National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Jaseem Jamal Panikkaveetil
Student ID: 21236828

School of Computing
National College of Ireland

Supervisor: Sasirekha Palaniswamy

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Jaseem Jamal Panikkaveetil

Student ID: 21236828

Programme: Msc in Data Analytics **Year:** 2023

Module: MSc Research Project

Lecturer: Sasirekha Palaniswamy

Submission Due Date: 14/12/2023

Project Title: Configuration Manual

Word Count: .....820..... **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: Jaseem Jamal Panikkaveetil

Date: 14/12/2023

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

Jaseem Jamal Panikkaveetil
Student ID: 21236828

1 Introduction

The configuration manual file describes the guidelines to implement the research project 'Ball Possession metrics-oriented Analysis to Predict Football League Rankings'. The hardware and software requirements of the project are also specified in this report.

2 Hardware Configuration

The hardware requirements for the project are given below.

- Processor: Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz 2.81 GHz
- RAM: 8GB to 16GB of RAM is usually preferred
- Storage: 128/ 500 GB SSD
- OS: Windows 10 Home Single Language version 20H2

3 Software Configuration

- Jupyter Notebook - Version: 6.4.6
- Microsoft Excel
- Python

4 Environment Setup

4.1 Setup Of Code

- This part will go over how to set up the Project. Anaconda has been downloaded and installed on your system. A new environment is established, and the most recent packages are installed. Jupyter Notebook and the Anaconda CMD prompt are both installed. Figure 2 shows the installed tools.
- Once the installation is complete, launch Anaconda Navigator from the Start menu. Click on the Jupyter Notebook icon to launch the application.

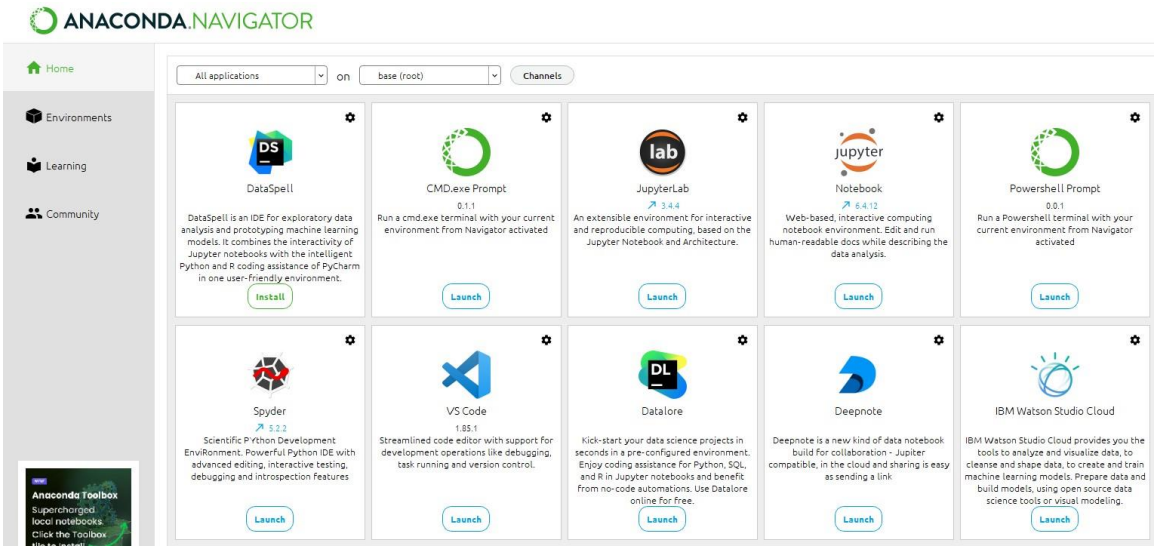

Figure 1: Anaconda Navigator

- Download python from official python website.
- Once the installation is complete, check the version by using the command “python -version”.

