

SOFTWARE

En aquest TFG s'han implementat tres mètodes de *Machine Learning* per tal de poder predir els casos d'obesitat. Aquests tres models són: Arbre de Decisió, *Random Forest* i Màquines de Suport Vectorial.

En el repositori de Github trobem tres codis diferents, un per a cada model.

La implementació dels 3 models predictius s'ha realitzat a partir de l'editor de codi *Visual Studio Code* amb el llenguatge de programació *Python*. S'ha fet ús de la llibreria *Sklearn*, una de les més utilitzades en *Machine Learning* dins l'entorn de *Python*. Permet realitzar algoritmes tant d'aprenentatge supervisat com de no supervisat i conté eines que serveixen per al preprocessat i anàlisi de dades i per avaluar la capacitat predictiva que mostren els diferents models. Aquesta llibreria conté d'altres que també s'han utilitzat per a la implementació.

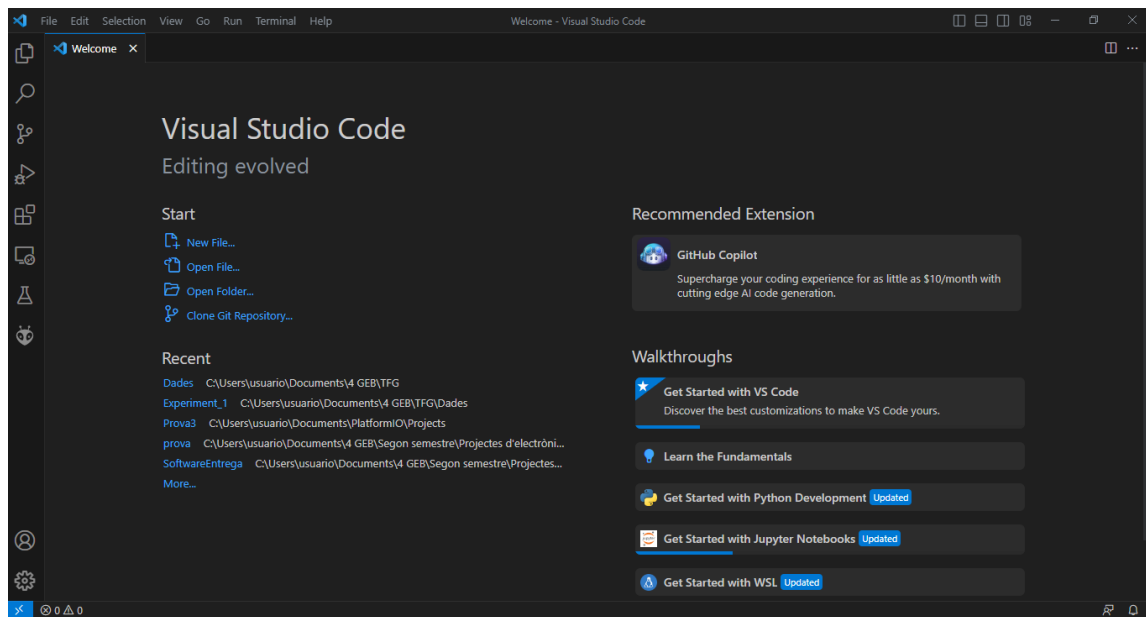
Aquestes són les que s'expliquen a continuació:

- **Pandas:** És una llibreria de codi obert molt popular dins de la programació en *Python* i dins de l'àrea de *Machine Learning*. Proporciona estructures flexibles per poder manipular i tractar dades de manera senzilla.
- **Numpy:** Aquesta llibreria proporciona un gran ventall d'operacions matemàtiques per a poder realitzar càlculs amb matrius i vectors.
- **Openpyxl:** És una llibreria de *Python* útil per poder llegir, escriure i modificar fitxers Excel.
- **Matplotlib:** És una llibreria de *Python* de codi obert que serveix per poder visualitzar dades. Permet crear gràfics i diagrames en diversos formats.
- **Seaborn:** Aquesta llibreria és un subconjunt de *matplotlib* i serveix per a la visualització de dades. Permet crear gràfics sofisticats amb menys línies de codi.

