức tối ưu hóa của Pandas.

🎲 Truy Cập Dữ Liệu

🔑 Sử dụng .loc[] cho truy cập theo nhãn và .iloc[] cho truy cập theo vị trí.

📉 Trực Quan Hóa

📊 Hãy kết hợp Pandas với Matplotlib hoặc Seaborn để tạo biểu đồ dễ hiểu từ dữ liệu.

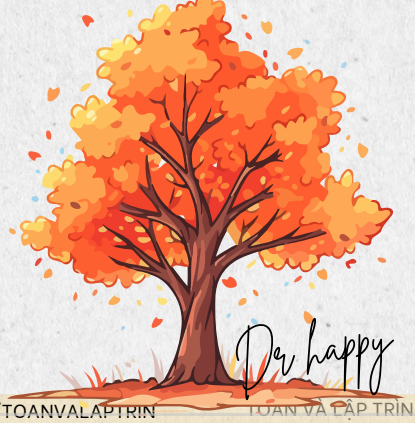
🚧 Cẩn Thận với Cảnh Báo

⚠️ Chú ý đến SettingWithCopyWarning và hiểu cách để tránh nó.

🌟 Cộng Đồng và Hỗ Trợ

💡 Tận dụng tài liệu hướng dẫn sẵn có và cộng đồng người dùng Pandas để tìm giải pháp và lời khuyên.

Cảm ơn bạn đã đồng hành cùng mình trong ngày 5 này, bạn thấy sao về ngày hôm nay ???



DAY 6: BÍ MẬT CỦA XÁC SUẤT



Xác suất cơ bản và sự kiện

Welcome! Phần 1: Cuộc Đối Thoại Giữa Cuộc Sống và Xác Suất

Bắt đầu câu chuyện từ một buổi sáng bình thường, khi bạn đứng trước cửa nhà, dắn đo không biết nên mang theo ô hay không. Bạn nhìn lên bầu trời, đám mây đen kịt báo hiệu cơn mưa sắp tới. Đây chính là lúc xác suất bước vào cuộc sống của bạn, dù bạn có nhận ra hay không. Bạn đang đánh giá xác suất mưa để quyết định có mang theo ô hay không.

Tại Sao Xác Suất Lại Quan Trọng?

Tại Sao Xác Suất Lại Quan Trọng?

Xác suất không chỉ giới hạn ở những quyết định cá nhân nhỏ nhặt; nó còn ảnh hưởng đến mọi khía cạnh của đời sống và xã hội, từ y học, kinh tế, khoa học, đến công nghệ. Mỗi khi chúng ta đứng trước lựa chọn, xác suất là người bạn đồng hành giúp chúng ta làm quen với bất định, giúp chúng ta 'đoán' tương lai dựa trên thông tin hiện tại.

Ví Dụ Thực Tế: Xác Suất trong Đời Sống

Dự Đoán Thời Tiết: Mỗi buổi sáng, bạn kiểm tra dự báo thời tiết để quyết định mặc gì và mang theo những gì. Dự đoán thời tiết chính là ứng dụng của xác suất, dựa trên dữ liệu về thời tiết trong quá khứ và hiện tại.

Quyết Định Mua Bảo Hiểm: Khi bạn quyết định mua bảo hiểm, dù là cho xe hơi, nhà cửa, hay sức khỏe, bạn đang sử dụng xác suất để đánh giá rủi ro có thể xảy ra và quyết định xem việc bảo vệ tài chính có đáng giá không.

Lựa Chọn Đầu Tư: Trong lĩnh vực tài chính và đầu tư, xác suất giúp nhà đầu tư đánh giá rủi ro và kỳ vọng lợi nhuận, từ đó đưa ra quyết định đầu tư thông minh.

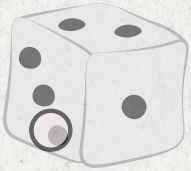
Học Xác Suất để Làm Gì?

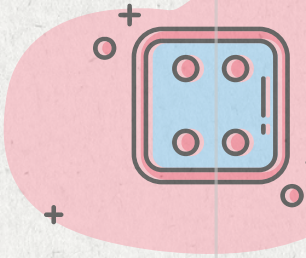
Học xác suất không chỉ là việc nắm bắt các công thức và tính toán. Đó còn là quá trình hiểu cách thế giới hoạt động, làm thế nào để đưa ra quyết định dựa trên không chắc chắn và không hoàn hảo. Điều này giúp chúng ta phát triển tư duy phản biện, cải thiện khả năng giải quyết vấn đề và, quan trọng nhất, học cách sống chung với bất định.

Tầm Quan Trọng Của Xác Suất

Xác suất là ngôn ngữ của bất định, một công cụ giúp chúng ta đối mặt với không chắc chắn của cuộc sống. Từ những quyết định hàng ngày đến những lựa chọn quan trọng trong kinh doanh và khoa học, xác suất giúp chúng ta đưa ra các quyết định sáng suốt, dựa trên thông tin và dữ liệu. Học xác suất không chỉ giúp bạn hiểu biết thêm về toán học, mà còn về cuộc sống và cách chúng ta tương tác với thế giới.







Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 2: Cơ Bản về Xác Suất và Sự Kiện

Bạn đang chuẩn bị cho một buổi picnic cuối tuần với gia đình. Một trong những yếu tố quan trọng nhất để quyết định liệu buổi picnic có thể diễn ra hay không chính là thời tiết. Bạn mở ứng dụng dự báo thời tiết và thấy rằng có 70% khả năng trời sẽ nắng. Đây chính là một ví dụ đơn giản về việc áp dụng xác suất trong cuộc sống hàng ngày - bạn đang sử dụng xác suất để đưa ra quyết định về việc có nên tiến hành kế hoạch của mình hay không.

Cơ Bản về Xác Suất và Sự Kiện
Trong xác suất, một sự kiện là một kết quả hoặc một nhóm kết quả từ một thí nghiệm ngẫu nhiên, như việc tung một đồng xu hay thăm dò dự báo thời tiết. Khi chúng ta nói về sự kiện "trời sẽ nắng", chúng ta đang đề cập đến một kết quả thuận lợi từ thí nghiệm dự báo thời tiết.

Tính Xác Suất của Một Sự Kiện
Xác suất của một sự kiện được tính bằng công thức:

Xác suất của Sự Kiện = Số lượng kết quả thuận lợi / Tổng số kết quả có thể

Ví Dụ Minh Họa: Tung Đồng Xu
Giả sử bạn tung một đồng xu cân đối, có hai kết quả có thể xảy ra: ngửa (H) và sấp (T). Nếu sự kiện bạn quan tâm là "tung được mặt ngửa", thì:

Số lượng kết quả thuận lợi cho sự kiện này là 1 (chỉ có một mặt ngửa).
Tổng số kết quả có thể là 2 (ngửa hoặc sấp).
Vậy xác suất để tung được mặt ngửa sẽ là: có 50% cơ hội để bạn tung được mặt ngửa.

Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 2: Cơ Bản về Xác Suất và Sự Kiện

Thông qua các ví dụ đơn giản như thế này, chúng ta thấy xác suất không chỉ là một khái niệm toán học xa lạ mà còn là một công cụ hữu ích giúp chúng ta đưa ra các quyết định hằng ngày dựa trên dữ liệu và thông tin có sẵn. Đây chính là lý do tại sao việc hiểu biết và ứng dụng xác suất có tầm ảnh hưởng lớn đến cuộc sống của chúng ta.

Bạn và người bạn thân đang lên kế hoạch đi du lịch đến một thành phố mơ ước. Bạn đã mơ về chuyến đi này từ lâu, nhưng giữa lòng phấn khích và mong chờ, có một nỗi lo nhỏ nhoi: bạn sợ hãi việc bay. Mỗi khi nghĩ đến việc phải ngồi trên máy bay, trái tim bạn bắt đầu đập nhanh và bạn cảm thấy lo lắng. Nhưng đây cũng là cơ hội để xác suất thể hiện sức mạnh của mình, giúp bạn vượt qua nỗi sợ hãi này.

$$\sigma^2 = \frac{\sum(x_i - \mu)^2}{n}$$

$$\sigma^2 = \frac{\sum(x_i - \mu)^2}{n}$$

$$\mu = \frac{\sum x_i}{n}$$

Câu Chuyện Xác Suất và Chuyến Bay

Bạn bắt đầu tìm kiếm thông tin và thống kê về an toàn hàng không. Bạn khám phá ra rằng xác suất gặp phải một tai nạn hàng không là cực kỳ thấp, khoảng 1 trong 11 triệu chuyến bay. Bạn so sánh con số này với những rủi ro hàng ngày khác mà bạn thường xuyên đối mặt mà không mảy may lo lắng, như lái xe đi làm mỗi ngày.

Phân Tích Sự Kiện và Xác Suất

Trong trường hợp này, sự kiện bạn quan tâm là:

Sự Kiện A: An toàn đến điểm đến - Xác suất này rất cao, gần như là 100% khi xét đến thống kê.

Sự Kiện B: Gặp phải tai nạn hàng không - Xác suất này cực kỳ thấp, 1 trong 11 triệu.

Khi nhìn vào con số, bạn nhận ra rằng mình thực sự an toàn khi bay, và rằng nỗi sợ của bạn không được hỗ trợ bởi dữ liệu và xác suất.

Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 2: Cơ Bản về Xác Suất và Sự Kiện

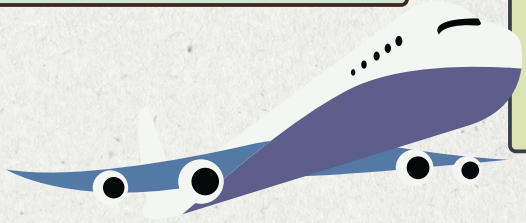
Đưa Ra Quyết Định Dựa Trên Xác Suất

Bằng cách áp dụng lý thuyết xác suất vào tình huống thực tế này, bạn bắt đầu cảm thấy yên tâm hơn. Dù nỗi sợ không thể biến mất hoàn toàn chỉ qua một đêm, nhưng nhận thức mới này giúp bạn giảm bớt căng thẳng và lo lắng. Bạn quyết định đặt vé máy bay, mở lòng với những trải nghiệm mới và bắt đầu hành trình mà bạn và người bạn thân đã mơ ước.

Kết Quả: Một Chuyến Đi Đáng Nhớ

Khi máy bay cất cánh, bạn cảm thấy một chút hồi hộp nhưng cũng rất phấn khích. Bạn nhìn qua cửa sổ, ngắm nhìn thế giới bên dưới từ trên cao, và bỗng nhiên, mọi lo lắng dường như tan biến. Chuyến đi không chỉ là một kỳ nghỉ tuyệt vời mà còn là một bài học quý giá về cách xác suất và thống kê có thể giúp chúng ta đối mặt và vượt qua nỗi sợ.

Qua ví dụ này, chúng ta thấy rằng xác suất không chỉ hữu ích trong việc đưa ra quyết định dựa trên lý trí, mà còn có thể trở thành nguồn động viên tinh thần, giúp chúng ta vượt qua những nỗi sợ và bước ra khỏi vùng an toàn của mình, mở rộng chân trời và khám phá những khả năng mới trong cuộc sống.





Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 3: Xác Suất trong Kinh Doanh

Bài Toán: Ra Mắt Sản Phẩm Mới

Tình Huống

Công ty của bạn đang phát triển một ứng dụng di động mới với mục tiêu giúp người dùng cải thiện thói quen sống hàng ngày. Trước khi quyết định đầu tư vào sản xuất và marketing, đội ngũ quản lý muốn ước lượng xác suất thành công của sản phẩm dựa trên một số yếu tố quan trọng.

Yếu Tố Cần Xem Xét

Thị Trường Mục Tiêu: Thị trường mục tiêu đã được xác định và nghiên cứu chưa? Làm thế nào để biết thị trường mục tiêu sẽ tiếp nhận sản phẩm như thế nào?

Đối Thủ Cạnh Tranh: Có bao nhiêu đối thủ cạnh tranh trực tiếp và gián tiếp? Sản phẩm của bạn có ưu điểm gì nổi bật so với họ?

Năng Lực Sản Xuất: Liệu công ty có đủ năng lực sản xuất và phân phối sản phẩm không? Có nguồn lực tài chính đủ để hỗ trợ dự án này không?

Chiến Lược Marketing: Chiến lược marketing đã được xác định rõ ràng chưa? Bạn dự định sử dụng kênh nào để tiếp cận khách hàng mục tiêu?

Xác Định Xác Suất Thành Công

Để tính toán xác suất thành công của sản phẩm, bạn có thể sử dụng phương pháp định tính và định lượng kết hợp:

Phân tích SWOT (Strengths, Weaknesses, Opportunities, Threats) để đánh giá ưu nhược điểm, cơ hội và thách thức.

Phân tích Thị Trường: Sử dụng dữ liệu từ nghiên cứu thị trường để ước lượng phần trăm thị trường mục tiêu có khả năng tiếp nhận sản phẩm.







CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINHVN TOÁN VÀ LẬP TRÌNH

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 3: Xác Suất trong Kinh Doanh

Phân tích Đối Thủ: Xác định xác suất có thể vượt qua được đối thủ cạnh tranh dựa trên các ưu điểm nổi bật của sản phẩm.

Đánh Giá Năng Lực Sản Xuất và Tài Chính: Đảm bảo rằng công ty có đủ nguồn lực để hỗ trợ sản phẩm từ giai đoạn sản xuất đến khi đưa ra thị trường.

Dựa trên các phân tích trên, hãy đưa ra một ước lượng tổng thể về xác suất thành công của sản phẩm. Nếu xác suất này đủ cao (ví dụ, trên 75%), đội ngũ quản lý có thể quyết định tiếp tục với dự án. Ngược lại, nếu xác suất thành công thấp, có thể cần xem xét lại dự án hoặc điều chỉnh chiến lược.

Bằng cách tính toán và phân tích kỹ lưỡng, bạn không chỉ giúp công ty tránh được những rủi ro không cần thiết mà còn đóng góp vào việc hình thành các chiến lược hiệu quả, tối ưu hóa cơ hội thành công cho sản phẩm mới.



Xác suất, trong trường hợp này, không chỉ là một con số mà còn là một công cụ quyết định giúp dẫn dắt chiến lược kinh doanh tổng thể.

```
# Định nghĩa điểm số cho từng yếu tố
market_score = 0.8
competitor_score = 0.5
production_capability_score = 0.7
marketing_strategy_score = 0.9

# Tính xác suất thành công của sản phẩm
# Điểm đối thủ cạnh tranh được điều chỉnh bởi (1 - competitor_score) vì điểm càng thấp có nghĩa là cạnh tranh càng ít
success_probability = (market_score + (1 - competitor_score) + production_capability_score + marketing_strategy_score) / 4

success_probability
```

Do happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 3: Xác Suất trong Kinh Doanh

Định nghĩa điểm số cho từng yếu tố:

market_score = 0.8: Điểm này cho thấy mức độ hấp dẫn của thị trường mục tiêu, với 0.8 cho thấy thị trường khá hứa hẹn.
competitor_score = 0.5: Điểm này phản ánh mức độ cạnh tranh trong thị trường, với 0.5 là một mức độ cạnh tranh trung bình.

production_capability_score = 0.7: Điểm này đánh giá khả năng sản xuất và cung ứng sản phẩm, với 0.7 cho thấy công ty có năng lực sản xuất khá tốt.
marketing_strategy_score = 0.9: Điểm này đánh giá chiến lược marketing, với 0.9 cho thấy công ty có một kế hoạch marketing mạnh



Tính xác suất thành công của sản phẩm:

Đoạn code sau đây tính toán xác suất thành công dựa trên trung bình cộng của bốn điểm số, nhưng với một điều chỉnh đối với competitor_score: (market_score + (1 - competitor_score) + production_capability_score + marketing_strategy_score) / 4.
Điểm đối thủ cạnh tranh (competitor_score) được điều chỉnh bằng cách lấy 1 - competitor_score. Điều này được thực hiện vì một điểm đối thủ cạnh tranh thấp (nghĩa là ít cạnh tranh) là tốt cho sản phẩm, do đó cần phải đảo ngược giá trị để phản ánh đúng mức độ tích cực của nó.
Kết quả cuối cùng success_probability là giá trị trung bình của các điểm số sau khi đã điều chỉnh, cho chúng ta một ước lượng tổng thể về khả năng thành công của sản phẩm.

success_probability được tính toán ra là 0.725, hoặc 72.5%, cho thấy sản phẩm có khả năng thành công khá cao dựa trên các yếu tố được phân tích.

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINHVN TOÁN VÀ LẬP TRÌNH

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

Bước 1 là một phần quan trọng của quy trình phân tích và lập kế hoạch cho một dự án kinh doanh mới như mở quán cà phê. Dưới đây là một cái nhìn chi tiết hơn về cách tiếp cận mỗi yếu tố:

Vị trí (Location)
Tính quan trọng: Vị trí là một trong những yếu tố quyết định thành công của một quán cà phê. Một vị trí tốt có thể thu hút lượng khách hàng lớn và đảm bảo khả năng tiếp cận dễ dàng.
Đánh giá: Cần xem xét sự gần gũi với khách hàng mục tiêu, sự dễ dàng trong việc đi lại và đậu xe, và mức độ hấp dẫn của vị trí đó. Cân nhắc sự hiện diện của các điểm thu hút khách hàng như trường học, văn phòng, hay khu mua sắm.



Cạnh tranh (Competition)
Tính quan trọng: Hiểu biết về cạnh tranh giúp bạn nhận biết về thị trường và cơ hội để nổi bật.
Đánh giá: Xác định số lượng và loại hình của các đối thủ cạnh tranh trong khu vực. Phân tích điểm mạnh và điểm yếu của họ so với dự án của bạn.

Chất lượng sản phẩm (Product Quality)
Tính quan trọng: Sản phẩm chất lượng cao giúp tạo ra sự khác biệt và tạo dựng lòng trung thành của khách hàng.
Đánh giá: Cân nhắc về nguồn gốc và chất lượng của hạt cà phê, đa dạng của menu, và khả năng đáp ứng các yêu cầu đặc biệt của khách hàng (như các lựa chọn thức uống sức khỏe hoặc dành cho người ăn chay).
Chiến lược marketing (Marketing Strategy)
Tính quan trọng: Một chiến lược marketing hiệu quả giúp tăng nhận thức thương hiệu và thu hút khách hàng.

Be happy

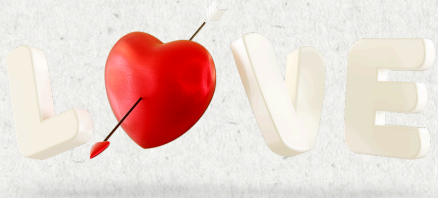
DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

Đánh giá: Xác định các kênh truyền thông và quảng cáo phù hợp với khách hàng mục tiêu của bạn, từ mạng xã hội đến quảng cáo truyền thống. Đề xuất các chiến dịch khuyến mãi hoặc sự kiện đặc biệt để tạo sự chú ý.

Dịch vụ khách hàng (Customer Service)
Tính quan trọng: Dịch vụ khách hàng xuất sắc giúp tạo ra trải nghiệm tích cực, khuyến khích khách hàng quay lại và chia sẻ với người khác.
Đánh giá: Đảm bảo đội ngũ nhân viên được đào tạo tốt và có khả năng cung cấp dịch vụ nhanh chóng, thân thiện và chuyên nghiệp. Cân nhắc việc áp dụng các chính sách dịch vụ khách hàng để giải quyết vấn đề và phản hồi.



Sau khi xác định và phân tích các yếu tố này, bạn sẽ có cái nhìn tổng quan và chi tiết về những gì cần thiết để quán cà phê mới của bạn thành công. Bước tiếp theo sẽ bao gồm việc thu thập dữ liệu cụ thể, phát triển các giải pháp chiến lược cho mỗi yếu tố

Bước 2: Thu Thập Dữ Liệu
Chúng ta thu thập dữ liệu định lượng và định tính cho mỗi yếu tố:

Vị trí: Điểm 0.8 - Vị trí trung tâm, gần nhiều văn phòng và trường học.
Cạnh tranh: Điểm 0.4 - Có 5 quán cà phê khác trong bán kính 500m.
Chất lượng sản phẩm: Điểm 0.9 - Sử dụng hạt cà phê chất lượng cao, thức uống đa dạng.
Chiến lược marketing: Điểm 0.7 - Kế hoạch marketing trên mạng xã hội và phát tờ rơi.
Dịch vụ khách hàng: Điểm 0.85 - Đội ngũ phục vụ thân thiện và chuyên nghiệp.

Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

Với dữ liệu đã thu thập ở Bước 2, bạn đã có những đánh giá cụ thể về từng yếu tố ảnh hưởng đến sự thành công của quán cà phê mới. Điểm số cho mỗi yếu tố phản ánh mức độ tốt hoặc thách thức mà quán cà phê có thể gặp phải trong mỗi lĩnh vực cụ thể. Giờ, chúng ta sẽ tiến tới Bước 3 - Phát Triển Mô Hình Tính Xác Suất.

Bước 3 - Phát Triển Mô Hình Tính Xác Suất

Ở bước này, chúng ta sẽ phát triển một mô hình để tính toán xác suất thành công của quán cà phê mới dựa trên dữ liệu đã thu thập. Mô hình này sẽ giúp chúng ta hiểu rõ hơn về cách các yếu tố khác nhau tác động đến sự thành công và làm thế nào chúng ta có thể sử dụng thông tin này để đưa ra quyết định.



1. Định Nghĩa Xác Suất Thành Công:

Xác suất thành công của quán cà phê mới có thể được định nghĩa như là khả năng đạt được một mục tiêu cụ thể, chẳng hạn như đạt được một lượng doanh thu nhất định, số lượng khách hàng hàng tháng, hoặc mức độ nhận biết thương hiệu trong thị trường mục tiêu.

2. Xây Dựng Công Thức Tính Toán:

Mô hình có thể dựa trên việc tính trung bình cộng điểm số của các yếu tố, nhưng cũng có thể phức tạp hơn bằng cách áp dụng các trọng số cho từng yếu tố tùy thuộc vào mức độ quan trọng của chúng đối với sự thành công của quán cà phê.

3. Áp Dụng Mô Hình Tính Toán:

Sử dụng công thức đã xây dựng, áp dụng các điểm số và trọng số (nếu có) từ dữ liệu đã thu thập để tính toán xác suất thành công.

Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

4. Đánh Giá và Điều Chỉnh Mô Hình:
Mô hình cần được đánh giá để đảm bảo rằng nó phản ánh chính xác mối quan hệ giữa các yếu tố và sự thành công của quán cà phê. Điều này có thể đòi hỏi việc điều chỉnh các trọng số hoặc thậm chí là cách tính toán nếu cần thiết, dựa trên phản hồi hoặc dữ liệu bổ sung.

5. Sử dụng Mô Hình để Dự Đoán:
Cuối cùng, mô hình có thể được sử dụng để dự đoán sự thành công của quán cà phê dưới các điều kiện khác nhau, giúp đưa ra quyết định về việc thực hiện các chiến lược cụ thể hoặc điều chỉnh kế hoạch kinh doanh.



Mô hình tính xác suất này chỉ là một khung cơ bản và điểm khởi đầu. Trong thực tế, bạn có thể cần phải sử dụng các phương pháp phức tạp hơn như mô phỏng Monte Carlo hoặc phân tích độ nhạy để đánh giá chính xác hơn các yếu tố ảnh hưởng đến sự thành công của quán cà phê.

Bước 4: Tính Toán và Phân Tích
Sau khi đã xác định được mô hình tính xác suất thành công của quán cà phê, bước tiếp theo là thực hiện các tính toán và phân tích kết quả. Điều này sẽ giúp chúng ta hiểu rõ hơn về ảnh hưởng của từng yếu tố đến thành công của dự án và nhận diện được cơ hội cũng như thách thức.

Tính Toán Xác Suất Thành Công:

Dựa trên công thức đã đề xuất ở Bước 3 và dữ liệu thu thập được ở Bước 2, chúng ta sẽ thực hiện tính toán xác suất thành công của quán cà phê:

Dio Lappy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

```
# Định nghĩa điểm số và trọng số cho mỗi yếu tố (Giả định tất cả các yếu tố có trọng số như nhau trong ví dụ này)
scores = {'location': 0.8, 'competition': 0.4, 'product_quality': 0.9, 'marketing': 0.7, 'customer_service': 0.85}

# Điều chỉnh điểm cạnh tranh (1 - điểm cạnh tranh để phản ánh cạnh tranh càng ít thì càng tốt)
scores['competition'] = 1 - scores['competition']
```



```
# Tính xác suất thành công dựa trên trung bình cộng của các điểm số đã điều chỉnh
success_probability = sum(scores.values()) / len(scores)

print(f"Xác suất thành công của quán cà phê dựa trên mô hình và dữ liệu hiện tại là: {success_probability:.2f}")
```

Phân Tích Kết Quả:

Với xác suất thành công đã được tính toán, chúng ta cần phân tích kết quả để hiểu rõ:

Điểm Mạnh: Chất lượng sản phẩm và dịch vụ khách hàng là những điểm mạnh chính của quán cà phê, dựa trên điểm số cao. Đây có thể là những điểm nhấn quan trọng trong việc thu hút khách hàng.

Thách Thức: Mặc dù đã điều chỉnh, nhưng cạnh tranh vẫn là một thách thức lớn do có nhiều quán cà phê khác trong khu vực. Điều này đòi hỏi quán cà phê cần có chiến lược để nổi bật.

Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

Đưa ra Giả định và Kiểm tra:

Kiểm tra Giả định: Nếu thay đổi một số giả định, chẳng hạn như cải thiện chiến lược marketing (điểm tăng từ 0.7 lên 0.8) hoặc giảm bớt sự cạnh tranh (điểm tăng từ 0.6 lên 0.7 sau điều chỉnh), xác suất thành công sẽ thay đổi như thế nào?

Phân tích ảnh hưởng: Đánh giá xem việc thay đổi trong từng yếu tố ảnh hưởng đến xác suất thành công như thế nào và xác định những yếu tố nào quan trọng nhất đối với sự thành công của quán cà phê.
Bước này không chỉ giúp chúng ta đánh giá xác suất thành công dựa trên dữ liệu hiện tại, mà còn giúp nhìn nhận cơ hội và thách thức một cách toàn diện hơn, từ đó đưa ra các quyết định và điều chỉnh chiến lược phù hợp.



Dựa trên kết quả phân tích từ Bước 4, chúng ta có thể tạo ra một báo cáo tổng hợp và đề xuất cụ thể nhằm tăng cơ hội thành công cho quán cà phê mới.

Báo Cáo Tổng Hợp:
Tổng Quan:
Quán cà phê mới đã được đánh giá dựa trên các yếu tố chính ảnh hưởng đến thành công bao gồm vị trí, cạnh tranh, chất lượng sản phẩm, chiến lược marketing và dịch vụ khách hàng. Mỗi yếu tố đã được gán điểm dựa trên đánh giá định lượng và định tính.

Xác suất thành công: Dựa trên mô hình tính toán, quán cà phê có xác suất thành công khoảng 0.77. Điều này cho thấy một tiềm năng khá cao, nhưng vẫn còn không gian để cải thiện.
Điểm mạnh: Chất lượng sản phẩm và dịch vụ khách hàng được đánh giá cao, là những lợi thế có thể tận dụng để thu hút và giữ chân khách hàng.

Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 5: Đánh giá dự án coffee

Thách thức: Cạnh tranh cao trong khu vực là một thách thức lớn, cần có chiến lược để nổi bật so với các đối thủ.
Đề Xuất:
Tối ưu hóa Vị Trí và Thiết Kế Quán Cà Phê:

Tăng cường marketing online, nhất là trên mạng xã hội, để xây dựng cộng đồng và tăng sự nhận biết về thương hiệu.
Tổ chức các sự kiện hoặc chương trình khuyến mãi để thu hút khách hàng và tạo sự chú ý.
Tập trung vào Dịch Vụ Khách Hàng:

Đào tạo nhân viên về dịch vụ khách hàng xuất sắc, tạo ra một môi trường thân thiện và mời gọi.
Xem xét phản hồi của khách hàng và cải thiện dịch vụ dựa trên những phản hồi đó.

Phân Tích Đối Thủ và Định Vị Thương Hiệu:

Thực hiện phân tích chi tiết về các đối thủ cạnh tranh và tìm cách để định vị quán cà phê sao cho nổi bật.
Xem xét các chiến lược giá cả, sản phẩm, và dịch vụ để đảm bảo quán cà phê cung cấp giá trị độc đáo cho khách hàng.
Bằng cách thực hiện các đề xuất trên, quán cà phê có thể cải thiện cơ hội thành công và tạo ra một vị thế vững chắc trong thị trường cạnh tranh.



Be happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 6: Lưu ý

Lưu ý: Khi áp dụng xác suất vào lập trình, đặc biệt trong việc phân tích và dự đoán kết quả cho các dự án kinh doanh, có một số điểm quan trọng cần lưu ý:

Trong Xác Suất:
Giả định Cơ bản: Đảm bảo rằng các giả định cơ bản của mô hình xác suất phù hợp với bối cảnh thực tế. Ví dụ, giả định về độc lập của các sự kiện cần được xem xét cẩn thận.

Phạm vi Áp dụng: Xác định rõ ràng phạm vi áp dụng và giới hạn của các mô hình xác suất, đảm bảo rằng chúng phù hợp với bối cảnh và mục tiêu cụ thể của dự án.

Dữ liệu và Độ chính xác: Dữ liệu đầu vào cho phân tích xác suất cần được thu thập và xử lý một cách chính xác. Độ chính xác và độ tin cậy của dữ liệu đầu vào có ảnh hưởng trực tiếp đến kết quả.

Biến động và Rủi ro: Hiểu rõ về biến động và rủi ro liên quan đến các sự kiện được phân tích. Cần xem xét đến các phân phối xác suất và độ lệch tiềm ẩn trong dữ liệu.

Trong Lập Trình:
Lựa chọn Ngôn ngữ và Thư viện: Chọn ngôn ngữ lập trình và thư viện phù hợp có hỗ trợ mạnh mẽ cho phân tích xác suất và thống kê, như Python với các thư viện như NumPy, SciPy, và Pandas.

Tính Tái sử dụng và Mở rộng: Viết mã nguồn dễ đọc, tái sử dụng và có khả năng mở rộng. Sử dụng hàm và mô-đun để tạo ra mã nguồn có cấu trúc tốt và dễ bảo trì.



Do happy

DAY 6: BÍ MẬT CỦA XÁC SUẤT

Xác suất cơ bản và sự kiện

Welcome! Phần 6: Lưu ý

Kiểm Tra và Xác minh Mô hình: Kiểm tra và xác minh kỹ lưỡng mô hình xác suất trong code để đảm bảo rằng nó hoạt động đúng như mong đợi và không có lỗi lập trình.

Đánh giá Hiệu suất: Đánh giá hiệu suất của code, đặc biệt là với các phân tích lớn và phức tạp, để đảm bảo rằng ứng dụng hoạt động một cách hiệu quả.

Giao diện Người dùng: Nếu cần, phát triển giao diện người dùng thân thiện để người dùng có thể dễ dàng nhập dữ liệu và hiểu kết quả, mà không cần phải tham gia trực tiếp vào lập trình.

Bảo mật và Bảo vệ Dữ liệu: Đảm bảo rằng dữ liệu được xử lý và lưu trữ một cách an toàn, tuân thủ các quy định về bảo mật và bảo vệ dữ liệu.

Bằng cách lưu ý đến những điểm trên, bạn có thể xây dựng một hệ thống phân tích dựa trên xác suất không chỉ chính xác và đáng tin cậy mà còn thân thiện với người dùng và hiệu quả từ góc độ lập trình.

Thế là chúng ta đã hoàn thành xong ngày thứ 5 rồi, Aiko cảm ơn bạn đã học tập, chúc bạn một ngày thứ 6 đầy thú vị nhé.



Do happy

DAY 7

PHÂN PHỐI VÀ RỦI RO



BIẾN NGẪU NHIÊN, PHÂN PHỐI XÁC SUẤT, MÔ PHỎNG RỦI RO, LỢI NHUẬN CHO DANH MỤC ĐẦU TƯ

Học về biến ngẫu nhiên và phân phối xác suất không chỉ là nền tảng quan trọng trong nhiều lĩnh vực khoa học và kỹ thuật mà còn mang lại những điều tốt đẹp trong cuộc sống hàng ngày của chúng ta.

Trong thế giới đầy rẫy những bất định và không chắc chắn, hiểu biết về biến ngẫu nhiên và phân phối xác suất trở thành một ngọn hải đăng soi sáng, giúp chúng ta dự đoán và định hình những sự kiện có khả năng xảy ra trong tương lai. Biến ngẫu nhiên, với bản chất không thể dự đoán trước được một cách chính xác, là những yếu tố cơ bản trong cuộc sống hàng ngày, từ việc tung một đồng xu cho đến những quyết định lớn lao trong kinh doanh hay khoa học. Phân phối xác suất, một khái niệm chặt chẽ và đầy tính toán, giúp chúng ta hiểu được cách thức mà các biến ngẫu nhiên này phân bố và hoạt động. Nó không chỉ đơn thuần là một công thức toán học mà còn là một công cụ giúp chúng ta phân tích và dự đoán các kết quả có thể xảy ra.Phân phối xác suất cho phép chúng ta đặt ra những dự đoán thông minh về tương lai, dựa trên dữ liệu quá khứ và hiện tại, từ đó giúp chúng ta chuẩn bị và lên kế hoạch cho những bất định có thể xảy ra.

Qua việc nghiên cứu và áp dụng các phân phối xác suất khác nhau, chúng ta có thể đạt được hiểu biết sâu sắc hơn về thế giới xung quanh mình. Từ những ứng dụng trong khoa học tự nhiên, như nghiên cứu về hành vi của các phần tử hóa học, cho đến những ứng dụng trong khoa học xã hội, như dự đoán xu hướng kinh tế hay tâm lý học đám đông, phân phối xác suất giúp chúng ta tiếp cận và giải quyết những vấn đề phức tạp một cách hệ thống và khoa học.

Nhưng quan trọng hơn cả, hiểu biết về biến ngẫu nhiên và phân phối xác suất giúp chúng ta chấp nhận sự không chắc chắn như một phần tất yếu của cuộc sống. Thay vì sợ hãi và tránh né, chúng ta học cách đối mặt và quản lý nó, biến những bất định thành cơ hội để phát triển và trưởng thành. Trong thế giới đầy rẫy những bất ngờ, khả năng hiểu và quản lý rủi ro là một kỹ năng vô giá, giúp chúng ta điều hướng cuộc sống một cách thông minh và tự tin hơn.

Dr happy

PHẦN I: KIẾN THỨC BIẾN NGẪU NHIÊN, PHÂN PHỐI XÁC SUẤT.

Khái niệm biến ngẫu nhiên

Hãy tưởng tượng biến ngẫu nhiên như một hòm bất ngờ, mỗi lần bạn mở ra, bạn sẽ nhận được một món quà khác nhau. Điều này giống như trong thế giới thực, khi kết quả của một sự kiện không thể dự đoán trước một cách chính xác, và biến ngẫu nhiên giúp chúng ta ghi lại hoặc mô tả những kết quả đó.

BIẾN NGẪU NHIÊN RỜI RẠC

Hãy nghĩ về việc tung một đồng xu. Có hai khả năng: sấp hoặc ngửa. Nếu chúng ta tung đồng xu này 10 lần, số lần xuất hiện mặt sấp có thể là 0, 1, 2, và cứ thế cho đến 10. Số lần này là một ví dụ về biến ngẫu nhiên rời rạc - nó có thể nhận một số lượng hữu hạn các giá trị.

Tóm lại, biến ngẫu nhiên giúp chúng ta mô tả sự không chắc chắn trong các sự kiện xung quanh. Loại "biến" này có thể là "rời rạc" khi nó chỉ nhận một số lượng giới hạn các giá trị cụ thể, hoặc là "liên tục" khi nó có thể nhận bất kỳ giá trị nào trong một khoảng nhất định. Hiểu về chúng giúp chúng ta có cái nhìn sâu sắc hơn về thế giới và làm cơ sở cho nhiều quyết định quan trọng trong cuộc sống cũng như trong các lĩnh vực khoa học.

BIẾN NGẪU NHIÊN LIÊN TỤC

Giờ, hãy tưởng tượng bạn đo chiều cao của người lớn. Giá trị này có thể rất đa dạng và không giới hạn, ví dụ như 1.75m, 1.752m, 1.7523m, và cứ thế mở rộng. Điều này cho thấy biến ngẫu nhiên liên tục - nó có thể nhận bất kỳ giá trị nào trong một khoảng, chẳng hạn như từ 0 đến vô cùng, và không giới hạn ở các giá trị rời rạc.

RỜI RẠC, LIÊN TỤC

VÍ DỤ VỀ BIẾN NGẪU NHIÊN

Hãy tưởng tượng bạn là chủ một quán cà phê nhỏ xinh đẹp nằm ở góc phố. Mỗi ngày, bạn đều tự hỏi: "Hôm nay quán mình sẽ đón bao nhiêu vị khách?" và "Doanh thu sẽ như thế nào?". Đây chính là lúc biến ngẫu nhiên và phân phối xác suất thể hiện giá trị của mình trong việc dự đoán những câu hỏi hàng ngày như thế này.

KIẾN THỨC BIẾN NGẪU NHIÊN

Be happy

PHẦN 1: KIẾN THỨC BIẾN NGẪU NHIÊN, PHÂN PHỐI XÁC SUẤT.

Tình huống Kinh doanh:
Hãy mường tượng mỗi ngày, số lượng khách hàng ghé thăm quán cà phê của bạn biến động từ 50 đến 100 người - một ví dụ điển hình của biến ngẫu nhiên rời rạc, vì bạn biết rằng không thể có 50.5 người ghé qua quán trong một ngày. Mặt khác, mỗi khách hàng chi tiêu một số tiền nhất định khi họ rời quán, có thể là 15 USD, 20 USD, hoặc thậm chí là 25.75 USD - một ví dụ của biến ngẫu nhiên liên tục, vì số tiền này có thể rơi vào bất kỳ giá trị nào trong một khoảng.

Mô phỏng Doanh thu:
Dựa trên hai yếu tố trên, bạn muốn ước lượng doanh thu hàng ngày của mình. Để làm được điều này, bạn quyết định sử dụng một chút phép thuật của Python.

```
import numpy as np

# Khoảng dự đoán số lượng khách hàng mỗi ngày
khach_toi_thieu, khach_toi_da = 50, 100
# Giá trị trung bình và độ lệch chuẩn của tiền chi tiêu trên mỗi đơn hàng
chi_tieu_tb, do_lech_chuan = 20, 5

# Sử dụng ma thuật numpy để 'dự đoán' số lượng khách hàng
so_luong_khach = np.random.randint(khach_toi_thieu, khach_toi_da + 1)
# Và 'dự đoán' chi tiêu trung bình của mỗi khách hàng
chi_tieu_trung_binh = np.random.normal(chi_tieu_tb, do_lech_chuan)

# Tính toán 'doanh thu mơ ước' cho ngày hôm nay
doanh_thu_ao_tuong = so_luong_khach * chi_tieu_trung_binh

# In ra 'dự báo' mơ mộng của bạn
print(f"Hôm nay, chúng ta 'dự báo' sẽ có {so_luong_khach} vị thực khách ghé thăm.")
print(f"Mỗi người trong số họ sẽ 'tưởng tượng' chi {chi_tieu_trung_binh:.2f} USD.")
print(f"Vậy, 'trong mơ', doanh thu của chúng ta sẽ là: ${doanh_thu_ao_tuong:.2f} USD!")
```

Với đoạn mã này, mỗi lần bạn chạy, bạn sẽ nhận được một "bức tranh" mới về ngày làm việc tiếp theo của mình, đầy màu sắc và bất ngờ. Đây là cách mà biến ngẫu nhiên và phân phối xác suất không chỉ giúp bạn hiểu và dự đoán về kinh doanh một cách khoa học hơn, mà còn mang lại cho bạn niềm vui và sự hứng khởi trong mỗi ngày làm việc.