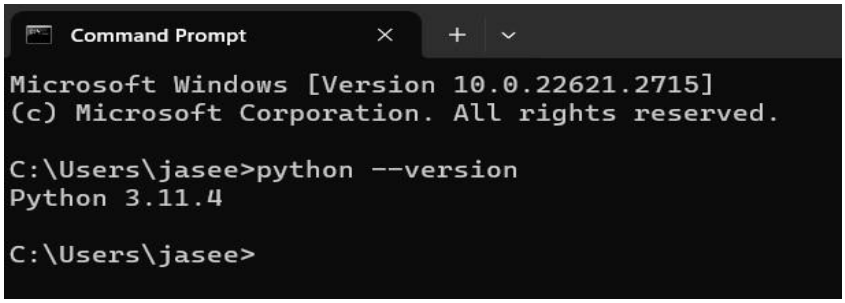

Figure 2: Python Version

- Find the folder named ‘21236828_CodeArtifacts’.
- There are python files and input, input1 folder that contains all datasets, Open the “1_Segregation_File1” file and run the commands similarly do it for the rest of the all the files except “1_Segregation_File-4-Seasons”. This will give prediction of 2023/2024 season in test files. “1_Segregation_File-4-Seasons” is loaded to get predictions of the 2021/2022 season.

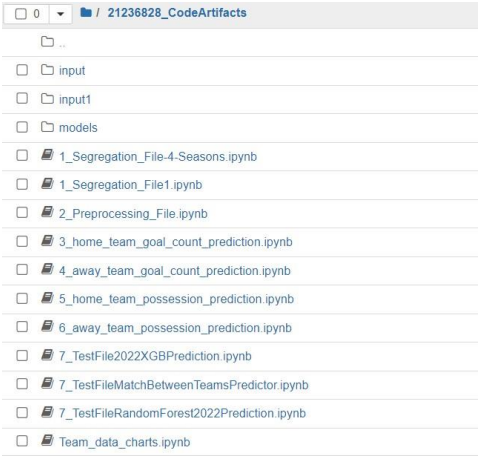

Figure 3 Folder Structure

5 Data Selection

The dataset is sourced from a public website, which provides Football related datasets. The dataset used here is Premier League Dataset: https://footystats.org/england/premierleague

6 Implementation

6.1 Data pre-processing

- The data is loaded as a data frame and saved as df, selecting the necessary columns and finally converting the time object to time data as shown in Figure 6.

Data Loading

```
df = pd.read_csv("match_data.csv")
df.head()
```

Figure 4 Load Data

- Only the required columns are selected. The rest are discarded in the pre-processing step.

```
selected_columns = ['home_team_name', 'away_team_name', 'home_team_goal_count', 'away_team_goal_count',
                    'total_goal_count', 'home_team_corner_count', 'away_team_corner_count', 'home_team_shots',
                    'away_team_shots', 'home_team_shots_on_target', 'away_team_shots_on_target',
                    'home_team_shots_off_target', 'away_team_shots_off_target', 'home_team_possession',
                    'away_team_possession', 'team_a_xg', 'team_b_xg']

df_selected = df[selected_columns]
```

Figure 5 Selecting Necessary columns

- The filetype of “date_GMT “ is converted from object to usable format.

```
df['date_GMT'] = pd.to_datetime(df['date_GMT'], format='%b %d %Y - %I:%M%p')
df.head()
```

Figure 6 Converting date object to date format

```
df_selected.to_csv("preprocessed_data.csv", index=False)
```

Figure 7 The preprocessed dataset is obtained

6.2 Model Building

First, the 1_Segregation_File1 is opened and click on run all. The data set is divided to training and testing dataset here. With train test split function with a test size of 20% and random state of 42, dataset acquired from the preprocessed data.csv file was suitably divided into training and testing sets. The train and test sizes are given below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(1520, 16) (380, 16) (1520, 1) (380, 1)
```

Figure 8 Splitting the dataset

The four Models are trained as shown below.

6.2.1 Random Forest

Algorithm: 1 RandomForestRegressor

