Project Title:

La Liga Match Result Predictions

Module Title: Artificial Intelligence and Machine Learning
Module Code: CSMAI21
Lecturer responsible: Dr Yevgeniya Kovalchuk
Student Name: Jasper, Ching Yat Cheng
Student ID: 30801690
Word count: 5023
Date: 11 March 2022
Hours spent for this assignment: 40 hours

Abstract

The Football match result prediction causes many assets involved in the betting procedure; therefore, it is an expanding research area that is not less than cryptocurrency and the stock market. Commonly, The statistical models proved by mathematics experts have been used for predicting the match results. However, the method takes issue with complex analysis since the rising volume of football-related information. For this reason, this research study examined the Spanish La Liga football match statistics and betting odds data from six different betting companies within three seasons from 2019-2020 to 2021-2022. The probabilities of the full-time result (Home Win, Draw, Away Win) of the match were estimated according to three machine learning algorithms, including Support Vector Classification (SVC), Random Forest Classifier (RF) and Multi-Layer Perceptron Neural Network Models (MLP). 10-fold cross-validation had been applied to verify the accuracy of the prediction model. By comparing the datasets and models, The SVC models with data containing betting odds achieved the highest accuracy and f1 score, which are 60.09% and 58.18%.The proposed prediction result is hoped to contribute as a benchmark for the research studies of the prediction of football match results.

1. Background

Football (called soccer in the US) is one of the most popular and well-known sports globally. Hence, football match prediction has caught many people's interests, such as betting crowds and companies, researchers, sports teams' management, etc. However, plenty of factors like weather, location, teams' lineup, yellow or red cards, home advantage or any other factor could be affected the result of the football match. Therefore, the probability of the result will keep changing within 90 minutes of a football match.

Traditionally, the experts had been predicted the results by different statistical models. One of the most popular statistical models, Markov Chain Monte Carlo [1], attempts to evaluate the game's results by analyzing the strength difference between players, the psychological impact of under-estimate the opponent, calculating the attack intensity, Etc. This mathematical model can reflect a certain extent of the match information. Nevertheless, with the increasing number of football-related information referenced, statistical models are hard to handle a huge amount of data with too many other factors.

Due to the limitation of the statistical models, this research attempted to estimate the abilities of three different machine learning models, SVC, RF and MLP, for predicting football match results which shows a better performance for predicting football matches [2]. The dataset for training the models was acquired by https://www.football-data.co.uk/spainm.php.

A literature review of the previous related work in Section 2. The detail of the dataset description in Section 3. The experiments, exploration data analysis and model evaluation are in Section 4. The comparison of different model and data performances in Section 5.

2. Literature Review

Razali et al. [3] proposed a Bayesian Networks (BN) model for forecasting the match result. The open-source software WEKA founded by General Public License (GNU) had been applied for this research study, which can separately consider different factors in each match. Match results and statistical data had been selected as a dataset for training the model. The method that the study measured the function of the model was K-fold cross-validation. This research got a 75.09% average accuracy for predicting the match. However, the dataset in the research contained the final goal of each team inside, and it must have a high correlation which almost represented the match result. Therefore, the outcome may not have enough representation.

Samba [4] combined 12 seasons of leagues data and extracted 21 features such as a referee, odds, division, and statistical information. The 12 multilayer perceptron had been developed for training the model. The data split into 60% of the train set, 20% validation set and 20% test set. The study results show 54% accuracy of Premier League, 44% accuracy of Championship, 44% accuracy of League 1 and 43% accuracy of League 2. The study shows sufficient representation of the validation, but the accuracy rates are relatively low than another research. The researcher could also compare other models to evaluate.

Azhari et al. [5] used a statistical model for predicting the match outcomes, which applied Poisson regression models composed the estimation of the home team's advantage, the opposing team's defensive ability, the team's attacking ability and the average number of goals scored in a game. The result of the study shows the probabilities of the winning teams. Also, the researchers' estimated the final ranking of just one season. The overall accuracy reached 61%. The dataset did not show clearly in this report and the amount of data was relatively insufficient than another research. However, the research shows a new perspective for data scientist to increase their machine learning model.

3. Dataset description

There are three seasons of La Liga from 2019-2020 to 2021-2022 had been extracted from the website Football-Data.co.uk. The 2019-2020, 2020-2021, and 2021-2022 files contain 380, 380 and 258 matches. Each match originally had 105 columns of data such as match results and statistics, result odds (Home Win, Draw, Away Team), total goals odds, Asian handicap odds and closing odds provided by six different companies. Except for result odds, the odds data were not related to predicting the result by observation. In addition, since the matches were independent of each other rather than a time series relationship, the date, time, and the unrelated betting odds had been dropped first. The shape of the initial dataset is 1015 matches and 38 columns of data after combining all dataset.

3.1 The Context and Content of three datasets

This dataset contains results data, match Statistics, odds data of result from six betting companies across three season.

(H=Home Win, D=Draw, A=Away Win)
HomeTeam – Name of Home Team
AwayTeam = Name of Away Team
FTHG = Number of Full Time Home Goals
FTAG = Number of Full Time Away Goals
FTR = Full Time Result - (H, D, A)
HTHG = Number of Half Time Home Goals
HTAG = Number of Half Time Away Goals
HTR = Half Time Result - (H, D, A)
HS = Number of Home Team Shots
AS = Number of Away Team Shots
HST = Number of Home Team Shots on Target
AST = Number of Away Team Shots on Target
HC = Number of Home Team Corners
AC = Number of Away Team Corners
HF = Number of Home Team Fouls Committed
AF = Number of Away Team Fouls Committed
HO = Number of Home Team Offsides
AO = Number of Away Team Offsides
HY = Number of Home Team Yellow Cards
AY = Number of Away Team Yellow Cards
HR = Number of Home Team Red Cards
AR = Number of Away Team Red Cards
B365H = Bet365 home win odds
B365D = Bet365 draw odds
B365A = Bet365 away win odds
BWH = Bet&Win home win odds
BWD = Bet&Win draw odds
BWA = Bet&Win away win odds
IWH = Interwetten¬ home win odds
IWD = Interwetten draw odds
IWA = Interwetten away win odds
PSH = Pinnacle home win odds
PSD = Pinnacle draw odds
PSA = Pinnacle away win odds
VCH = VC Bet home win odds
VCD = VC Bet draw odds
VCA = VC Bet away win odds
WHH = William Hill home win odds
WHD = William Hill draw odds
WHA = William Hill away win odds

4. Machine learning model

Every Machine learning model and technique owned different advantages and weaknesses. A detailed understanding of the field of the problem and other models themselves is required to choose the most appropriate method to make the prediction. The book [7] shows an in-depth introduction to different machine learning techniques. After the basic understanding of the models, the machine learning methods used in this analysis were:

Support Vector Machines (SVC) – is a non-probabilistic binary linear classifier, which accepts an input dataset and forecasting result. it is also a quick and memory-efficient process algorithm that can generate useful results for many learning problems [7]. It offers a method that is suitable for both classification and regression.

Random Forest (RF) – is a classification algorithm composed of many decision trees. The feature randomness and bagging are used in the algorithm to build up an unrelated forest of decision trees. The result of this algorithm assumes to be more accurate than any individual tree.

Multi-Layer Perceptron Neural Network Models (MLP) – is a branch of the feedforward artificial neural networks (ANN). MLP is composed of a minimum of three different layers such as input, hidden, and output layer. every node inside the neural networks except the input node is a neuron that applied a non-linear function. The backpropagation, a supervised learning method had been applied to the MLP for training the data. The linear perceptron can be distinguishing non-linear activation and its multiple layers.

4.1 Summary of the approach

The study preprocesses previously selected datasets to remove extraneous data. Afterward, the data will be divided into input variables (X) and output variables (y). In this study, y is the full-time result of the match. Then, three models were trained to predict the game results. Finally, the prediction accuracy of the three models in two datasets was compared.

4.1.1 Data pre-processing approach

In the first part of the experiment, exploratory data analysis was used for observing the different relationships of each column of the data. By exploring the data, it can be gaining more understanding about the correlation of each data, how many unrelated, redundant, and too much influence data. Different visualization graphs had been plotted to shows the linkage between the data.

4.1.2 Model validation approach

Even though the paper [3] shows the incomplete representation of the result, the evaluation method, k-fold cross-validation is perhaps one of the reasons that can generate a higher mark than the average score in other reports. Therefore, the experiment applied the same technique for evaluating the prediction models. The K fold cross validation performed the k times fitting procedure to training the data. The validation method split the training set and validating set as random each time.

