PREDICTING ENGLISH PREMIER LEAGUE MATCH RESULTS BY USING SUPERVISED MACHINE LEARNING TECHNIQUES

Non-Thesis M.Sc. Project Report by
Arda Karabiber

Submitted to the
Graduate School of Engineering
In Partial Fulfillment of the Requirements for
the Degree of
Master of Science

in the
Department of Data Science

Özyeğin University
January 2024

Approved by:

Professor Okan Örsan Özener
Advisor, Department of Industrial Engineering
Özyeğin University

Date Approved:

ABSTRACT

Betting in sports started to become very common since it is easily accessible thanks to computers and mobile phones. Nowadays, bookmakers in sports betting can have a lot of customers and the amount of money circulating in this sector is huge. Because of that, bookmakers started to use advanced machine learning techniques to predict match results and increase their profit with this method. They have access to enormous amount of data, and they achieve great results by combining this data with complex machine learning models.

The purpose of this project is, understanding how machine learning models are used to predict the results of football matches. Supervised machine learning techniques are chosen for this task. Several of these techniques has been used and their performances are compared regarding to different performance metrics. Since the amount of data is used is not a match to the ones bookmakers use, performance of the models might not be as high as their predictions. Still, with different data processing and hyperparameter tuning methods, performance of the models achieved good results considering the uncertain nature of predicting matches.

ÖZETÇE

Bilgisayar ve cep telefonlarının yaygınlaşması ile birlikte sporda bahis oldukça yaygın bir pratik olmaya başladı. Bu günlerde, bahis şirketleri çok fazla müşteri bulabiliyor ve bu sayede oldukça yüksek gelir elde edebiliyorlar. Sektörün hacmi de son yıllarda oldukça artmış durumda. Bu sebeple, bahis şirketleri gelirlerini daha da arttırmak ve kayıplarını azaltmak için maç tahminlerinde gelişmiş makine öğrenmesi metotlarını kullanmaya başladılar. Ellerindeki devasa boyutlardaki verileri kompleks modeller ile birleştirerek çok iyi sonuçlar elde etmeye başladılar.

Bu projenin amacı, makine öğrenmesi metotları ile futbol maçlarının sonuçlarının nasıl tahmin edilebileceğini anlamak ve göstermektir. Bu projede kullanmak üzere gözetimli makine öğrenmesi modelleri seçilmiştir. Birkaç farklı model kullanılıp sonuçları farklı performans metrikleri ile karşılaştırılmıştır. Burada kullanılan veri seti bahis şirketlerinin kullandıklarına oranla çok daha basit ve küçük olduğu için modellerin sonuçları da bahis şirketleri kadar başarılı olmayabilir. Ancak farklı veri işleme ve model parametreleri seçim yöntemleri ile iyi sayılabilecek sonuçlar elde edilmeye çalışılmıştır.

TABLE OF CONTENTS

INTRODUCTION **6**

DATA PROCESSING AND CLEANING **8**

2.1. Data Characteristics 8

2.2. Data Processing and Cleaning 9

2.2.1. Dropping rows or columns *9*

2.2.2. Structural Changes on Features *9*

2.2.3. Train – Test Split *9*

FITTING AND EVALUATING MODELS **10**

3.1. Random Forest 10

3.2. Bagging Classifier 11

3.3. XGBoost 12

TESTING IN A MORE REALISTIC SCENARIO **14**

4.1. Removing Features that Indicate Target Value 14

4.2. Adding Two New Features 15

CONCLUSION **17**

INTRODUCTION

Football is one of the most watched sports in the world with billions of fans worldwide and become an industry itself. It is such an enormous sector that it includes lots of smaller side sectors within itself. Betting is one, and probably largest, of them. There are lots of betting companies around the world that make huge profits from the game. Sports betting industry has become so big that it was banned or restricted in some countries. Betting companies, or bookmakers, have to make their calculations carefully in order to maximize their profits in this environment where there are excessive amount of people betting and doing their best to win their bets. That's why, bookmakers started to use machine learning algorithms to predict match results before while adjusting the odds. With complex algorithms and excess amount of data in their hands, bookmakers make better predictions and increase their profit thanks to that. However, this amount of data is not accessible for everyone, and the models that bookmakers are used are unknown to public.