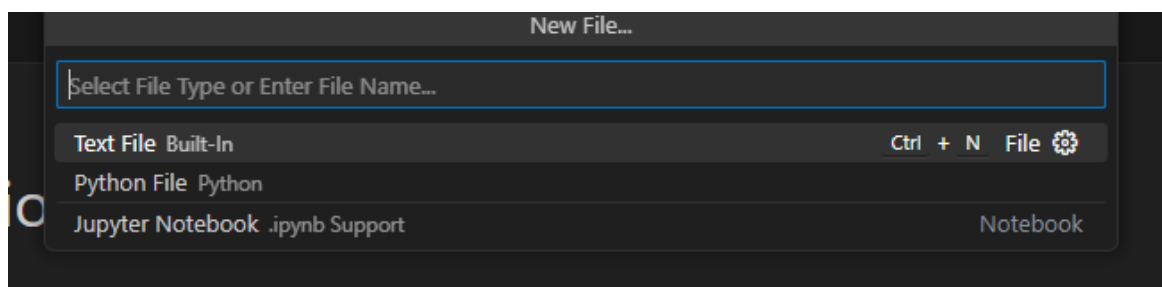
Tot el software utilitzat és de caràcter públic així que està a l'abast de qualsevol persona que vulgui fer ús d'ell.

MANUAL D'INSTAL·LACIÓ

Per poder fer servir el material utilitzat en aquest TFG caldria descarregar els tres arxius de codi i instal·lar el programa *Visual Studio Code*.

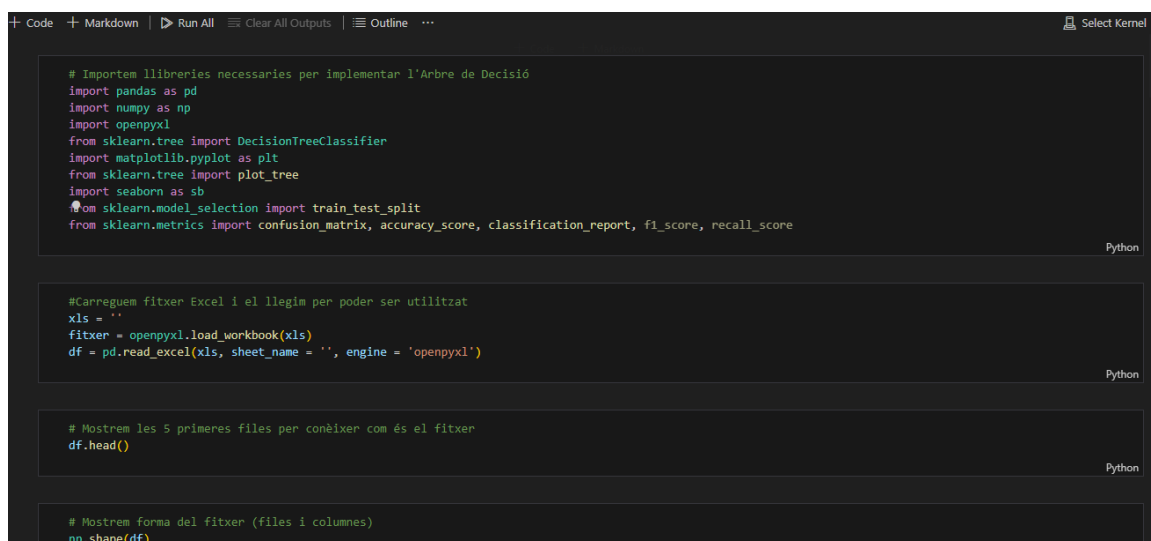


Un cop descarregat el programa s'hauria de clicar A *File* i obrir els fitxers descarregats. Seguidament, s'hauria d'escollir el tipus *Jupyter Notebook*, ja que els fitxers està en aquest format.