Be happy

PHẦN I: KIẾN THỨC BIẾN NGẪU NHIÊN, PHÂN PHỐI XÁC SUẤT.

Khái niệm phân phối xác suất:

Chúng ta có thể xem phân phối xác suất như là công thức làm bánh của cuộc sống. Trong cuộc sống hàng ngày, chúng ta thường gặp phải nhiều tình huống không chắc chắn, giống như khi bạn đang chuẩn bị một món bánh mà không biết chính xác nó sẽ ra sao. Phân phối xác suất giúp chúng ta dự đoán "hương vị" cuối cùng của món bánh này, dựa trên "nguyên liệu" (các kết quả có thể xảy ra) và "công thức" (xác suất của mỗi kết quả).

Để hiểu rõ hơn, hãy tưởng tượng bạn đang cân nhắc việc mua một ngôi nhà Khi bạn xem xét việc mua một ngôi nhà, bạn phải đối mặt với một loạt các quyết định phức tạp, được hình thành bởi nhiều yếu tố khác nhau. Trong số đó, vị trí của ngôi nhà là yếu tố quan trọng, vì nó không chỉ ảnh hưởng đến tiện ích hàng ngày của bạn mà còn ảnh hưởng đến giá trị tài sản dài hạn. Giá cả là một yếu tố khác quyết định khả năng mua của bạn và cũng phản ánh chất lượng, kích thước, và nhiều đặc điểm khác của ngôi nhà. Bên cạnh đó, tình trạng kinh tế tổng thể, bao gồm lãi suất vay mua nhà, tình hình việc làm, và tăng trưởng kinh tế, cũng ảnh hưởng đến giá nhà và quyết định mua của bạn.

Phân phối xác suất giúp bạn đánh giá một cách hệ thống khả năng giá nhà tăng hoặc giảm trong tương lai dựa trên các yếu tố này. Ví dụ, bạn có thể phân tích dữ liệu lịch sử và xu hướng hiện tại để ước lượng rằng có 70% khả năng giá nhà sẽ tăng do sự phục hồi của thị trường kinh tế và nhu cầu nhà ở tăng lên. Ngược lại, có thể có 30% khả năng giá nhà sẽ giảm do các yếu tố như lãi suất tăng hoặc thị trường bão hòa.

Thông qua việc sử dụng phân phối xác suất, bạn không chỉ dựa vào cảm tính mà có thể dựa vào một phương pháp định lượng để đánh giá khả năng xảy ra của các tình huống khác nhau. Điều này giúp bạn có cái nhìn tổng quan và đưa ra quyết định mua nhà một cách thông minh, dựa trên cơ sở dữ liệu và phân tích, thay vì chỉ dựa vào trực giác.

PHÂN PHỐI XÁC SUẤT

KIẾN THỨC PHÂN PHỐI XÁC SUẤT



CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH

PHẦN I: KIẾN THỨC BIẾN NGẪU NHIÊN, PHÂN PHỐI XÁC SUẤT.

Bạn là một nhà buôn bán quần áo, và bạn muốn phân tích khả năng bán được một lượng hàng nhất định trong một ngày dựa trên dữ liệu bán hàng của những ngày trước đó. Ví dụ, bạn có dữ liệu bán hàng của 7 ngày qua như sau:

Ngày 1: Bán được 10 chiếc
Ngày 2: Bán được 15 chiếc
Ngày 3: Bán được 7 chiếc
Ngày 4: Bán được 12 chiếc
Ngày 5: Bán được 20 chiếc
Ngày 6: Bán được 8 chiếc
Ngày 7: Bán được 14 chiếc

Giả sử bạn muốn ước lượng xác suất bán được ít nhất 12 chiếc quần áo trong một ngày. Chúng ta sẽ sử dụng Python để tính toán:

Xác định số ngày bán được ít nhất 12 chiếc. Tính xác suất bán được ít nhất 12 chiếc quần áo trong một ngày dựa trên dữ liệu đã cho

PHÂN PHỐI XÁC SUẤT

```
# Dữ liệu bán hàng của 7 ngày
ban_hang = [10, 15, 7, 12, 20, 8, 14]

# Số ngày bán được ít nhất 12 chiếc
so_ngay_ban_duoc_12 = sum(1 for so_luong in ban_hang if so_luong >= 12)

# Tổng số ngày
tong_so_ngay = len(ban_hang)

# Tính xác suất bán được ít nhất 12 chiếc trong một ngày
xacsuat_ban_duoc_12 = so_ngay_ban_duoc_12 / tong_so_ngay

# In xác suất
print(f"Xác suất bán được ít nhất 12 chiếc quần áo trong một ngày là: {xacsuat_ban_duoc_12 * 100}%")
```

Xác suất bán được ít nhất 12 chiếc quần áo trong một ngày là: 57.14285714285714%

Dựa trên dữ liệu bán hàng của 7 ngày, chúng ta đã tính toán được xác suất bán được ít nhất 12 chiếc quần áo trong một ngày là khoảng 57.14%. Điều này có nghĩa là, dựa trên dữ liệu đã cho, trong bất kỳ ngày nào, bạn có khoảng 57.14% khả năng bán được ít nhất 12 chiếc quần áo.

KIẾN THỨC PHÂN PHỐI XÁC SUẤT

PHẦN 2: KIẾN THỨC TOÁN HỌC LIÊN QUAN TỚI BIẾN NGẪU NHIÊN VÀ PHÂN PHỐI XÁC SUẤT.

Trong phân phối xác suất, có nhiều kiến thức và công thức toán học quan trọng được sử dụng để mô tả và tính toán các sự kiện ngẫu nhiên. Dưới đây là một số khái niệm và công thức cơ bản:

1. Xác Suất
Xác suất của một sự kiện, ký hiệu là P(A), là khả năng sự kiện đó xảy ra. Nó được tính bằng số lần sự kiện mong muốn xảy ra chia cho tổng số lần thử nghiệm, và giá trị của nó nằm trong khoảng từ 0 đến 1 (0% đến 100%).

2. Biến Ngẫu Nhiên
Biến ngẫu nhiên là một biến số mà giá trị của nó là kết quả của một sự kiện ngẫu nhiên.

Có hai loại biến ngẫu nhiên chính:

Biến ngẫu nhiên rời rạc: Nhận một số giới hạn các giá trị. Ví dụ: số lần tung đồng xu cho ra mặt sấp.
Biến ngẫu nhiên liên tục: Có thể nhận bất kỳ giá trị nào trong một khoảng. Ví dụ: chiều cao của người trưởng thành.

toán học

3. Phân Phối Xác Suất
Phân phối rời rạc: Mô tả xác suất của từng giá trị mà biến ngẫu nhiên rời rạc có thể nhận. Ví dụ phổ biến là phân phối Bernoulli, phân phối Binomial, và phân phối Poisson.
Phân phối liên tục: Mô tả xác suất của tất cả giá trị trong một khoảng nào đó mà biến ngẫu nhiên liên tục có thể nhận. Ví dụ phổ biến là phân phối chuẩn (phân phối Gauss), phân phối Exponential, và phân phối Uniform.

Chúng ta sẽ xem xét hai ví dụ, một cho phân phối rời rạc và một cho phân phối liên tục, để minh họa cách cài đặt chúng trong Python.

Ví dụ 1: Phân phối Binomial (Rời rạc)
Phân phối Binomial mô tả số lần thành công trong n lần thử nghiệm độc lập, khi mỗi thử nghiệm chỉ có hai kết quả: thành công hoặc thất bại. Ví dụ: nếu bạn tung một đồng xu 10 lần, xác suất để xuất hiện đúng 5 lần mặt sấp là gì?

Ví dụ 2: Phân phối Chuẩn (Liên tục)
Phân phối chuẩn là một trong những phân phối xác suất quan trọng nhất trong thống kê. Nó được sử dụng để mô tả nhiều biến ngẫu nhiên liên tục trong tự nhiên, xã hội và kinh tế. Ví dụ: chiều cao của người trưởng thành thường được mô tả bằng phân phối chuẩn.

Kiến thức toán học

PHẦN 2: KIẾN THỨC TOÁN HỌC LIÊN QUAN TỚI BIẾN NGẪU NHIÊN VÀ PHÂN PHỐI XÁC SUẤT.

Nếu bạn có một hộp bi với các bi có giá trị tiền tệ là [10, 20, 30, 40] và mỗi viên bi có xác suất tương ứng là [0.1, 0.4, 0.3, 0.2], bạn có thể tính kỳ vọng của số tiền bạn sẽ nhận được bằng cách nhân mỗi giá trị bi với xác suất tương ứng và lấy tổng:

$E(X)=(10\times0.1)+(20\times0.4)+(30\times0.3)+(40\times0.2)$

Kết quả sẽ cho biết giá trị trung bình mà bạn có thể mong đợi nhận được từ mỗi lần tham gia vào cuộc chơi, giúp bạn hiểu rõ hơn về khả năng kiếm được tiền trong trò chơi này.

ví dụ về việc áp dụng khái niệm kỳ vọng vào một tình huống về cà phê:

Giả sử bạn đến một quán cà phê và bạn muốn tính kỳ vọng của số tiền bạn sẽ phải trả cho mỗi ly cà phê bạn mua. Bạn biết rằng quán cà phê này có một chính sách đặc biệt: họ sẽ tung một con xúc xắc để quyết định giá của mỗi ly cà phê. Mỗi mặt của xúc xắc sẽ tương ứng với một giá khác nhau như sau:

Mặt số 1: 10.000 đồng
Mặt số 2: 15.000 đồng
Mặt số 3: 20.000 đồng
Mặt số 4: 25.000 đồng
Mặt số 5: 30.000 đồng
Mặt số 6: 35.000 đồng

Bạn muốn tính kỳ vọng của số tiền bạn sẽ trả cho mỗi ly cà phê.

$E(X) = 1/6 * 10.000 + 1/6*15.000 + 1/6 * 20.000 + 1/6 * 25.000 + 1/6 *30.000 + 1/6 *35.000$

Kết quả của phép tính này sẽ là kỳ vọng của số tiền bạn sẽ trả cho mỗi ly cà phê. Điều này giúp bạn hiểu được mức giá trung bình dự kiến mỗi khi bạn đặt ly cà phê tại quán này.

Để mô tả rõ hơn về tính phương sai Var(X), hãy xem xét một ví dụ cụ thể về việc tính phương sai của điểm thi cuối kỳ của một lớp học.

Giả sử bạn là một giáo viên và bạn muốn đánh giá sự biến động của điểm số cuối kỳ của học sinh trong lớp học của bạn. Bạn có một danh sách điểm số của 10 học sinh, và bạn muốn tính phương sai để đo lường mức độ biến động của các điểm số này so với điểm trung bình của lớp.

Dưới đây là điểm số của 10 học sinh:

[75, 82, 90, 68, 88, 95, 72, 79, 85, 91]

Đầu tiên, chúng ta sẽ tính kỳ vọng của điểm số cuối kỳ bằng cách tính trung bình của các điểm số này:

$E(X) = 1/10 * (75 + 82 + 90 + 68 + 88 + 95 + 72 + 79 + 85 + 91)$

Tiếp theo, chúng ta sẽ tính phương sai của điểm số cuối kỳ. Để làm điều này, chúng ta sẽ thực hiện các bước sau:

CÀI ĐẶT

PHẦN 2: KIẾN THỨC TOÁN HỌC LIÊN QUAN TỚI BIẾN NGẪU NHIÊN VÀ PHÂN PHỐI XÁC SUẤT.

1. Tính độ lệch mỗi điểm số từ kỳ vọng: X - E(X)
2. Bình phương độ lệch của mỗi điểm số: $(X - E(X))^2$
3. Tình trung bình của các bình phương độ lệch: $VAR(X) = E((X - E(X))^2)$

```
import numpy as np

# Điểm số của học sinh
X = np.array([75, 82, 90, 68, 88, 95, 72, 79, 85, 91])

# Tính kỳ vọng của điểm số
E_X = np.mean(X)
print("Kỳ vọng của điểm số: ", E_X)

# Tính phương sai của điểm số
Var_X = np.mean((X - E_X)**2)
print("Phương sai của điểm số: ", Var_X)
```

Định Lý Giới Hạn Trung Tâm

Định lý giới hạn trung tâm: Giới thiệu bằng ví dụ

Hãy tưởng tượng bạn có một cái lọ đựng rất nhiều viên bi màu sắc khác nhau, và mỗi viên bi có trọng lượng khác nhau. Bạn muốn biết trung bình mỗi viên bi nặng bao nhiêu, nhưng việc cân từng viên một thì mất quá nhiều thời gian. Vậy bạn làm thế nào?

Bạn lấy ra một nhóm nhỏ các viên bi, cân chúng và tính trung bình trọng lượng của chúng. Bạn lặp lại việc này nhiều lần với các nhóm viên bi khác nhau. Cuối cùng, bạn sẽ thấy rằng trung bình trọng lượng của các nhóm này sẽ gần với trung bình trọng lượng của tất cả các viên bi trong lọ.

Cách Định lý giới hạn trung tâm hoạt động

Đây chính là ý tưởng cơ bản của Định lý giới hạn trung tâm. Nó cho biết khi bạn lấy một số lượng lớn mẫu nhỏ từ một tập hợp lớn dữ liệu (ở đây là các viên bi), trung bình của các mẫu này sẽ hình thành một hình dạng đặc biệt được gọi là phân phối chuẩn, hay còn gọi là hình chuông.

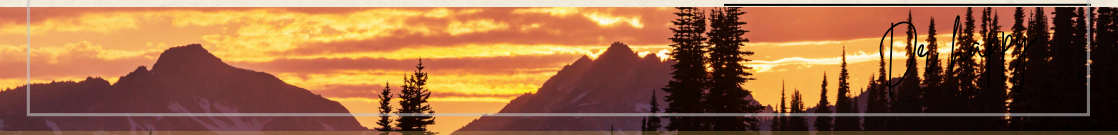
Tại sao nó quan trọng?

Điều này rất hữu ích vì phân phối chuẩn có nhiều tính chất đẹp giúp chúng ta dễ dàng phân tích và đưa ra kết luận. Ví dụ, nếu bạn biết một nhóm mẫu tuân theo phân phối chuẩn, bạn có thể dễ dàng tính toán xem một mẫu nào đó có phải là "bình thường" hay không chỉ bằng cách xem nó nằm ở đâu trong hình chuông.

Ví dụ cụ thể

Giả sử bạn đang làm việc tại một công ty sản xuất bánh kẹo và bạn muốn đảm bảo mỗi gói bánh đều có trọng lượng đúng như quảng cáo. Bạn không thể cân từng gói một vì có quá nhiều.

CÀI ĐẶT



PHẦN 3:TÍNH XÁC SUẤT THÀNH CÔNG CHO DỰ ÁN KINH DOANH MỚI

Case Study: Tính Xác Suất Thành Công cho Dự án Kinh Doanh Cà Phê Mới

Trong bối cảnh thị trường cà phê cạnh tranh và đa dạng, việc đánh giá và tính toán xác suất thành công cho một dự án kinh doanh cà phê mới trở nên hết sức quan trọng. Dự án này có thể là việc mở một quán cà phê mới với một ý tưởng độc đáo hoặc một loại cà phê sáng tạo. Tính toán xác suất thành công giúp hiểu rõ hơn về cơ hội và rủi ro, từ đó lập kế hoạch và phân bổ nguồn lực hiệu quả.

Định Nghĩa Thành Công

Trước hết, chúng ta cần xác định rõ "thành công" trong kinh doanh cà phê này có nghĩa là gì. Điều này có thể bao gồm các yếu tố như doanh thu, lợi nhuận, số lượng khách hàng, đánh giá tích cực, hoặc một sự kết hợp của những yếu tố này.

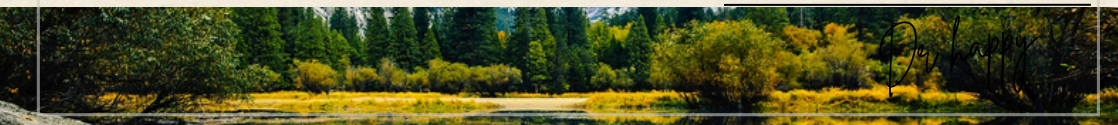
Phân Tích các Yếu Tố Ảnh Hưởng
Các yếu tố ảnh hưởng đến thành công của dự án kinh doanh cà phê có thể bao gồm:

1. Địa Điểm
Đánh Giá: Phân tích địa điểm dựa trên lưu lượng người qua lại, đối tượng khách hàng trong khu vực, và sự gần gũi với các địa điểm hấp dẫn khác (như trường học, văn phòng).
Đo Lường: Sử dụng dữ liệu từ các nghiên cứu thị trường và các công cụ phân tích địa lý để ước lượng lượng khách hàng tiềm năng.

2. Đặc Điểm của Sản Phẩm
Đánh Giá: Xem xét tính độc đáo, chất lượng của cà phê và mức độ đa dạng của menu. Phản hồi từ nhóm tiêu dùng mục tiêu thông qua khảo sát hoặc thử nghiệm thị trường.
Đo Lường: Đánh giá dựa trên số lượng phản hồi tích cực, xếp hạng và đánh giá từ nhóm tiêu dùng mục tiêu.

3. Xu Hướng Thị Trường
Đánh Giá: Nghiên cứu và phân tích xu hướng tiêu dùng hiện tại, bao gồm sở thích về loại cà phê, phong cách phục vụ, và yêu cầu về không gian quán.
Đo Lường: Sử dụng báo cáo ngành, dữ liệu từ các tổ chức nghiên cứu thị trường để xác định xu hướng đang phát triển và phản ứng của thị trường.

4. Chiến Lược Tiếp Thị và Thương Hiệu
Đánh Giá: Xây dựng chiến lược tiếp thị để xây dựng và tăng cường nhận thức thương hiệu. Chiến lược này bao gồm quảng cáo trực tuyến, truyền thông xã hội, sự kiện, và chương trình khuyến mãi.
Đo Lường: Theo dõi tăng trưởng lượng theo dõi trên truyền thông xã hội, lượng truy cập website, và tần suất xuất hiện của thương hiệu trong các phương tiện truyền thông.



PHẦN 3:TÍNH XÁC SUẤT THÀNH CÔNG CHO DỰ ÁN KINH DOANH MỚI

5. Quản Lý và Hoạt Động
Đánh Giá: Hiệu quả quản lý, kỹ năng của nhân viên, và quy trình vận hành. Tổ chức đào tạo nhân viên và phát triển hệ thống quản lý chất lượng.
Đo Lường: Đánh giá thông qua sự hài lòng của khách hàng, hiệu suất nhân viên, và độ trôi chảy của quy trình vận hành hàng ngày.

Tính Toán Xác Suất Thành Công

Dựa trên phân tích trên, chúng ta có thể sử dụng một mô hình toán học như hồi quy tuyến tính hoặc phân tích rủi ro Monte Carlo để tính toán xác suất thành công. Mỗi yếu tố sẽ được gán một trọng số dựa trên mức độ ảnh hưởng của nó đến sự thành công của dự án. Dữ liệu thu thập được từ phân tích trước đó sẽ được đưa vào mô hình để ước lượng.

Ví dụ, nếu một địa điểm có lượng khách hàng tiềm năng cao, sản phẩm được đánh giá tốt, xu hướng thị trường đang tăng trưởng, chiến lược tiếp thị hiệu quả, và quản lý vận hành suôn sẻ, thì xác suất thành công của dự án có thể được mô hình hóa là cao.

Tuy nhiên, quan trọng là phải nhớ rằng mô hình và tính toán chỉ có thể cung cấp một ước lượng dựa trên dữ liệu và giả định hiện tại. Việc thực tế có thể biến động do nhiều yếu tố không thể dự đoán trước được.

Bước 1: Thu Thập Dữ Liệu

Tiếp theo, chúng ta cần thu thập dữ liệu liên quan đến các yếu tố này từ nghiên cứu thị trường, phân tích cạnh tranh, khảo sát khách hàng, và các nguồn dữ liệu nội bộ.

