Python 機器學習項目——用 XGBoost 檢測帕金森病

在這個 Python 機器學習項目中，我們將構建一個模型，我們可以使用該模型準確檢測人體內是否存在帕金森病，讓我們從介紹使用的術語開始。

* **什麼是帕金森病？**

帕金森病是一種中樞神經系統進行性疾病，會影響運動並引起震顫和僵硬，它有 5 個階段，每年影響印度超過 100 萬人，這是慢性的，目前還沒有治愈方法，它是一種神經退行性疾​​病，影響大腦中產生多巴胺的神經元。

* **什麼是 XGBoost？**

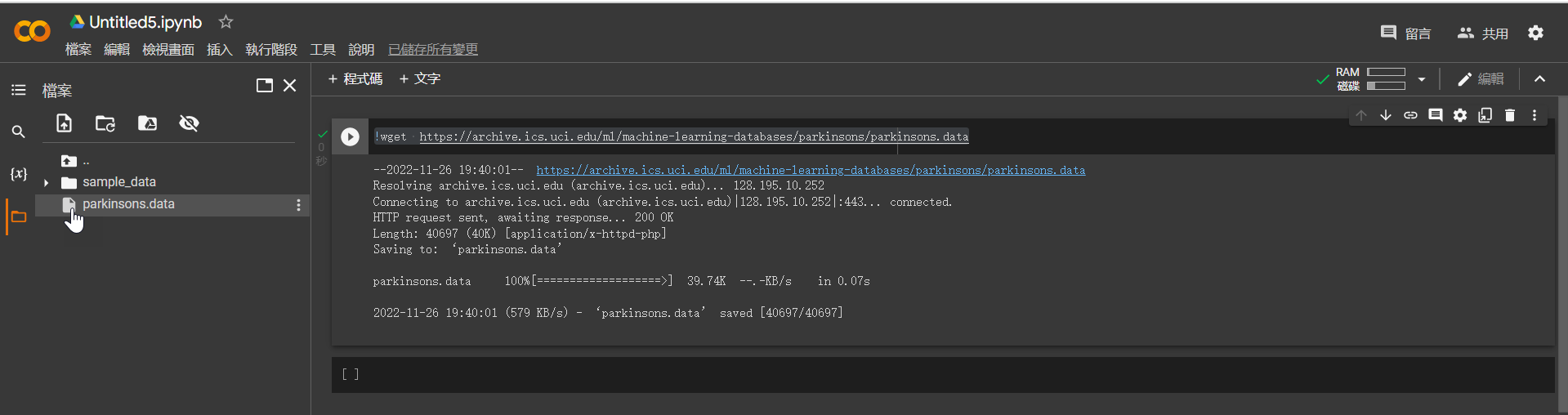
XGBoost 是一種新的機器學習算法，在設計時考慮了速度和性能，**XGBoost** 代表 **eXtreme Gradient Boosting**，它基於決策樹。在這個項目中，我們將從 xgboost 庫中導入 XGBClassifier；這是用於 XGBoost 分類的 scikit-learn API（\*查） 來實現檢測帕金森病之目標。

在這個 Python 機器學習項目中，我們將使用 ***Python libraries--*scikit-learn、numpy、pandas** 和 **xgboost** 構建一個使用 XGBClassifier 的模型，整個實作中我們將加載數據、獲取特徵和標籤、縮放特徵、拆分數據集、構建 XGBClassifier，然後去計算此模型的準確性。

