|  |  |  |
| --- | --- | --- |
| **Serial. No** | **Chapter** | **Page.No** |
| **1.** | **Acknowledgment** | **3** |
| **2.** | **Abstract** | **6** |
| **3.** | **Chapter - 1 Introduction** | **7** |
| **4.** | **Chapter - 2 Requirement and proposed system** | **8** |
| **5.** | **Chapter - 3 Module description** | **15** |
| **7.** | **Chapter - 4 Result and discussion** | **17** |
| **8.** | **Chapter - 5 Conclusion** | **20** |
| **9.** | **Chapter - 6 Reference** | **21** |

**ABSTRACT**

In sports, football is one of the largest sporting events in the world. People from all over the world love and follow this sport with passion. With this passion, comes the urge to bet for the upcoming fixtures. So we aim to develop a classification model for predicting the outcome for a match.

The goal of this project is to predict match outcomes of European football matches more accurately than bookkeepers based on the real time factors associated with the match rating, previous meeting H2H results and home-away advantage bases.The data we use are web scrapped FIFA data and from betting sites hence, thereby beating the odds and in the end, generate a positive return on investment. Our model predicts the outcome – win, draw, or defeat.

1. **INTRODUCTION**

Football in British English or Soccer in American English is one of the most followed sports in the world of sports. People enjoy both playing and watching football matches.Fanbases of some football clubs are so mighty that the clubs primary income is through certain merchandise , broadcasting fees and through the odd betting sites which based on previous data stored in their record predicts the match and does the book keeping.

Analytics of sports is so vast such that even small information regarding individual players and the whole team is readily available for doing feature engineering. But this data is being used for the organizations benefit mostly.

Our prediction model does the necessary feature extraction from the real time data of the whole team and the players fifa rating along with data of previous win, draw or defeat of the home and away team which have been web scrapped from some of the famous bookkeeping sites.

By using the bookkeeper’s outcomes the outcome which was computed by algorithm is compared, the accuracy percentage of each algorithm is plotted. The average accuracy of the whole system is computed from this.

After plotting the accuracy rates, confusion matrix is plotted for each algorithm which gives the detailed comparison between the truth value and the predicted value.

1. **Requirement and proposed system**

**2.1 DATASET DESCRIPTION**

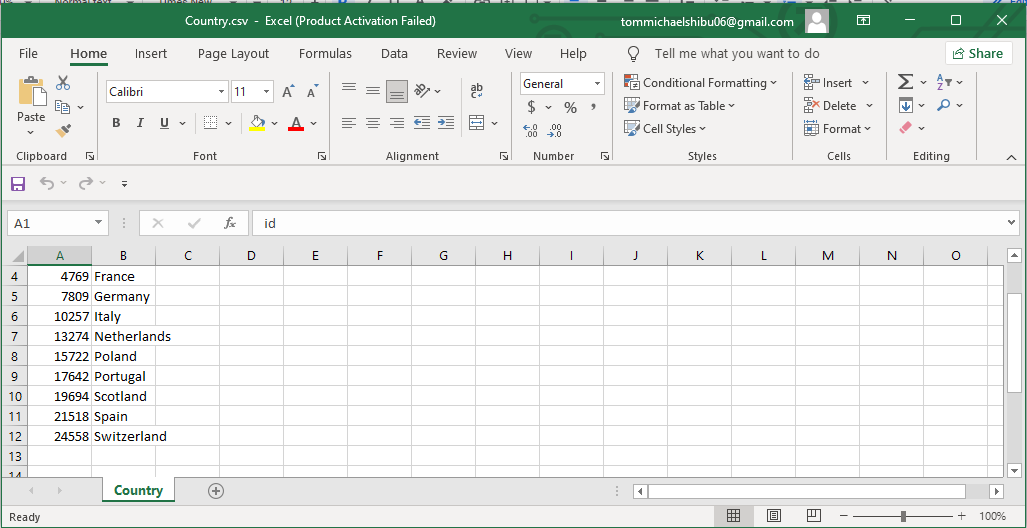
In this chapter we will be discussing the datasets we use for our project.

Let us discuss the dataset before we go to requirements for this project.