Để tạo ra một bộ dữ liệu mẫu cho việc phân tích, chúng ta có thể mô phỏng dữ liệu dựa trên các yếu tố ảnh hưởng như Địa Điểm, Đặc Điểm của Sản Phẩm, Xu Hướng Thị Trường, Chiến Lược Tiếp Thị và Thương Hiệu, cũng như Quản Lý và Hoạt Động. Dưới đây là cách bạn có thể tạo một bộ dữ liệu mẫu trong Python:

```
import pandas as pd
import numpy as np

# Số lượng mẫu dữ liệu
n_samples = 100
# Tạo dữ liệu ngẫu nhiên mô phỏng cho các yếu tố ảnh hưởng
np.random.seed(42) # Đảm bảo kết quả được tái tạo
dia_diem = np.random.randint(1, 6, n_samples) # Địa điểm được đánh giá từ 1 đến 5
dac_diem_san_pham = np.random.randint(1, 6, n_samples) # Đặc điểm sản phẩm từ 1 đến 5
xu_huong_thi_truong = np.random.randint(1, 6, n_samples) # Xu hướng thị trường từ 1 đến 5
```



PHẦN 3:TÍNH XÁC SUẤT THÀNH CÔNG CHO DỰ ÁN KINH DOANH MỚI

```
chien_luoc_tiep_thi = np.random.randint(1, 6, n_samples) # Chiến lược tiếp thị từ 1 đến 5
quan_ly_va_hoat_dong = np.random.randint(1, 6, n_samples) # Quản lý và hoạt động từ 1 đến 5

# Giả sử tỉ lệ thành công là một hàm của tất cả các yếu tố trên, cộng thêm một lượng nhiễu ngẫu nhiên
success_rate = 0.2 * dia_diem + 0.2 * dac_diem_san_pham + 0.2 * xu_huong_thi_truong + 0.2 * chien_luoc_tiep_thi + 0.2 * quan_ly_va_hoat_dong + np.random.normal(0, 0.5, n_samples)
success_rate = np.clip(success_rate, 0, 5) # Giới hạn tỉ lệ thành công trong khoảng từ 0 đến 5

# Tạo DataFrame từ dữ liệu
data = pd.DataFrame({
  'Dia Diem': dia_diem,
  'Dac Diem San Pham': dac_diem_san_pham,
  'Xu Huong Thi Truong': xu_huong_thi_truong,
  'Chien Luoc Tiep Thi': chien_luoc_tiep_thi,
  'Quan Ly va Hoat Dong': quan_ly_va_hoat_dong,
  'Ti Le Thanh Cong': success_rate
})
# Xuất dữ liệu ra tệp CSV
data.to_csv('du_an_cafe.csv', index=False)
```

Mã này tạo ra một tệp CSV có tên là du_an_cafe.csv với 100 mẫu dữ liệu. Mỗi mẫu bao gồm các giá trị cho các yếu tố ảnh hưởng và một giá trị cho tỉ lệ thành công, mô phỏng dựa trên sự kết hợp của các yếu tố này cộng thêm một lượng nhiễu ngẫu nhiên. Bạn có thể điều chỉnh số lượng mẫu dữ liệu (n_samples) và phạm vi của các biến để phản ánh tình huống cụ thể của dự án kinh doanh cà phê.

Bước 2: Đọc và Xử lý Dữ liệu

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Đọc dữ liệu từ tệp CSV
data = pd.read_csv('du_an_cafe.csv')

# Giả sử cột 'success_rate' là tỉ lệ thành công (ví dụ: doanh thu, lượng khách,...)
# và các cột khác là các yếu tố ảnh hưởng đã được đo lường và chuẩn hóa
X = data.drop('success_rate', axis=1)
y = data['success_rate']
```




PHẦN 3:TÍNH XÁC SUẤT THÀNH CÔNG CHO DỰ ÁN KINH DOANH MỚI

Bước 3: Chia Dữ liệu và Huấn luyện Mô hình

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Xây dựng và huấn luyện mô hình hồi quy tuyến tính
model = LinearRegression()
model.fit(X_train, y_train)
```

```
# Dự đoán trên tập kiểm tra
y_pred = model.predict(X_test)
```

Bước 4: Đánh giá Mô hình

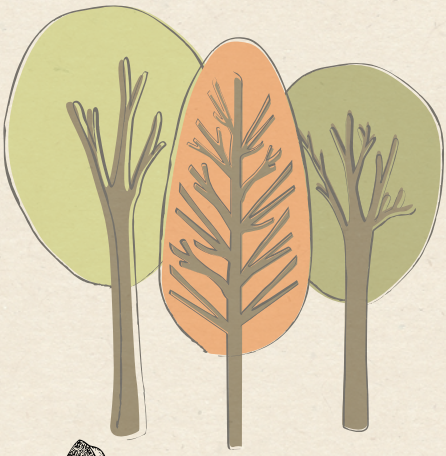
```
# Tính toán sai số bình phương trung bình và hệ số xác định (R^2)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'MSE: {mse:.2f}')
print(f'R^2: {r2:.2f}')
```

Bước 5: Sử dụng Mô hình để Ước lượng Xác Suất Thành Công

Dựa vào kết quả phân tích, đánh giá cơ hội và rủi ro của dự án. Sử dụng thông tin này để tối ưu hóa kế hoạch kinh doanh, bao gồm việc xác định các điểm mạnh cần phát huy và điểm yếu cần khắc phục.

```
# Ước lượng xác suất thành công cho một dự án mới dựa trên các yếu tố ảnh hưởng
# Giả sử đây là các giá trị cho một dự án mới
new_project = np.array([[4, 5, 3, 4, 5]]) # Điểm cho các yếu tố từ 1-5
predicted_success = model.predict(new_project)
print(f'Xác suất thành công dự kiến cho dự án mới: {predicted_success[0]*100:.2f}%')
```

Vậy là chúng mình đã đi qua ngày thứ 7 cùng nhau, ngày hôm nay ad mong bạn đạt được một kết quả thật tốt nhé.





DAY 8: SỐ LIỆU NÓI LÊN ĐIỀU GÌ?

Hôm nay, trong ngày thứ tám của hành trình học tập của chúng mình, mình rất vui được đồng hành cùng bạn. Hy vọng rằng, những kiến thức và trải nghiệm mới mà bạn sẽ khám phá hôm nay sẽ mở ra nhiều cơ hội và ý tưởng mới trong cuộc sống. Hôm nay, chúng mình sẽ tập trung vào hai phần chính



PHẦN 1: THỐNG KÊ MÔ TẢ CƠ BẢN

Trong phần này, chúng ta sẽ bắt đầu bằng việc tìm hiểu về thế giới thú vị của Thống kê Mô Tả. Chúng mình sẽ cùng nhau đọc qua kiến thức về giới thiệu về thống kê mô tả, đây là nền tảng giúp chúng ta hiểu và diễn giải dữ liệu một cách có ý nghĩa. Chúng ta sẽ tìm hiểu tại sao thống kê mô tả lại quan trọng đến vậy trong phân tích dữ liệu.

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Áp dụng những kiến thức đã học, chúng mình sẽ tiến hành một nhiệm vụ thực tế: Phân tích doanh số bán hàng của một công ty, hiểu được dữ liệu doanh số, xử lý dữ liệu, phân tích xu hướng trung tâm và phân tán, phân tích theo thời gian và phân khúc.



Be happy

PHẦN 1: THỐNG KÊ MÔ TẢ CƠ BẢN

Điểm danh các loại dữ liệu:

Trong cuộc hành trình học tập hôm nay, một trong những điểm dừng chân đầu tiên của chúng mình sẽ là "Các Loại Dữ Liệu". Chúng ta sẽ bắt đầu bằng việc làm quen với hai người bạn thân thiết trong ngôn ngữ dữ liệu: Dữ Liệu Định Tính và Định Lượng.
Dữ Liệu Định Tính, hay còn gọi là dữ liệu phân loại, giống như câu chuyện kể về sắc màu và kết cấu của thế giới xung quanh. Nó nói về "loại" hơn là "số lượng". Tưởng tượng bạn đang đứng trong một quán cà phê, Định Tính sẽ là hương thơm của cà phê, màu sắc của bức tranh trên tường, hoặc giai điệu của bản nhạc đang phát. Đây là những thuộc tính mà chúng ta không thể đo lường bằng số, nhưng lại tạo nên bản sắc đặc biệt cho không gian đó.

Đi sâu hơn nữa, chúng ta sẽ gặp Dữ Liệu Định Lượng, người bạn mô tả thế giới qua con số, qua "bao nhiêu". Trong quán cà phê ấy, Định Lượng sẽ là số lượng khách hàng đang ngồi, số tách cà phê đã được phục vụ, hay mức độ âm thanh đo bằng decibel. Định Lượng cho chúng ta biết về kích thước, số lượng, hoặc mức độ một cách chính xác và khách quan.

Khi chúng ta tiếp tục khám phá, chúng ta sẽ phân biệt thêm giữa Dữ Liệu Liên Tục và Rời Rạc trong vùng đất của Định Lượng. Dữ liệu Liên Tục, như làn sóng biển, tồn tại một cách mượt mà và không gián đoạn. Nó có thể nhận bất kỳ giá trị nào trong một khoảng, giống như chiều cao của khách hàng trong quán cà phê, đo bằng centimet - luôn luôn có một giá trị ở giữa hai số mà bạn chọn.

Ngược lại, Dữ liệu Rời Rạc giống như những viên đá trên bờ biển, rõ ràng và đếm được. Nó bao gồm những giá trị cụ thể, không có gì ở giữa. Trong ví dụ của quán cà phê, số lượng khách hàng là dữ liệu rời rạc - bạn không thể có 3.5 khách hàng, chỉ có thể là 3 hoặc 4.

Cuộc hành trình khám phá các loại dữ liệu này không chỉ giúp chúng ta hiểu biết thêm về thế giới dữ liệu mà còn là bước đệm quan trọng để chúng ta xử lý và phân tích chúng một cách chính xác. Bây giờ, khi bạn nhìn vào một tập dữ liệu, bạn sẽ biết cách định danh và đối thoại với các thành viên trong gia đình dữ liệu rộng lớn này.

Dr happy

PHẦN 1: THỐNG KÊ MÔ TẢ CƠ BẢN

Đo lường dữ liệu:

Chúng mình thường nghe nói về việc "đo lường" trong mọi lĩnh vực cuộc sống, từ kết quả học tập đến hiệu suất công việc. Nhưng bạn đã bao giờ tự hỏi làm thế nào để "đo lường" những thứ không hề cầm nắm được như dữ liệu? Hãy cùng nhau khám phá ba thước đo cực kỳ thú vị và quan trọng: trung bình, trung vị, và mode.

Chúng mình thử tìm hiểu 3 thước đo này từ mô tả khu vườn và các loại hoa nhé. Loài hoa đầu tiên, Trung Bình, được biết đến với vẻ đẹp cân xứng của nó, luôn cho chúng ta cái nhìn tổng quan về khu vườn dữ liệu. Bằng cách cộng tất cả giá trị lại và chia cho số lượng, Trung Bình khéo léo tổng hợp sự phong phú của khu vườn vào một điểm tập trung, cho chúng ta thấy một cảnh quan hài hòa.
Tiếp theo, chúng ta gặp Trung Vị, loài hoa mọc ở chính giữa con đường dữ liệu khi tất cả các giá trị được sắp xếp từ thấp đến cao. Trung Vị giống như một cây cầu, nối liền hai bờ của khu vườn; nó cho chúng ta biết giá trị nằm ngay giữa, giúp ta cảm nhận được sự cân bằng, không bị lệch lạc bởi những giá trị ngoại lai.

Cuối cùng, không thể bỏ qua loài hoa Mode, với sức sống mãnh liệt nhất trong khu vườn, luôn tỏa sáng rực rỡ bởi nó xuất hiện nhiều lần nhất. Mode cho chúng ta thấy điểm gặp gỡ yêu thích của dữ liệu, nơi tập trung nhiều giá trị nhất, làm lộ diện một mẫu số chung, một xu hướng nổi bật.

Mình có thể lấy một ví dụ khác để bạn dễ tiếp cận hơn nhé.

Be happy

PHẦN 1: THỐNG KÊ MÔ TẢ CƠ BẢN

Trung bình, bạn có thể đã quen thuộc với nó từ những ngày còn ngồi trên ghế nhà trường. Nó chính là "ngôi sao" trong việc cung cấp cái nhìn tổng quát về một tập hợp dữ liệu. Hãy nghĩ về nó như một người bạn thân thiện, luôn sẵn sàng cho bạn biết điều gì đang xảy ra "trên bình diện tổng thể".

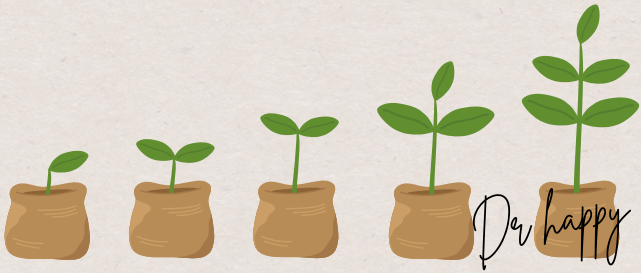
Tiếp theo, trung vị không phải lúc nào cũng nhận được sự chú ý xứng đáng như trung bình, nhưng lại là một chỉ số cực kỳ hữu ích khi bạn muốn biết giá trị "ở giữa" của tập dữ liệu. Khi các số liệu của bạn nhảy múa quá sôi nổi và không thể hiện rõ ràng xu hướng chung, trung vị chính là người bạn đáng tin cậy giúp bạn "tĩnh tâm".

Cuối cùng, mode có thể không xuất hiện thường xuyên như hai người bạn kia, nhưng khi nó làm, mode chính là "ngôi sao sáng". Mode cho bạn biết giá trị nào xuất hiện nhiều nhất trong tập dữ liệu của bạn, mang lại cái nhìn sâu sắc về xu hướng phổ biến nhất.

Khi kết hợp ba thước đo này, chúng mình không chỉ có cái nhìn toàn diện về dữ liệu mà còn có thể nắm bắt được những chi tiết tinh tế nhất, giúp đưa ra quyết định chính xác hơn trong mọi tình huống. Thật thú vị, phải không nào?

Thước đo phân tán:
Trong thế giới rộng lớn của dữ liệu, đôi khi chúng ta cần một chiếc la bàn để chỉ ra cách dữ liệu được phân tán trên bản đồ của thông tin. Đó chính là lúc các thước đo phân tán xuất hiện và tỏa sáng, giống như những người hướng dẫn du lịch dày dạn kinh nghiệm, giúp chúng ta hiểu được địa hình phức tạp của dữ liệu.

Đầu tiên, hãy nghĩ về "phạm vi" như khoảng cách giữa hai điểm xa nhất trên bản đồ dữ liệu của bạn. Đó là một cách trực quan để nhìn thấy sự chênh lệch lớn nhất, giống như khoảng cách từ điểm thấp nhất đến điểm cao nhất của một ngọn núi.



Tiếp theo, “phương sai“ và “độ lệch chuẩn“ là cặp bài trùng thể hiện sự dao động của dữ liệu quanh điểm trung tâm. Hãy tưởng tượng bạn đang đứng ở trung tâm của một quảng trường và phương sai là mức độ rộng lớn của không gian xung quanh bạn, còn độ lệch chuẩn giống như bước chân trung bình mà bạn cần thực hiện để khám phá mỗi góc của quảng trường đó.

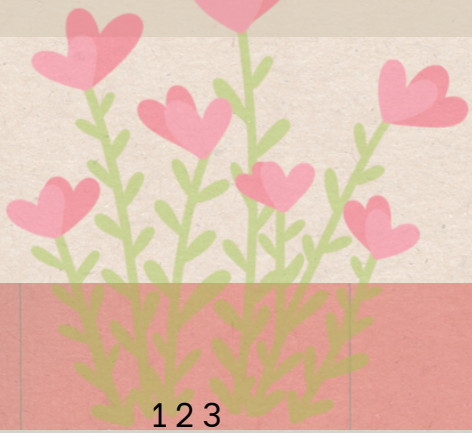
Cuối cùng, “biên độ tứ phân vị“ giống như một cuộc phiêu lưu khám phá, chia bản đồ dữ liệu của bạn thành bốn phần. Nó cho bạn biết về sự phân bố của dữ liệu trong từng phần, giúp bạn hiểu được những khu vực dữ liệu tập trung nhiều nhất và khu vực nào ít người lui tới hơn.

Bạn thấy đó phân tán không chỉ kể cho chúng ta nghe về câu chuyện của dữ liệu mà còn vẽ ra bản đồ vô cùng sinh động về sự đa dạng và phong phú của chúng.

Biểu đồ

Trong hành trình khám phá thế giới dữ liệu của mình, chúng mình đã gặp gỡ nhiều người bạn thú vị như các con số, thước đo, và giờ đây, chúng ta sẽ được gặp gỡ những người bạn đầy màu sắc và sinh động: các biểu đồ và đồ thị. Chúng không chỉ làm cho dữ liệu trở nên hấp dẫn mà còn giúp chúng ta “nhìn thấy“ câu chuyện đằng sau các con số một cách rõ ràng hơn.

Hãy tưởng tượng, chúng mình đang ở trong một khu vườn dữ liệu rộng lớn. Trong khu vườn ấy, các biểu đồ và đồ thị giống như những tấm bản đồ minh họa, giúp chúng mình không chỉ nhìn thấy mà còn “cảm nhận“ được cảnh vật xung quanh. Từ biểu đồ cột giống như những tòa nhà cao chọc trời, biểu đồ hình quạt giống như những lát bánh pizza đầy màu sắc, cho đến biểu đồ hộp và histogram giống như những dải phân cấp của địa hình, mỗi loại đều kể một câu chuyện riêng.



Be happy

PHẦN 1: THỐNG KÊ MÔ TẢ CƠ BẢN

Khi tạo ra một biểu đồ, chúng mình giống như những họa sĩ, sử dụng dữ liệu làm mực vẽ để tạo nên những bức tranh sinh động. Mỗi loại biểu đồ sẽ phù hợp với một loại câu chuyện nhất định: biểu đồ cột sẽ giúp so sánh giữa các nhóm, biểu đồ hình quạt thể hiện tỷ lệ của một toàn thể, còn histogram và biểu đồ hộp sẽ kể về phân phối và phạm vi của dữ liệu.

Nhưng công việc không dừng lại ở việc tạo ra chúng. Điều thú vị nhất nằm ở việc "đọc" và diễn giải những bức tranh dữ liệu này. Chúng ta sẽ học cách nhìn vào từng chi tiết, từ màu sắc đến hình dáng và vị trí, để hiểu sâu hơn về thông tin mà chúng muốn truyền đạt.

Qua đó, chúng mình không chỉ trang bị cho mình khả năng phân tích dữ liệu mà còn phát triển tư duy phê phán và khả năng diễn đạt, biến dữ liệu từ những con số trừu tượng thành những câu chuyện có ý nghĩa, dễ hiểu và dễ chia sẻ với mọi người.



Do happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Áp dụng những kiến thức đã học, chúng mình sẽ tiến hành một nhiệm vụ thực tế: Phân tích doanh số bán hàng của một công ty.

Công việc bao gồm: Bạn note lại nhé

Hiểu Dữ Liệu Doanh Số: Chúng mình sẽ cùng nhau khám phá cấu trúc và các đặc điểm chính của dữ liệu doanh số bán hàng.

Xử Lý Dữ Liệu: Bạn sẽ học cách làm sạch và chuẩn bị dữ liệu cho phân tích, để dữ liệu của chúng ta sẵn sàng cho các bước tiếp theo.

Phân Tích Xu Hướng Trung Tâm và Phân Tán: Áp dụng các thước đo mà bạn đã học để khám phá và diễn giải doanh số bán hàng.

Phân Tích Theo Thời Gian và Phân Khúc: Chúng ta sẽ xem xét xu hướng theo thời gian và phân tích doanh số dựa trên các phân khúc khác nhau như khu vực địa lý hoặc loại sản phẩm.

Kết Luận và Đề Xuất: Cuối cùng, chúng sẽ rút ra kết luận từ phân tích và đề xuất các biện pháp có thể giúp cải thiện doanh số bán hàng dựa trên bằng chứng từ dữ liệu.

Áp dụng những kiến thức đã học, chúng mình sẽ tiến hành một nhiệm vụ thực tế: Phân tích doanh số bán hàng của một công ty.

