Fundamentals of Data Science

Project's Report

Predicting Football Results Using Machine Learning Techniques

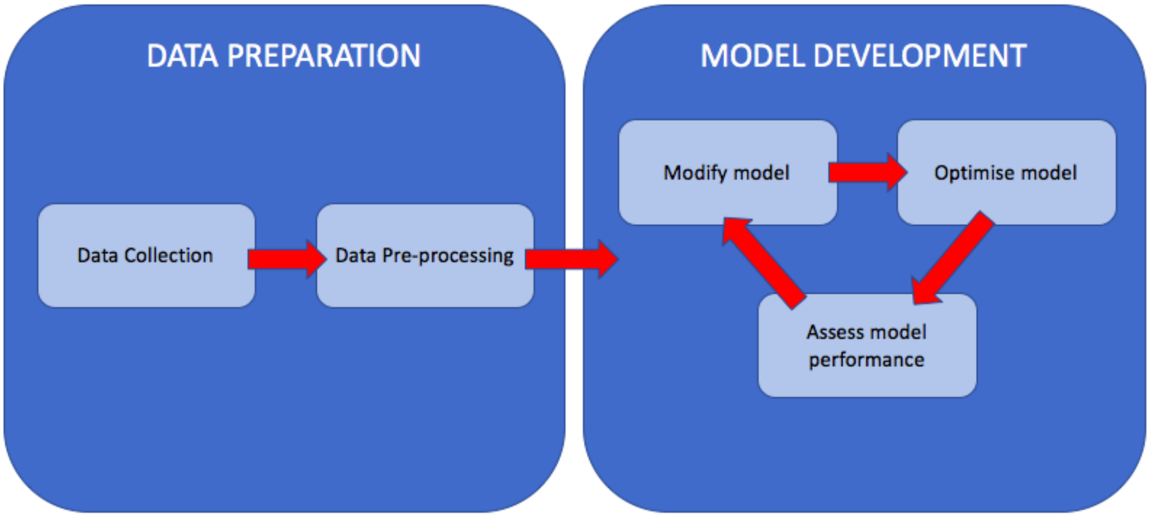
Görkem Ayten

380807

I. Introduction

This project uses Machine Learning to predict the results of a football match when given some stats from previous matches. In order to predict the result different Machine Learning models were used and the accuracy of them was compared.

The workflow of the project is as shown below.



II. Dataset

i. Data Origin

I have obtained a dataset from the Kaggle Data Science website. This database contains:

- Match scores
- Players and team attributes from EA Sports FIFA games
- Data from more than 25,000 men's professional football games
- Seasons 2008 to 2016
- Betting odds from up to 10 providers
- Detailed match events

I chose English Premiere League for match events that I needed to build my expected goals models. In addition, I used the data from 2008 season to 2016.

ii. Data Pre-processing

The most important step before building my model is to analyze and pre-process the data to make sure that it is in a usable format for training and testing different models.

First of all, I fetched the necessary data from database. I chose the English Premiere League.

```
matches = pd.read_sql_query("SELECT season, date,
home_team_api_id,away_team_api_id,home_team_goal,away_team_goal FROM Match WHERE
league_id is 1729", database)
```

I fetch the "season", "home team", "away team", "home team goal", "away team goal" information from table.

Then I checked if there is any missing data in the dataset I have. I needed to confirm that I didn't have missing data to train my model incorrectly.

```
matches.isnull().sum()
```

```
season 0
date 0
home_team_api_id 0
away_team_api_id 0
home_team_goal 0
away_team_goal 0
```

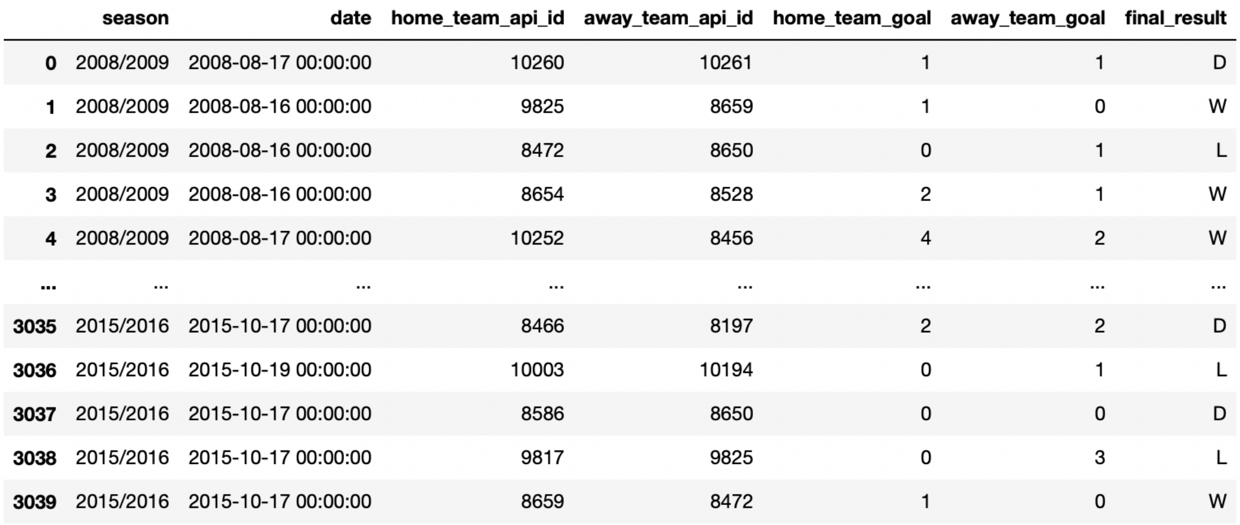
After making sure that I did not have any missing information, I made statistical calculations. I calculated the mean, max and min value of home team's and away team's goals.

```
Average of home team goal: 1.550986842105263
Average of away team goal: 1.1595394736842106
Maximum goals scored by the home team: 9
Maximum goals scored by the away team: 6

Total number of matches: 3040
Number of matches won by home team: 1390
Win rate: 45.723684210526315%
```

Just by looking at the past match data I have, I calculated that the home team wins about 46%.

