# Configuration Manual

MSc Research Project
Data Analytics

Michael Dunne
Student ID: x15420892

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Michael Dunne | | |
| **Student ID:** | X15420892 | | |
| **Programme:** | Data Analytics | **Year:** | 2021 |
| **Module:** | Research Project | | |
| **Lecturer:** | Noel Cosgrave | | |
| **Submission Due Date:** | 16/12/2021 | | |
| **Project Title:** | Configuration Manuel | | |
| **Word Count:** | **Page Count:** 9 | | |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Michael Dunne

**Date:** 16/12/2021

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Michael Dunne
Student ID: x15420892

## 1 Requirements Guide

Please follow all the steps in this manual in order to run the code that has been used in the research project document that compliments this configuration manual. This document will discuss the requirements both hardware and software that is minimally needed in order to reproduce this code.

## 2 Machine Hardware requirements

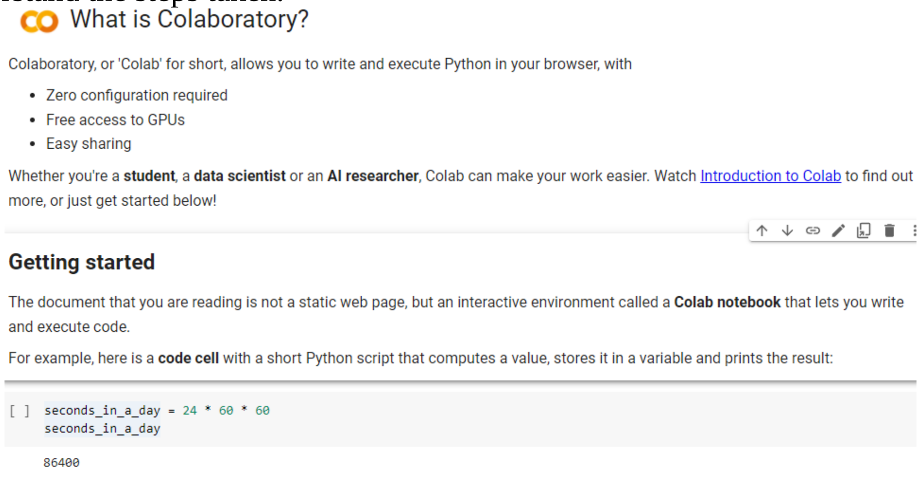
In order to run the code, the following machine hardware will need to be met for the project to work. The machine that preformed this research configuration is: 8gb RAM, AMD RADEON R5 3 GHz processor, 64-bit OS, windows 10.

## 3 Machine Software requirements

The following software requirements are needed to run the code and are the ones that have been used. Google Colaboratory is used as the environment for the research code. Python was used as the language for this project and version 3.7 respectfully. Google drive/Gmail account is used to link to notebook. Jupyter notebook is needed, Microsoft excel is needed as data stored as csv file. Microsoft word was used to write the research project report and this manual.

## 4 Environment set up

Here is where the setting up of the colab environment is done, following these steps will allow the code to run for our research project. The steps are also accompanied by images to better understand the steps taken.



What is Colaboratory?

Colaboratory, or 'Colab' for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to find out more, or just get started below!