Công việc bao gồm: Bạn note lại nhé

Hiểu Dữ Liệu Doanh Số: Chúng mình sẽ cùng nhau khám phá cấu trúc và các đặc điểm chính của dữ liệu doanh số bán hàng.

Xử Lý Dữ Liệu: Bạn sẽ học cách làm sạch và chuẩn bị dữ liệu cho phân tích, để dữ liệu của chúng ta sẵn sàng cho các bước tiếp theo.

Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Hiểu Dữ Liệu Doanh Số
Dữ liệu doanh số bán hàng giống như một cuốn sách lịch sử, ghi chép lại hành trình của mỗi sản phẩm từ khi chào đời đến khi tìm thấy chủ nhân mới. Trong cuốn sách này, mỗi trang, mỗi dòng, mỗi từ đều mang thông tin quý giá về ngày bán, số lượng bán ra, giá bán, và đôi khi là phản hồi từ khách hàng.

Dữ Liệu Ví Dụ
Hãy tưởng tượng chúng ta có một bảng dữ liệu đơn giản về doanh số bán hàng như sau:

Ngày Sản Phẩm Số Lượng Doanh Thu
2021-01-01 Cà Phê Việt 10 100
2021-01-01 Cà Phê Mỹ 5 75
2021-01-02 Cà Phê Việt 15 150
2021-01-02 Cà Phê Brazil 7 105

Trong bảng này, mỗi hàng đại diện cho một giao dịch, với thông tin về ngày bán, loại sản phẩm, số lượng sản phẩm được bán và tổng doanh thu từ giao dịch đó.

Code Python để Đọc và Phân Tích Dữ Liệu
Giả sử bảng dữ liệu trên được lưu trong một tệp CSV có tên là doanh_so_ban_hang.csv. Dưới đây là cách bạn có thể sử dụng Python để đọc và phân tích dữ liệu:



PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
import pandas as pd

# Đọc dữ liệu từ tệp CSV
doanh_so = pd.read_csv('doanh_so_ban_hang.csv')

# Hiển thị 5 hàng đầu tiên để kiểm tra dữ liệu
print(doanh_so.head())

# Phân tích cơ bản: Tính tổng doanh thu và tổng số lượng sản phẩm bán ra
tong_doanh_thu = doanh_so['Doanh Thu'].sum()
tong_so_luong = doanh_so['Số Lượng'].sum()

print(f"Tổng doanh thu: {tong_doanh_thu}")
print(f"Tổng số lượng sản phẩm bán ra: {tong_so_luong}")

# Phân tích doanh số bán hàng theo sản phẩm
doanh_thu_san_pham = doanh_so.groupby('Sản Phẩm')['Doanh Thu'].sum()
print("Doanh thu theo sản phẩm:")
print(doanh_thu_san_pham)
```





PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
import pandas as pd

# Đọc dữ liệu từ tệp CSV
doanh_so = pd.read_csv('doanh_so_ban_hang.csv')

# Hiển thị 5 hàng đầu tiên để kiểm tra dữ liệu
print(doanh_so.head())

# Phân tích cơ bản: Tính tổng doanh thu và tổng số lượng sản phẩm bán ra
tong_doanh_thu = doanh_so['Doanh Thu'].sum()
tong_so_luong = doanh_so['Số Lượng'].sum()

print(f"Tổng doanh thu: {tong_doanh_thu}")
print(f"Tổng số lượng sản phẩm bán ra: {tong_so_luong}")

# Phân tích doanh số bán hàng theo sản phẩm
doanh_thu_san_pham = doanh_so.groupby('Sản Phẩm')['Doanh Thu'].sum()
print("Doanh thu theo sản phẩm:")
print(doanh_thu_san_pham)
```

Trong đoạn code trên, chúng ta đã sử dụng thư viện pandas để đọc và thao tác với dữ liệu. Sau khi đọc dữ liệu, chúng ta in ra 5 hàng đầu tiên để kiểm tra, sau đó tính tổng doanh thu và tổng số lượng sản phẩm bán ra. Cuối cùng, chúng ta phân tích doanh thu theo từng loại sản phẩm để xem sản phẩm nào mang lại doanh thu cao nhất.

Như vậy, thông qua việc khám phá và phân tích dữ liệu doanh số, chúng ta có thể hiểu rõ hơn về hiệu suất kinh doanh và tìm ra những cơ hội để tối ưu hóa doanh thu.

Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Xử lý dữ liệu:
Bước xử lý dữ liệu là một nhiệm vụ không thể thiếu, giống như việc chúng ta chuẩn bị hành trang trước khi lên đường. Mục tiêu là đảm bảo rằng "hành trang" dữ liệu của chúng ta sẽ sạch sẽ, gọn gàng và dễ dàng sử dụng nhất có thể.

Dữ Liệu Ví Dụ
Hãy tưởng tượng chúng ta có một bảng dữ liệu về doanh số bán hàng với một số vấn đề cần được xử lý:

Ngày Sản Phẩm Số Lượng Doanh Thu Ghi Chú
2021-01-01 Cà Phê Việt 10 100
2021-01-01 Cà Phê Mỹ - 75 Số lượng thiếu
2021-01-02 15 150 Sản phẩm thiếu
2021-01-02 Cà Phê Brazil 7 - Doanh thu thiếu
Cà Phê Việt 20 200 Ngày thiếu

Trong bảng này, chúng ta thấy một số vấn đề như giá trị thiếu ở các cột "Ngày", "Sản Phẩm", "Số Lượng" và "Doanh Thu".

Code Python để Làm Sạch và Chuẩn bị Dữ Liệu
Để xử lý dữ liệu này, chúng ta có thể sử dụng Python và thư viện pandas như sau:





PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
import pandas as pd
import numpy as np

# Giả sử dữ liệu được lưu trong 'doanh_so_ban_hang.csv'
doanh_so = pd.read_csv('doanh_so_ban_hang.csv')

# Kiểm tra dữ liệu thiếu
print(doanh_so.isnull().sum())

# Xử lý dữ liệu thiếu
doanh_so['Ngày'].fillna(method='ffill', inplace=True) # Điền giá trị thiếu ở cột 'Ngày' bằng giá trị ngày trước đó
doanh_so['Sản Phẩm'].fillna('Thông tin thiếu', inplace=True) # Điền 'Thông tin thiếu' cho Sản Phẩm thiếu
doanh_so['Số Lượng'].replace('-', np.nan, inplace=True) # Chuyển '-' thành NaN
doanh_so['Số Lượng'].fillna(doanh_so['Số Lượng'].median(), inplace=True) # Điền giá trị thiếu bằng trung vị của cột 'Số Lượng'
doanh_so['Doanh Thu'].replace('-', np.nan, inplace=True) # Chuyển '-' thành NaN
doanh_so['Doanh Thu'].fillna(doanh_so['Doanh Thu'].median(), inplace=True) # Điền giá trị thiếu bằng trung vị của cột 'Doanh Thu'

# Loại bỏ cột 'Ghi Chú' nếu không cần thiết
doanh_so.drop('Ghi Chú', axis=1, inplace=True)

# Kiểm tra lại dữ liệu sau khi xử lý
print(doanh_so)
```

Trong đoạn code trên, chúng ta sử dụng phương thức fillna() để điền giá trị thiếu, sử dụng replace() để thay thế các giá trị không hợp lệ như '-', và cuối cùng là loại bỏ cột "Ghi Chú" nếu như nó không cần thiết cho phân tích. Điều này giúp dữ liệu của chúng ta trở nên sạch sẽ và đầy đủ hơn, sẵn sàng cho những bước phân tích tiếp theo.

Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

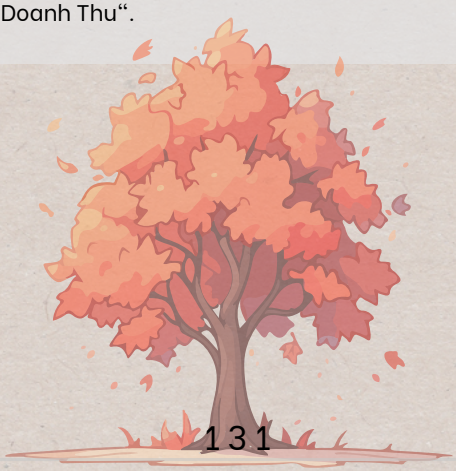
Phân tích xu hướng trung tâm và phân tán

Chúng ta sẽ bước vào giai đoạn phân tích xu hướng trung tâm và phân tán để khám phá những bí mật ẩn giấu trong doanh số bán hàng. Đây là bước giúp chúng ta hiểu rõ mức độ tập trung và sự đa dạng của dữ liệu, giống như cách bạn nhìn vào một bức tranh và cố gắng hiểu ý nghĩa đằng sau những nét vẽ.

Dữ Liệu Ví Dụ
Giả sử chúng ta có một bảng dữ liệu doanh số bán hàng như sau, sau khi đã được làm sạch và chuẩn bị:

Ngày Sản Phẩm Số Lượng Doanh Thu
2021-01-01 Cà Phê Việt 10 100
2021-01-01 Cà Phê Mỹ 5 75
2021-01-02 Cà Phê Việt 15 150
2021-01-02 Cà Phê Brazil 7 105
2021-01-03 Cà Phê Việt 20 200
Bảng này đã sẵn sàng cho việc phân tích, với các thông tin cơ bản về ngày bán, loại sản phẩm, số lượng sản phẩm được bán, và tổng doanh thu từ mỗi giao dịch.

Code Python để Phân Tích
Chúng ta sẽ sử dụng Python để tính các thước đo xu hướng trung tâm như trung bình và trung vị, cũng như các thước đo phân tán như phạm vi, phương sai và độ lệch chuẩn cho cột “Doanh Thu“.



Do happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
import pandas as pd

# Tạo DataFrame từ dữ liệu ví dụ
doanh_so = pd.DataFrame({
'Ngày': ['2021-01-01', '2021-01-01', '2021-01-02', '2021-01-02', '2021-01-03'],
'Sản Phẩm': ['Cà Phê Việt', 'Cà Phê Mỹ', 'Cà Phê Việt', 'Cà Phê Brazil', 'Cà Phê Việt'],
'Số Lượng': [10, 5, 15, 7, 20],
'Doanh Thu': [100, 75, 150, 105, 200]
})

# Tính các thước đo xu hướng trung tâm
trung_binh = doanh_so['Doanh Thu'].mean()
trung_vi = doanh_so['Doanh Thu'].median()

# Tính các thước đo phân tán
pham_vi = doanh_so['Doanh Thu'].max() - doanh_so['Doanh Thu'].min()
phuong_sai = doanh_so['Doanh Thu'].var()
do_lech_chuan = doanh_so['Doanh Thu'].std()

# Hiển thị kết quả
print(f'Trung bình Doanh Thu: {trung_binh}')
print(f'Trung vị Doanh Thu: {trung_vi}')
print(f'Phạm vi Doanh Thu: {pham_vi}')
print(f'Phương sai Doanh Thu: {phuong_sai}')
print(f'Độ lệch chuẩn Doanh Thu: {do_lech_chuan}')
```



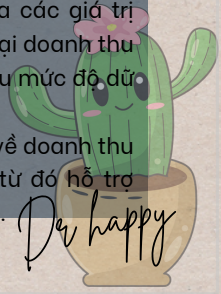
Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
import pandas as pd

# Tạo DataFrame từ dữ liệu ví dụ
doanh_so = pd.DataFrame({
'Ngày': ['2021-01-01', '2021-01-01', '2021-01-02', '2021-01-02', '2021-01-03'],
'Sản Phẩm': ['Cà Phê Việt', 'Cà Phê Mỹ', 'Cà Phê Việt', 'Cà Phê Brazil', 'Cà Phê Việt'],
'Số Lượng': [10, 5, 15, 7, 20],
'Doanh Thu': [100, 75, 150, 105, 200]
})

# Tính các thước đo xu hướng trung tâm
trung_binh = doanh_so['Doanh Thu'].mean()
trung_vi = doanh_so['Doanh Thu'].median()

# Tính các thước đo phân tán
pham_vi = doanh_so['Doanh Thu'].max() - doanh_so['Doanh Thu'].min()
phuong_sai = doanh_so['Doanh Thu'].var()
do_lech_chuan = doanh_so['Doanh Thu'].std()

# Hiển thị kết quả
print(f'Trung bình Doanh Thu: {trung_binh}')
print(f'Trung vị Doanh Thu: {trung_vi}')
print(f'Phạm vi Doanh Thu: {pham_vi}')
print(f'Phương sai Doanh Thu: {phuong_sai}')
print(f'Độ lệch chuẩn Doanh Thu: {do_lech_chuan}')
```

Qua việc tính toán, chúng ta có thể thấy mức độ tập trung (trung bình, trung vị) và mức độ đa dạng (phạm vi, phương sai, độ lệch chuẩn) của doanh thu từ bán hàng. Trung bình cho ta biết giá trị "trung bình" của doanh thu, trong khi trung vị giúp chúng ta hiểu giá trị "ở giữa" của dữ liệu, giảm thiểu ảnh hưởng của các giá trị ngoại lệ. Phạm vi cho chúng ta biết khoảng cách giữa giao dịch mang lại doanh thu cao nhất và thấp nhất, phương sai và độ lệch chuẩn giúp chúng ta hiểu mức độ dữ liệu phân tán quanh giá trị trung bình.

Phân tích này không chỉ giúp chúng ta hiểu được "bức tranh tổng thể" về doanh thu mà còn cung cấp cái nhìn sâu sắc về cách dữ liệu "chuyển động", từ đó hỗ trợ chúng ta trong việc đưa ra quyết định và chiến lược kinh doanh phù hợp.



PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Phân Tích Theo Thời Gian và Phân Khúc:

Chúng ta sẽ khám phá doanh số bán hàng qua thời gian và theo phân khúc, như một chuyến du hành qua thời gian và không gian để tìm hiểu nơi sản phẩm được yêu thích nhất và lúc nào chúng được săn đón nhiều nhất.

Dữ Liệu Ví Dụ

Giả sử chúng ta có bảng dữ liệu doanh số bán hàng như sau, bao gồm thông tin về ngày, khu vực địa lý, loại sản phẩm và doanh thu:

Ngày Khu Vực Sản Phẩm Doanh Thu
2021-01-01 Bắc Cà Phê Việt 100
2021-01-01 Nam Cà Phê Mỹ 150
2021-01-02 Bắc Cà Phê Việt 120
2021-01-02 Nam Cà Phê Brazil 180
2021-01-03 Bắc Cà Phê Việt 200

Code Python để Phân Tích

Chúng ta sẽ sử dụng Python để phân tích xu hướng theo thời gian và phân tích doanh số theo các phân khúc khác nhau, như khu vực địa lý và loại sản phẩm.

```
import pandas as pd
import matplotlib.pyplot as plt

# Tạo DataFrame từ dữ liệu ví dụ
doanh_so = pd.DataFrame({
'Ngày': ['2021-01-01', '2021-01-01', '2021-01-02', '2021-01-02', '2021-01-03'],
'Khu Vực': ['Bắc', 'Nam', 'Bắc', 'Nam', 'Bắc'],
'Sản Phẩm': ['Cà Phê Việt', 'Cà Phê Mỹ', 'Cà Phê Việt', 'Cà Phê Brazil', 'Cà Phê Việt'],
'Doanh Thu': [100, 150, 120, 180, 200]
})
```

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH TOÁN VÀ LẬP TRÌNH

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
# Chuyển đổi 'Ngày' sang định dạng datetime
doanh_so['Ngày'] = pd.to_datetime(doanh_so['Ngày'])

# Phân tích theo thời gian
doanh_so_theo_ngay = doanh_so.groupby('Ngày')['Doanh Thu'].sum()

# Phân tích theo phân khúc
doanh_so_theo_khuvuc = doanh_so.groupby('Khu Vực')['Doanh Thu'].sum()
doanh_so_theo_sanpham = doanh_so.groupby('Sản Phẩm')['Doanh Thu'].sum()

# Vẽ biểu đồ doanh số theo thời gian
plt.figure(figsize=(10, 6))
plt.plot(doanh_so_theo_ngay.index, doanh_so_theo_ngay.values, marker='o',
linestyle='-', color='b')
plt.title('Doanh Số Bán Hàng Theo Thời Gian')
plt.xlabel('Ngày')
plt.ylabel('Doanh Thu')
plt.grid(True)
plt.show()

# Vẽ biểu đồ doanh số theo khu vực
doanh_so_theo_khuvuc.plot(kind='bar', title='Doanh Số Bán Hàng Theo Khu Vực',
color='orange')
plt.xlabel('Khu Vực')
plt.ylabel('Doanh Thu')
plt.show()

# Vẽ biểu đồ doanh số theo sản phẩm
doanh_so_theo_sanpham.plot(kind='bar', title='Doanh Số Bán Hàng Theo Sản
Phẩm', color='green')
plt.xlabel('Sản Phẩm')
plt.ylabel('Doanh Thu')
plt.show()
```

Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

```
# Chuyển đổi 'Ngày' sang định dạng datetime
doanh_so['Ngày'] = pd.to_datetime(doanh_so['Ngày'])
```

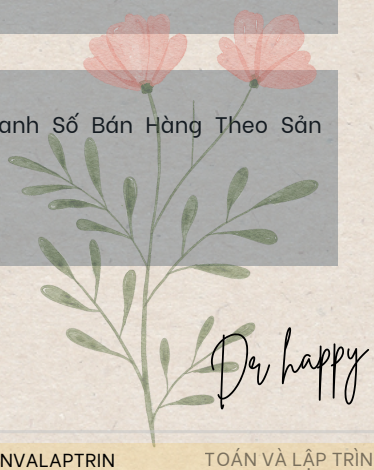
```
# Phân tích theo thời gian
doanh_so_theo_ngay = doanh_so.groupby('Ngày')['Doanh Thu'].sum()
```

```
# Phân tích theo phân khúc
doanh_so_theo_khuvuc = doanh_so.groupby('Khu Vực')['Doanh Thu'].sum()
doanh_so_theo_sanpham = doanh_so.groupby('Sản Phẩm')['Doanh Thu'].sum()
```

```
# Vẽ biểu đồ doanh số theo thời gian
plt.figure(figsize=(10, 6))
plt.plot(doanh_so_theo_ngay.index, doanh_so_theo_ngay.values, marker='o', linestyle='-', color='b')
plt.title('Doanh Số Bán Hàng Theo Thời Gian')
plt.xlabel('Ngày')
plt.ylabel('Doanh Thu')
plt.grid(True)
plt.show()
```

```
# Vẽ biểu đồ doanh số theo khu vực
doanh_so_theo_khuvuc.plot(kind='bar', title='Doanh Số Bán Hàng Theo Khu Vực', color='orange')
plt.xlabel('Khu Vực')
plt.ylabel('Doanh Thu')
plt.show()
```

```
# Vẽ biểu đồ doanh số theo sản phẩm
doanh_so_theo_sanpham.plot(kind='bar', title='Doanh Số Bán Hàng Theo Sản Phẩm', color='green')
plt.xlabel('Sản Phẩm')
plt.ylabel('Doanh Thu')
plt.show()
```



PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Trong đoạn code trên, chúng ta đã sử dụng thư viện pandas để tổ chức và phân tích dữ liệu, và matplotlib để vẽ các biểu đồ, giúp chúng ta có cái nhìn trực quan về doanh số theo thời gian, theo khu vực và theo loại sản phẩm. Điều này giúp chúng ta dễ dàng nhận ra xu hướng và đặc điểm nổi bật trong dữ liệu, từ đó đưa ra quyết định tối ưu cho doanh nghiệp.

Vẽ các dạng biểu đồ cho chúng ta những thông tin gì?

Dạng 1: Doanh số bán hàng theo thời gian

Xu hướng: Biểu đồ giúp xác định xu hướng chung của doanh số bán hàng qua các ngày. Bạn có thể thấy doanh số tăng lên, giảm xuống, hay ổn định qua thời gian. Một đường cong đi lên cho thấy sự tăng trưởng, trong khi một đường cong đi xuống cho thấy sự suy giảm.

