大數據資料分析實作

數據收集與清理技術

博雅(科技)課程

常見資料清理問題

- 缺失值處理：
  - 定義：為什麼資料中會有缺失值？
    - 缺失值是指資料中某些欄位的數據缺失。這可能是由於資料收集時的錯誤或資料丟失。
  - 處理方法：填補、刪除、插值法等
    - 用其他數據的平均值、中位數或眾數進行填補。
    - 如果缺失數據太多，可以選擇刪除相關行或列。
    - 高級方法：使用插值法或基於機器學習的填補方法。

常見資料清理問題

- 重複值處理：
  - 定義：重複數據的影響
    - 重複的數據會造成分析結果的偏差，尤其是在計算統計指標或進行建模時。
  - 檢查與刪除重複資料
  - 處理方法：使用Pandas的.duplicated()與.drop_duplicates()方法
    - 可以使用Pandas中的.duplicated()方法來檢查資料中是否有重複的行，並使用.drop_duplicates()方法刪除。

常見資料清理問題

- 資料格式處理：
  - 資料格式不一致的問題（日期格式、數字格式等）
    - 資料的格式不一致（如日期欄位格式不同、數字欄位有額外的符號等）會影響數據分析的準確性。

  - 處理方法：使用Pandas的.to_datetime()與.astype()方法來處理格式
    - 轉換日期欄位為標準日期格式（例如：datetime格式）。
    - 確保數字欄位的格式正確並轉換為數值型態。

常見資料清理問題

- 異常值檢測與處理：
  - 定義：異常值是什麼？如何識別異常值？
    - 異常值會扭曲分析結果，尤其是當資料範圍非常大或包含極端值時。
  - 處理方法：盒鬚圖（Boxplot）等視覺化技術來檢測異常值
    - 通常，異常值會顯示為超出正常範圍的數據（例如，一個年齡欄位中出現了負數或極大數值）。
    - 可以用視覺化工具（例如箱型圖、散佈圖）檢測異常值，並決定是否刪除或替換

資料清理工具與方法

- 進行資料清理的工具與方法，特別是Pandas庫，因為它是Python中處理數據清理最常用的工具之一。

- 介紹Pandas庫的常用方法：
  - .isnull(), .dropna(), .fillna()
  - .duplicated(), .drop_duplicates()
  - .astype(), .to_datetime()

資料清理工具與方法：處理缺失值

- 缺失值的處理是資料預處理中的首要任務，因為缺失數據的存在會直接影響後續分析。

- 處理缺失值：
  - isnull() 與 fillna()：
  - .isnull()：檢查哪些欄位有缺失值，返回布林值（True表示缺失）。
  - .fillna()：可以用均值、中位數或其他方法來填補缺失值。

資料清理工具與方法：處理缺失值

- 範例介紹：
  - 讀取資料集：引入一個有缺失值的資料集
  - 使用.isnull()檢查缺失值的存在。
  - 使用.fillna()填補缺失值。
  - 選擇使用不同的方法（例如，用列的均值填補）。
- Python原始碼：

```python
import pandas as pd

# 讀取資料集
data = pd.read_csv("example_data.csv")

# 檢查缺失值
print(data.isnull().sum())

# 使用平均數填補缺失值
data.fillna(data.mean(), inplace=True)

# 檢查填補後的缺失值
print(data.isnull().sum())
```

資料清理工具與方法：處理重複資料

- 重複資料的存在會影響模型的預測效果，尤其是計算統計量和建立模型時。

- 處理重複值：
  - duplicated() 與 drop_duplicates()：
  - .duplicated()：檢查資料中是否有重複行。
  - .drop_duplicates()：刪除重複的資料行。

資料清理工具與方法：處理重複資料

- 範例介紹：
  - 讀取資料集並檢查是否有重複資料。
  - 使用.duplicated()來檢查重複行
  - 使用.drop_duplicates()來刪除。
- Python原始碼：

```python
# 檢查重複值
print(data.duplicated().sum())

# 刪除重複值
data.drop_duplicates(inplace=True)

# 查看刪除後的資料
print(data)
```

資料清理工具與方法：格式處理

- 如何處理資料格式，尤其是如何處理日期格式不一致的問題。將日期轉換為統一格式是資料清理過程中的常見需求。

- 格式處理：
  - astype() 與 to_datetime()：
  - .astype()：用來將欄位轉換為指定的數據類型。
  - .to_datetime()：將非標準格式的日期轉換為datetime格式。

資料清理工具與方法：格式處理

- 範例介紹：
  - 使用.to_datetime()方法將非標準格式的日期轉換為datetime格式。

- Python原始碼：

```
# 假設有一個日期欄位是非標準格式
data['date'] = pd.to_datetime(data['date'], format='%Y-%m-%d')


# 查看轉換後的資料類型
print(data['date'].dtype)
```