4.1.3 Neural Network Models building approach

In the 1996 Summer Olympics, Condon et [8] used the MLP to estimate the scores of different countries, and the results were better than the regression model. Rothstein et al. [9] predicted the football matches in Finland by fuzzy model with neural networks. Silva et al. [10] applied non-linear models constructed by MLP, and the predicted result and the actual result only had a minor difference. Referring to the analysis and experience of previous studies, MLP models will be applied as the part of neural network models in this experiment.

```
Model: "sequential_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_100 (Dense)            (None, 64)                1152
dense_101 (Dense)            (None, 32)                2080
dropout_60 (Dropout)         (None, 32)                0
dense_102 (Dense)            (None, 16)                528
dropout_61 (Dropout)         (None, 16)                0
dense_103 (Dense)            (None, 8)                 136
dropout_62 (Dropout)         (None, 8)                 0
dense_104 (Dense)            (None, 3)                 27
=================================================================
Total params: 3,923
Trainable params: 3,923
Non-trainable params: 0
_________________________________________________________________
```

Figure 26 The summary of the neural network models

4.2 Data visualisation, pre-processing, feature selection

The data was visualized step by step in this section to find out the correlation between the data, and the performance of each team in different areas. The research attempts to chart the data's results and interpret the meaning under the images. Furthermore, the experiment will select the most appropriate data for training the model.

4.2.1 Exploration data analysis (EDA)

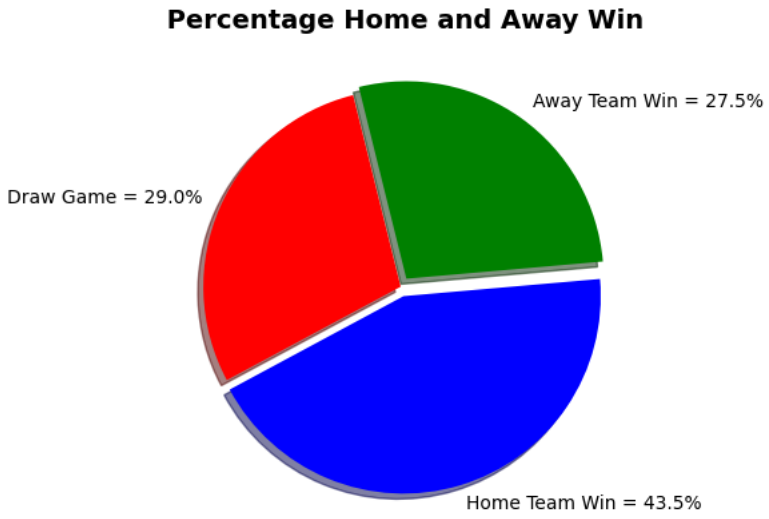

Figure 1 The Percentage of Home and Away Team Win

Figure 1 shows that a home team has a high win rate of 43.5%, while a draw and away team has a 29% and 27.5% chance of winning respectively. In the La Liga league, every team has an equal opportunity to be the home or away team. Because of this, the familiarity of the venue and the weather, the impact of traffic, and the audience's support on the court can affect the performance of the players and have a particular effect on the results of the game.

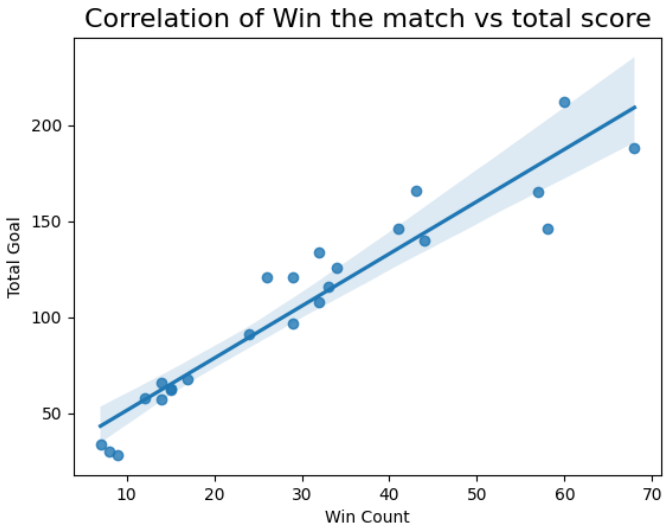

Figure 2 The Correlation between the result of win and total goal

The strong correlation between the number of wins and goal scores can be observed in Figure 2. Therefore, the next step in the study will explore the relationship between score and average corners/shots on each team.

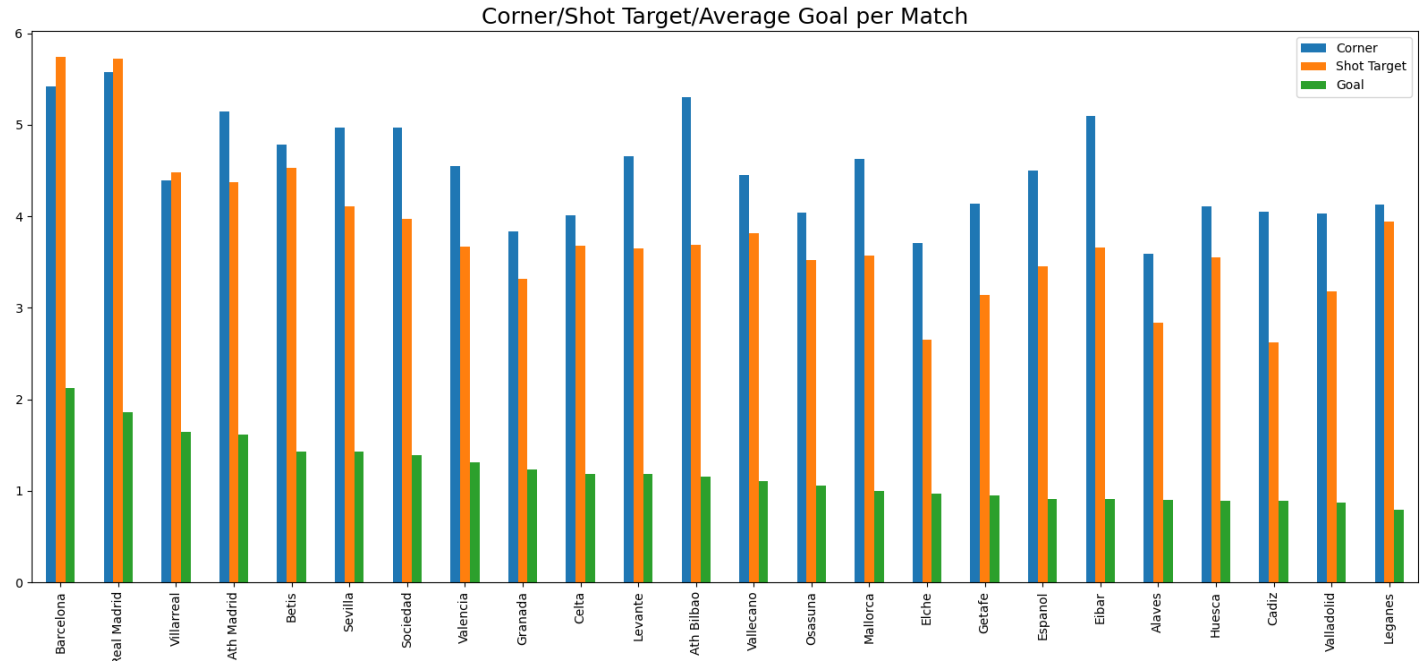

Figure 3 The relationship between goals, average Shot on target and corner

The Figure 3 shows that Barcelona has the highest goal-scoring average, and the team has a high standard of corners and shots on target per game. However, the average corner relationship is less than the average shot on target relationship. For example, Ath Bilbao, Eibar and Mallorca all have high average corner kicks, but average goals scored in the middle and lower levels. The average shot on target has a certain degree of correlation with the goal.