Python 機器學習項目的數據集

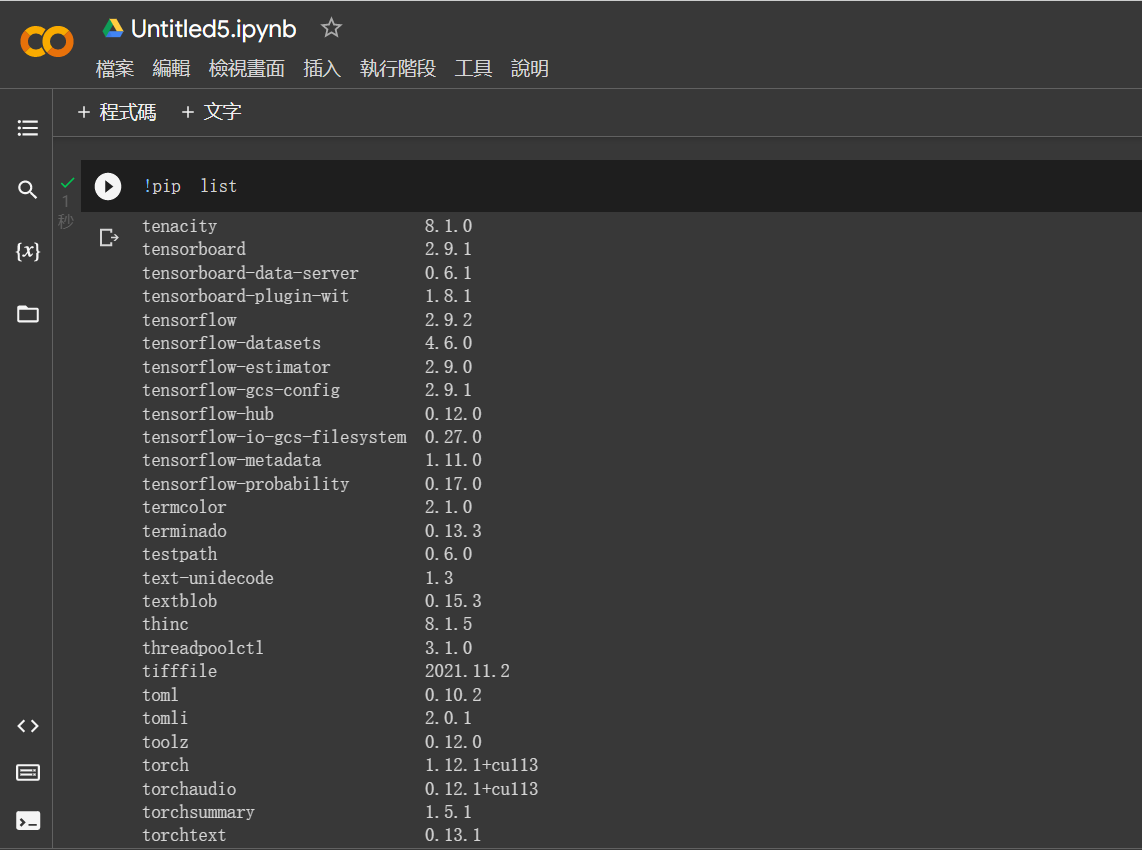
* 為此，您需要 UCI ML Parkinsons 數據集[parkinsons.data](https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/parkinsons.data)；你可以在 [***https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons***](https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons)下載取得，該數據集有 24 列和 195 條記錄，只有 39.7 KB。

!wget https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/parkinsons.data



先決條件

首先在開發平台上檢查是否包含如下模組



如沒有則需要使用 pip 進行安裝，如下：

**!pip install numpy pandas sklearn xgboost**

* 使用 XGBoost 檢測帕金森病的步驟

以下是練習 Python 機器學習項目所需的一些步驟(Below are some steps required to practice Python Machine Learning Projec)---