The aim of this project is creating a supervised machine learning algorithm that can predict football match results with a reasonable accuracy. Since it is impossible to obtain a large dataset with hundreds of features that bookmakers used, a public dataset that includes match results and key statistics from matches is used. Also, as mentioned above, models that bookmakers use are unknown; therefore, the supervised machine learning techniques that we have learned in this program are utilized to achieve results.

In this project, English Premier League results between 2011 and 2021 are used as the dataset. The dataset includes match results, the teams are playing, the season, the referee, and several match statistics like shot numbers of home team or number of corners away team uses. As for the models used, three classification models are selected. These are random forest, bagging classifier, and XGBoost. Grid search method is implemented for hyperparameter tuning to all these three models and their performances are compared by using accuracy, precision, recall, and F1 score metrics.

Lastly, the best performed model among these three one is selected to be used in a more realistic scenario where the team statistics are not known. Since these statistics are unknown while predicting match results in real life, best performed model is tested in a scenario that is closer to real life situations.

DATA PROCESSING AND CLEANING

In this part, the data used for models will be explained and data processing steps will be clarified.

2.1. Data Characteristics

Dataset includes 11,113 samples with 23 features. These samples are all matches played in English Premier League between start of 1993-94 season and end of 2021-22 season.

Table 1: Features and their explanation

Column Name	Explanation
Season	The season that the match played
DateTime	Exact date of the match
HomeTeam	Name of home team
AwayTeam	Name of away team
FTHG	Full time home goals
FTAG	Full time away goals
FTR	Full time result (Home win, away win or draw)
HTHG	Half time home team goals
HTAG	Half time away team goals
HTR	Half time result (Home win, away win or draw)
Referee	Name of referee in that match
HS	Home team total shots
AS	Away team total shots
HST	Home team total shots on target
AST	Away team total shots on target
HC	Home team corners
AC	Away team corners
HF	Home team fouls
AF	Away team fouls
HY	Home team yellow cards
AY	Away team yellow cards
HR	Home team red cards
AR	Away team red cards

2.2. Data Processing and Cleaning

2.2.1. Dropping rows or columns

First, the matches played before 2011-12 season are dropped from the dataset in order to acquire a more compact dataset. Structure of Premier League was different and there were too many different teams that relegated and could not climb back after that season. Furthermore, statistics from older seasons might not be consistent reliable, so these seasons are not included for running the models.

After that, features can have strong impact on results are excluded. These features are “FTHG”, “FTAG”, “HTHG”, “HTAG”, and “HTR”. These can directly indicate the result of the matches; therefore, they are not included to prevent trivializing the performance of the models. “DateTime” feature is also excluded since there is another feature “Season” that also states the time period of the match played.

2.2.2. Structural Changes on Features

In this part, the structure of “Season” is changed. This feature shows in which season the match played (Ex: 2020-21). However, it is not possible to run classification models with this structure. Thus, this column is changed to “SeasonStartYear” that shows only the year of the season that match is played (Ex: For 2020-21 Season, it is 2020 now).

For the next part, dummy variables are added for string values in order to run models properly. Only three features were in string shape. These are “HomeTeam”, “AwayTeam”, and “Referee”. All these three features are transformed into dummy variables. In addition to that, the target variable “FTR” that indicates the result of the match is encoded by using label encoder from sklearn library.

2.2.3. Train – Test Split

For this step, the dataset is divided into X and y where X includes independent variables and y includes dependent (target) variable. Then, the dataset is split into train and test with test size of 0.2.

FITTING AND EVALUATING MODELS

Three models are used on the dataset and their results are compared. These three models are random forest classifier, bagging classifier, and XGBoost classifier. Grid search method as hyperparameter tuning process is used for all these models. The results are below:

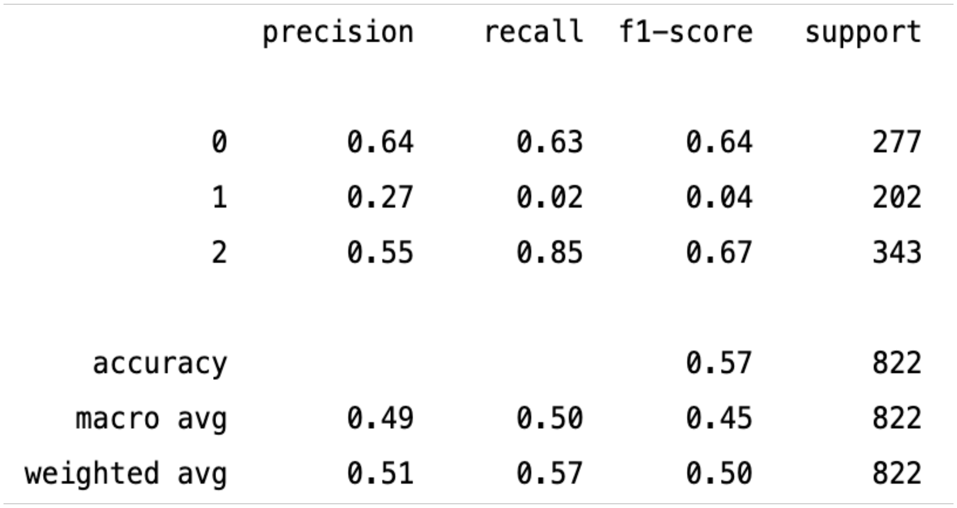
3.1. Random Forest

First model used on data is random forest classifier. Grid search is used to determine best parameters to use in the model. Best parameters are:

```
Best Hyperparameters: {'max_depth': 30, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 200}
```

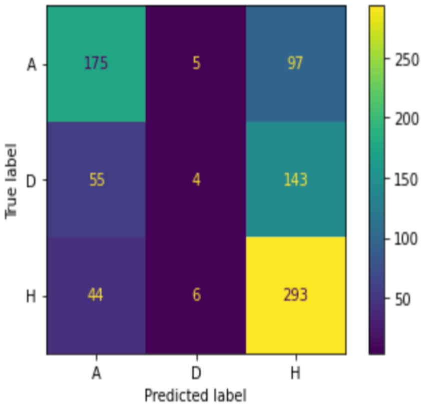
Performance of the model is below:

Figure 1: Classification report of Random Forest



```
              precision    recall  f1-score   support

           0       0.64      0.63      0.64       277
           1       0.27      0.02      0.04       202
           2       0.55      0.85      0.67       343

    accuracy                           0.57       822
   macro avg       0.49      0.50      0.45       822
weighted avg       0.51      0.57      0.50       822
```

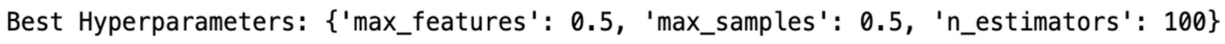
Figure 1: Confusion Matrix of Random Forest



Random forest achieved 0.57 accuracy. The most apparent problem in this model is predictions for class “D” which indicates matches resulted with draw. The performance of the model is very problematic specific to class “D” predictions.

3.2. Bagging Classifier

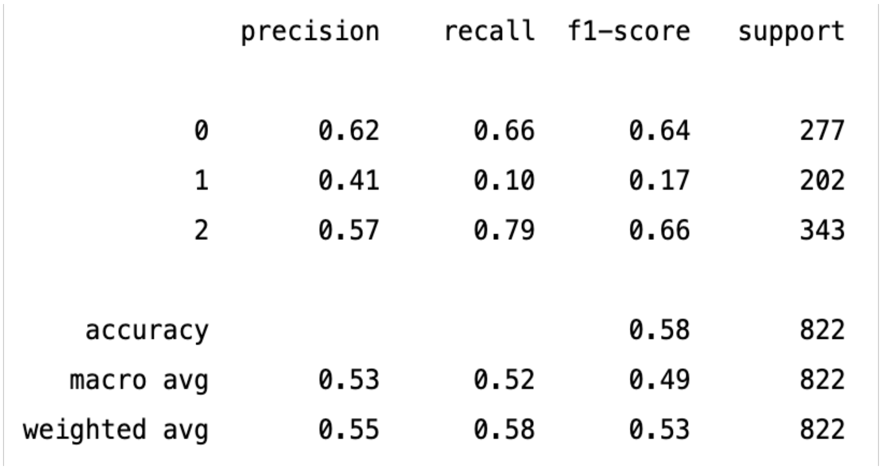
Second model used is bagging classifier. Again, grid search is used for parameter selection and best parameters from that operation are used in the model. Best parameters for this model are:



```
Best Hyperparameters: {'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 100}
```