**Getting started**

The document that you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

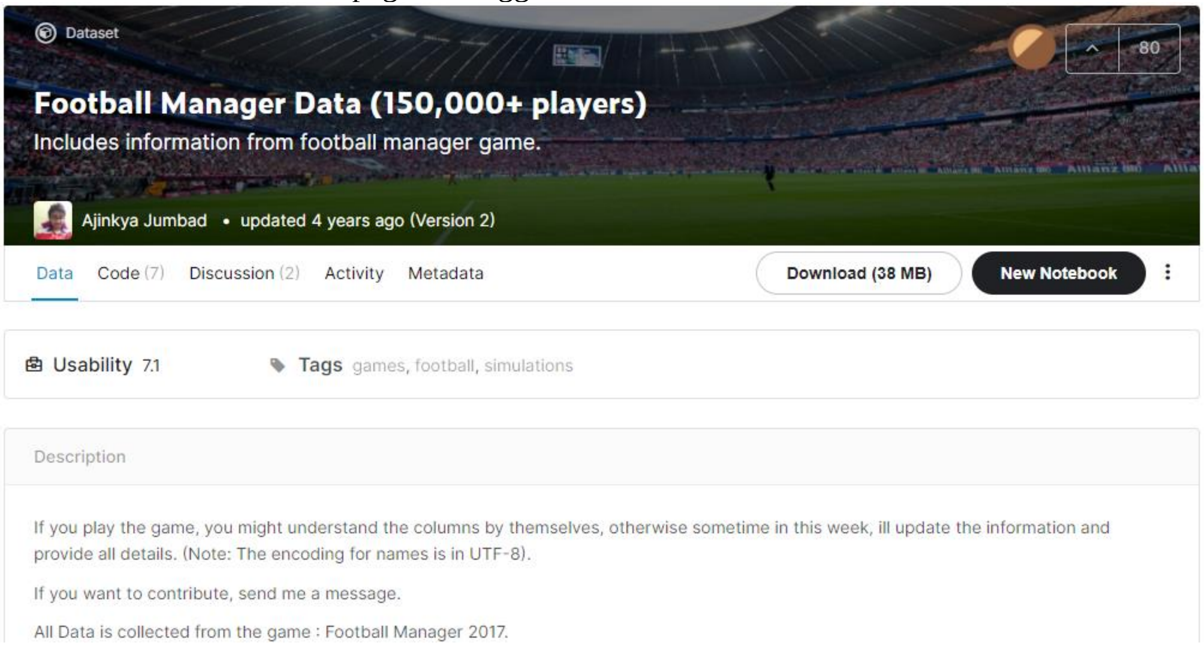
For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day

    86400
```

Figure 1: Google Colab environment

## 5 Data selection process

The dataset that was used for this research is from the open source dataset website Kaggle.com. The dataset is called Football Manager Data (150,000+ players). Figure 2 shows the overview of the dataset page on Kaggle.


Dataset
Football Manager Data (150,000+ players)
Includes information from football manager game.
Ajinkya Jumbad • updated 4 years ago (Version 2)
80
Data
Code (7)
Discussion (2)
Activity
Metadata
Download (38 MB)
New Notebook
Usability 7.1
Tags games, football, simulations
Description

If you play the game, you might understand the columns by themselves, otherwise sometime in this week, ill update the information and provide all details. (Note: The encoding for names is in UTF-8).

If you want to contribute, send me a message.

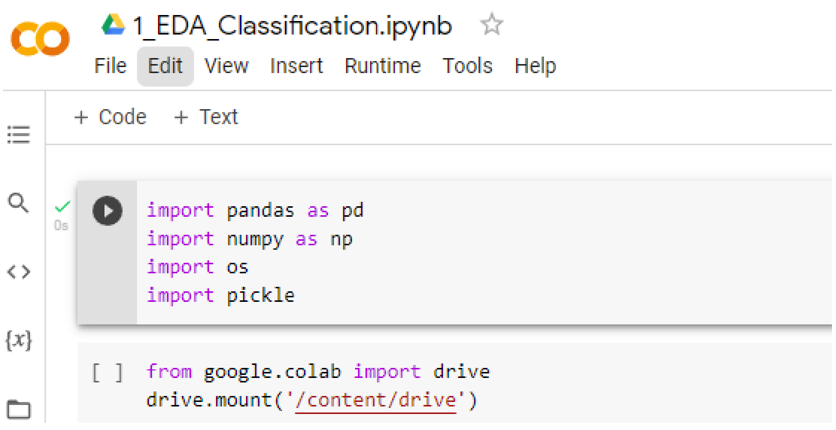
All Data is collected from the game : Football Manager 2017.

Figure 2. Football Manager dataset

## 6 Install Libraries

For this research the following libraries need to be installed in order for the research to completely work, otherwise results may differ depending on some libraries not being installed correctly. The below list details all libraries used, with example of ipynb. File 1.

1. Pandas
2. Numpy
3. Tensorflow
4. Scikit-learn
5. Matplotlib
6. Lightgbm
7. OS
8. Pickle