Mùa vụ và Sự Kiện: Biểu đồ cũng giúp bạn nhận diện mẫu mùa vụ hoặc ảnh hưởng của các sự kiện đặc biệt (như ngày lễ, chiến dịch quảng cáo) đến doanh số bán hàng. Ví dụ, bạn có thể thấy doanh số tăng vọt vào dịp lễ Tết hoặc Giáng Sinh.

So sánh: Nếu bạn có dữ liệu từ nhiều năm, biểu đồ cho phép bạn so sánh doanh số giữa các năm hoặc các tháng khác nhau để xác định bất kỳ mẫu lặp lại nào hoặc sự khác biệt đáng chú ý.

Điểm Đột Phá và Sự Cố: Biểu đồ giúp bạn nhận biết các điểm đột phá hoặc sự cố bất thường, nơi doanh số bất ngờ tăng vọt hoặc giảm sâu. Điều này có thể yêu cầu điều tra thêm để hiểu nguyên nhân - có thể là do sự cố về hàng tồn kho, thay đổi trong chiến lược kinh doanh, hoặc các yếu tố bên ngoài.

Quyết định Chiến lược: Cuối cùng, phân tích biểu đồ này giúp người quản lý và nhà phân tích đưa ra quyết định chiến lược dựa trên dữ liệu, như điều chỉnh sản xuất, tối ưu hóa hàng tồn kho, hoặc điều chỉnh chiến lược tiếp thị và bán hàng.

Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY



Dạng 2: Doanh số bán hàng theo khu vực

Phân bổ Doanh số: Biểu đồ giúp xác định khu vực nào có doanh số cao nhất và thấp nhất, cho phép bạn nhìn thấy sự phân bổ doanh số một cách rõ ràng.

Khu Vực Chính: Bạn có thể xác định được khu vực nào là "động lực" chính đằng sau doanh số bán hàng của công ty, nơi tập trung phần lớn doanh thu.

Cơ hội Tăng trưởng: Biểu đồ cũng giúp phát hiện các khu vực có tiềm năng tăng trưởng cao mà có thể chưa được khai thác triệt để, dựa trên so sánh doanh thu giữa các khu vực.

So sánh Khu vực: Cho phép so sánh hiệu suất bán hàng trực tiếp giữa các khu vực, giúp định hình chiến lược kinh doanh và tiếp thị để tối ưu hóa doanh thu ở từng khu vực.

Đánh giá Chiến lược Khu vực: Dựa trên thông tin từ biểu đồ, bạn có thể đánh giá hiệu quả của các chiến lược tiếp thị và phân phối hiện tại ở các khu vực khác nhau, và điều chỉnh chúng nếu cần.

Cân nhắc Về Ngân Sách và Tài nguyên: Biểu đồ giúp quyết định cách phân bổ ngân sách và tài nguyên một cách hiệu quả, tập trung vào khu vực có doanh thu cao hoặc có tiềm năng phát triển mạnh.

Nhìn chung, biểu đồ doanh số theo khu vực cung cấp cái nhìn toàn diện về cách thức doanh thu được tạo ra trên địa bàn, giúp bạn đưa ra quyết định dựa trên dữ liệu cụ thể và chiến lược hóa việc tăng trưởng doanh thu ở các khu vực khác nhau.

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Dạng 3: Biểu đồ doanh số bán hàng theo sản phẩm.

Biểu đồ doanh số bán hàng theo sản phẩm, cung cấp một cái nhìn trực quan và dễ hiểu về hiệu suất bán hàng của từng sản phẩm trong danh mục của công ty. Dưới đây là một số thông tin và hiểu biết quan trọng mà biểu đồ này mang lại:

Hiệu Suất Bán Hàng của Từng Sản Phẩm: Biểu đồ giúp xác định sản phẩm nào đang bán chạy và sản phẩm nào kém hiệu suất. Điều này quan trọng trong việc định hình chiến lược sản phẩm và marketing.

Ưu Tiên Sản Xuất và Tồn Kho: Dựa vào doanh số, công ty có thể quyết định sản xuất nhiều hơn các sản phẩm bán chạy và giảm bớt hoặc tái thiết kế sản phẩm kém hiệu suất.

Phân Khúc Thị Trường: Sự phổ biến của các sản phẩm khác nhau có thể phản ánh sở thích của khách hàng và phân khúc thị trường mà công ty đang phục vụ. Điều này giúp công ty điều chỉnh sản phẩm và dịch vụ để phù hợp hơn với nhu cầu của khách hàng.

Chiến Lược Giá và Khuyến Mãi: Biểu đồ cho thấy sản phẩm nào có thể cần giảm giá hoặc chương trình khuyến mãi để tăng doanh số. Ngược lại, sản phẩm nào có doanh số tốt có thể không cần chiết khấu mạnh.

Đầu Tư vào Phát Triển Sản Phẩm: Những sản phẩm có doanh số cao có thể là ứng viên sáng giá cho việc đầu tư thêm vào R&D để cải tiến hoặc phát triển phiên bản mới, trong khi những sản phẩm kém hiệu suất có thể cần được đánh giá lại hoặc ngừng sản xuất.

So Sánh và Đối Chiếu: Nếu có dữ liệu từ nhiều kỳ, biểu đồ giúp so sánh hiệu suất bán hàng của sản phẩm qua các kỳ khác nhau, từ đó nhìn nhận được xu hướng tăng trưởng hay suy giảm của từng sản phẩm.

Biểu đồ doanh số theo sản phẩm là một công cụ quan trọng giúp công ty hiểu rõ hơn về cách khách hàng tương tác với sản phẩm của mình, qua đó đưa ra những quyết định chiến lược nhằm tối ưu hóa doanh số và tăng trưởng.

Be happy

PHẦN 2: PHÂN TÍCH DOANH SỐ BÁN HÀNG CÔNG TY

Thật tuyệt vời khi cùng bạn đồng hành qua ngày thứ 8 đầy thú vị và học hỏi này! Mỗi bước đi trên hành trình khám phá dữ liệu đã mở ra những cánh cửa kiến thức mới, giúp chúng ta nhìn thấy và hiểu hơn về thế giới thông tin quanh ta. Mình hy vọng rằng những gì chúng ta đã cùng nhau tìm hiểu và khám phá hôm nay không chỉ giúp bạn tích lũy thêm kiến thức mà còn truyền cảm hứng để bạn tiếp tục hành trình học hỏi của mình.

Chúng mình còn 2 ngày nữa để hoàn thành chương trình, và mình tin rằng những ngày tiếp theo sẽ còn nhiều điều thú vị và bổ ích đang chờ đợi. Mỗi ngày là một cơ hội mới để phát triển và mở rộng tầm nhìn của bản thân.

Cảm ơn bạn đã dành thời gian cùng mình hôm nay. Mình mong chờ được gặp lại bạn vào ngày thứ 9, nơi chúng ta sẽ tiếp tục hành trình khám phá và học hỏi này. Hãy nghỉ ngơi thật tốt và chuẩn bị sẵn sàng cho những kiến thức mới vào ngày mai. Hẹn gặp lại bạn!

Dr happy

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Matplotlib: Là một thư viện vẽ đồ họa 2D trong Python, cho phép bạn tạo ra nhiều loại biểu đồ khác nhau, từ đơn giản đến phức tạp, với khả năng tùy chỉnh cao

1

Matplotlib

2

Seaborn

🎨 Phần 1: Matplotlib/Seaborn cho dữ liệu đồ họa

Trong phần này, bạn sẽ được giới thiệu đến hai thư viện phổ biến trong Python dành cho trực quan hóa dữ liệu: Matplotlib và Seaborn.

Matplotlib: Là một thư viện vẽ đồ họa 2D trong Python, cho phép bạn tạo ra nhiều loại biểu đồ khác nhau, từ đơn giản đến phức tạp, với khả năng tùy chỉnh cao. Bạn sẽ học cách tạo các biểu đồ cơ bản như biểu đồ đường, biểu đồ cột, và biểu đồ phân tán.

Seaborn: Là một thư viện dựa trên Matplotlib, được tối ưu hóa để tạo ra các biểu đồ thống kê đẹp mắt một cách dễ dàng. Seaborn cung cấp một giao diện cao cấp, giúp trực quan hóa dữ liệu phức tạp trở nên đơn giản và trực quan hơn.

📉 Phần 2: Vẽ biểu đồ xu hướng thị trường chứng khoán

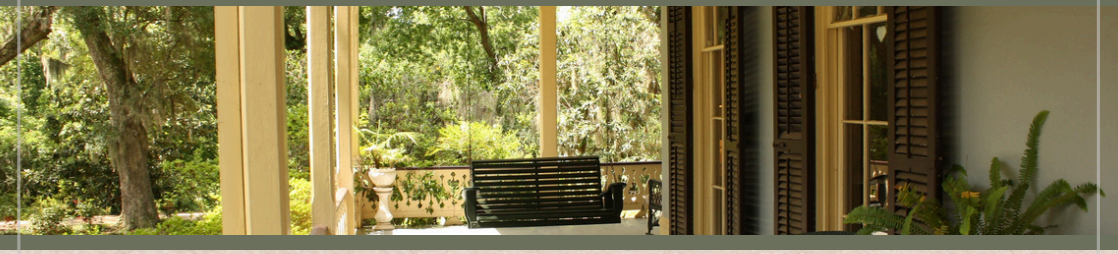
Đây là một bài toán thực tế, nơi bạn sẽ sử dụng dữ liệu về giá cổ phiếu qua thời gian để: Tạo biểu đồ đường thể hiện xu hướng giá cổ phiếu. Sử dụng biểu đồ cột để phân tích khối lượng giao dịch.

Khám phá mối quan hệ giữa các chỉ số tài chính khác nhau thông qua biểu đồ phân tán.

Qua đó, bạn không chỉ rèn luyện kỹ năng trực quan hóa dữ liệu mà còn có cơ hội áp dụng vào phân tích tài chính, một ứng dụng thực tiễn và giá trị của trực quan hóa.

Be happy

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Phần 1: Matplotlib/Seaborn cho dữ liệu đồ họa

Giới thiệu về Matplotlib: Matplotlib là một thư viện rất mạnh mẽ và linh hoạt trong Python, cho phép bạn tạo ra các biểu đồ đồ họa 2D chất lượng cao. Từ các biểu đồ cơ bản như biểu đồ đường để thể hiện xu hướng dữ liệu, biểu đồ cột để so sánh giữa các nhóm, đến biểu đồ phân tán để khám phá mối quan hệ giữa hai biến, Matplotlib có thể giúp bạn thực hiện tất cả với vài dòng code.

Dữ Liệu Ví Dụ

Giả sử chúng ta có dữ liệu doanh số bán hàng hàng tháng cho hai sản phẩm: Cà Phê Việt và Cà Phê Mỹ, từ tháng 1 đến tháng 6 như sau:

Tháng Cà Phê Việt Cà Phê Mỹ

1 120 150

2 135 160

3 150 155

4 145 170

5 160 165

6 175 180

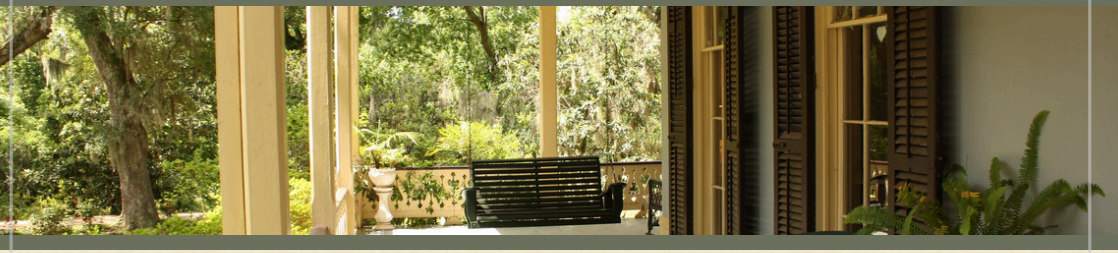
Mô tả Biểu Đồ

Dựa trên dữ liệu này, chúng ta sẽ tạo một biểu đồ đường để thể hiện xu hướng doanh số bán hàng hàng tháng cho hai sản phẩm. Mỗi dòng trên biểu đồ sẽ đại diện cho một sản phẩm, với trục x là tháng và trục y là doanh số.

```
import matplotlib.pyplot as plt
# Dữ liệu
thang = ['Tháng 1', 'Tháng 2', 'Tháng 3', 'Tháng 4', 'Tháng 5', 'Tháng 6']
ca_phe_viet = [120, 135, 150, 145, 160, 175]
ca_phe_my = [150, 160, 155, 170, 165, 180]
# Tạo biểu đồ
plt.figure(figsize=(10, 6))
plt.plot(thang, ca_phe_viet, marker='o', linestyle='-', color='b', label='Cà Phê Việt')
plt.plot(thang, ca_phe_my, marker='s', linestyle='--', color='r', label='Cà Phê Mỹ')

# Đặt tiêu đề và nhãn
plt.title('Doanh Số Bán Hàng Hàng Tháng')
plt.xlabel('Tháng')
plt.ylabel('Doanh Số')
plt.legend()
plt.grid(True)

# Hiển thị biểu đồ
plt.show()
```



DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Phần 1: Matplotlib/Seaborn cho dữ liệu đồ họa

Giới thiệu về Matplotlib

Matplotlib là một thư viện rất mạnh mẽ và linh hoạt trong Python, cho phép bạn tạo ra các biểu đồ đồ họa 2D chất lượng cao. Từ các biểu đồ cơ bản như biểu đồ đường để thể hiện xu hướng dữ liệu, biểu đồ cột để so sánh giữa các nhóm, đến biểu đồ phân tán để khám phá mối quan hệ giữa hai biến, Matplotlib có thể giúp bạn thực hiện tất cả với vài dòng code.

Dữ Liệu Ví Dụ

Giả sử chúng ta có dữ liệu doanh số bán hàng hàng tháng cho hai sản phẩm: Cà Phê Việt và Cà Phê Mỹ, từ tháng 1 đến tháng 6 như sau:

Tháng	Cà Phê Việt	Cà Phê Mỹ
1	120	150
2	135	160
3	150	155
4	145	170
5	160	165
6	175	180

Mô tả Biểu Đồ

Dựa trên dữ liệu này, chúng ta sẽ tạo một biểu đồ đường để thể hiện xu hướng doanh số bán hàng hàng tháng cho hai sản phẩm. Mỗi dòng trên biểu đồ sẽ đại diện cho một sản phẩm, với màu sắc khác nhau để dễ

```
import matplotlib.pyplot as plt
# Dữ liệu
thang = ['Tháng 1', 'Tháng 2', 'Tháng 3', 'Tháng 4', 'Tháng 5', 'Tháng 6']
ca_phe_viet = [120, 135, 150, 145, 160, 175]
ca_phe_my = [150, 160, 155, 170, 165, 180]
```

```
# Tạo biểu đồ
plt.figure(figsize=(10, 6))
plt.plot(thang, ca_phe_viet, marker='o', linestyle='-', color='b', label='Cà Phê Việt')
plt.plot(thang, ca_phe_my, marker='s', linestyle='--', color='r', label='Cà Phê Mỹ')
# Đặt tiêu đề và nhãn
plt.title('Doanh Số Bán Hàng Hàng Tháng')
plt.xlabel('Tháng')
plt.ylabel('Doanh Số')
plt.legend()
plt.grid(True)
# Hiển thị biểu đồ
plt.show()
```

Trong đoạn code này, chúng ta đã sử dụng plt.plot() để tạo biểu đồ đường, với marker để đánh dấu điểm dữ liệu, linestyle để xác định kiểu đường nối, và color để chọn màu cho mỗi dòng. plt.legend() được sử dụng để thêm chú giải và plt.grid(True) giúp thêm lưới vào biểu đồ, làm cho dữ liệu dễ đọc hơn.

Biểu đồ này giúp chúng ta dễ dàng nhận ra xu hướng tăng trưởng của doanh số bán hàng qua các tháng cho cả hai loại cà phê, cũng như so sánh hiệu suất bán hàng giữa chúng.

CONTACT: (+84) 9110 2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRIN TOÁN VÀ LẬP TRÌNH

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Phần 1: Matplotlib/Seaborn cho dữ liệu đồ họa

Ví Dụ Data:

Giả sử bạn có dữ liệu về doanh số bán hàng của 3 loại cà phê khác nhau trong một tháng, như sau:

```
Loại Cà Phê Doanh Số (túi)
Cà Phê Việt 150
Cà Phê Brazil 120
Cà Phê Mỹ 180
```

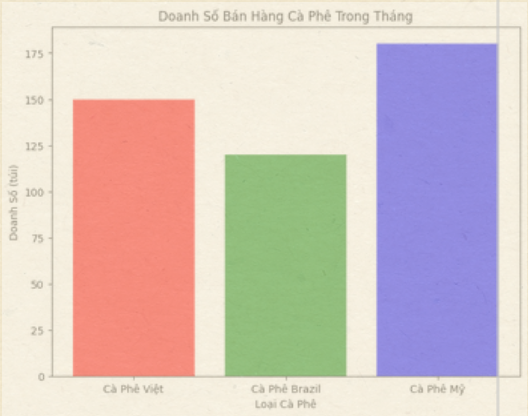
Mô Tả Hình Vẽ:

Biểu đồ cột sẽ có 3 cột, mỗi cột đại diện cho một loại cà phê. Chiều cao của cột "Cà Phê Việt" sẽ tương ứng với 150 túi, "Cà Phê Brazil" là 120 túi, và "Cà Phê Mỹ" là 180 túi. Các cột có thể được tô màu khác nhau để dễ dàng phân biệt.

Code Python:

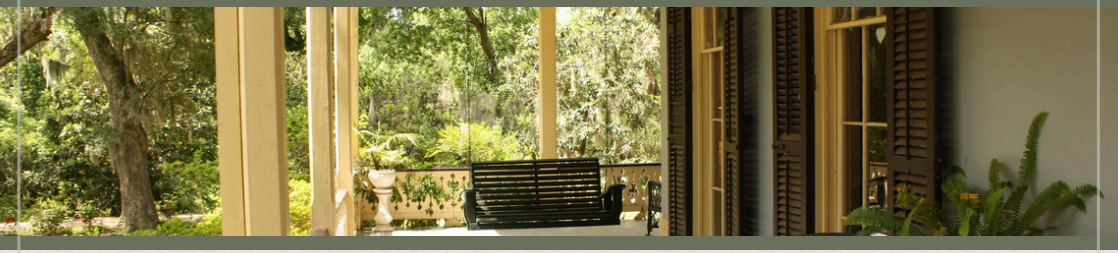
Sử dụng thư viện Matplotlib để vẽ biểu đồ cột từ dữ liệu trên:

```
Python Copy code
import matplotlib.pyplot as plt

# Dữ liệu
loai_ca_phe = ['Cà Phê Việt', 'Cà Phê Brazil', 'Cà Phê Mỹ']
doanh_so = [150, 120, 180]

# Tạo biểu đồ cột
plt.figure(figsize=(8, 6))
plt.bar(loai_ca_phe, doanh_so, color=['red', 'green', 'blue'])

# Đặt tiêu đề cho biểu đồ và các trục
plt.title('Doanh Số Bán Hàng Cà Phê Trong Tháng')
plt.xlabel('Loại Cà Phê')
plt.ylabel('Doanh Số (túi)')

# Hiển thị biểu đồ
plt.show()
```



Do happy

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Phần 1: Matplotlib/Seaborn cho dữ liệu đồ họa

Giới thiệu về Seaborn:

Seaborn là một thư viện trực quan hóa dữ liệu trong Python dựa trên Matplotlib, nhưng cung cấp một giao diện cao cấp hơn, giúp tạo ra các biểu đồ thống kê đẹp mắt một cách dễ dàng hơn. Seaborn tập trung vào việc tạo ra các biểu đồ thống kê nhanh chóng và dễ dàng, với giao diện và cách sử dụng dễ hiểu, cùng khả năng tùy chỉnh cao.

Ví Dụ Data:

Giả sử bạn có dữ liệu về điểm số của học sinh trong các bài kiểm tra Toán và Văn từ một lớp học, như sau:

Học Sinh Điểm Toán Điểm Văn
HS1 8 7
HS2 7 8
HS3 9 6
HS4 6 9
HS5 8 8

Code Python:

Chúng ta sẽ sử dụng Seaborn để vẽ biểu đồ phân tán (scatter plot) thể hiện mối quan hệ giữa điểm Toán và điểm Văn của học sinh.