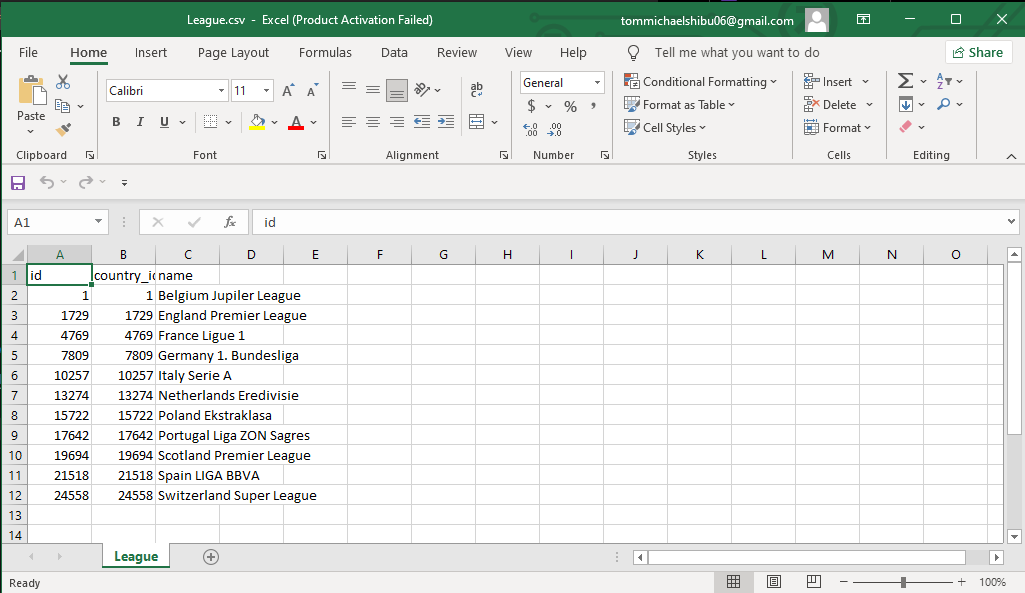
In this chapter we will be discussing briefly about the model , its functionality and its workflow.

Our database is divided into 8 sub dataset:

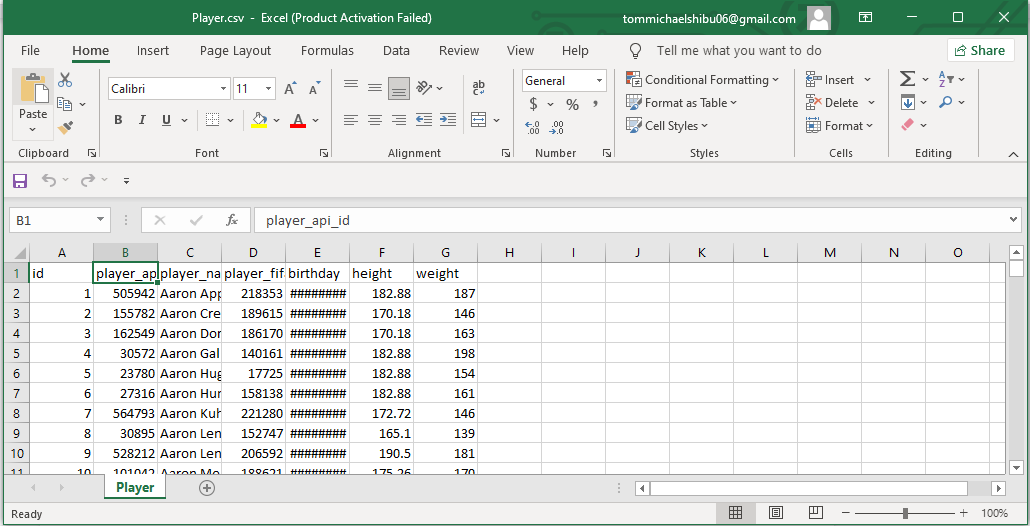
1. Country
2. League
3. Player
4. Player\_Attributes
5. Team
6. Team\_Attributes
7. sqlite\_Sequence
8. Bookkeepers data
9. **Country :** Contains all the Major European nations which have an ID associated with it to map to the Player and Team datasets which can be used to uniquely identify each set of players or associate them with their respective nations.