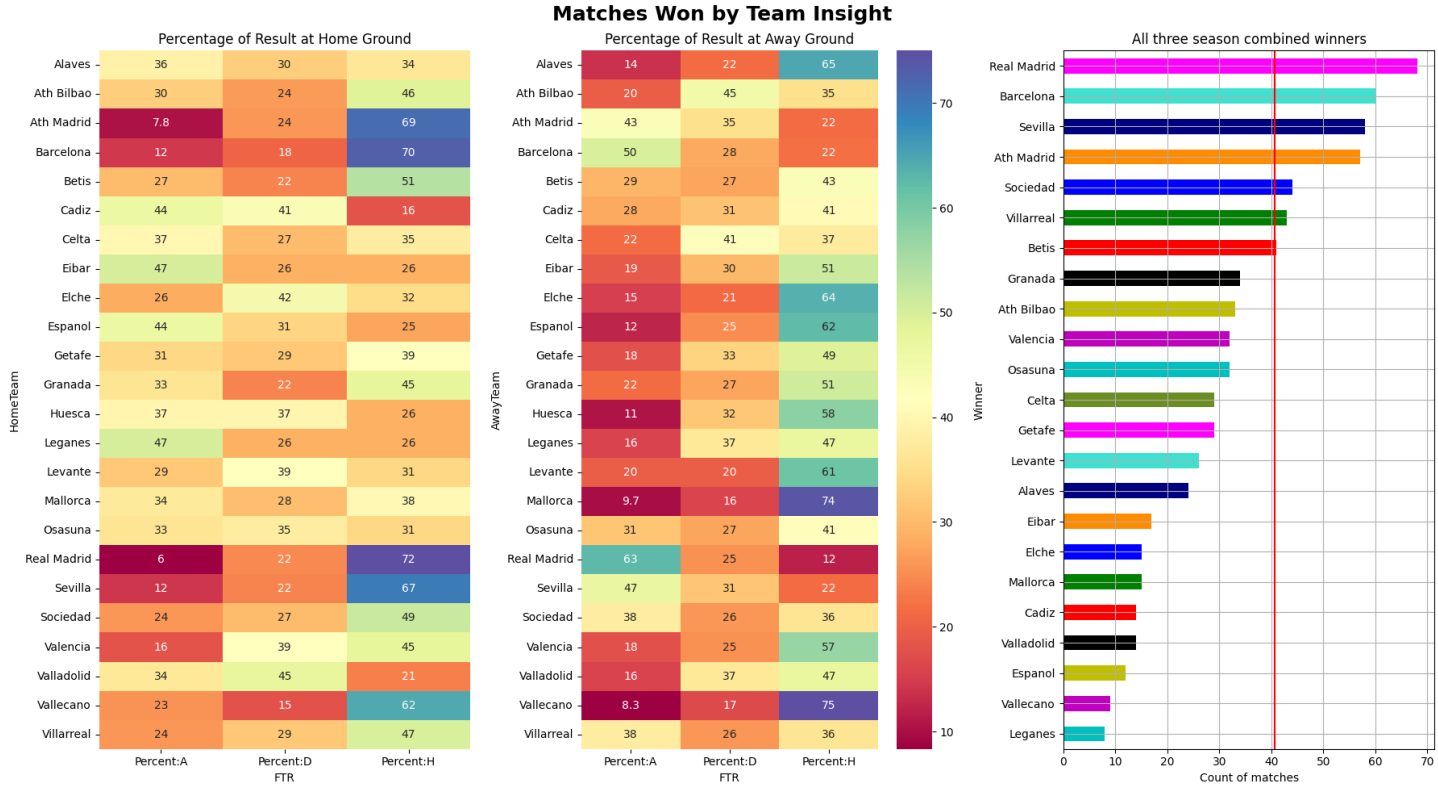

Figure 4 Percentage of winning the game by each team/ the most winning team

The heat map on the right of figure 5 shows the percentage of results at the home ground. Real Madrid, Barcelona, Ath Madrid, Sevilla and Vallecano shows the highest percentage of winning the game respectively in order, which is all over 60%. Cadiz, Valladolid, Espanol, Eibar, Huesca and Laganes show the lowest performance, which shows the winning rate under 30%. The team that happened the most of the drawn game on the Home ground is Valladolid.

The heat map on the right of figure 5 shows the percentage of results at the away ground. Real Madrid is the only team that got over 60% of the match in the away ground. Huesca, Espanol, Alaves, Elche, Leganes, Valladolid, Gatage, Valencia and Elbar show the lowest performance, which a winning rate under 20%. The team that happened the most of the drawn game on the away ground is Ath Bilbao.

The right side of figure 5 shows the most winning team in all three seasons. Real Madrid, Barcelona, Sevilla, Ath Madrid, Sociedad and Villarreal show the best performance within all units.

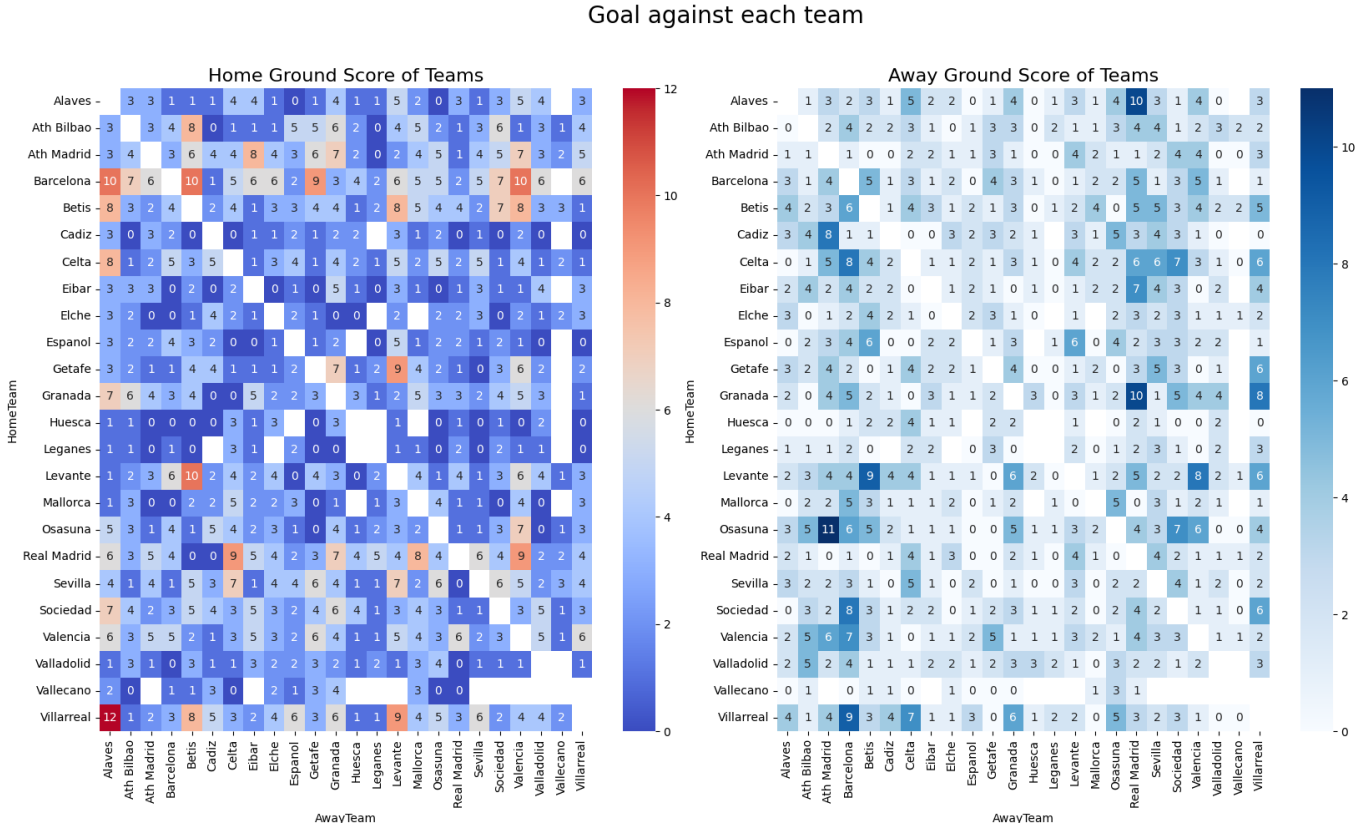

Figure 5 Team score how many goals against which team

In Figure 5, y axis teams mean the goal team on home ground, x axis teams mean the goal team on away ground. The highest number of goals (12 goals) in home ground are Villarreal vs Alaves. Barcelona vs Alaves & Betis & Sociedad also over 10 goals. the team Huesca got the most 0 goals against 8 other teams.

On the other hand, the total number of the goals in the matches of Osasuna vs Ath Madrid is 11, which is the highest number of the goals in away ground. Real Madrid got 10 goals in the match against the team of Alaves and Granada. Vallecano had a lowest number of against to other teams.