```
import pandas as pd
import numpy as np
import os
import pickle
```

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

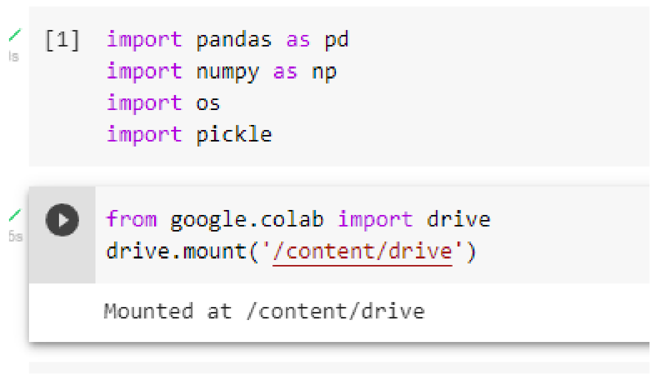
Figure 3. Import libraries for file.1 (EDA classfication task)

## 7 Implementation and using the code files.

The code files are quite straight forward to use. A quick breakdown of the specifc files that are in the folder. There is 6 jupyter notebook files and also two .py files. The .py files have the two models and functions to load the data. These models and functions are imported in the relevant jupyter notebooks. This means that the .py files should NOT be run directly because jupyter notebooks are to be used instead. See each jupyter notebook file is numbered and we detail a description below of each.

- 0: This file performs EDA and feature selection for regression task.
- 1: This file performs EDA and feature selection for classification task.
- 2: This file runs machine learning algorithms for both regression and classification tasks. The results are saved in "pickle_Data" folder for use in visualization stage.
- 3: This file is for running CNN models on both regression and classification tasks. The results are shown in the notebook and are not saved.
- 4: This file is used for comparing the regression models performances.
- 5: This file compares the classification models performances.

Once the data is downloaded off Kaggle and uploaded to google drive as well as all the files. Mounting the google drive must then be done as soon below in figure 4. Please accept access from Gmail to access google drive files.



```
[1] import pandas as pd
    import numpy as np
    import os
    import pickle
```

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

Mounted at /content/drive
```

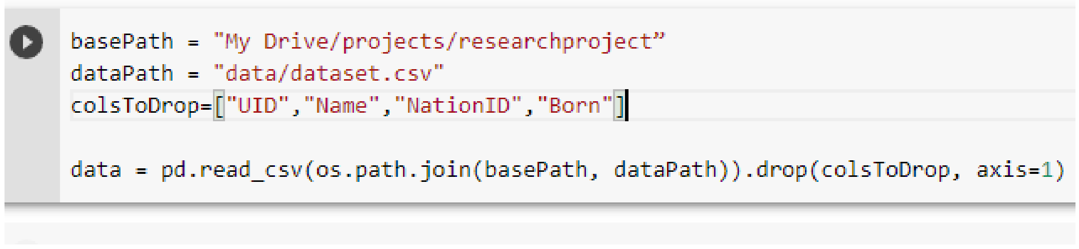
Figure 4. Mounted drive

To run the Jupyter notebook directly on google colab. Just set the base path to your drive. For example, if you have places all the files on your google drive (as suggested) at location:
My Drive/projects/researchproject,

Then the basebath would be: basePath =”My Drive/projects/researchproject”

This should be set on all the notebook files as basepath respectfully, this will then run on colab. However, while running the machine learning models, you must upload the models.py file. For running the DNN or CNN, you also need to upload cnn_models.py file to colab. Screenshots later will show this example.

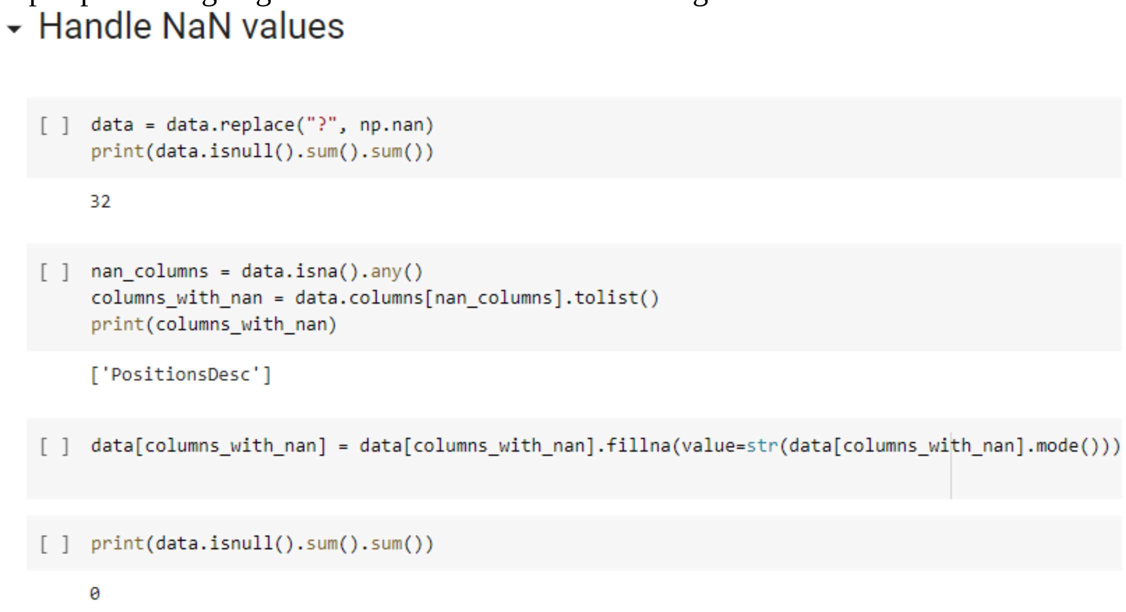
Once this is done, the data can now be read in as the datapath, and the pre-processing stage can begin. See figure 5.



```
basePath = "My Drive/projects/researchproject”
dataPath = "data/dataset.csv"
colsToDrop=["UID","Name","NationID","Born"]

data = pd.read_csv(os.path.join(basePath, dataPath)).drop(colsToDrop, axis=1)
```