****

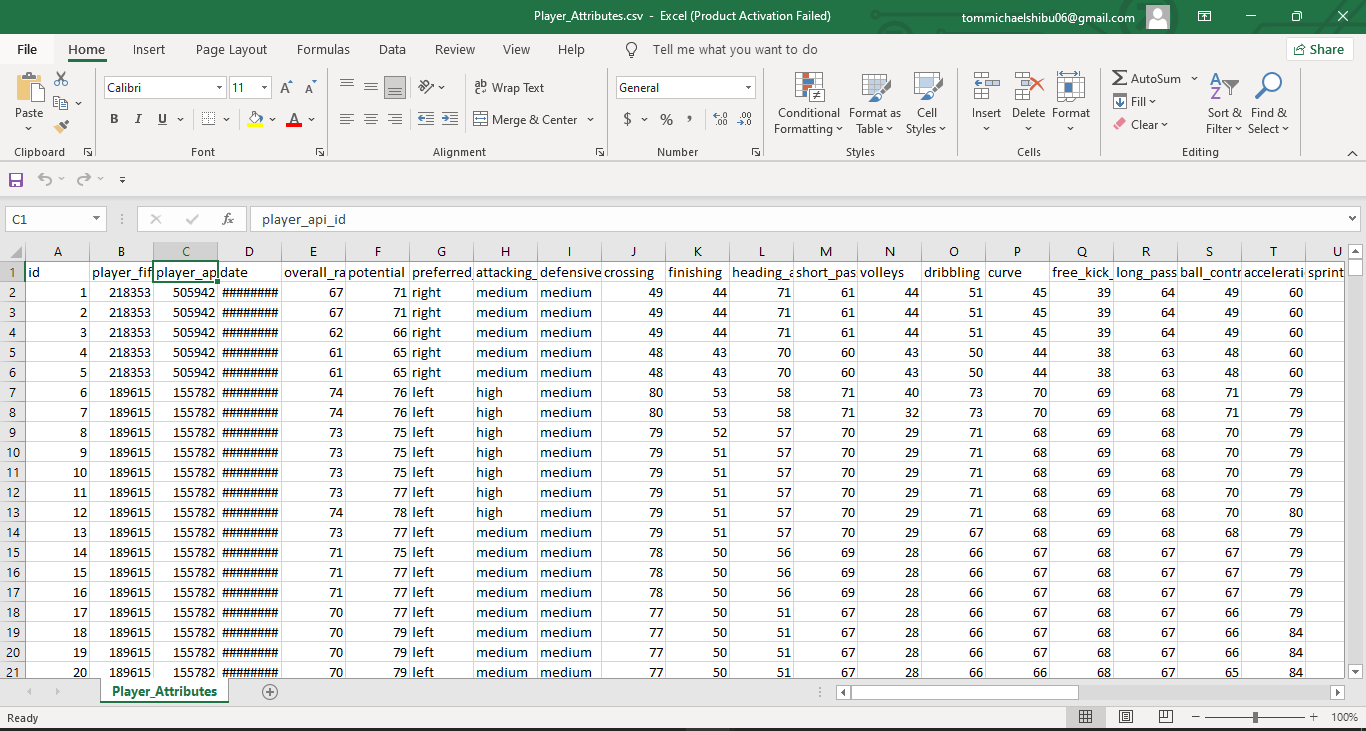
1. **League :** Contains name of all the major leagues in Europe and it is used to map them or associate them with their respective clubs who play in the league.