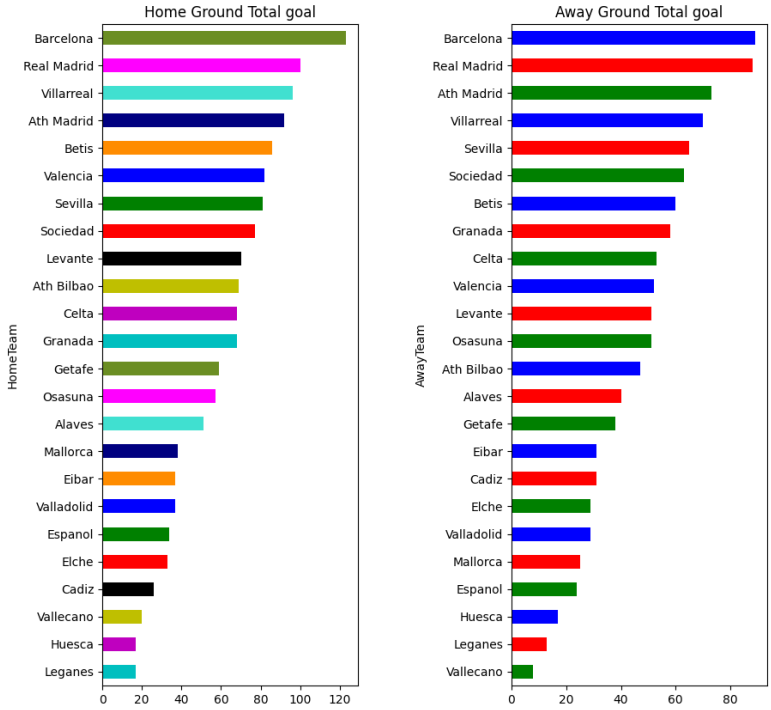

Figure 6 Goal made in Home Ground and Away Ground by team

Barcelona is the only team that can get over 120 goals in all matches on home ground, whereas the Leganes Huesca got under 20 goals. In away ground, Barcelona, Real Madrid performed the greatest number of goals over 80. On the other hand, Vallecano, Leganes and Huesca are still the lowest performance teams on the total number of goals. However, the most deficient performance team in the La Liga League will be replaced by another league's highest performance. Therefore, the figure of the total match count will be performed.

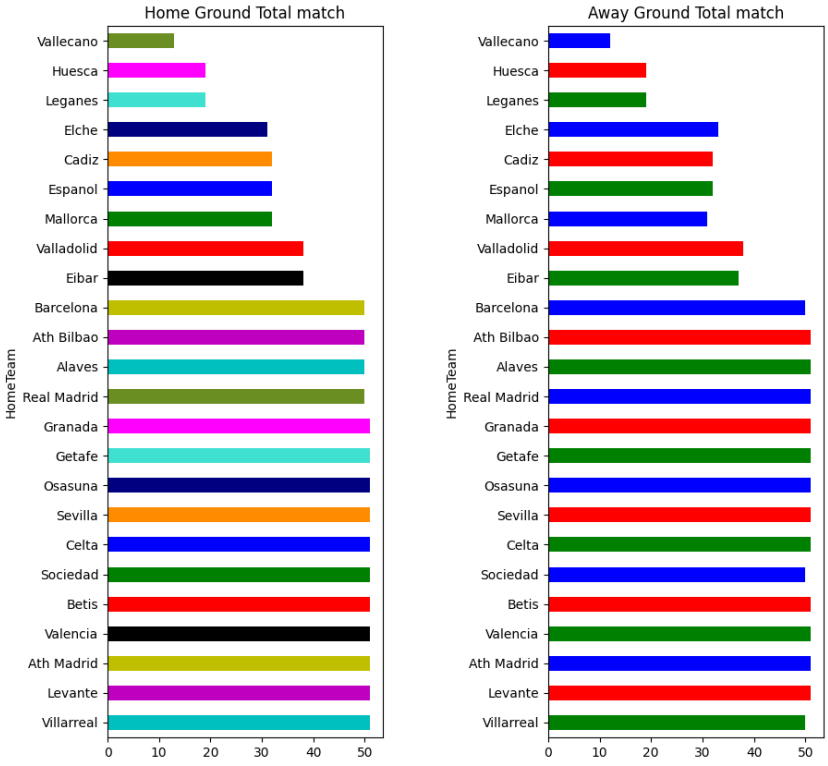

Figure 7 The matches count of each team

Figure 7 shows that Vallecano, Huesca, Leganes, Elche, Cadiz, Espanol, Mallorca, Valladolid and Eibar were not participated in all seasons. It can reflect why all those teams got the lowest matches to win in Figure 4 to 6.

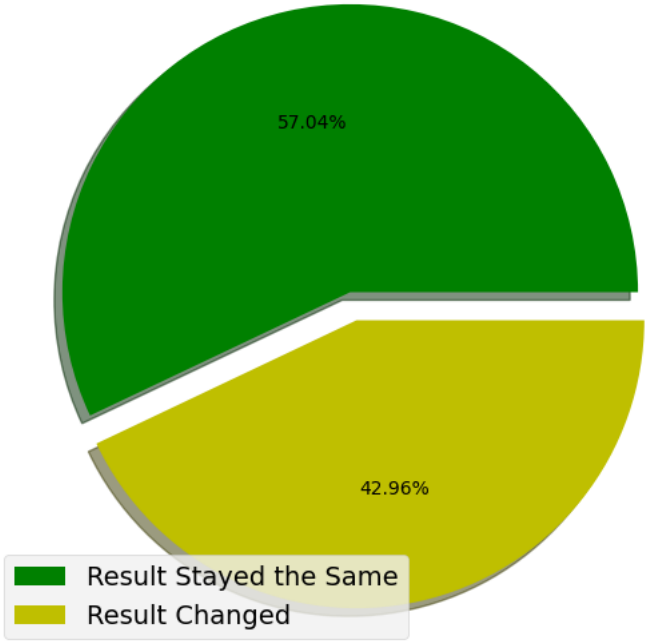

Figure 8 The percentage of changing the result after half of game

The figure 8 shows that 57.04% of the matches within three season will not change after half game of the match, whereas 42.96% of the match will be changed.

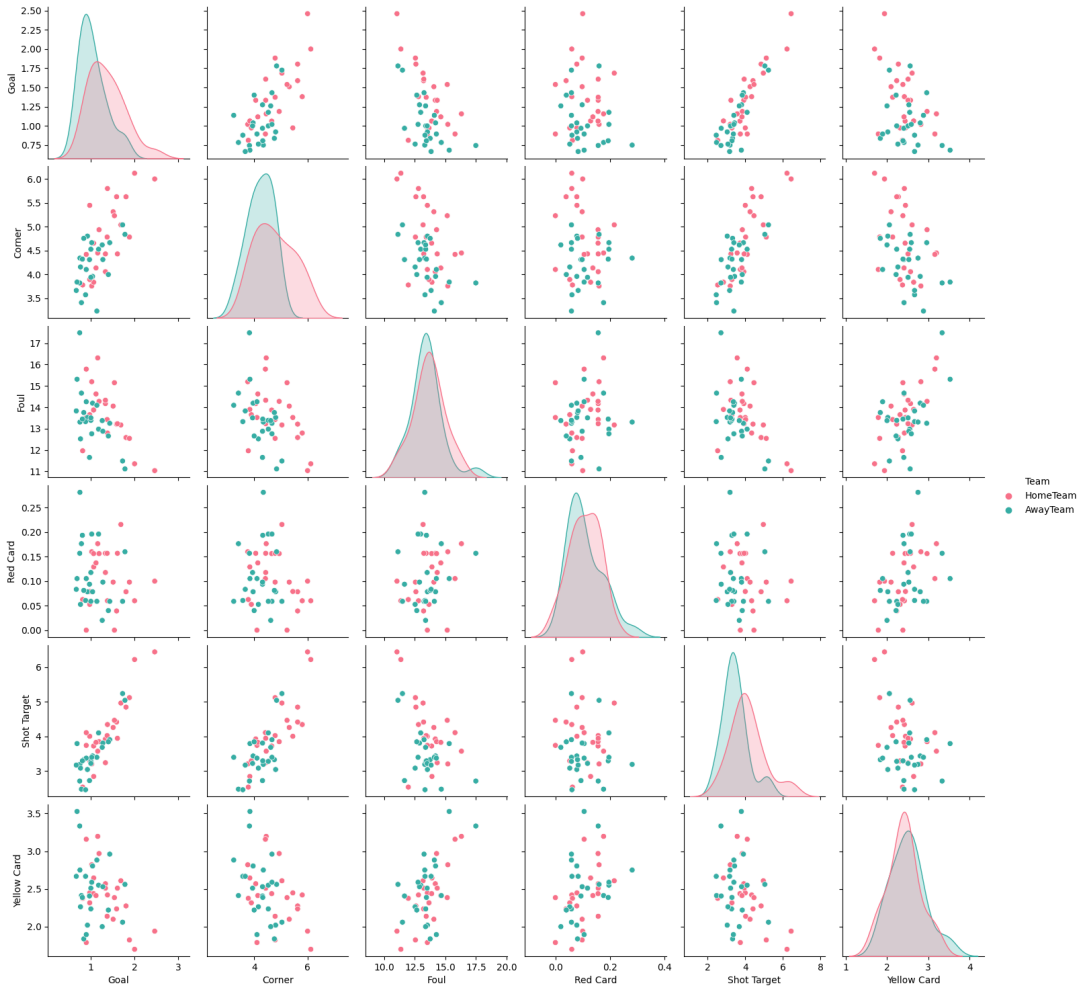

Figure 9 Correlation between different variables

The correlation between different statistical variables is shown in figure 9. The number of goals, corners, and shot on target is usually higher in the home ground. The number of goals positively correlated with corners shot on target, whereas the goals had a negative correlation with fouls red and yellow cards. The number of corners had a little negative correlation with fouls and yellow cards and a positive correlation with the shot target. The number of fouls and yellow cards also had a little negative correlation. In other relationships, there is no very clear correlation between the data.