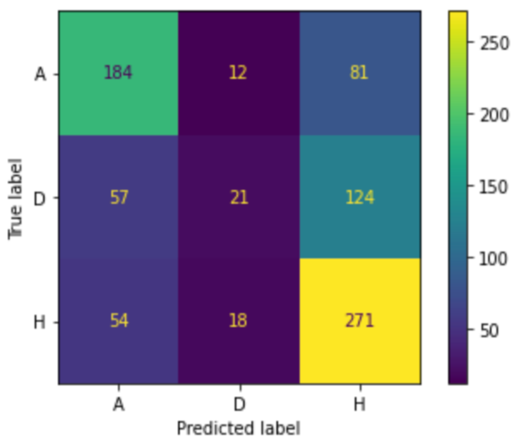
Performance of the model is below:

Figure 3: Classification report of Bagging Classifier



```
              precision    recall  f1-score   support

           0       0.62      0.66      0.64       277
           1       0.41      0.10      0.17       202
           2       0.57      0.79      0.66       343

    accuracy                           0.58       822
   macro avg       0.53      0.52      0.49       822
weighted avg       0.55      0.58      0.53       822
```

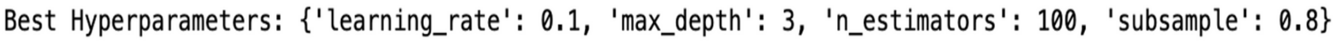
Figure 4: Confusion matrix of Bagging Classifier



Bagging classifier performed slightly better than random forest with 0.58 accuracy. In addition to that, it has achieved much better results while predicting class “D”. It still not a good result, but we should consider that it is very difficult to predict draws in football matches.

3.3. XGBoost

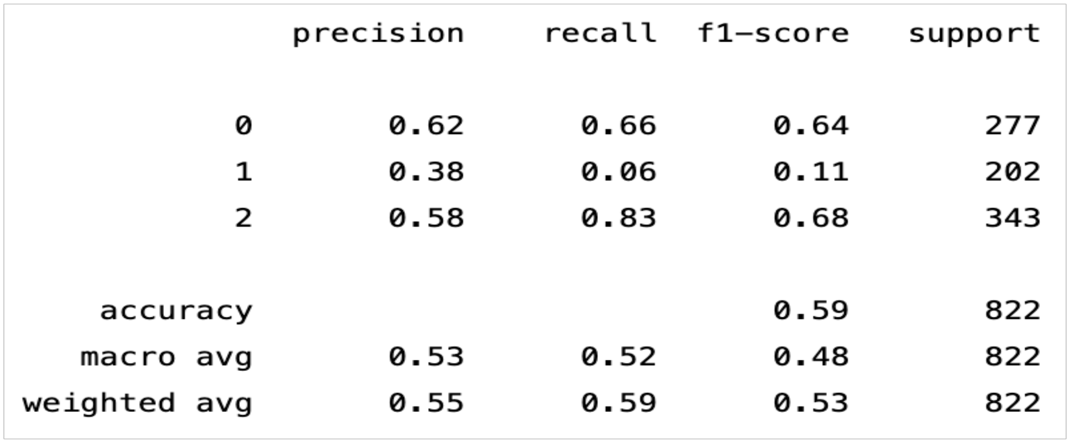
XGBoost classifier is the last model used on the data. For parameter selection, same procedure is used with the previous two models. Best parameters selected for this model are:



```
Best Hyperparameters: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.8}
```

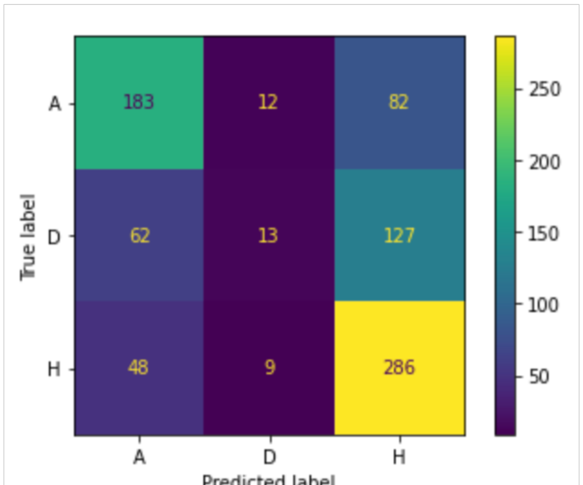
Performance of the model is below:

Figure 5: Classification report of XGBoost



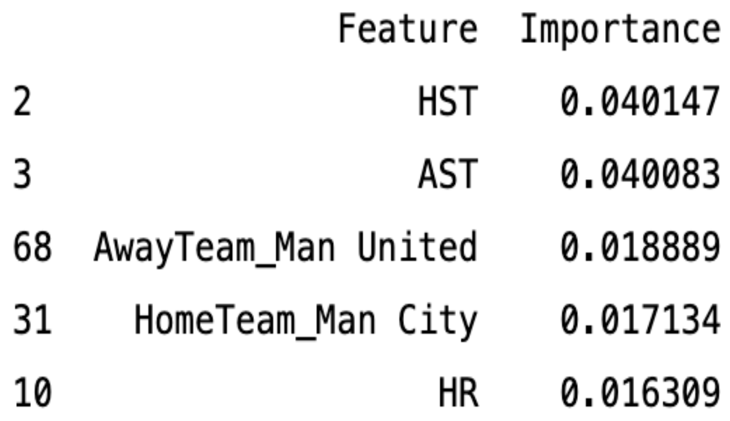
```
              precision    recall  f1-score   support

           0       0.62      0.66      0.64       277
           1       0.38      0.06      0.11       202
           2       0.58      0.83      0.68       343

    accuracy                           0.59       822
   macro avg       0.53      0.52      0.48       822
weighted avg       0.55      0.59      0.53       822
```

Figure 6: Confusion matrix of XGBoost



XGBoost performed best based on accuracy with a very small margin. Still, it could not predict draws as good as bagging classifier. However, our aim is predicting as many as games correctly, so accuracy is a more important metric here. XGBoost is selected as the best model on this dataset based on accuracy as a performance metric. So, the model is observed a bit more to understand how it performed. For that, the most impactful features in the model are examined by using feature importance method. The results are below:

Figure 7: Feature importance list for XGBoost



```
                Feature  Importance
2                   HST    0.040147
3                   AST    0.040083
68  AwayTeam_Man United    0.018889
31     HomeTeam_Man City    0.017134
10                   HR    0.016309
```

It is fair to say that there are two distinct variables with higher importance than other features. These are "HST" (Home team shots on target) and "AST" (Away team shots on target). Number of shots on target can be parallel to goals scored in football matches since these shots have higher probability of becoming goal than off target shots. Therefore, existence of these features in the dataset helped models to predict the results better. Nevertheless, we do not have access to statistics like these while predicting match results in real life since the matches have not been played yet. In the next phase, predicting match results in a more realistic scenario will be discussed.

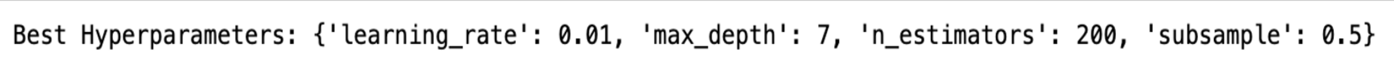
TESTING IN A MORE REALISTIC SCENARIO

In previous part, a dataset with many features that indicate the result was used for predictions. However, it is not a feasible practice in real life since it is impossible to know the statistics of matches before they are played. Thus, a few changes are made in the dataset to make the scenario closer to real life situation.

4.1. Removing Features that Indicate Target Value

In order to achieve that goal, all statistical columns are dropped from dataset. Only "Season", "HomeTeam", "AwayTeam", and "Referee" columns are kept for running the model. Since XGBoost was the best performed model in previous part, it is selected for this dataset.