Python Copy code
```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# Tạo DataFrame từ dữ liệu
data = pd.DataFrame({
'Học Sinh': ['HS1', 'HS2', 'HS3', 'HS4', 'HS5'],
'Điểm Toán': [8, 7, 9, 6, 8],
'Điểm Văn': [7, 8, 6, 9, 8]
})

# Vẽ biểu đồ phân tán sử dụng Seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='Điểm Toán', y='Điểm Văn', hue='Học Sinh', palette='deep', s=100)

# Đặt tiêu đề và nhãn cho biểu đồ
plt.title('Mối Quan Hệ giữa Điểm Toán và Điểm Văn của Học Sinh')
plt.xlabel('Điểm Toán')
plt.ylabel('Điểm Văn')

plt.legend(title='Học Sinh')
plt.show()
```

Trong đoạn code trên, sns.scatterplot được sử dụng để tạo biểu đồ phân tán, với trục x là điểm Toán, trục y là điểm Văn, và màu sắc của các điểm biểu diễn từng học sinh cụ thể. Biểu đồ giúp chúng ta thấy mối quan hệ giữa hai môn học và cách điểm số phân bố giữa các học sinh. Seaborn tự động cung cấp một biểu đồ đẹp mắt và dễ hiểu, làm nổi bật các mối quan hệ và xu hướng trong dữ liệu.

Do happy

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Phần 2: Vẽ biểu đồ xu hướng thị trường chứng khoán

Để vẽ biểu đồ xu hướng thị trường chứng khoán, chúng ta cần dữ liệu giá cổ phiếu theo thời gian. Dưới đây là một ví dụ giả định về dữ liệu giá cổ phiếu hàng ngày của một công ty trong một tuần.

Dữ Liệu Ví Dụ:
Ngày Giá Đóng Cửa
2021-01-01 100
2021-01-02 105
2021-01-03 102
2021-01-04 108
2021-01-05 110

Giả sử dữ liệu này được lưu trong một tệp CSV có tên là gia_co_phieu.csv.

Các Bước Làm:

Đọc Dữ Liệu: Sử dụng pandas để đọc dữ liệu từ tệp CSV.

Chuẩn bị Dữ Liệu: Chuyển đổi cột 'Ngày' thành kiểu dữ liệu datetime để dễ dàng vẽ biểu đồ thời gian.

Vẽ Biểu Đồ: Sử dụng Matplotlib và Seaborn để vẽ biểu đồ đường thể hiện xu hướng giá đóng cửa theo thời gian.

Code Python:

Sử dụng Matplotlib:

```
import pandas as pd
import matplotlib.pyplot as plt

# Đọc dữ liệu
df = pd.read_csv('gia_co_phieu.csv')
df['Ngày'] = pd.to_datetime(df['Ngày'])

# Vẽ biểu đồ sử dụng Matplotlib
plt.figure(figsize=(10, 6))
plt.plot(df['Ngày'], df['Giá Đóng Cửa'], marker='o', linestyle='-', color='b')
plt.title('Xu Hướng Giá Cổ Phiếu Theo Thời Gian - Matplotlib')
plt.xlabel('Ngày')
plt.ylabel('Giá Đóng Cửa')
plt.grid(True)
plt.show()
```

Sử dụng Seaborn:

Python Copy code

```
import seaborn as sns

# Vẽ biểu đồ sử dụng Seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x='Ngày', y='Giá Đóng Cửa', marker='o', color='r')
plt.title('Xu Hướng Giá Cổ Phiếu Theo Thời Gian - Seaborn')
plt.xlabel('Ngày')
plt.ylabel('Giá Đóng Cửa')
plt.grid(True)
plt.show()
```



Phần 2: Vẽ biểu đồ xu hướng thị trường chứng khoán

Trong cả hai đoạn code, chúng ta đều vẽ biểu đồ đường thể hiện giá đóng cửa theo thời gian. Sự khác biệt chính giữa Matplotlib và Seaborn nằm ở cách thức và mức độ tùy chỉnh: Seaborn cung cấp một cách tiếp cận dễ dàng hơn và tự động hóa nhiều tùy chỉnh, trong khi Matplotlib cho phép điều chỉnh tinh tế hơn nhưng đòi hỏi nhiều mã code hơn.

Dưới đây là một ví dụ về dữ liệu chứng khoán giả định, nơi chúng ta sẽ xem xét không chỉ giá đóng cửa mà còn khối lượng giao dịch hàng ngày của một cổ phiếu cụ thể. Điều này sẽ cung cấp cho chúng ta cái nhìn sâu sắc hơn về hoạt động thị trường, bao gồm sự quan tâm và tính thanh khoản của cổ phiếu.

Dữ Liệu Ví Dụ:
Ngày Giá Đóng Cửa Khối Lượng
2021-01-01 100 2500
2021-01-02 105 3000
2021-01-03 102 2000
2021-01-04 108 3500
2021-01-05 110 4000
Giả sử dữ liệu này được lưu trong một tệp CSV có tên là chung_khoan.csv.

Các Bước Thực Hiện:
Đọc Dữ Liệu: Dùng pandas để đọc dữ liệu từ tệp CSV.
Chuẩn bị Dữ Liệu: Chuyển đổi cột 'Ngày' sang kiểu datetime.
Vẽ Biểu Đồ Đường: Sử dụng Matplotlib và Seaborn để vẽ biểu đồ đường thể hiện giá đóng cửa theo thời gian.
Vẽ Biểu Đồ Cột: Sử dụng Matplotlib và Seaborn để vẽ biểu đồ cột thể hiện khối lượng giao dịch theo thời gian.

Code Python:
Sử dụng Matplotlib:
Python Copy code

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Đọc và chuẩn bị dữ liệu
df = pd.read_csv('chung_khoan.csv')
df['Ngày'] = pd.to_datetime(df['Ngày'])

# Vẽ biểu đồ giá đóng cửa
plt.figure(figsize=(14, 7))
plt.subplot(2, 1, 1) # 2 hàng, 1 cột, vị trí thứ 1
plt.plot(df['Ngày'], df['Giá Đóng Cửa'], marker='o', linestyle='-', color='b')
plt.title('Giá Đóng Cửa Theo Thời Gian')
plt.xlabel('Ngày')
plt.ylabel('Giá Đóng Cửa')
```



Phần 2: Vẽ biểu đồ xu hướng thị trường chứng khoán

```
# Vẽ biểu đồ khối lượng giao dịch
plt.subplot(2, 1, 2) # 2 hàng, 1 cột, vị trí thứ 2
plt.bar(df['Ngày'], df['Khối Lượng'], color='g')
plt.title('Khối Lượng Giao Dịch Theo Thời Gian')
plt.xlabel('Ngày')
plt.ylabel('Khối Lượng')

plt.tight_layout()
plt.show()
```

Sử dụng Seaborn:

```
import seaborn as sns
# Vẽ biểu đồ giá đóng cửa với Seaborn
plt.figure(figsize=(14, 7))
plt.subplot(2, 1, 1)
sns.lineplot(data=df, x='Ngày', y='Giá Đóng Cửa', marker='o', color='r')

plt.tight_layout()
plt.show()
```

Trong cả hai đoạn code, biểu đồ đầu tiên thể hiện giá đóng cửa của cổ phiếu theo thời gian, cung cấp thông tin về xu hướng giá. Biểu đồ thứ hai thể hiện khối lượng giao dịch, giúp ta hiểu về mức độ hoạt động của thị trường và sự quan tâm của nhà đầu tư đối với cổ phiếu này.



Be happy

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Trong ngày thứ 9 của chương trình học, chúng ta đã khám phá thế giới đầy màu sắc và sáng tạo của nghệ thuật trực quan hóa dữ liệu. Dưới đây là những điểm chính mà chúng ta đã học:

📝 Take Note Ngày 9: Nghệ Thuật Trực Quan Hóa Dữ Liệu

🎨 Thư Viện Matplotlib:

Giới Thiệu: Thư viện vẽ đồ họa 2D trong Python, hỗ trợ nhiều loại biểu đồ với khả năng tùy chỉnh cao.

Biểu Đồ Cơ Bản:

Biểu đồ đường: plt.plot(), thể hiện xu hướng dữ liệu.

Biểu đồ cột: plt.bar(), so sánh giữa các nhóm.

Biểu đồ phân tán: plt.scatter(), mối quan hệ giữa các biến.

Customization:

Thêm tiêu đề: plt.title()

Nhãn trục: plt.xlabel(), plt.ylabel()

Chú thích: plt.legend()

Màu sắc và định dạng: thông qua các tham số trong hàm vẽ.

🌈 Thư Viện Seaborn:

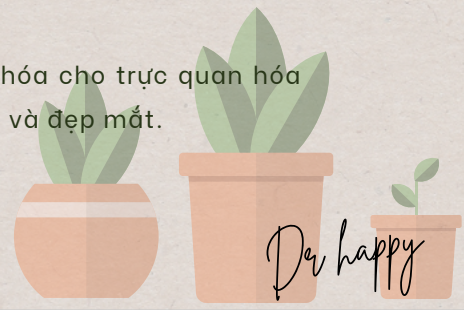
Giới Thiệu: Dựa trên Matplotlib, tối ưu hóa cho trực quan hóa dữ liệu thống kê, với giao diện đơn giản và đẹp mắt.

Biểu Đồ Cơ Bản:

Biểu đồ phân tán: sns.scatterplot()

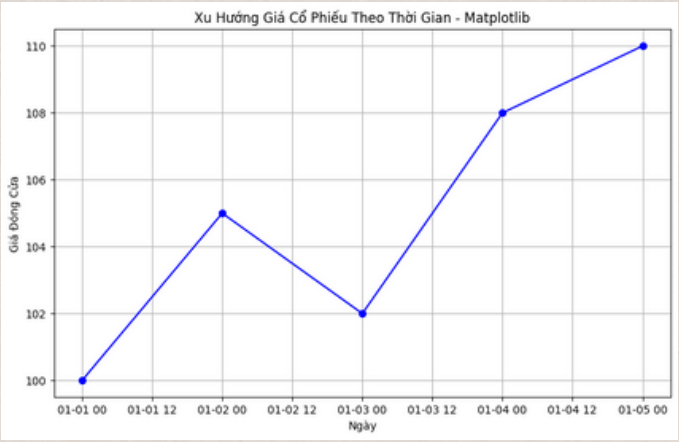
Biểu đồ đường: sns.lineplot()

Biểu đồ cột: sns.barplot()

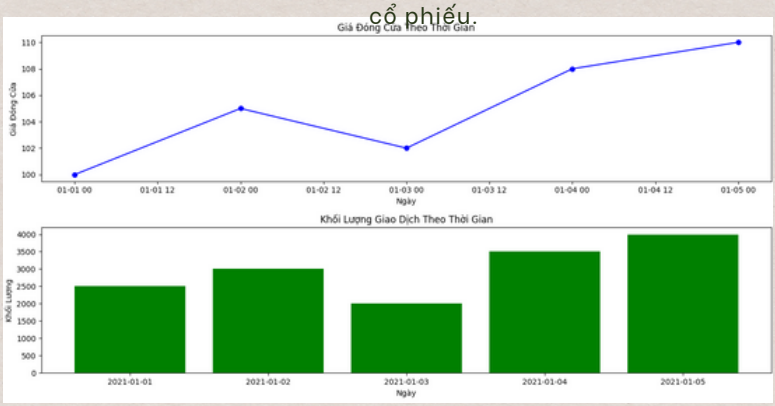


DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU





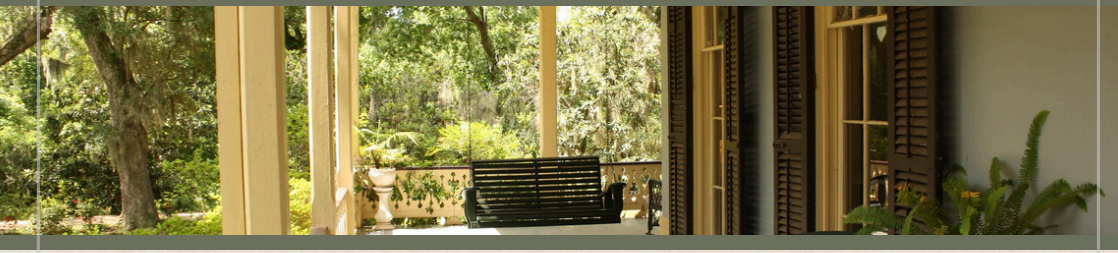
Kết quả về dữ liệu chứng khoán giả định, biểu đồ sẽ cung cấp cho chúng ta cái nhìn sâu sắc hơn về hoạt động thị trường, bao gồm sự quan tâm và tính thanh khoản của cổ phiếu.



Kết quả về dữ liệu chứng khoán thể hiện khối lượng giao dịch, giúp ta hiểu về mức độ hoạt động của thị trường và sự quan tâm của nhà đầu tư đối với cổ phiếu này.

Be happy

DAY 9: NGHỆ THUẬT TRỰC QUAN HÓA DỮ LIỆU



Phần 2: Vẽ biểu đồ xu hướng thị trường chứng khoán

🌈 Thư Viện Seaborn:

Các Tính Năng Đặc Biệt:

Tự động tính toán và vẽ mô hình hồi quy: sns.regplot()

Vẽ biểu đồ phân phối: sns.distplot()

Biểu đồ hộp: sns.boxplot(), phân phối và giá trị ngoại lệ của dữ liệu.

Ứng Dụng Trong Phân Tích Thị Trường Chứng Khoán:

Phân Tích Xu Hướng: Sử dụng biểu đồ đường để thể hiện xu hướng giá cổ phiếu theo thời gian.

Khối Lượng Giao Dịch: Biểu đồ cột thể hiện khối lượng giao dịch hàng ngày, phản ánh mức độ hoạt động của thị trường.

So Sánh Giữa Các Sản Phẩm: Dùng biểu đồ cột hoặc biểu đồ phân tán để so sánh giữa các loại cổ phiếu hoặc chỉ số.

💡 Tips:

Luôn kiểm tra và làm sạch dữ liệu trước khi vẽ biểu đồ.

Sử dụng màu sắc và hình dạng điểm để phân biệt giữa các loại dữ liệu.

Tận dụng chú thích và tiêu đề để biểu đồ dễ hiểu hơn.

Ngày thứ 9 đã mang đến một cái nhìn sâu sắc về cách thức trực quan hóa dữ liệu không chỉ làm cho dữ liệu trở nên sống động hơn mà còn giúp chúng ta hiểu sâu hơn về dữ liệu mà chúng ta đang phân tích. Hy vọng bạn đã tìm thấy niềm vui và giá trị từ những kiến thức hôm nay! Hẹn gặp lại vào ngày tiếp theo! 🚀







Be happy



DỰ ÁN VÀ TÌNH HUỐNG

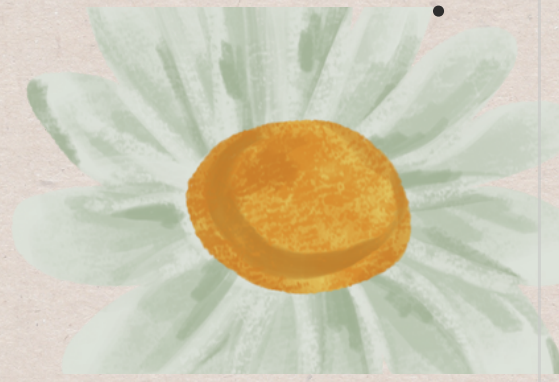
Dự án lần này sẽ tập trung vào việc phân tích và dự báo doanh số bán hàng của một công ty thương mại điện tử, sử dụng những kiến thức mà chúng ta đã học trong 9 ngày qua.

day 10

Do happy

TÌNH HUỐNG DỮ LIỆU MINH HỌA





📝 Tình Huống:

Công ty ABC là một công ty thương mại điện tử, kinh doanh nhiều loại sản phẩm khác nhau từ điện tử, gia dụng, thời trang đến thực phẩm. Với dữ liệu doanh số bán hàng hàng tháng của một số sản phẩm trong năm vừa qua, công ty muốn phân tích hiệu suất kinh doanh, xác định các sản phẩm chính và dự báo doanh số cho tháng tiếp theo.

📈 Dữ Liệu Minh Họa:

Tháng	Điện Tử	Gia Dụng	Thời Trang	Thực Phẩm	Tổng Doanh Số
Tháng 1	10000	5000	15000	2000	32000
Tháng 2	12000	5500	16000	2500	36000
Tháng 3	13000	6000	17000	3000	39000
...	...	...	...	...	...
Tháng 12	15000	8000	20000	4000	47000

Do happy

CÁC BƯỚC DỮ LIỆU



🚀 Bước 1: Khám Phá và Xử Lý Dữ Liệu

Đọc và Hiểu Dữ Liệu: Sử dụng Pandas để đọc dữ liệu từ CSV.

Xử Lý Dữ Liệu Thiếu: Kiểm tra và điền dữ liệu thiếu nếu có.

Phân Tích Thống kê Mô Tả: Tính trung bình, trung vị, độ lệch chuẩn cho mỗi loại sản phẩm.

📊 Bước 2: Phân Tích Dữ Liệu và Trực Quan Hóa

Phân Tích Doanh Số Theo Thời Gian: Vẽ biểu đồ đường cho tổng doanh số bán hàng theo tháng sử dụng Matplotlib.

So Sánh Doanh Số Giữa Các Loại Sản Phẩm: Vẽ biểu đồ cột để so sánh doanh số giữa các loại sản phẩm sử dụng Seaborn.

📈 Bước 3: Dự Báo Doanh Số

Mô Hình Hóa: Sử dụng một mô hình đơn giản như hồi quy tuyến tính để dự báo doanh số cho tháng tiếp theo.

Đánh Giá Mô Hình: Kiểm tra độ chính xác của mô hình thông qua các chỉ số như R-squared, MSE.

📝 Code Minh Họa:

```
python
Copy code
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model
import LinearRegression
from sklearn.metrics
import mean_squared_error, r2_score
```

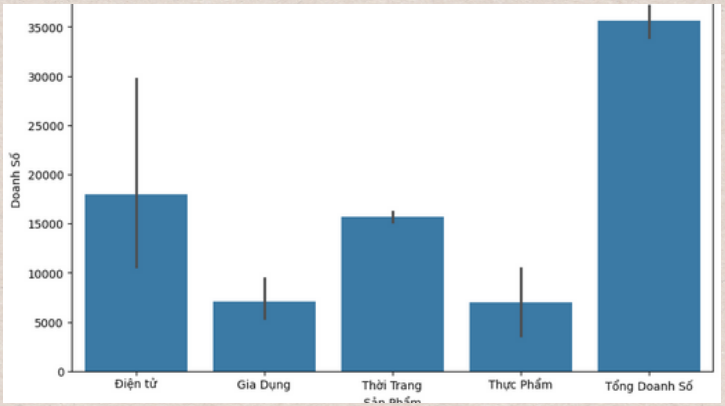
```
# Bước 1: Đọc và Xử Lý Dữ Liệu
df = pd.read_csv('data_ban_hang.csv')
df.fillna(method='ffill', inplace=True) # Điền dữ liệu thiếu
```

```
# Bước 2: Phân Tích và Trực Quan Hóa
plt.figure(figsize=(10, 6))
plt.plot(df['Tháng'], df['Tổng Doanh Số'], marker='o', linestyle='-')
plt.title('Doanh Số Bán Hàng Theo Tháng')
plt.xlabel('Tháng')
plt.ylabel('Doanh Số')
plt.show()
plt.figure(figsize=(10, 6))
sns.barplot(data=df.melt(id_vars=['Tháng'], var_name='Sản Phẩm', value_name='Doanh Số'), x='Sản Phẩm', y='Doanh Số')
plt.title('So Sánh Doanh Số Giữa Các Loại Sản Phẩm')
plt.show()
```

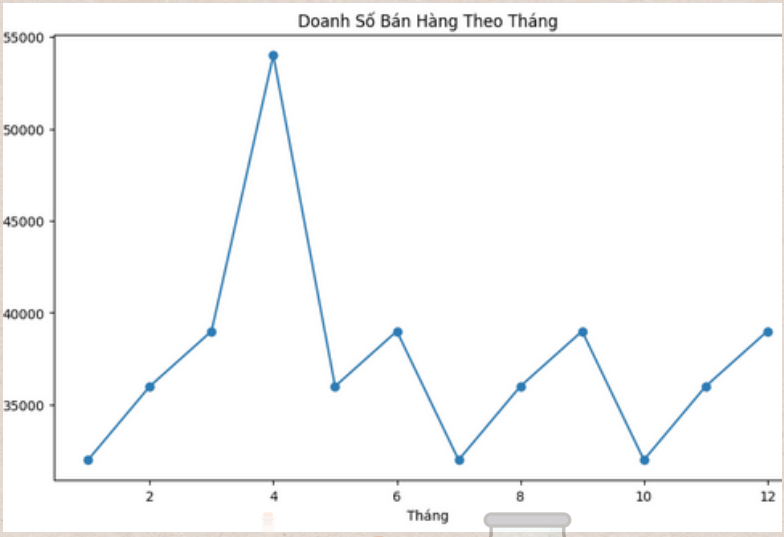
Dr happy

CÁC BƯỚC DỮ LIỆU

Kết quả về so sánh doanh số giữa các loại sản phẩm



Kết quả về doanh số bán hàng theo tháng.





Dr happy

CÁC BƯỚC DỮ LIỆU



Đây là một dự án tổng hợp giúp bạn áp dụng toàn bộ kiến thức đã học, từ việc xử lý và phân tích dữ liệu đến trực quan hóa và mô hình hóa dữ liệu. Hy vọng qua dự án này, bạn sẽ có cái nhìn toàn diện và thực tế về cách áp dụng Python vào phân tích kinh doanh.

Dự án lần này sẽ tập trung vào việc phân tích ảnh hưởng của thời tiết và các sự kiện địa phương lên hành vi mua sắm của khách hàng trong lĩnh vực bán lẻ.