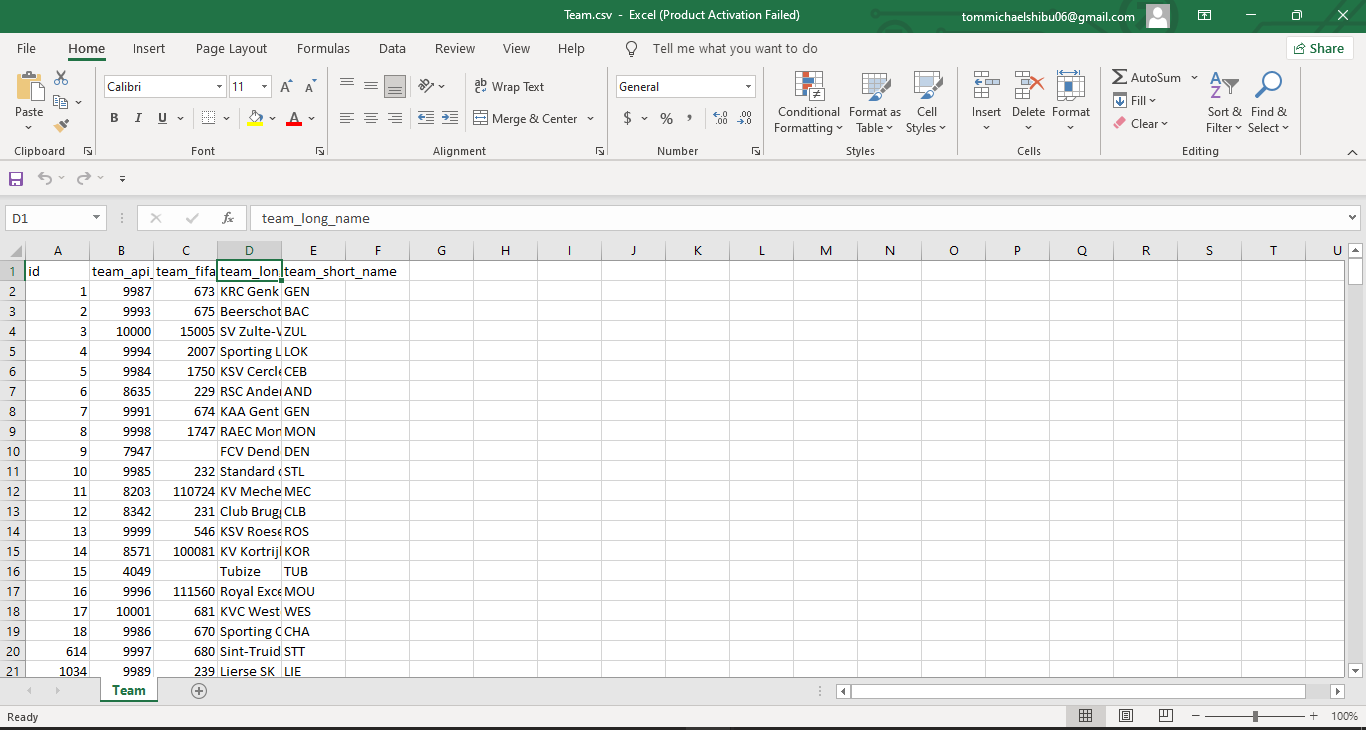
1. **Player :** Contains name of the player with their physique details (Height, Weight, Age ,DOB) the unique ID used for mapping this is the player\_api\_id and the player\_fifa\_api\_id which is used for mapping to fifa rating.