Figure 5. basepath and datapath

Data pre-processing begins with the removal and handling of NaN values.

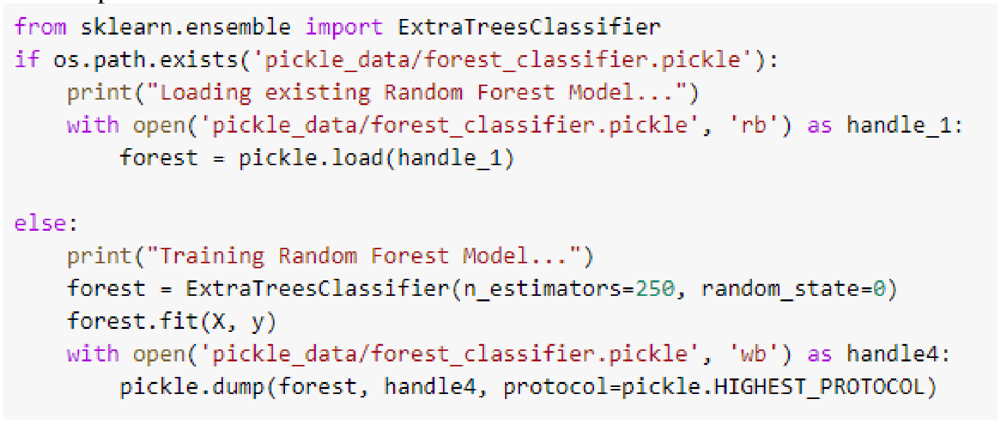


Handle NaN values

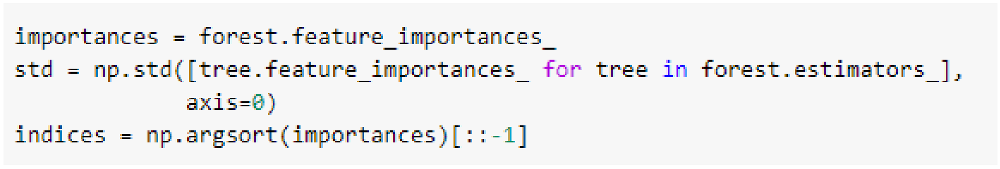
```
[ ] data = data.replace("?", np.nan)
    print(data.isnull().sum().sum())

    32

[ ] nan_columns = data.isna().any()
    columns_with_nan = data.columns[nan_columns].tolist()
    print(columns_with_nan)

    ['PositionsDesc']

[ ] data[columns_with_nan] = data[columns_with_nan].fillna(value=str(data[columns_with_nan].mode()))

[ ] print(data.isnull().sum().sum())

    0
```