資料清理工具與方法：異常值檢測

- 異常值會扭曲數據分析結果，因此需要檢測並處理。
- 透過使用視覺化技術（如箱型圖）來檢測資料中的異常值。
- 異常值檢測：
  - 視覺化工具：利用Matplotlib和Seaborn繪製箱型圖來檢測異常值。

資料清理工具與方法：異常值檢測

- 範例介紹：
  - 使用箱型圖檢測異常值：

    使用Seaborn繪製箱型圖（又稱為盒鬚圖），檢查資料中的異常值。
- Python原始碼：

```
import seaborn as sns
import matplotlib.pyplot as plt


# 使用箱型圖檢查異常值
sns.boxplot(data['age'])
plt.show()
```

課堂實作範例操作流程

課堂實作範例操作流程

- 我們將使用台灣的公開資料庫進行實作，來幫助大家了解如何進行資料清理，特別是處理缺失值、重複資料、格式處理以及異常值檢測。以下是具體的操作步驟與程式碼示範，並針對每個步驟提供詳細的解說。

- 針對沒有安裝 Python 軟體的學生，可以選擇使用 **Google Colab** 來進行學習，這樣就不需要在自己電腦上安裝任何程式，直接使用瀏覽器即可執行程式碼。

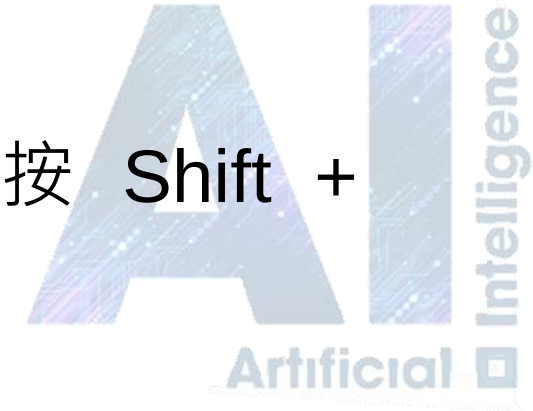
介紹 Google Colab 環境與介面操作

- Google Colab 是一個基於雲端的 Jupyter Notebook 環境，提供免費的 Python 編程和執行環境。使用者可以直接在瀏覽器中撰寫、執行程式碼，並輕鬆使用 Google 提供的資源（如 GPU 和 TPU）。

- 如何進入 Google Colab
  - 打開瀏覽器，進入 Google Colab，或者可以直接搜尋 "Google Colab"。
  - 若沒有 Google 帳號，請先註冊一個 Google 帳號並登入。
  - 在主頁上，會看到「新建筆記本」選項，點擊即可創建一個新的 Python 筆記本。

介紹 Google Colab 環境與介面操作

- Google Colab 的介面介紹
  - 程式碼區塊：這是用來輸入 Python 程式碼的地方。每一個程式碼區塊可以執行獨立的 Python 代碼。
  - 文字區塊：用來撰寫 Markdown 格式的說明文字，可以讓你在程式碼之間加入說明或註解。
  - 執行按鈕：點擊程式碼區塊左邊的播放按鈕（或按 Shift + Enter），就能執行該區塊的程式碼。



介紹 Google Colab 環境與介面操作

- Colab 內建的常用功能
  - 安裝套件：可以在 Colab 內使用 **!pip install** 安裝 Python 套件，這樣就可以用 pandas、numpy 等庫進行資料處理。
  - 掛載 Google Drive：使用 Google Drive 來儲存資料，方便存取和處理大檔案。

```
# 安裝 pandas 套件（如果尚未安裝）
!pip install pandas
```

- 儲存與下載筆記本
  - 儲存筆記本：點選頁面左上方的「檔案」>「儲存」，可以將筆記本儲存在 Google Drive 中。
  - 下載筆記本：點選「檔案」>「下載」>「下載為 .ipynb」，可以將整個筆記本下載到本地端。

資料集選擇

- 我們將以「台灣政府資料開放平台」提供的「**YouBike2.0臺北市公共自行車即時資訊**」為範例資料集。

實作步驟：下載資料並掛載 Google Drive

- 下載資料集
  - 下載YouBike的CSV格式資料，並將檔案命名為 youbike_data.csv
  - 下載後，請將資料存儲在 Google Drive 中，並將資料夾共享給 Colab 使用。
- 掛載 Google Drive
  - 在 Colab 中，需要掛載 Google Drive，這樣才能直接讀取儲存在 Google Drive 上的資料。
  - 在新的程式碼區塊中，輸入以下程式碼並執行，這將讓 Colab 存取你的 Google Drive 資料。

```
from google.colab import drive
drive.mount('/content/drive')
```