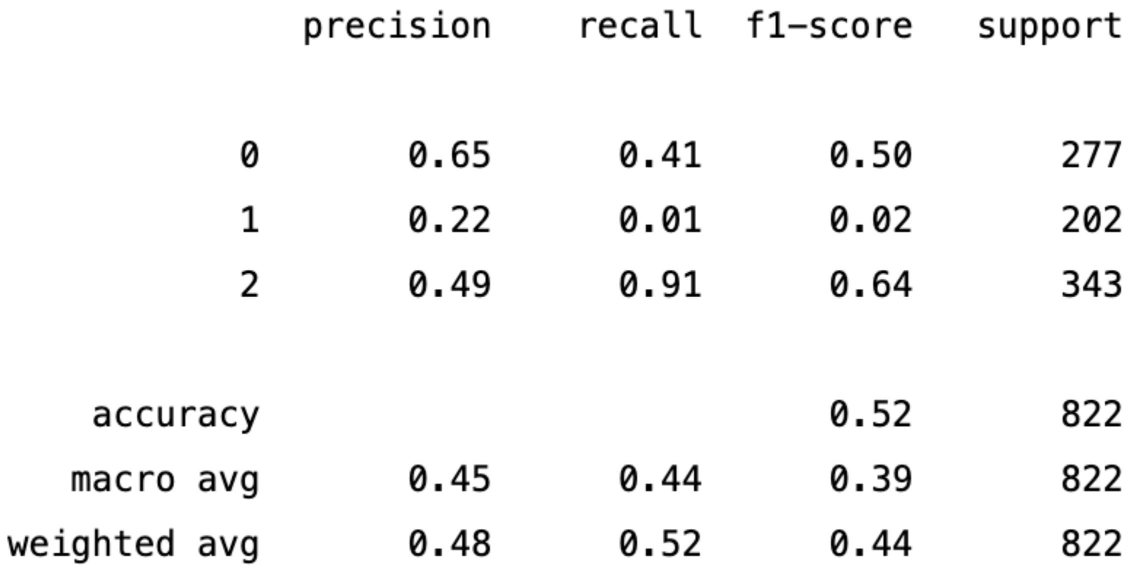
Grid search is used for selecting parameters of the model. Best parameters selected for this model are:



```
Best Hyperparameters: {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 200, 'subsample': 0.5}
```

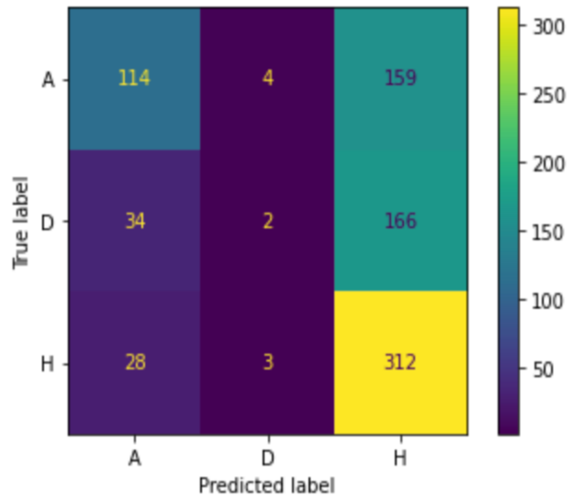
Performance of the model is below:

Figure 8: Classification report of XGBoost on new dataset



```
              precision    recall  f1-score   support

           0       0.65      0.41      0.50       277
           1       0.22      0.01      0.02       202
           2       0.49      0.91      0.64       343

    accuracy                           0.52       822
   macro avg       0.45      0.44      0.39       822
weighted avg       0.48      0.52      0.44       822
```

Figure 9: Confusion matrix of XGBoost on new dataset



As expected, the performance decreased significantly with scarce of features. There are only 4 features and none of them indicate the result on their own. It was an inevitable result, and the model cannot perform well in the present case.

4.2. Adding Two New Features

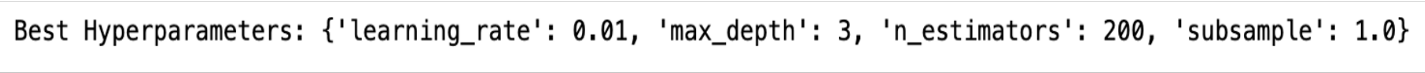
For this reason, two new features are added to this new dataset to improve the performance of the model. However, these features are known before the matches have been played. So, they do not change the real-life setting.

First of two features is form of teams. It states win percentage of teams in their last 5 matches before the selected match. It consists of two columns: one for home team, other one for away team. This feature is calculated number of wins of the team divided by total matches played in this period, which is 5 in this case.

Second feature is head-to-head results which introduces performances of teams playing against in previous their encounters. It is the historical win percentage of home team against away team based on their previous meetings. It consists of one column.