Figure 6. Handling NaN values

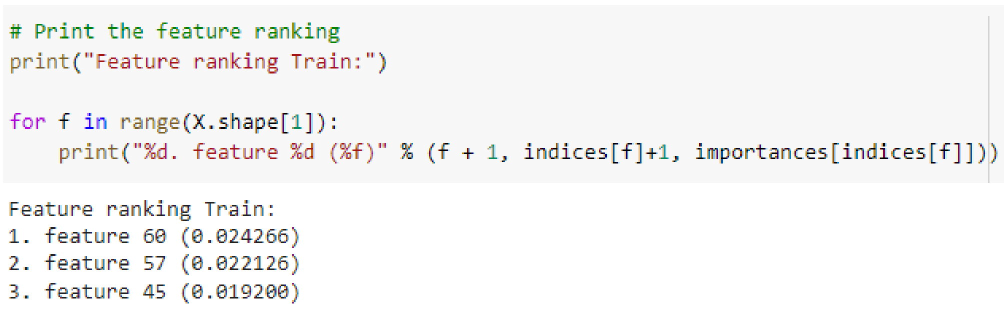
Next in this file, discretize label Y into classes of “high”, “medium” and “low” for the risk of injury proneness. Based off of threshold. Next then the labels are separated before feature importance can begin. This was done using random forest feature importance as seen in figure 7. The top 50 features were used for our models.



```
from sklearn.ensemble import ExtraTreesClassifier
if os.path.exists('pickle_data/forest_classifier.pickle'):
    print("Loading existing Random Forest Model...")
    with open('pickle_data/forest_classifier.pickle', 'rb') as handle_1:
        forest = pickle.load(handle_1)

else:
    print("Training Random Forest Model...")
    forest = ExtraTreesClassifier(n_estimators=250, random_state=0)
    forest.fit(X, y)
    with open('pickle_data/forest_classifier.pickle', 'wb') as handle4:
        pickle.dump(forest, handle4, protocol=pickle.HIGHEST_PROTOCOL)
```

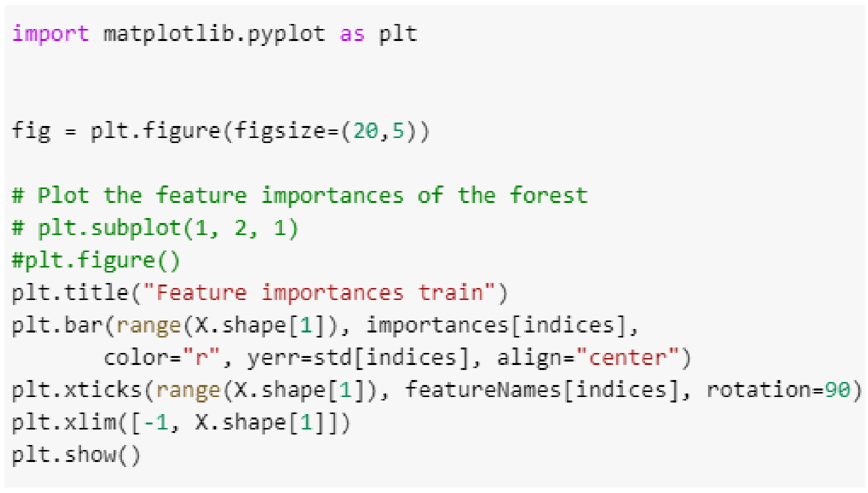


```
importances = forest.feature_importances_
std = np.std([tree.feature_importances_ for tree in forest.estimators_],
             axis=0)
indices = np.argsort(importances)[::-1]
```

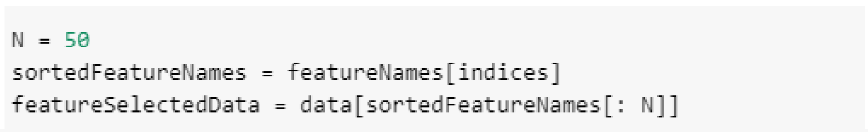


```
# Print the feature ranking
print("Feature ranking Train:")

for f in range(X.shape[1]):
    print("%d. feature %d (%f)" % (f + 1, indices[f]+1, importances[indices[f]]))
```