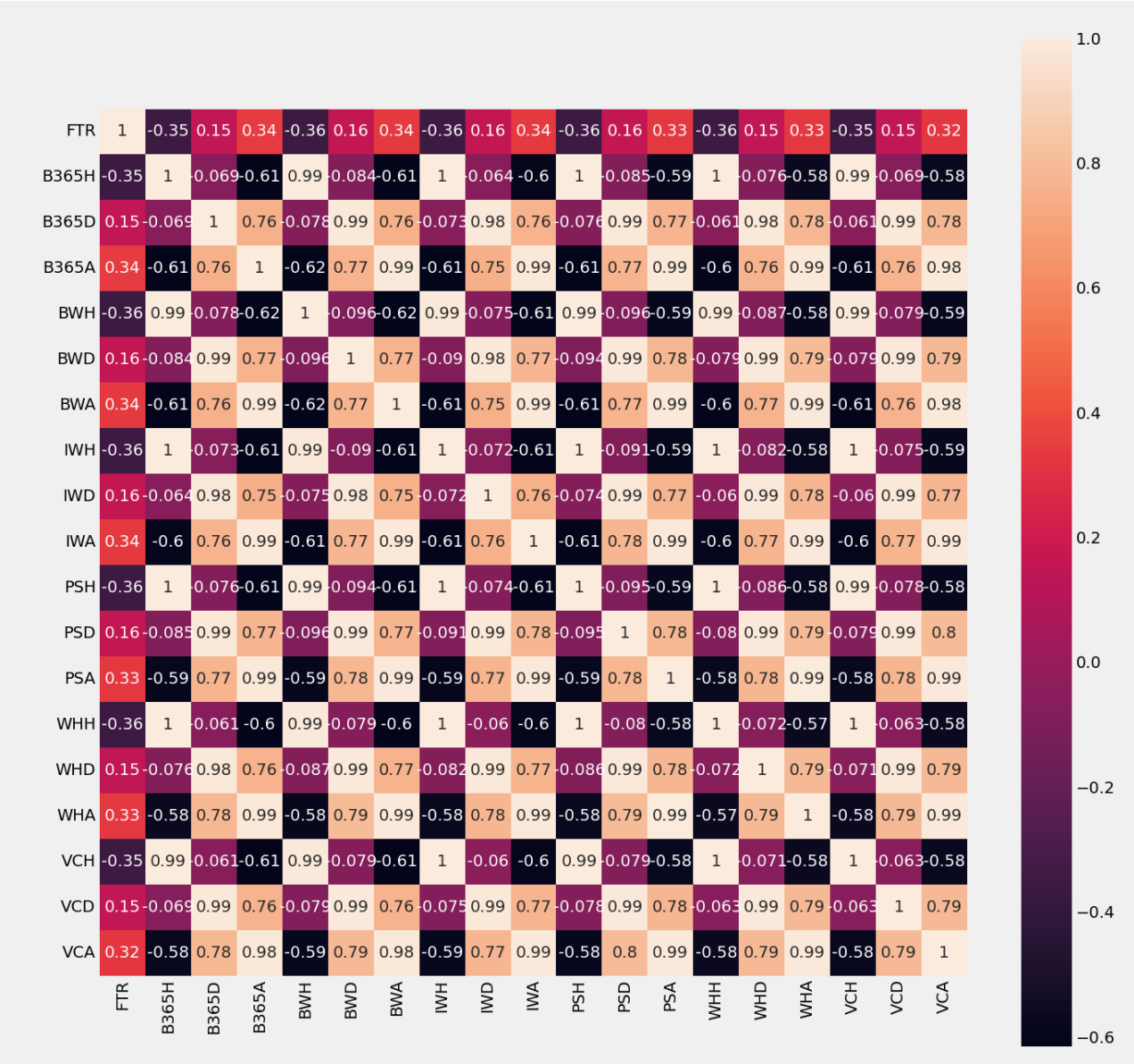

Figure 10 The relation between Betting Data and final result of the game

In this figure, all odds in home win, draw and away win provided an extremely high correlation, which means the companies sharing a similar perspective on all matches. For result of the match with betting odds, it shows a negative relation (-0.35 to -0.36)with home win odds and positive relation with away win (0.32 to 0.34) and draw (0.15 to 0.16). Therefore, the betting company odds have a certain reference value.

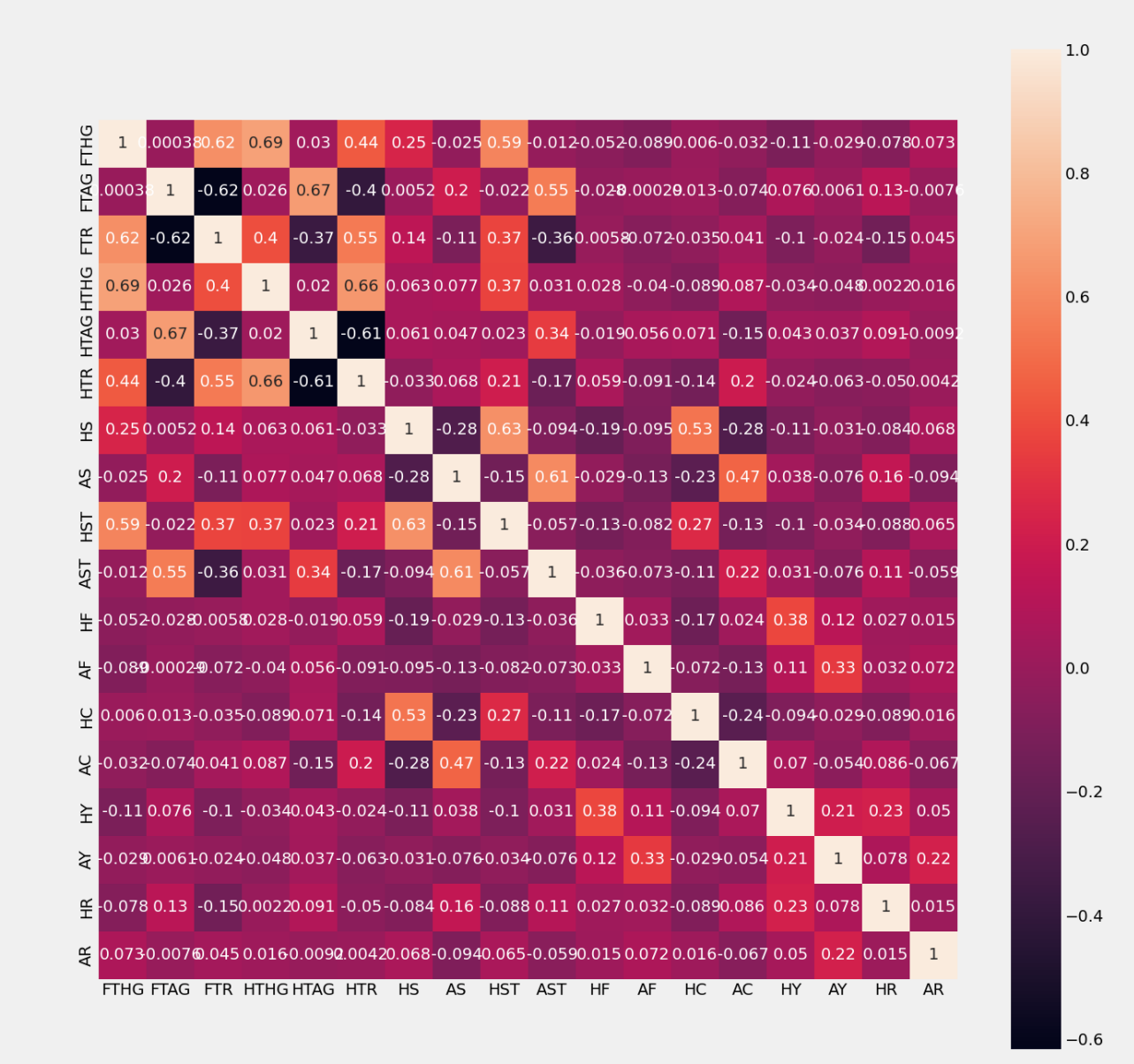

Figure 11 The relation between Betting Data and final result of the game

The main reason of figure 11 is attempt to found out the most influence columns on the result. The heat map shows that the Full game goals of home and away shows the highest correlation with the match result. In this figure, the most highest influence of the match result shows on result related data.