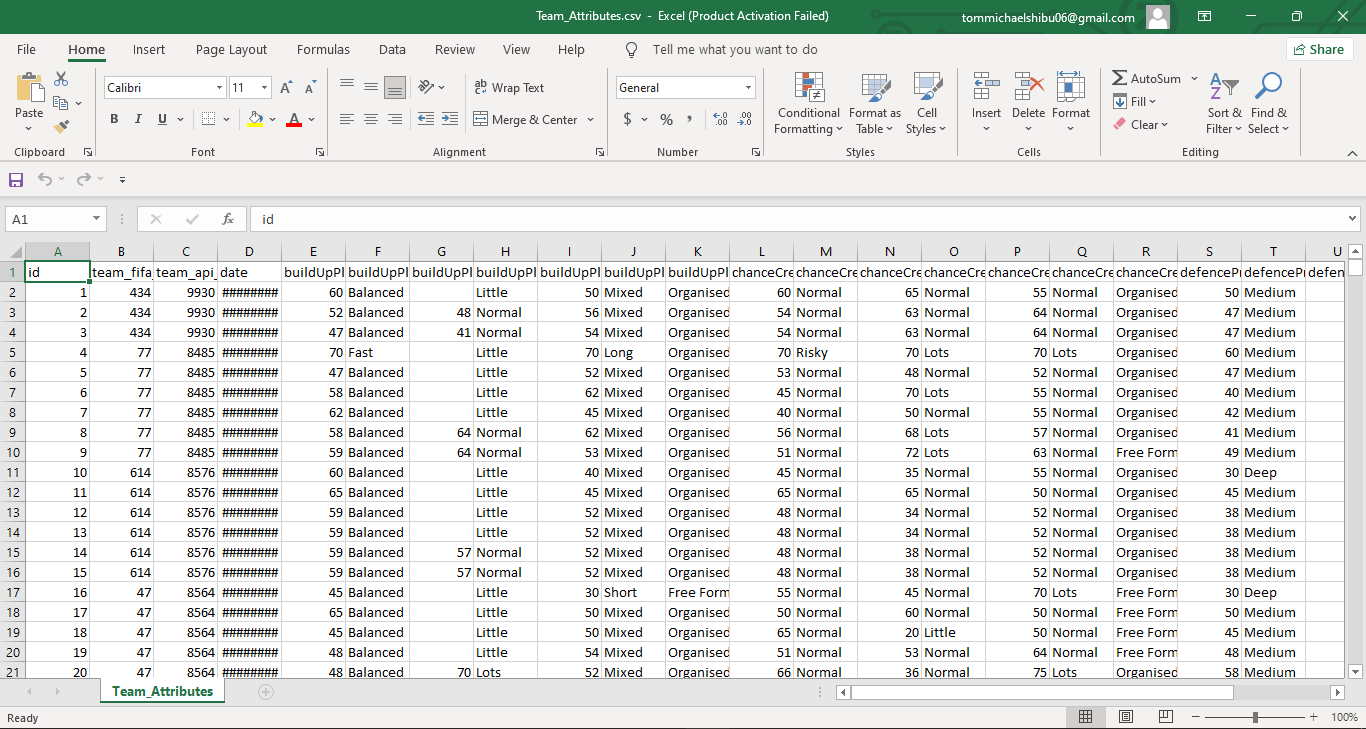
1. **Player\_Attribute :** In this dataset the most important features which we use for our project are stored which have been discussed in the above chapter (5). Here the player\_api\_id and the player\_fifa\_api\_id are used to associate with the Player dataset.

****

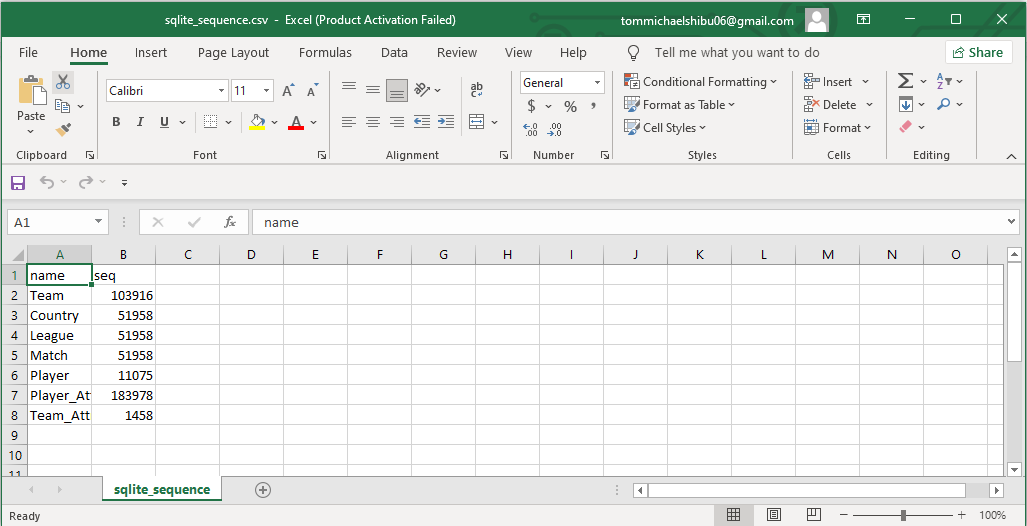
1. **Team :** Basic Team information such as team short name and long name are stored also in this dataset there are api\_id which are the primary key in this dataset and foreign key in team\_attribute dataset which can be used to map the linkage between both the datasets.