1. 進行必要的導入（Make necessary imports）：

import numpy as np

import pandas as pd

import os, sys

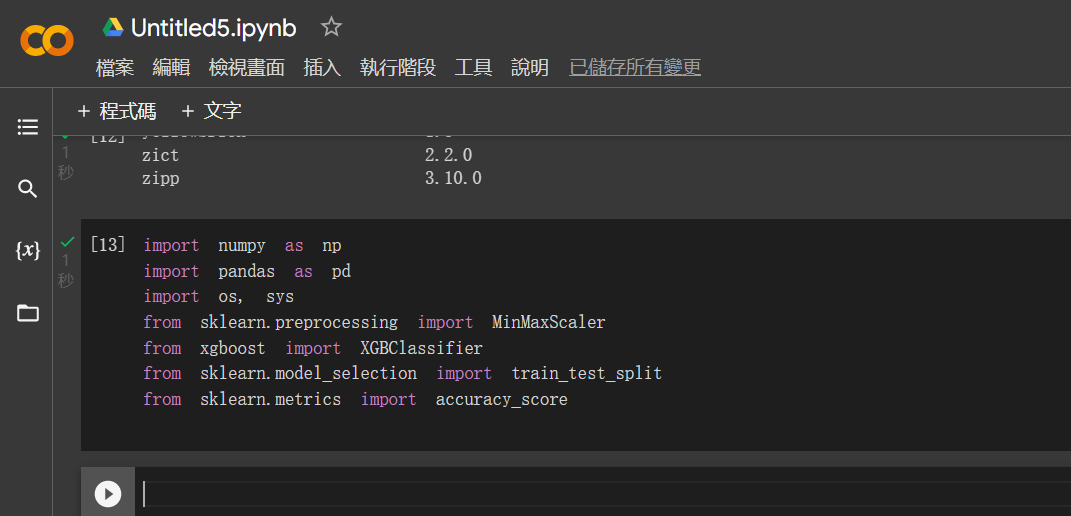
from sklearn.preprocessing import MinMaxScaler

from xgboost import XGBClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

**截屏Screenshot：**



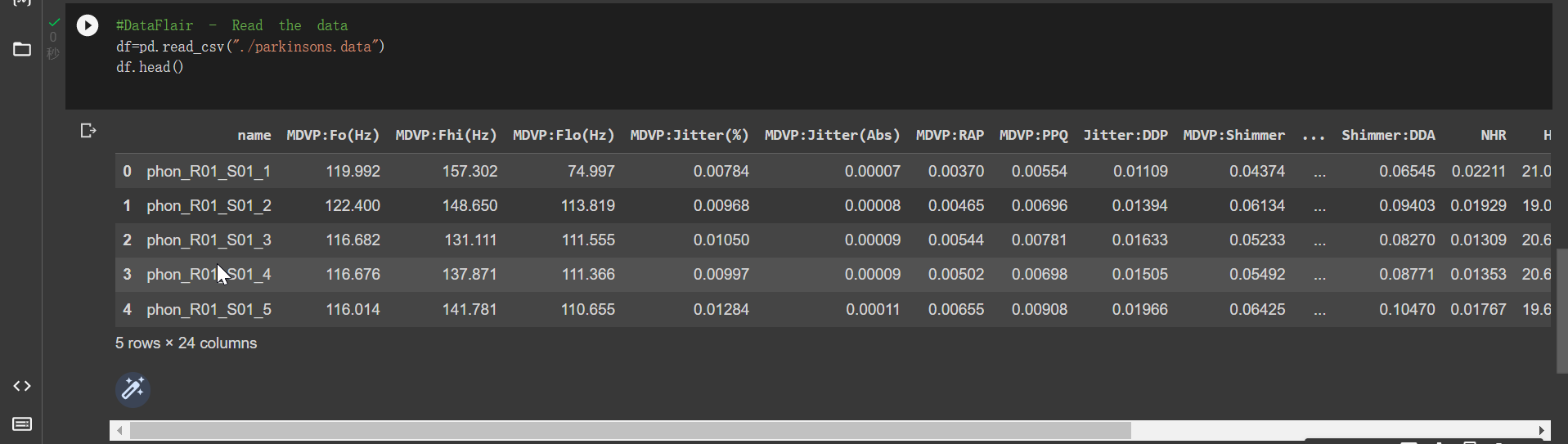
2. 現在，讓我們將數據讀入 DataFrame 並獲取前 5 條記錄（Now, let’s read the data into a DataFrame and get the first 5 records）。

#DataFlair - Read the data

df=pd.read\_csv("./parkinsons.data")

df.head()