MANUAL PER L'USUARI

Un cop obert el fitxer que ens interessa ens trobaríem en el següent punt:



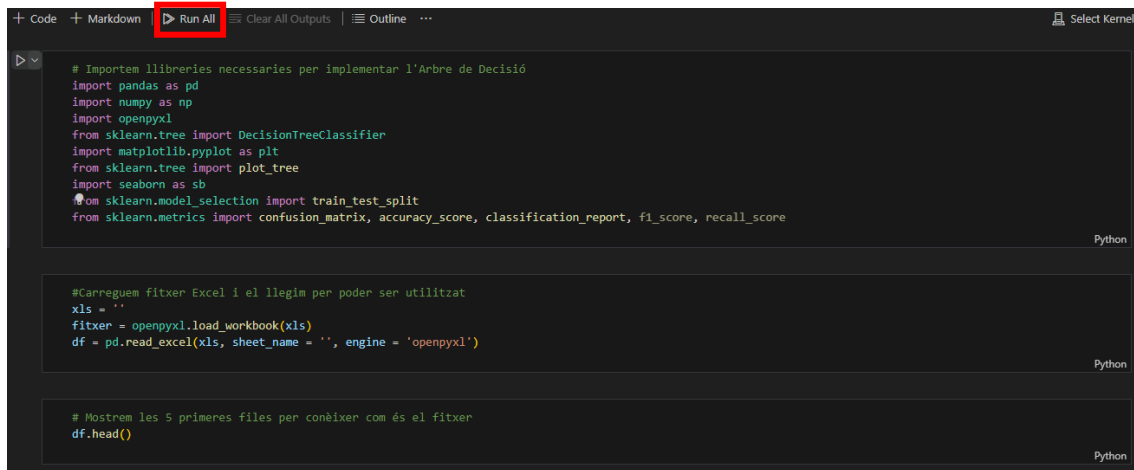
El primer que caldria fer és introduir la direcció del fitxer que es vulgues llegir. Concretament en el segon bloc de codi, per emplenar la comanda `xls = ''`.

S'hauria de completar les següents línies de codi introduint el nom de la variable *target* entre les cometes, o on esta situada la X en el cas de la segona imatge.

```
# Agrupem per la variable Target per saber com d'equilibrat està el conjunt de dades
df.groupby('').size()
```

```
#Mostrem plot de la variable X en relació a la quantitat que hi ha a la base de dades
sb.histplot(x = df.X, hue = df.OBE1TY_5Y_grup)
```

Finalment, donant-li a la següent instrucció es compilaria el codi.



The screenshot shows a Jupyter Notebook interface with a dark theme. At the top, there is a toolbar with buttons for '+ Code', '+ Markdown', 'Run All' (highlighted with a red box), 'Clear All Outputs', 'Outline', and 'Select Kernel'. Below the toolbar, the notebook contains three code cells, each with a 'Python' label on the right. The first cell contains a block of import statements for pandas, numpy, openpyxl, sklearn.tree, matplotlib.pyplot, sklearn.tree, sklearn.model_selection, and sklearn.metrics. The second cell contains code to load an Excel file into a pandas DataFrame. The third cell contains a single line of code to display the first five rows of the DataFrame.

```
+ Code + Markdown Run All Clear All Outputs Outline ... Select Kernel

# Importem llibreries necessaries per implementar l'Arbre de Decisió
import pandas as pd
import numpy as np
import openpyxl
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
import seaborn as sb
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, f1_score, recall_score

#Carreguem fitxer Excel i el llegim per poder ser utilitzat
xls = ''
fitxer = openpyxl.load_workbook(xls)
df = pd.read_excel(xls, sheet_name = '', engine = 'openpyxl')

# Mostrem les 5 primeres files per conèixer com és el fitxer
df.head()
```

Caldria seguir el mateix procés pels 3 fitxers, a partir d'aquí el codi seria capaç de llegir el fitxer i aniria generant els diferents models i mostrant totes les mètriques extretes per poder avaluar el model i extreure totes les conclusions pròpies de cada un d'ells.