4.2.2 Pre-processing

After the exploration data analysis, the betting data proved that it had a certain relationship with the match result, but the odds data from different companies shows a extremely high correlation. Therefore, the mean of 6 companies' odds on the home win, draw, away win had been calculated to replace the original odds data, which mean the columns of odds data will be reduced from 18 columns to 3 columns. On the other hand, figure 11 shows that there are high relationship between match result and other result related data.

After the data selection, the betting odds data was reduced from 18 columns to 3 columns, and results in data had been cleared except the full-time result. The shape of the final dataset for training the models is 1015 matches and 18 columns of data after combining all datasets.

HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR
Ath Bilbao	Barcelona	1	0	H	0	0	D
Celta	Real Madrid	1	3	A	0	1	A
Valencia	Sociedad	1	1	D	0	0	D
Mallorca	Eibar	2	1	H	1	0	H
Leganes	Osasuna	0	1	A	0	0	D
...	...	...	...	..	...	...	..
Villarreal	Espanol	5	1	H	3	0	H
Sevilla	Betis	2	1	H	2	0	H
Sociedad	Osasuna	1	0	H	0	0	D
Barcelona	Ath Bilbao	4	0	H	1	0	H
Granada	Cadiz	0	0	D	0	0	D

Figure 12 Python snippets of initial data frame

Home team, away team, FTR, HTR columns is not numerical data, it cannot fit into the prediction models. Therefore, LabelEncoder function from Sklearn pre-processing library applied in these columns.

4.2.3 Feature selection

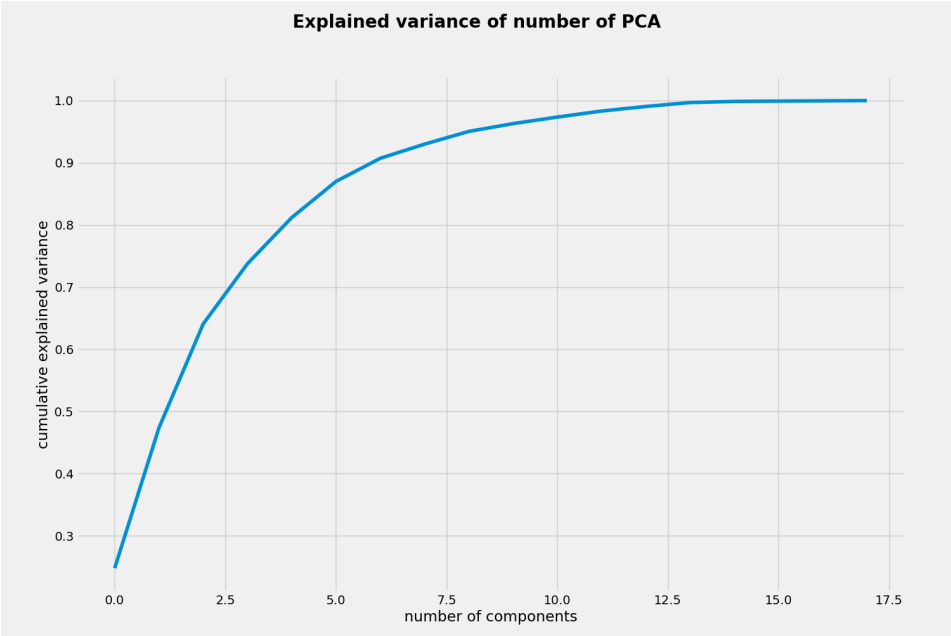

Figure 13 Explained variance of number of PCA

PCA 1	0.2479	**PCA 6**	0.8698	**PCA 11**	0.9734	**PCA 16**	0.9991
PCA 2	0.4735	**PCA 7**	0.9070	**PCA 12**	0.9831	**PCA 17**	0.9995
PCA 3	0.6407	**PCA 8**	0.9298	**PCA 13**	0.9904	**PCA 18**	
PCA 4	0.7375	**PCA 9**	0.9503	**PCA 14**	0.9968	**PCA 19**	
PCA 5	0.8117	**PCA 10**	0.9628	**PCA 15**	0.9986	**PCA 20**	

Table 1 Explained variance of number of PCA

The feature selection methods assume to reduce the number of input columns (variables) and keeping the most valuable data for the model to predict the target. Principal Component Analysis (PCA) is one of the best methods to reduce the number of variables. In this experiment, the size of the initial dataset was decreased in the pre-processing procedure, and input columns are not a considerable data size. Therefore, the data after pre-processing had been used in this case. The input data were numerical and the output data categorical.

4.3 Model training and evaluation

Model Creation:

```
# Instantiate the machine learning classifiers
svc_model = SVC(coef0=5, kernel='poly')
rfc_model = RandomForestClassifier()
kfolds = KFold(n_splits=10)
```

Figure 14 Python snippets of model creation

For the SVM, Support Vector Classification (SVC) had been created with coef0 = 5 and kernel = poly. The polynomial kernel allows a non-linear model learning. The meaning of the kernel is representing the similarity of train samples in the feature area on top of polynomials of original columns. For RF, Random Forest Classifier (RFC) had been created with default setting.

```
def baseline_model(number_of_features=17):
    # create model
    model = Sequential()
    model.add(Dense(64, input_dim=number_of_features, activation='relu'))
    model.add(Dense(32, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(16, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(8, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(3, activation='sigmoid'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

Figure 15 Python snippets of model creation (2)

For the Multi-Layer Perceptron Neural Network Models, there are total 8 layers add into the model. There is a input layer in the first .add() function, the input dimension was required for set up a layer and connects to the input variables. The layers 2 to 7 is a hidden layers which between the input layer and the output layer. The final layer is a output layer which can produce the output variables. The activation set up as rectified linear activation function (relu) which is a piecewise linear function that can directly to output the data input. activation in the output layer is sigmoid, it can guarantee the output will always stay in the range between 0 and 1.

Model Training:

```
def models_evaluation(X, y):
    X = X
    y = y
    estimator = keras_estimator(X, y)
    # Perform cross-validation to each machine learning classifier
    svc_val = cross_validate(svc_model, X, y, cv=kfolds, scoring=scoring)
    rfc_val = cross_validate(rfc_model, X, y, cv=kfolds, scoring=scoring)
    mlp_val = mlp_function(X, y)
```

Figure 16 Python snippets of model training

```
def mlp_function(X, y):
    # evaluate the keras model
    estimator = keras_estimator(X, y)
    results = cross_validate(estimator, X, y, cv=kfolds, scoring=scoring)
    return results
```

Figure 16 Python snippets of model training (2)

For training the models, k-fold cross-validation had been applied which the k = 10. X is the dataset feature which contain the statistical data and the mean of betting odds data. y is the target dataset (Match Result).

Model evaluation:

```
# Define dictionary with performance metrics
scoring = {'accuracy': make_scorer(accuracy_score),
           'precision': make_scorer(precision_score, average='macro'),
           'recall': make_scorer(recall_score, average='macro'),
           'f1_score': make_scorer(f1_score, average='macro')}
```

Figure 17 Python snippets of model scoring