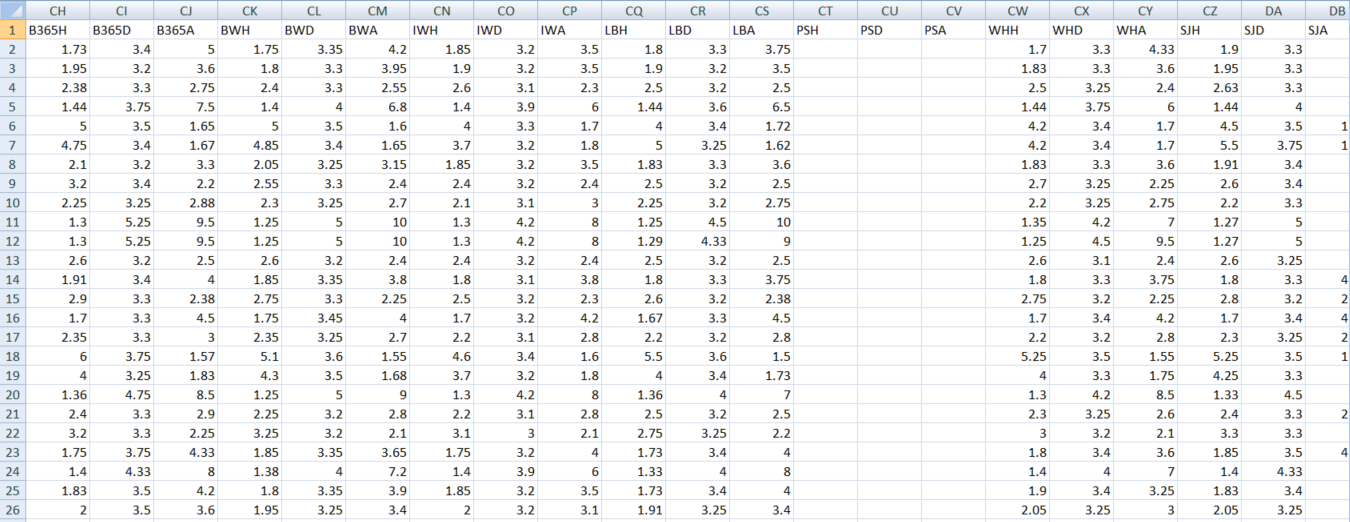
1. **Team\_Attribute :** Contains details about sports associated characters of a team and the football philosophy used in a particular club.



1. **Sqlite Sequence :** SQL sequencing for arranging all the dataset in an order for extracting features according to the need of the model.



1. **Bookkeepers data :** Contains record of the win, draw and defeat odds of the famous betting sites.



**2.2Proposed System:**

In order to get best prediction results, we need to try different classifiers such as GaussianNB, RandomForest and KNeighbors. Our model is built for the sole purpose of prediction of the match outcome based on the features extracted from the bookkeepers data which we acquire from kaggle under the name of [European Soccer Database](https://www.kaggle.com/hugomathien/soccer). Model mainly focuses on the following parameters for prediction.

* home\_team\_goals\_difference
* away\_team\_goals\_difference
* home\_player\_overall\_rating(1-11)
* away\_player\_overall\_rating(1-11)
* Betting\_siteWin( in terms of odds to win)
* Betting\_siteDraw(in terms of odds to draw)
* Betting\_siteDefeat(in terms of odds to defeat)

The above mentioned features are extracted from the whole database which are stored under the X\_train dataframe for the purpose of training efficiently.