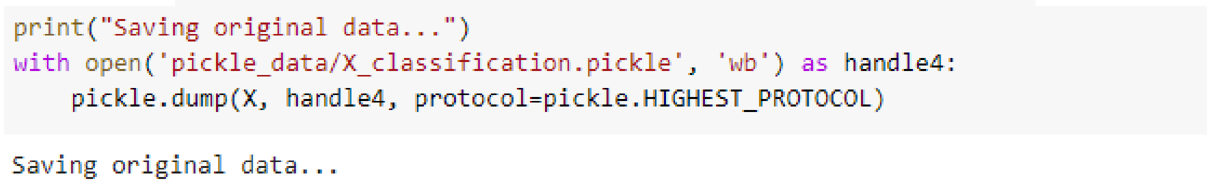
```
Feature ranking Train:
1. feature 60 (0.024266)
2. feature 57 (0.022126)
3. feature 45 (0.019200)
```



```
import matplotlib.pyplot as plt


fig = plt.figure(figsize=(20,5))

# Plot the feature importances of the forest
# plt.subplot(1, 2, 1)
#plt.figure()
plt.title("Feature importances train")
plt.bar(range(X.shape[1]), importances[indices],
       color="r", yerr=std[indices], align="center")
plt.xticks(range(X.shape[1]), featureNames[indices], rotation=90)
plt.xlim([-1, X.shape[1]])
plt.show()
```

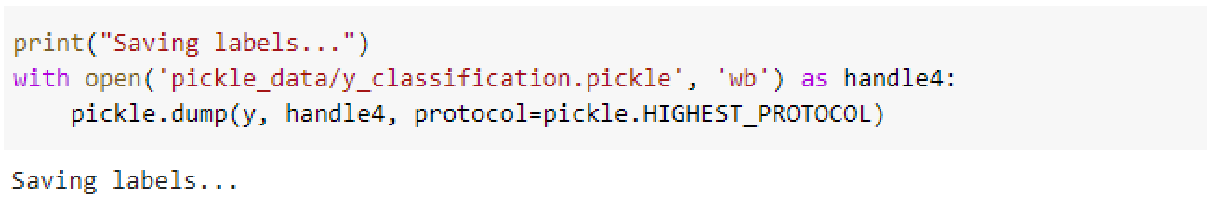


```
N = 50
sortedFeatureNames = featureNames[indices]
featureSelectedData = data[sortedFeatureNames[: N]]
```



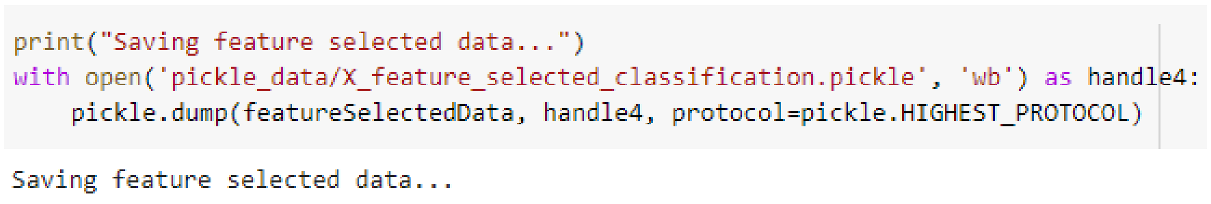
```
print("Saving original data...")
with open('pickle_data/X_classification.pickle', 'wb') as handle4:
    pickle.dump(X, handle4, protocol=pickle.HIGHEST_PROTOCOL)
```

```
Saving original data...
```



```
print("Saving labels...")
with open('pickle_data/y_classification.pickle', 'wb') as handle4:
    pickle.dump(y, handle4, protocol=pickle.HIGHEST_PROTOCOL)
```

```
Saving labels...
```



```
print("Saving feature selected data...")
with open('pickle_data/X_feature_selected_classification.pickle', 'wb') as handle4:
    pickle.dump(featureSelectedData, handle4, protocol=pickle.HIGHEST_PROTOCOL)
```

```
Saving feature selected data...
```