```
svc_predict = cross_val_predict(svc_model, X, y, cv=kfolds)
rfc_predict = cross_val_predict(rfc_model, X, y, cv=kfolds)
mlp_predict = cross_val_predict(estimator, X, y, cv=kfolds)
print("SVC prediction confusion matrix")
print(confusion_matrix(y, svc_predict))
plot_cm(confusion_matrix(y, svc_predict), 'Reds')
plt.show()
print("RFC prediction confusion matrix")
print(confusion_matrix(y, rfc_predict))
plot_cm(confusion_matrix(y, rfc_predict), 'Blues')
plt.show()
print("MLP prediction confusion matrix")
print(confusion_matrix(y, mlp_predict))
plot_cm(confusion_matrix(y, mlp_predict), 'Greens')
plt.show()

print("SVC prediction classification report")
print(classification_report(list(y), svc_predict))
print("RF prediction classification report")
print(classification_report(list(y), rfc_predict))
print("MLP prediction classification report")
print(classification_report(list(y), mlp_predict))
```

Figure 18 Python snippets of model evaluation (CONFUSION MATRIX AND REPORT)

After the training process, the result had been set up as 4 different scores such as accuracy, precision, recall and f1 score. For generating the confusion matrix and classification report, the prediction of y must be exist. Therefore, cross_val_predict() function will be applied to training the model again. Afterward, the confusion matrix and classification reports will be built for each models.

4.3 Results and discussion

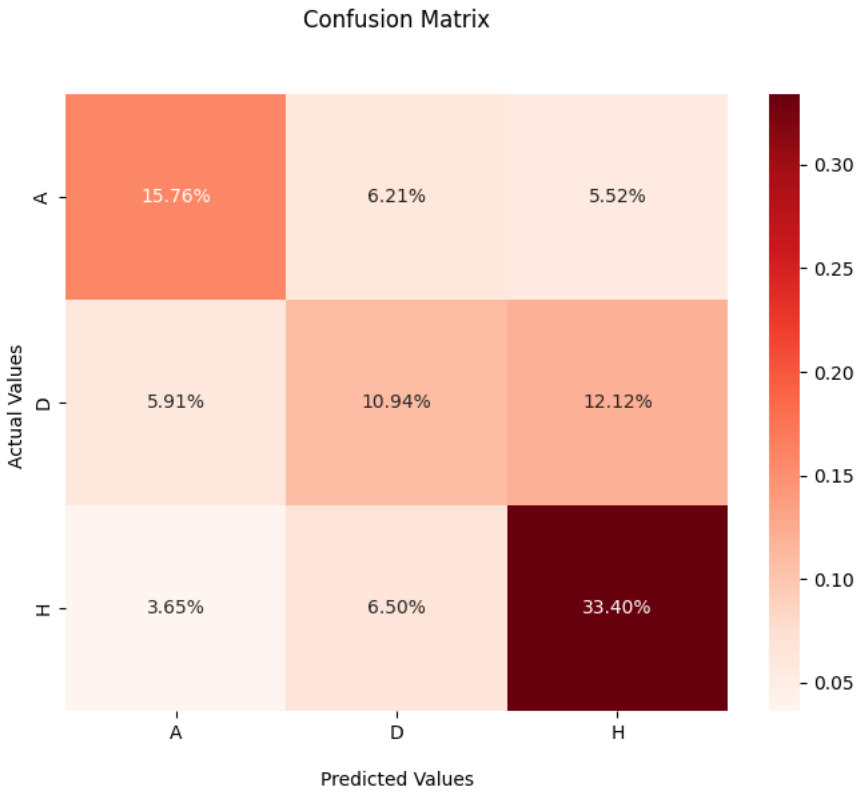

Figure 18 confusion matrix of SVM

SVC prediction classification report

	precision	recall	f1-score	support
A	0.62	0.57	0.60	279
D	0.46	0.38	0.42	294
H	0.65	0.77	0.71	442
accuracy			0.60	1015
macro avg	0.58	0.57	0.57	1015
weighted avg	0.59	0.60	0.59	1015

Figure 19 Classification Report of SVM

There are a total of 25.32% of data classified into away team win (A), 23.65% of data classified into Draw (D), and 51.04% of data classified into home team win (H). The true positive of away team win is 15.76%, the draw is 10.94%, and home team win is 33.40%. The precision of A is 62%, D is 46% and H is 65%. The false negative (FN) of A is 11.73%, D is 18.03%, and H is 10.15%. Therefore, the recall percentage in A is 57%, D is 38%, and H is 77%. The f1 score of A is 60%, D is 42%, H is 71%. The weighted average precision is 59%, recall is 60%, f1 score is 59%, and the accuracy of the model is 60%. In this result, both precision, recall and f1-score percentages in Draw shows significantly lower than other class.

From the confusion matrix, the predicted values tends to classified more data to H, and the most error of the prediction is putting the D result to H, which is 12.12%. Half of the False positive (FP) of D equally placed into H and A.

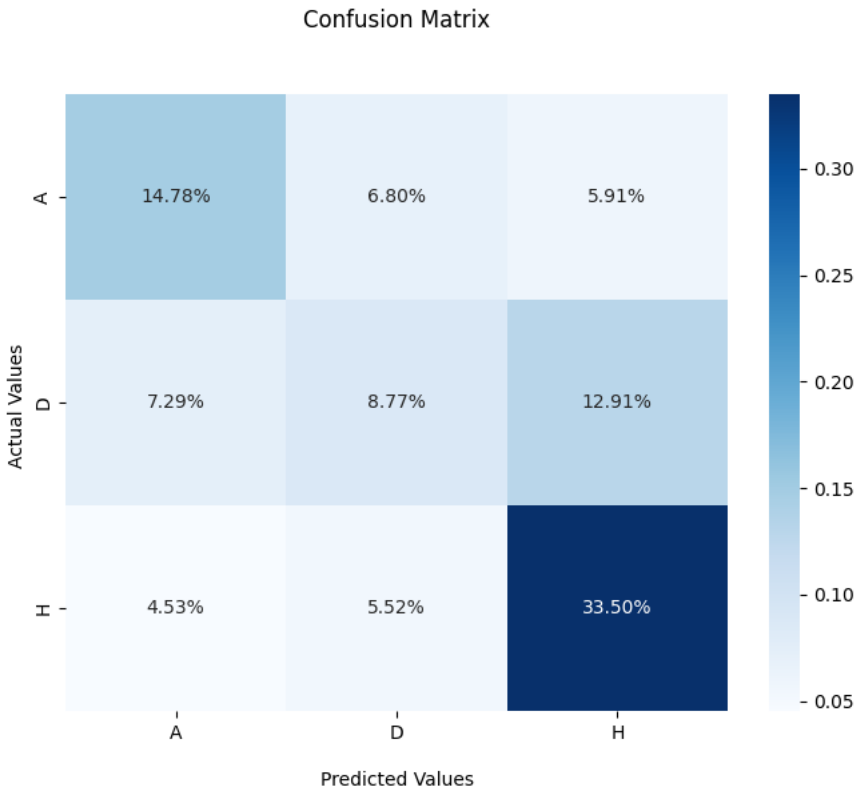

Figure 20 confusion matrix of RF

RF prediction classification report

	precision	recall	f1-score	support
A	0.56	0.54	0.55	279
D	0.42	0.30	0.35	294
H	0.64	0.77	0.70	442
accuracy			0.57	1015
macro avg	0.54	0.54	0.53	1015
weighted avg	0.55	0.57	0.56	1015

Figure 21 Classification Report of RF

There are a total of 26.6% of data classified into away team win (A), 15.76% of data classified into Draw (D), and 52.31% of data classified into home team win (H). The true positive of away team win is 14.78%, the draw is 8.77%, and home team win is 33.50%. The precision of A is 56%, D is 42% and H is 64%. The false negative (FN) of A is 12.71%, D is 20.2%, and H is 18.82%. Therefore, the recall percentage in A is 54%, D is 30%, and H is 70%. The f1 score of A is 55%, D is 35%, H is 70%. The weighted average precision is 55%, recall is 57%, f1 score is 56%, and the accuracy of the model is 57%. In this result, both precision, recall and f1-score percentages in Draw also shows significantly lower than other class.

From the confusion matrix, the predicted values tends to classified more data to H, and the most error of the prediction is putting the D result to H, which is 12.91%. Half of the False positive (FP) of D placed in A more than H, whereas A placed to D more than H.

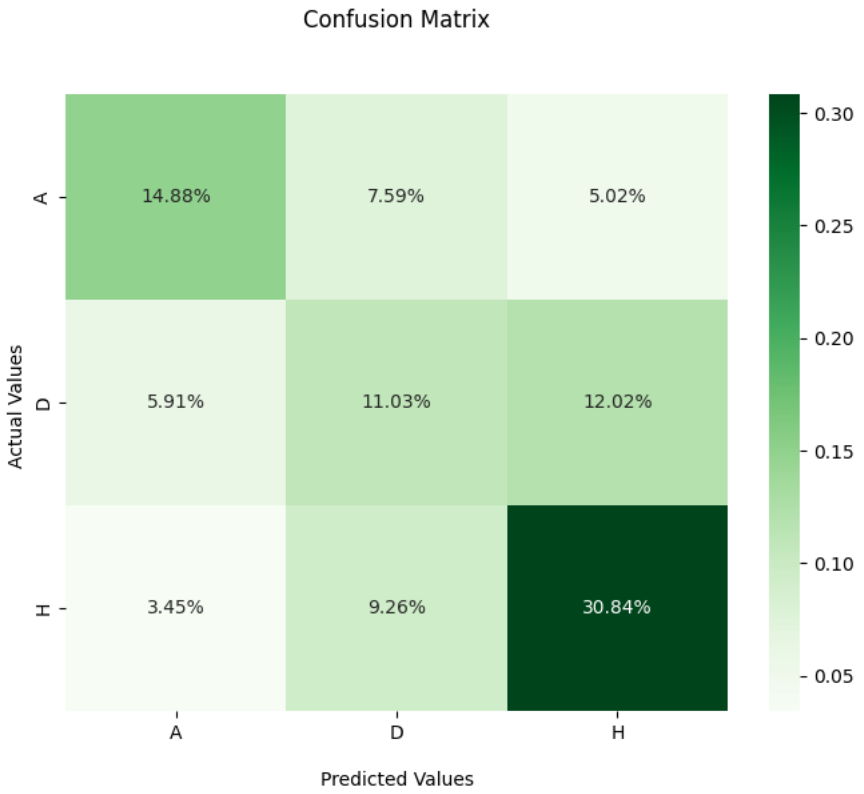

Figure 22 confusion matrix of MLP

MLP prediction classification report

	precision	recall	f1-score	support
A	0.61	0.54	0.58	279
D	0.40	0.38	0.39	294
H	0.64	0.71	0.67	442
accuracy			0.57	1015
macro avg	0.55	0.54	0.55	1015
weighted avg	0.56	0.57	0.56	1015

Figure 23 Classification Report of MLP

There are a total of 24.19% of data classified into away team win (A), 27.94% of data classified into Draw (D), and 47.88% of data classified into home team win (H). The true positive of away team win is 14.88%, the draw is 11.05%, and home team win is 30.84%. The precision of A is 61%, D is 40% and H is 64%. The false negative (FN) of A is 12.61%, D is 17.93%, and H is 12.73%. Therefore, the recall percentage in A is 54%, D is 38%, and H is 71%. The f1 score of A is 58%, D is 39%, H is 67%. The weighted average precision is 56%, recall is 57%, f1 score is 56%, and the accuracy of the model is 57%. In this result, both precision, recall and f1-score percentages in Draw also shows significantly lower than other class.

From the confusion matrix, the predicted values tends to classified more data to H, and the most error of the prediction is putting the D result to H, which is 12.91%. Half of the False positive (FP) of D placed in H more than A, whereas A placed to D more than H.

5. Results comparison across the models built