Functions:

1. get\_match\_label(): Derives a label for a given match.
2. get\_fifa\_stats(): Aggregates fifa stats for a given match.
3. get\_fifa\_data(): Gets fifa data for all matches.
4. get\_overall\_fifa\_rankings(): Get overall fifa rankings from fifa data.
5. get\_last\_matches(): Get the last x matches of a given team.
6. get\_last\_matches\_against\_eachother(): Get the last x matches of two given teams.
7. get\_goals(): Get the goals of a specific team from a set of matches.
8. get\_goals\_conceided(): Get the goals conceded of a specific team from a set of matches.
9. get\_wins():Get the number of wins of a specific team from a set of matches.
10. get\_match\_features():Create match specific features for a given match.
11. create\_feables(): Create and aggregate features and labels for all matches.
12. train\_classifier(): Fits a classifier to the training data.
13. predict\_labels(): Makes predictions using a fit classifier based on scorer.
14. train\_calibrate\_predict(): Train and predict using a classifier based on scorer.
15. convert\_odds\_to\_prob(): Converts bookkeeper odds to probabilities.
16. get\_bookkeeper\_data(): Aggregates bookkeeper data for all matches and bookkeepers.
17. get\_bookkeeper\_probs(): Get bookkeeper data and convert to probabilities for vertical aggregation.
18. compare\_probabilities(): Map bookkeeper and model probabilities.
19. find\_best\_classifier():Tune all classifier and dimensionality reduction combinations to find best classifier.
20. plot\_training\_results(): Plot results of classifier training.
21. plot\_confusion\_matrix(): Plot confusion matrix for given classifier and data.
22. plot\_bookkeeper\_cf\_matrix():''' Plot confusion matrix of bookkeeper predictions. '''
23. **MODULE DESCRIPTION**

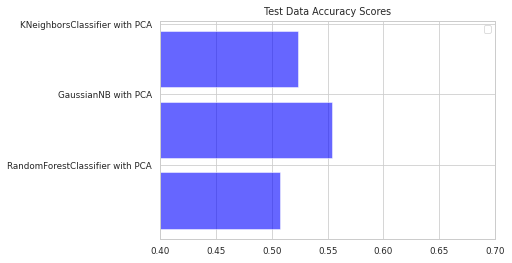
In this chapter, the modules used in this project are detailed and python is used as the programming language and google colab is used for the implementation of the code.

For data extraction and analyzing the data following libraries are used,

1. **sqlite -**  It is a self-contained, file based SQL database. It is a powerful part of the Python standard library which helps to work with a fully featured on-disk SQL database without installing any additional libraries.
2. **pandas** - pandas is an open source library in Python. It provides ready to use high-performance data structures and data analysis tools. It has a higher level interface. It also provides streamlined alignment of tabular data and powerful time series functionality. Data frames is the key data structure in Pandas which allows to store and manipulate tabular data as 2-D data structure.
3. **numpy** - numpy is the default library for working with numerical data in Python. it contains multidimensional array and matrix data structures. It provides ndarray, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to python that guarantee efficient calculations with arrays and matrices and it provides a high level mathematical function that operates on these arrays and matrices.
4. **Seaborn** - It is used for data visualization and exploratory data analysis and to plot functions which operate on dataframes and arrays containing complete datasets and to perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of the plots mean.
5. **Matplotlib -** In data analysis it helps to understand the large amount of data through different visualisations and graphical plotting, it is a numerical extension of Numpy. It can also act as an alternative to MATLAB.
6. **Sklearn** - It is one of the robust libraries in Python which provides a selection of efficient tools for statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. It implements several loss, score and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values or binary decision values.
7. **Itertools** - It is a module in Python that is used to iterate over data structures that can be stepped over using a for loop. Such data structures are also known as iterables. This module incorporates functions that utilize computational resources efficiently. using this module also tends to enhance the readability and maintainability of the code.