Figure 7. Feature Importance

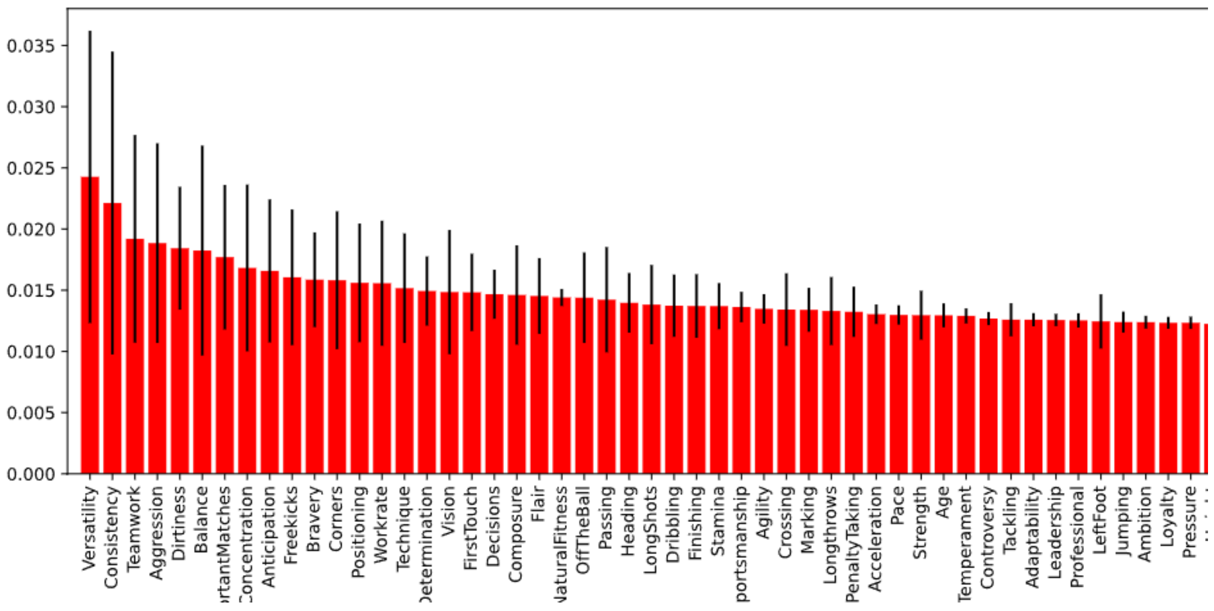
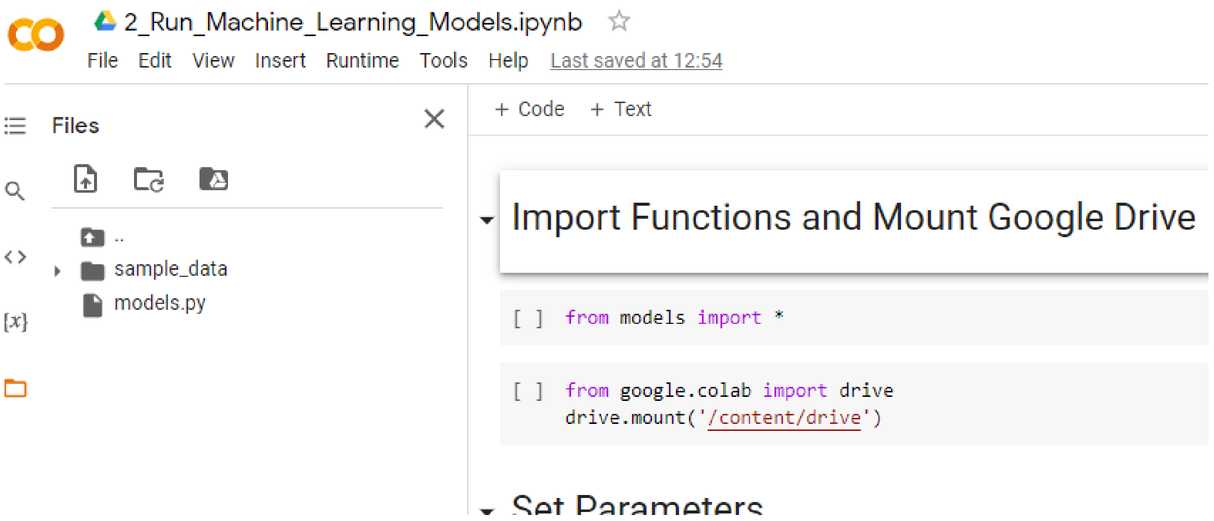


Figure 8. Feature Importance train

The same is done for SVM, linear regression, PCA and no feature selection in this file but for example shown is random forest for feature selection approach.

The figure 9 below relates to the earlier stage in which the models.py file must be uploaded to colab whilst running the 2: machine learning models. See drive files to left.



2_Run_Machine_Learning_Models.ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:54
+ Code + Text
Files
..
sample_data
models.py

Import Functions and Mount Google Drive

```
from models import *
```

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

Set Parameters

Figure 9. Running machine learning models.

The same must be done for the cnn_models.py file being uploaded to colab when running the 3: run_cnn_models file. As stated, do not run the .py files. Once these steps are complete the code will run smoothly and will be able to look at the model's performances and evaluation. Thank you for reading.

## References