```
# Bước 3: Dự Báo Doanh Số
X = df.index.values.reshape(-1, 1) # Thời gian (Tháng)
y = df['Tổng Doanh Số'].values # Doanh số

model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)

# Đánh giá mô hình
print(f'R-squared: {r2_score(y, y_pred)}')
print(f'MSE: {mean_squared_error(y, y_pred)}')

# Dự báo doanh số cho tháng tiếp theo
next_month = np.array([[len(df)]])
predict_next_month = model.predict(next_month)
print(f'Doanh số dự báo cho tháng tiếp theo: {predict_next_month[0]}')
```



Do happy

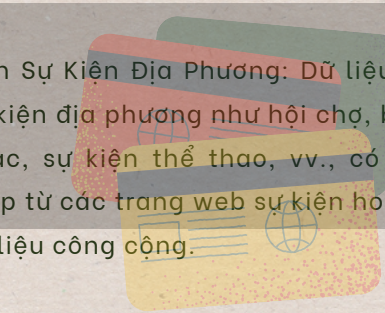


CÁC BƯỚC DỮ LIỆU

🚀 Bước 1: Thu Thập và Chuẩn bị Dữ liệu
Thu Thập Dữ liệu
Dữ liệu Khách Hàng: Bao gồm thông tin về lịch sử mua hàng, sở thích, và địa điểm mua sắm của khách hàng. Dữ liệu này có thể được thu thập từ cơ sở dữ liệu CRM (Quản lý quan hệ khách hàng) của công ty.

Dữ liệu Thời Tiết: Thông tin về nhiệt độ, lượng mưa, độ ẩm, vv., có thể được thu thập thông qua các API thời tiết như OpenWeatherMap, AccuWeather.

Lịch Sự Kiện Địa Phương: Dữ liệu về các sự kiện địa phương như hội chợ, buổi hòa nhạc, sự kiện thể thao, vv., có thể thu thập từ các trang web sự kiện hoặc cơ sở dữ liệu công cộng.



Chuẩn bị Dữ liệu
Làm Sạch Dữ liệu:

Loại bỏ các hàng trùng lặp.
Điền vào giá trị thiếu, ví dụ, sử dụng giá trị trung bình cho dữ liệu số hoặc phương pháp 'ffill'/'bfill' cho dữ liệu thời gian.

Xử Lý Dữ liệu Thời Tiết:

Chỉ giữ lại các thông số thời tiết cần thiết, ví dụ: nhiệt độ trung bình, tổng lượng mưa.
Chuẩn hóa dữ liệu nhiệt độ (nếu cần) để phù hợp với đơn vị sử dụng trong phân tích.

Chuyển Đổi Dữ liệu Thời Gian:

Chuyển đổi tất cả các cột thời gian sang định dạng datetime của Python để dễ dàng thao tác và phân tích.
Tạo các biến mới nếu cần, ví dụ: cột chỉ ra ngày nào có sự kiện.



Be happy

CÁC BƯỚC DỮ LIỆU API

📝 Ví Dụ Code:

```
python
Copy code
import pandas as pd
import requests
import json
# Thu thập dữ liệu thời tiết từ API
api_key = “YOUR_API_KEY“
city_name = “YOUR_CITY_NAME“
api_url = f“http://api.openweathermap.org/data/2.5/weather?q={city_name}&appid={api_key}“
response = requests.get(api_url)
weather_data = json.loads(response.text)

# Tạo DataFrame từ dữ liệu thời tiết
weather_df = pd.DataFrame({
'Date': [pd.Timestamp.now().normalize()],
'Temperature': [weather_data['main']['temp']],
'Humidity': [weather_data['main']['humidity']],
'Precipitation': [weather_data['rain']['1h'] if 'rain' in weather_data else 0]
})
```

```
# Làm sạch và chuẩn bị dữ liệu
# Giả sử 'customer_data.csv' là tệp chứa dữ liệu khách hàng
customer_df = pd.read_csv('customer_data.csv')
customer_df['Purchase Date'] = pd.to_datetime(customer_df['Purchase Date'])
```

```
# Điền giá trị thiếu
customer_df.fillna(method='ffill', inplace=True)
```

```
# Gộp dữ liệu thời tiết vào dữ liệu khách hàng dựa trên ngày
final_df = pd.merge(customer_df, weather_df, left_on='Purchase Date', right_on='Date', how='left')
```

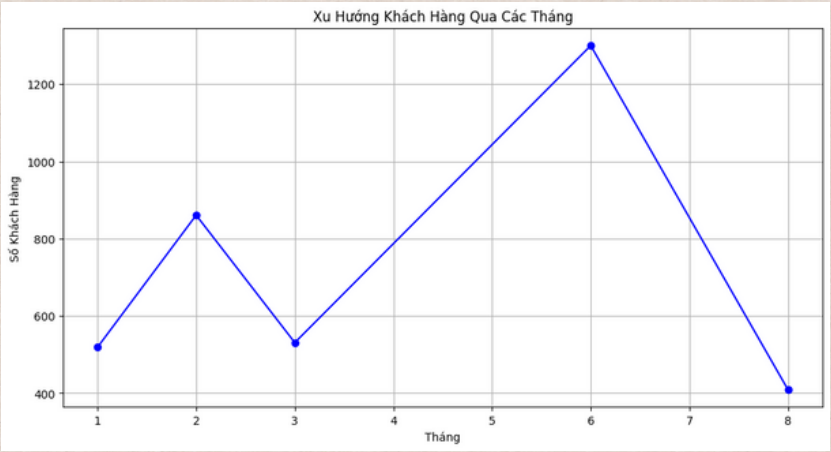
```
print(final_df.head())
```



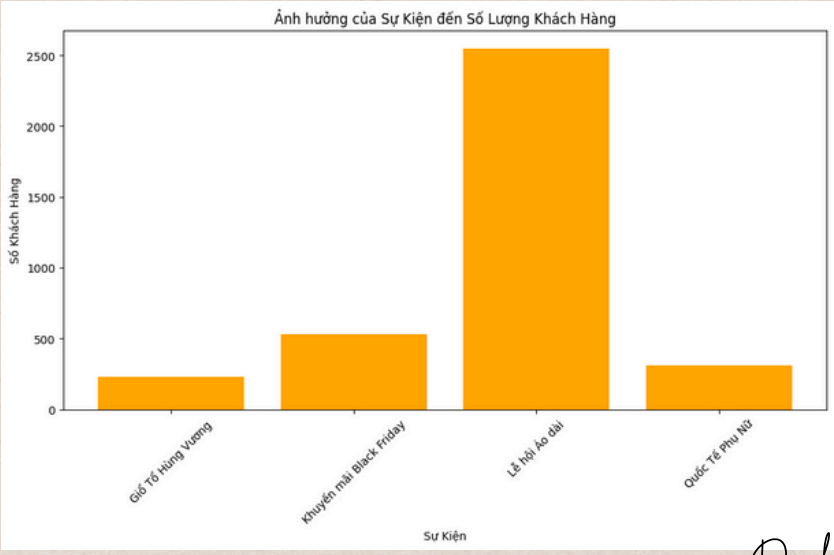
Be happy

CÁC BƯỚC DỮ LIỆU

Kết quả xu hướng khách hàng qua các tháng.



Ảnh hưởng của Sự Kiện đến Số Lượng Khách Hàng



Do happy



CÁC BƯỚC DỮ LIỆU

```
# Làm sạch và chuẩn bị dữ liệu
# Giả sử 'customer_data.csv' là tệp chứa dữ liệu khách hàng
customer_df = pd.read_csv('customer_data.csv')
customer_df['Purchase Date'] = pd.to_datetime(customer_df['Purchase Date'])

# Điền giá trị thiếu
customer_df.fillna(method='ffill', inplace=True)

# Gộp dữ liệu thời tiết vào dữ liệu khách hàng dựa trên ngày
final_df = pd.merge(customer_df, weather_df, left_on='Purchase Date', right_on='Date', how='left')

print(final_df.head())
```

Đoạn code này minh họa cách thu thập dữ liệu thời tiết từ API và chuẩn bị dữ liệu cho phân tích, bao gồm cách làm sạch và gộp dữ liệu. Đây chỉ là một phần trong quy trình thu thập và chuẩn bị dữ liệu; tùy vào nhu cầu cụ thể và nguồn dữ liệu, bạn có thể cần thêm nhiều bước xử lý dữ liệu hơn.



Dr happy



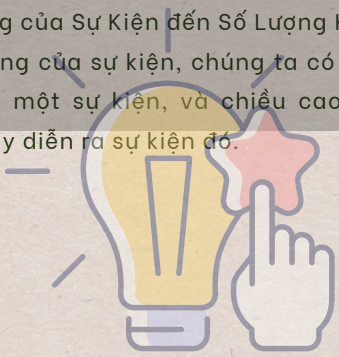
CÁC BƯỚC DỮ LIỆU

Phân Tích Xu Hướng Khách Hàng

Biểu Đồ Đường Xu Hướng Khách Hàng Qua Các Tháng:
Để vẽ biểu đồ đường thể hiện xu hướng khách hàng qua các tháng, chúng ta sẽ tính tổng số lượng khách hàng mua sắm trong mỗi tháng và vẽ chúng trên biểu đồ.

Ảnh hưởng của Thời Tiết
Biểu Đồ Phân Tán giữa Số Lượng Khách Hàng và Dữ Liệu Thời Tiết:
Vẽ biểu đồ phân tán để tìm mối quan hệ giữa điều kiện thời tiết (ví dụ: nhiệt độ) và số lượng khách hàng. Mỗi điểm trên biểu đồ đại diện cho một ngày, với trục x là nhiệt độ và trục y là số lượng khách hàng.

Sự Kiện và Số Lượng Khách Hàng
Biểu Đồ Cột Ảnh hưởng của Sự Kiện đến Số Lượng Khách Hàng:
Để phân tích ảnh hưởng của sự kiện, chúng ta có thể tạo một biểu đồ cột, với mỗi cột đại diện cho một sự kiện, và chiều cao của cột thể hiện số lượng khách hàng trong ngày diễn ra sự kiện đó.



Do happy

CONTACT: (+84)-9110-2191
HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH.VN
TOÁN VÀ LẬP TRÌNH

CÁC BƯỚC DỮ LIỆU

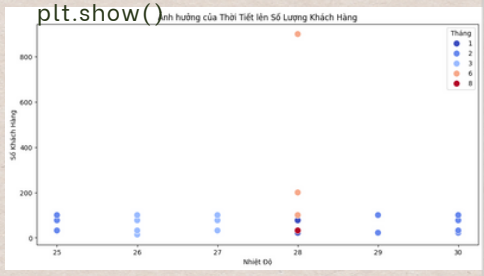
📝 Code Minh Họa:

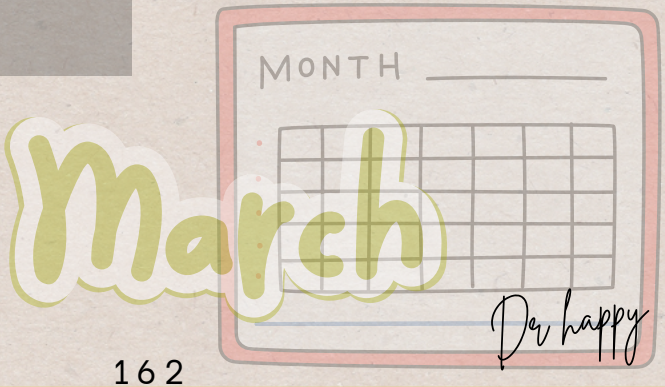
1. Biểu Đồ Đường Xu Hướng Khách Hàng

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Giả sử 'data' là DataFrame chứa dữ liệu đã được chuẩn bị
# Vẽ biểu đồ đường xu hướng khách hàng qua các tháng sử dụng Matplotlib
plt.figure(figsize=(12, 6))
plt.plot(data['Tháng'], data['Số Khách Hàng'], marker='o', linestyle='-', color='b')
plt.title('Xu Hướng Khách Hàng Qua Các Tháng')
plt.xlabel('Tháng')
plt.ylabel('Số Khách Hàng')
plt.grid(True)
plt.show()
```

2. Biểu Đồ Phân Tán Ảnh hưởng của Thời Tiết

```
# Vẽ biểu đồ phân tán ảnh hưởng của thời tiết sử dụng Seaborn
plt.figure(figsize=(12, 6))
sns.scatterplot(data=data, x='Nhiệt Độ', y='Số Khách Hàng', hue='Tháng', palette='coolwarm', s=100)
plt.title('Ảnh hưởng của Thời Tiết lên Số Lượng Khách Hàng')
plt.xlabel('Nhiệt Độ')
plt.ylabel('Số Khách Hàng')
plt.show()
```





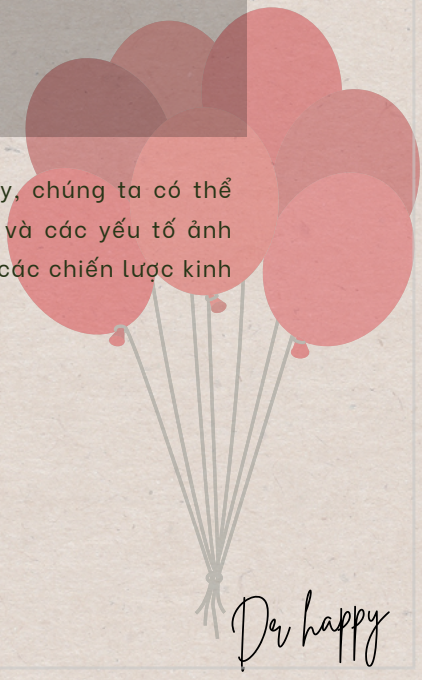


CÁC BƯỚC DỮ LIỆU

3. Biểu Đồ Cột Ảnh hưởng của Sự Kiện

```
# Giả sử 'event_data' là DataFrame chứa thông tin về sự kiện và số lượng khách hàng
# Vẽ biểu đồ cột ảnh hưởng của sự kiện đến số lượng khách hàng sử dụng Matplotlib
plt.figure(figsize=(12, 6))
plt.bar(event_data['Sự Kiện'], event_data['Số Khách Hàng'], color='orange')
plt.title('Ảnh hưởng của Sự Kiện đến Số Lượng Khách Hàng')
plt.xlabel('Sự Kiện')
plt.ylabel('Số Khách Hàng')
plt.xticks(rotation=45)
plt.show()
```

Qua các phân tích và trực quan hóa này, chúng ta có thể hiểu rõ hơn về hành vi của khách hàng và các yếu tố ảnh hưởng đến họ, từ đó giúp công ty đưa ra các chiến lược kinh doanh phù hợp.





CÁC BƯỚC DỮ LIỆU

📈 Dự Báo Số Lượng Khách Hàng
Mô Hình Hóa

Xây Dựng Mô Hình:

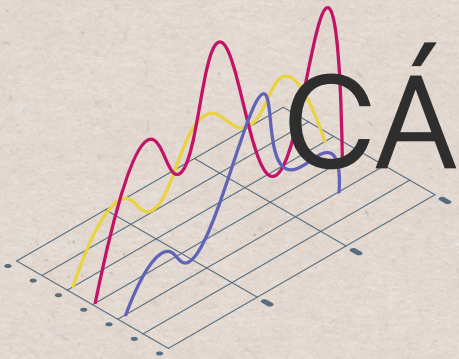
Sử dụng hồi quy tuyến tính để dự báo số lượng khách hàng. Các biến độc lập có thể bao gồm tháng, nhiệt độ trung bình, số lượng sự kiện trong tháng, vv.
Nếu dữ liệu có tính mùa vụ, có thể xem xét sử dụng mô hình thời gian như ARIMA.
Đánh Giá Mô Hình
Đánh Giá Hiệu Suất Mô Hình:
Sử dụng các chỉ số như RMSE (Root Mean Square Error) để đo lường sai số dự báo.
Sử dụng R-squared để đánh giá mức độ mô hình giải thích được biến động trong dữ liệu.

📝 Code Minh Họa:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Giả sử 'data' là DataFrame đã được chuẩn bị, chứa số lượng khách hàng và các yếu tố như tháng, nhiệt độ, số sự kiện
X = data[['Tháng', 'Nhiệt Độ', 'Số Sự Kiện']] # Biến độc lập
y = data['Số Khách Hàng'] # Biến phụ thuộc
```

Be happy

CONTACT: (+84)-9110-2191 HTTPS://WWW.FACEBOOK.COM/TOANVALAPTRINH.VN TOÁN VÀ LẬP TRÌNH



CÁC BƯỚC DỮ LIỆU

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Xây dựng mô hình hồi quy tuyến tính
model = LinearRegression()
model.fit(X_train, y_train)

# Dự báo số lượng khách hàng
y_pred = model.predict(X_test)

# Đánh giá mô hình
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print(f'RMSE: {rmse}')
print(f'R-squared: {r2}')

# Sử dụng mô hình để dự báo
new_data = [[5, 25, 2]] # Tháng 5, nhiệt độ 25 độ C, 2 sự kiện
predicted_customers = model.predict(new_data)
print(f'Số lượng khách hàng dự báo: {predicted_customers[0]}')
```

Qua việc áp dụng các kỹ thuật hồi quy tuyến tính, chúng ta có thể dự báo số lượng khách hàng dựa trên các yếu tố đã phân tích. Việc đánh giá mô hình qua RMSE và R-squared giúp chúng ta hiểu được độ chính xác và khả năng giải thích của mô hình, từ đó cung cấp thông tin hữu ích cho quyết định kinh doanh.

Dr happy

ISSUE 1 | VOL. 1 | MARCH 2024

PYTHON, TOÁN VÀ ỨNG DỤNG KINH TẾ

WRITER: DR HAPPY

AI ENGINEER: NGUYEN VU

EDITOR: LUCIE TRUC NGUYEN

ISSUE 1 | VOL. 1 | MARCH 2024



Hãy tiếp tục hành trình khám phá lập trình và toán học cùng chúng tôi tại Fanpage "Toán và Lập Trình" để nhận được kiến thức và hỗ trợ tốt nhất!

MATHEMATICS AND PROGRAMMING

Thông tin liên hệ:
Dr.Happy - 0911.118.758
Nguyên Vũ - 0971.844.698

TOÁN VÀ LẬP TRÌNH