**輸出截圖(Output Screenshot)：**



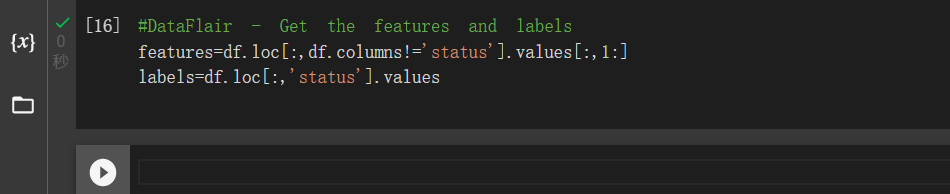
3. 從DataFrame（數據集）中獲取特徵和標籤；特徵是除“狀態”之外的所有列，標籤是“狀態”列中的那些。（Get the features and labels from the DataFrame (dataset). The features are all the columns except ‘status’, and the labels are those in the ‘status’ column.）

#DataFlair - Get the features and labels

features=df.loc[:,df.columns!='status'].values[:,1:]

labels=df.loc[:,'status'].values

**截屏Screenshot：**

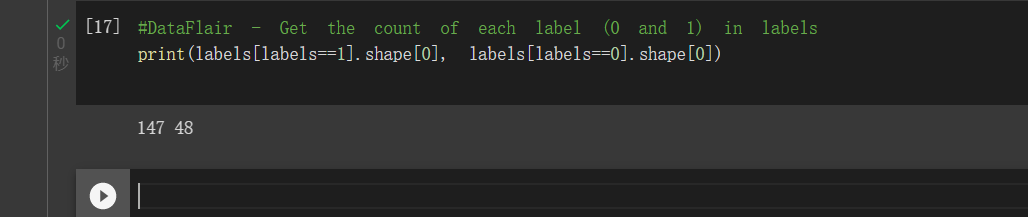


4. 'status' 列有值 0 和 1 作為標籤；讓我們為 0 和 1 獲取這些標籤的計數。（The ‘status’ column has values 0 and 1 as labels; let’s get the counts of these labels for both- 0 and 1）

#DataFlair - Get the count of each label (0 and 1) in labels

print(labels[labels==1].shape[0], labels[labels==0].shape[0])

**輸出截圖Output Screenshot：**



表示我們數據集中的狀態列中有 147 個 1 和 48 個 0(We have 147 ones and 48 zeros in the status column in our dataset.)。

5. 初始化一個 MinMaxScaler 並將特徵縮放到 -1 和 1 之間以對其進行歸一化，MinMaxScaler 通過將特徵縮放到給定範圍來轉換特徵，而fit\_transform() 方法適合數據，然後對其進行轉換，我們不需要縮放標籤。(Initialize a MinMaxScaler and scale the features to between -1 and 1 to normalize them. The MinMaxScaler transforms features by scaling them to a given range. The fit\_transform() method fits to the data and then transforms it. We don’t need to scale the labels.)

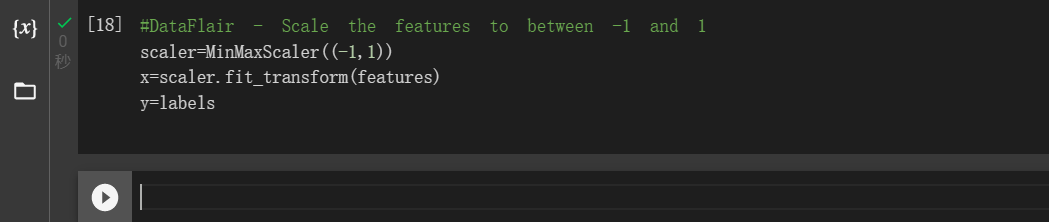
#DataFlair - Scale the features to between -1 and 1

scaler=MinMaxScaler((-1,1))

x=scaler.fit\_transform(features)

y=labels

**截屏（Screenshot）：**

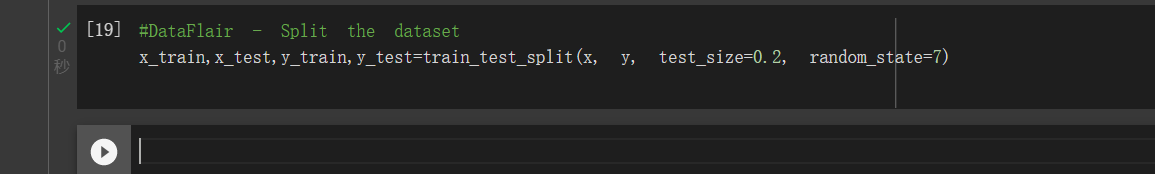


6. 現在，將數據集拆分為訓練集和測試集，保留 20% 的數據用於測試。（Now, split the dataset into training and testing sets keeping 20% of the data for testing.）