```
from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor()
rf_model = rf_model.fit(X_train.values, y_train.values.ravel())
```

```
rf_pred = rf_model.predict(X_test.values)
rf_pred = [int(goal) for goal in rf_pred]
print(rf_pred)
```

Figure 9 Random Forest

6.2.2 XGBoost

Algorithm: 2 XGBoostRegressor

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\91756\anaconda3\li
Requirement already satisfied: numpy in c:\users\91756\anaconda3\lib\
Requirement already satisfied: scipy in c:\users\91756\anaconda3\lib\
Note: you may need to restart the kernel to use updated packages.
```

```
from xgboost import XGBRegressor
xgb_model = XGBRegressor()
xgb_model = xgb_model.fit(X_train.values, y_train.values.ravel())
```

```
xgb_pred = xgb_model.predict(X_test.values)
xgb_pred = [int(goal) for goal in xgb_pred]
print(xgb_pred)
```

Figure 8 XGBoost

6.2.3

Algorithm: 3 LGBMRegressor

```
from lightgbm import LGBMRegressor
lgbm_model = LGBMRegressor()
lgbm_model = lgbm_model.fit(X_train.values, y_train.values.ravel())
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of t
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 806
[LightGBM] [Info] Number of data points in the train set: 1520, number of u
[LightGBM] [Info] Start training from score 1.530263
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

Figure 9 : LightGBM

6.2.4

Algorithm: 4 Ridge

```
from sklearn.linear_model import Ridge
ridge_model = Ridge()
ridge_model = ridge_model.fit(X_train.values, y_train.values.ravel())
```

```
ridge_pred = ridge_model.predict(X_test.values)
ridge_pred = [int(goal) for goal in ridge_pred]
print(ridge_pred)
```

Figure 10: Ridge

As shown in the above four sections, the code files 4_away_team_goal_count_prediction , 5_home_team_possession_prediction , 6_away_team_possession_prediction are run in the same way. This will give the models for away team goal count, home team possession and away team possession.

7. Testfiles

Run 7_TestFile2022XGBPrediction.

This will load the pickle module shown below.

```
In [115]: with open(file="models/XGBRegressor_hometeam_goal.pkl", mode="rb") as file:
              home_team_goal_model = pickle.load(file=file)

          Loading away team goal prediction model

In [116]: with open(file="models/XGBRegressor_awayteam_goal.pkl", mode="rb") as file:
              away_team_goal_model = pickle.load(file=file)

          Loading home team possession prediction model

In [117]: with open(file="models/XGBRegressor_hometeam_possession.pkl", mode="rb") as file:
              home_team_possession_model = pickle.load(file=file)

          Loading away team possession prediction model

In [118]: with open(file="models/XGBRegressor_awayteam_possession.pkl", mode="rb") as file:
              away_team_possession_model = pickle.load(file=file)
```

Figure 11: Loading pickle modules

The below image shows the possession based metrics conditions. This will give ranking prediction using XGBoost.

```
# Function to calculate points based on conditions
def calculate_points(home_team_name, away_team_name, home_goals, away_goals, home_possession, away_possession):
    if home_goals > away_goals and home_possession > away_possession:
        # Condition 1
        return home_team_name, 3  # Home team win
    elif home_goals < away_goals and home_possession < away_possession:
        # Condition 2
        return away_team_name, 3  # Away team win
    elif home_goals >= away_goals and home_possession <= away_possession:
         # Condition 3
        return 'Draw', 1  # Draw
    elif home_goals <= away_goals and home_possession >= away_possession:
        # Condition 4
        return 'Draw', 1  # Draw
    else:
        # Condition 5
        return 'No Result', 0  # No points
```

Figure 12: Possession based metrics oriented conditions Similarly, **7_TestFileRandomForest2022Prediction** is run to get the ranking prediction using Random Forest

Similarly, **Run 7_TestFileMatchBetweenTeamsPredictor**, this will give the individual matches prediction using Random Forest Model.