After these two new features are added, the dataset reached its final form and becomes ready for running the model. Again, XGBoost is used for this part as well.

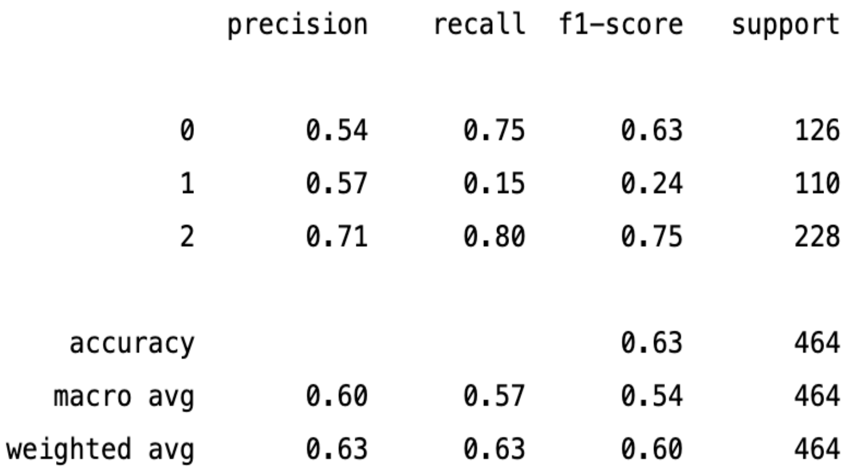
For parameter selection, same methodology is followed. Grid search is implemented, and best parameters come from grid search are:



```
Best Hyperparameters: {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200, 'subsample': 1.0}
```

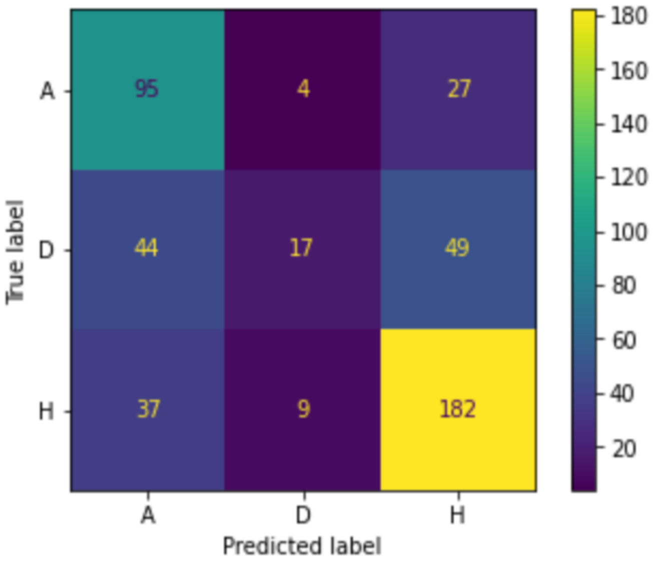
Performance of the model is below:

Figure 10: Classification report of XGBoost on final dataset



```
              precision    recall  f1-score   support

           0       0.54      0.75      0.63       126
           1       0.57      0.15      0.24       110
           2       0.71      0.80      0.75       228

    accuracy                           0.63       464
   macro avg       0.60      0.57      0.54       464
weighted avg       0.63      0.63      0.60       464
```

Figure 11: Confusion matrix of XGBoost on final dataset



This time, the model performed significantly better with accuracy of 0.63. Surprisingly, this model even surpassed older models that used statistical results of matches. We infer that recent form of teams and results of previous matches between two teams playing have strong effect on result of the match. With this result, it is fair to say that the model performs at an acceptable level even without using statistics of matches as features.

CONCLUSION

To conclude, predicting the outcome of football matches is not an easy task. There are a lot of unknown variables that can affect the results and gathering every feature that can have impact on outcome is nearly impossible. Even models that bookmakers use cannot give correct results 100% with the amount of data and resources they have. This project shows that predicting an acceptable number of matches is possible even with fewer data and less complex models. But still, feature selection performs a key role. At last part, adding two new features affected the result dramatically since the form of teams and results of previous matches between same two teams have a huge impact on result. Lastly, it is fair to say that it is possible to use machine learning models we have learned through this program in real life scenarios and with proper arrangements they can perform well.