#DataFlair - Split the dataset

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x, y, test\_size=0.2, random\_state=7)

**截屏(Screenshot)：**



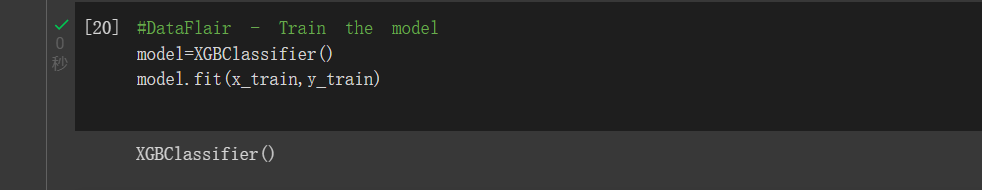
7. 初始化 XGBClassifier 並訓練模型，This classifies using eXtreme Gradient Boosting——使用***[梯度提升算法](https://data-flair.training/blogs/gradient-boosting-algorithm/)***解決現代數據科學問題，它屬於 ML 中的集成學習類別，我們使用許多模型進行訓練和預測以產生一個卓越的輸出。（Initialize an XGBClassifier and train the model. This classifies using eXtreme Gradient Boosting- using[***gradient boosting algorithms***](https://data-flair.training/blogs/gradient-boosting-algorithm/)for modern data science problems. It falls under the category of Ensemble Learning in ML, where we train and predict using many models to produce one superior output.）

#DataFlair - Train the model

model=XGBClassifier()

model.fit(x\_train,y\_train)

**輸出截圖（Output Screenshot）：**



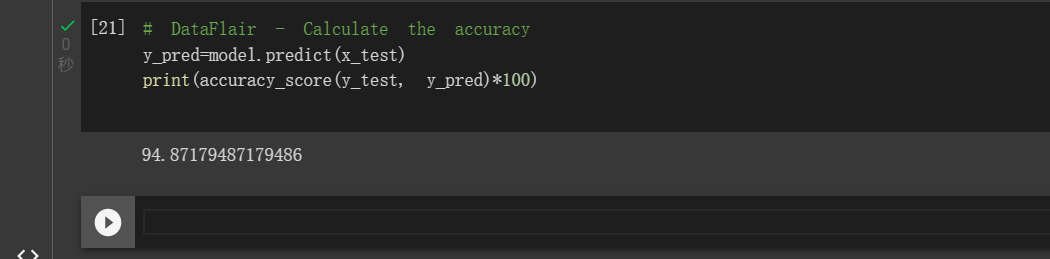
8. 最後，生成 y\_pred（x\_test 的預測值）併計算模型的準確度，打印出來。（Finally, generate y\_pred (predicted values for x\_test) and calculate the accuracy for the model. Print it out.）

# DataFlair - Calculate the accuracy

y\_pred=model.predict(x\_test)

print(accuracy\_score(y\_test, y\_pred)\*100)

**輸出截圖Output Screenshot：**



## 結論（Summary）

在這個 Python 機器學習項目中，我們學會了使用各種因素來檢測個體是否患有帕金森病。我們為此使用了 XGBClassifier，並使用 sklearn 庫來準備數據集。這給了我們 94.87% 的準確率，考慮到這個 python 項目中的代碼行數，這已經很不錯了。(In this Python machine learning project, we learned to detect the presence of Parkinson’s Disease in individuals using various factors. We used an XGBClassifier for this and made use of the sklearn library to prepare the dataset. This gives us an accuracy of 94.87%, which is great considering the number of lines of code in this python project.)

Hope you enjoyed this Python project. We have already provided you the links for more interesting Python Projects at series of themes.