實作步驟：讀取資料集

- 在 Colab 中，可以使用 pandas 來讀取 CSV 格式的資料。
- 請在 Google Drive 中找到剛才上傳的 YouBike 資料。
- pd.read_csv()：用來讀取 CSV 檔案，並將其轉換為 Pandas DataFrame格式。
- .head()：顯示資料集的前五筆資料，以了解資料的基本結構。

```
import pandas as pd


# 讀取資料集
data = pd.read_csv('/content/drive/MyDrive/你的資料夾路徑/youbike_data.csv')


# 顯示資料前五筆
data.head()
```

補充說明

數據分析的第一步！

- 開啟數據分析的大門，要使用python做數據分析，其實現有很多很好用的package可以讓我們輕易的上手
- 介紹數據分析會使用到的幾個package，包含：
  - Pandas
  - Numpy
  - Matplotlib

補充說明

- Pandas、Numpy與Matplotlib構成了資料科學的強大基礎
- Pandas是一個基於Numpy的package，在處理數據方面非常的好用簡單，透過標籤和索引，Pandas讓我們可以非常輕易的處理數據
- Numpy是一個提供矩陣運算非常好用的工具，具備平行處理的能力，可以將操作動作一次套用在大型陣列上，幫助我們做更多方法建立多維數據以及矩陣運算，像是Pandas就是建立在Numpy的基礎延伸的套件
- Matplotlib是Python繪圖的它包含了大量的工具，可以使用這些工具創建各種圖形，包括簡單的散點圖、直方圖，甚至是三維圖形，將資料轉成圖表，在python的數據分析中會經常使用Matplotlib完成數據可視化的工作

實作步驟：檢查資料的結構與缺失值

- 檢查資料的結構以及是否有缺失值。這是資料清理的第一步，了解資料型態以及是否有任何需要處理的缺失值。
- data.info()：顯示資料集的結構，包含每個欄位的資料型態、非空值的數量等資訊。
- data.isnull().sum()：計算每個欄位的缺失值數量。

```
# 檢查資料結構
data.info()


# 檢查缺失值
missing_values = data.isnull().sum()
print(missing_values)
```

實作步驟：填補缺失值

- 處理缺失值：
  - 填補缺失值：使用平均數或中位數填補缺失的數值欄位
  - .fillna(data['available_rent_bikes'].mean())：將[available_rent_bikes]欄位中的缺失值填補為該欄位的均值。
  - 這樣能確保資料不會因為缺失值而造成分析錯誤。

```
# 用均值填補缺失值
data['available_rent_bikes'] = data['available_rent_bikes']
.fillna(data['available_rent_bikes'].mean())
# 檢查缺失值是否處理完成
print(data.isnull().sum())
```

實作步驟：檢查與刪除重複資料

- 檢查資料中是否有重複的行，否則會影響統計計算或模型訓練。

- 檢查重複資料
  - data.duplicated().sum()：檢查資料集中的重複行，並計算出重複行的數量。

```
# 檢查重複資料
duplicates = data.duplicated().sum()
print(f"重複資料數量：{duplicates}")
```

實作步驟：檢查與刪除重複資料

- 刪除重複資料：若資料中存在重複行，要將其刪除。
  - .drop_duplicates()：刪除所有重複的資料行，inplace=True 表示直接在原資料中進行修改。

```
# 刪除重複資料行
data.drop_duplicates(inplace=True)

# 檢查重複資料是否刪除
print(f"重複資料數量：{data.duplicated().sum()}")
```

實作步驟：異常值檢測與處理

- 異常值可能會影響統計分析結果，我們可以使用箱型圖來檢查資料中的異常值。
  - sns.boxplot()：繪製箱型圖，能夠幫助我們檢查資料的分布情況，特別是異常值。

```
import seaborn as sns
import matplotlib.pyplot as plt


# 使用箱型圖檢查 '借車數' 欄位的異常值
sns.boxplot(data['available_rent_bikes'])
plt.show()
```

小結

- 實作資料清理的流程：
  ① 安裝與掛載：安裝 pandas，掛載 Google 雲端硬碟以讀取資料。
  ② 讀取數據：讀取 CSV 檔案，顯示基本數據資訊。
  ③ 處理缺失值：檢查缺失值並用平均值填補。
  ④ 處理重複數據：檢查並刪除重複數據。
  ⑤ 視覺化數據：繪製箱型圖來分析 available_rent_bikes 的分佈狀況。
- 這是一個完整的「數據清理與分析」流程，適用於初學者學習 Python 資料處理的基本概念。

「大數據資料分析實作」課程