```
def all_model_score_table(svc, rfc, mlp):
    # Create a data frame with the models perfoamnce metrics scores
    models_scores_table = pd.DataFrame({f'SVC': [svc['test_accuracy'].mean(),
                                                 svc['test_precision'].mean(),
                                                 svc['test_recall'].mean(),
                                                 svc['test_f1_score'].mean()],

                                        f'RF': [rfc['test_accuracy'].mean(),
                                                rfc['test_precision'].mean(),
                                                rfc['test_recall'].mean(),
                                                rfc['test_f1_score'].mean()],

                                        f'MLC': [mlp['test_accuracy'].mean(),
                                                 mlp['test_precision'].mean(),
                                                 mlp['test_recall'].mean(),
                                                 mlp['test_f1_score'].mean()]
                                        },

                                       index=['Accuracy', 'Precision', 'Recall', 'F1 Score'])
    # Add 'Best Score' column
    models_scores_table['Best Score'] = models_scores_table.idxmax(axis=1)
    # Return models performance metrics scores data frame
    return (models_scores_table)
```

Figure 24 Python snippets of creating a combined score table

```
Summary table
                SVC        RF        MLC Best Score
Accuracy   0.600922  0.570452  0.579208        SVC
Precision  0.581848  0.537227  0.572733        SVC
Recall     0.575020  0.540235  0.563333        SVC
F1 Score   0.571731  0.530903  0.556808        SVC
```

Figure 25 Python snippets of final result (MLC = MLP)

Because the results of figures 18 – 23 are trained by the function call cross_val_predict(), and the result of figure 25 is trained by the cross_validation() function, every training will return a different result. Therefore, the comparing table will be marked as an experiment result. In figure 25, the accuracy of SVC, RFC and MLP are 60.09%, 57.05% and 57.92%; the precision of SVC, RFC and MLP are 58.18%, 53.72% and 57.27%; the recall of SVC, RFC and MLP are 57.50%, 54.02% and 56.33%; the F1 score of SVC, RFC and MLP are 57.17%, 53.09% and 55.68%. By comparing different scores in three models, Support Vector Machines shows the best performance. MLP result shows a better performance than RF.

There is a lot of contrast and similarity of the models shown in Figures 18, 20 and 22. RF model tends to classify a Draw result to Away win, and the MLP model tends to predict a Draw result to Home win. All three models tend to predict Draw result as a Home result; For predicting the Away result, all models false negative in Draw result in more than Home result.

6. Conclusion

This study refers to the model design of the literature [3] and [4] and finds out the advantages and disadvantages of the related paper. At the same time, The research refers to the book's content [7] to find out the model that the researchers think is the most suitable for predicting the outcome of the football match and conducting experiments. By analysing the experimental results of three models, including SVM, RF and MLP, it is found that SVM is the most accurate in training this dataset, with an accuracy rate and f1 score of 60.09% and 57.17%. The experimental results show that SVM is more suitable for training football data than RF and MLP.

In the process of EDA, the research found that many data have a certain correlation. For example, Shot target and the number of corners have a particular relationship with the number of goals; the odds of the gambling company will have a significant correlation with the player's result; the home field has a more significant win rate than the away game, etc.

7.Recommendations and Future work

In Figure 4-7, the study found that the lower-ranked teams did not participate in the league every season. Therefore, if future research can weigh relevant problems, the data could become more accurate. In addition, if the study can further collect weather data, a Date can be used to train the model. Furthermore, the teams and players are different every season. Suppose scholars of related research want to reduce their reliance on data in the field. In that case, they should refer to more diverse data, including sentiment analysis on social media and players' information.

In future work, the research aims to analyse and collect more data including:

- Collect the football match on each team data from the football game called FIFA.
- Keep tracking the case happed in the live.
- Collect weather condition from the government website and add it into the dataset.
- Explore the relationship and history between teams.
- Extract sentiment score from the social media platform like Twitter, Facebook or Reddit.

8. Reference

[1] M. J. Dixon and S. G. Coles, "Modelling association football scores and inefficiencies in the football betting market," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 46, no. 2, pp. 265–280, 1997.

[2] M. P. da Silva, F. Gonçalves, and L. Ramos, "Football Classification Predications," *GitHub*. [Online]. Available: https://github.com/motapinto/football-classification-predications/blob/master/src/Supervised%20Learning%20Models.ipynb. [Accessed: 10-Mar-2022].

[3] N. Razali, A. Mustapha, F. A. Yatim, and R. Ab Aziz, "Predicting football matches results using Bayesian Networks for English Premier League (EPL)," *IOP Conference Series: Materials Science and Engineering*, vol. 226, p. 012099, 2017.

[4] S. Samba, "Football result prediction by deep learning algorithms." [Online]. Available: https://www.researchgate.net/profile/Stefan-Samba/publication/334415630_Football_Result_Prediction_by_Deep_Learning_Algorithms/links/5d2834b9458515c11c273ba3/Football-Result-Prediction-by-Deep-Learning-Algorithms.pdf. [Accessed: 10-Mar-2022].

[5] H. R. Azhari, Y. Widyaningsih, and D. Lestari, "Predicting final result of football match using Poisson regression model," *Journal of Physics: Conference Series*, vol. 1108, p. 012066, 2018.

[6] T. M. Mitchell, *Machine learning*. New York: McGraw Hill, 2017.

[7] A. Boz, "Large Scale Machine Learning using NVIDIA CUDA," *CodeProject*, 09-Mar-2012. [Online]. Available: https://www.codeproject.com/Articles/336147/Large-Scale-Machine-Learning-using-NVIDIA-CUDA. [Accessed: 10-Mar-2022].

[8] E. M. Condon, B. L. Golden, and E. A. Wasil, "Predicting the success of nations at the summer olympics using neural networks," *Computers & Operations Research*, vol. 26, no. 13, pp. 1243–1265, 1999.

[9] A. P. Rotshtein, M. Posner, and A. B. Rakityanskaya, "Football predictions based on a fuzzy model with genetic and neural tuning - cybernetics and Systems Analysis," *SpringerLink*. [Online]. Available: https://link.springer.com/article/10.1007/s10559-005-0098-4. [Accessed: 11-Mar-2022].

[10] A. J. Silva, A. M. Costa, P. M. Oliveira, V. M. Reis, J. Saavedra, J. Perl, A. Rouboa, and D. A. Marinho, "The use of neural network technology to model swimming performance," *Journal of sports science & medicine*, 01-Mar-2007. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3778687/. [Accessed: 11-Mar-2022].