1. **Result and Discussion**

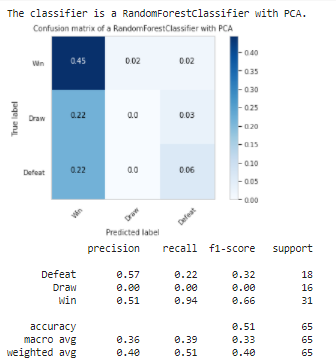
Here we see the accuracy of all of the three classifiers:

****

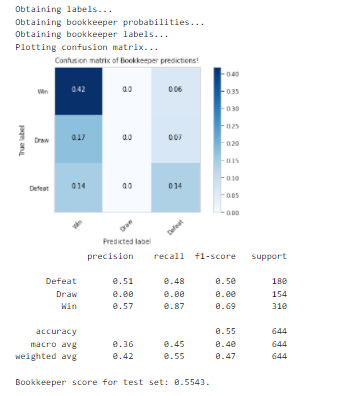
We observe that the accuracy is almost the same for each classifier. But, Gaussian Naive Bayes shows better accuracy by a small margin.

Below, we generate the Confusion matrices for each classifier and the bookkeepers data as well.

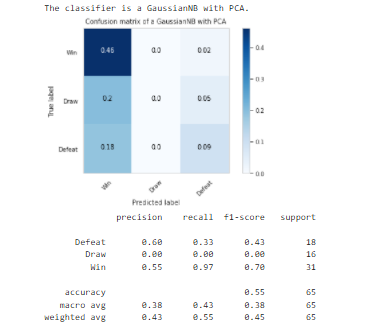
For the first case, we take the Random Forest classifier. We observe that the win prediction is good. but the loss and draw predictions are not that satisfying.

****

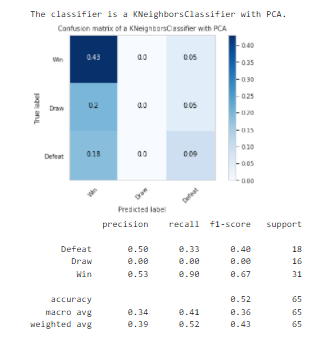
That is because classifiers are trying to imitate the bookkeeper's predictions. The confusion matrix of bookkeepers too shows low accuracy for loss and draw prediction.

****

Now we take a look at GaussianNB, which shows similar results, i.e, good results for win prediction but less for loss and draw prediction.

****

At last we come to KNeighbour Classifier, which has similar results as others.

****

In the end, we have replicated the prediction model classifier used by the bookkeepers. If the model is provided with better data for training, we may be able to get better results for loss and draw predictions.

1. **CONCLUSION**

In summary, the results of our project reflect a solid first effort in prediction match outcomes in football using machine learning techniques. By comparing multiple classification algorithms in combination with dimensionality reduction, it was discovered that a Gaussian Naives Bayes regression model in combination with a principal component analysis has the best performance. Combining the model selection the robustness and replicability of results is ensured. We firmly believe that this model selection and tuning process can easily be applied to all sorts of classification problems. Another important aspect of the project is the feature creation process since the variety of features that could be created based on the underlying raw data is extremely high. Since existing approaches to football predictions are considered, We created a variety of features that resulted in final model performance superior to bookkeeper prediction accuracy. Creating these features, choosing what aspects of a football match to take was certainly the most challenging part of this project. It remains that football matches are affected by countless factors as well as a high degree of randomness, which makes it impossible to achieve perfect prediction accuracy in my opinion. However, We believe that given more data sources, it is certainly possible to further increase the performance of the model.

1. **REFERENCE**

[1]<https://www.digitalocean.com/community/tutorials/how-to-use-the-sqlite3-module-in-python-3>

[2]<https://www.journaldev.com/29055/python-pandas-module-tutorial>

[3]<https://numpy.org/doc/stable/user/absolute_beginners.html#:~:text=NumPy%20can%20be%20used%20to,on%20these%20arrays%20and%20matrices>.

[4]<https://seaborn.pydata.org/introduction.html>

[5]<https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/#:~:text=Matplotlib%20is%20a%20cross%2Dplatform,embed%20plots%20in%20GUI%20applications>.

[6]<https://medium.com/mlpoint/matplotlib-for-machine-learning-6b5fcd4fbbc7>

[7]<https://www.tutorialspoint.com/scikit_learn/index.htm#:~:text=Scikit%2Dlearn%20(Sklearn)%20is,a%20consistence%20interface%20in%20Python>.

[8]<https://www.educative.io/edpresso/what-are-itertools-in-python>