The dataset didn't have a column showing whether the home team won or not. For this reason, I created a new column named “final_result” by looking at the final score of matches. Then I got the following dataset:

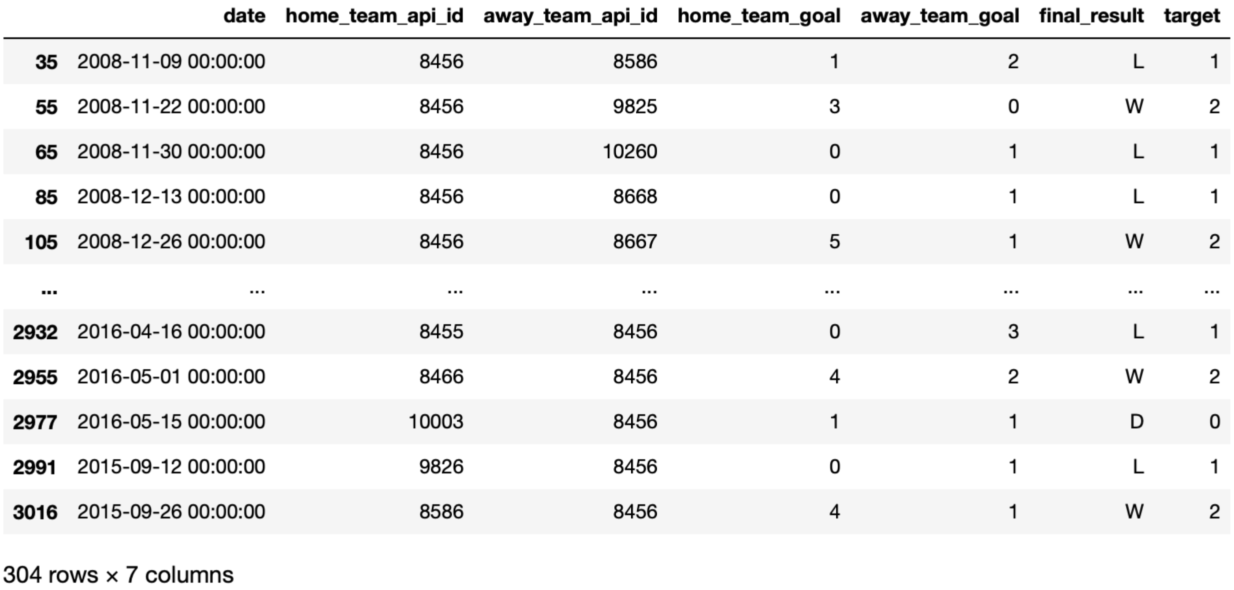


	season	date	home_team_api_id	away_team_api_id	home_team_goal	away_team_goal	final_result
0	2008/2009	2008-08-17 00:00:00	10260	10261	1	1	D
1	2008/2009	2008-08-16 00:00:00	9825	8659	1	0	W
2	2008/2009	2008-08-16 00:00:00	8472	8650	0	1	L
3	2008/2009	2008-08-16 00:00:00	8654	8528	2	1	W
4	2008/2009	2008-08-17 00:00:00	10252	8456	4	2	W
...	...	...	...	...	...	...	...
3035	2015/2016	2015-10-17 00:00:00	8466	8197	2	2	D
3036	2015/2016	2015-10-19 00:00:00	10003	10194	0	1	L
3037	2015/2016	2015-10-17 00:00:00	8586	8650	0	0	D
3038	2015/2016	2015-10-17 00:00:00	9817	9825	0	3	L
3039	2015/2016	2015-10-17 00:00:00	8659	8472	1	0	W

Then I looked at how many matches each team played in total to decide which football team I will use to train my model. I chose Arsenal, one of the teams that played the most matches, to train my model.

I got rid of columns that would not work for me while training my model such as “season”.

Finally, I converted the match results from string value to int value and added it as a new column to my dataset. The final version of my dataset as follows:



	date	home_team_api_id	away_team_api_id	home_team_goal	away_team_goal	final_result	target
35	2008-11-09 00:00:00	8456	8586	1	2	L	1
55	2008-11-22 00:00:00	8456	9825	3	0	W	2
65	2008-11-30 00:00:00	8456	10260	0	1	L	1
85	2008-12-13 00:00:00	8456	8668	0	1	L	1
105	2008-12-26 00:00:00	8456	8667	5	1	W	2
...	...	...	...	...	...	...	...
2932	2016-04-16 00:00:00	8455	8456	0	3	L	1
2955	2016-05-01 00:00:00	8466	8456	4	2	W	2
2977	2016-05-15 00:00:00	10003	8456	1	1	D	0
2991	2015-09-12 00:00:00	9826	8456	0	1	L	1
3016	2015-09-26 00:00:00	8586	8456	4	1	W	2

304 rows × 7 columns

L: 1 W: 2 D: 0

I did some statistical calculations about Arsenal such as win rate and average goal. I calculated the home match win rate as about 53% by just looking at the past matches.

```
Total number of matches that Arsenal played: 304
Number of matches won by Arsenal: 137
Win rate: 45.06578947368421%

The home goal average of Arsenal scored: 1.6578947368421053
The away goal average of Arsenal scored: 1.2796052631578947
The maximum home goals scored by Arsenal: 8
The maximum away goals scored by Arsenal: 6
```

III. Design

I used all previous matches of Arsenal for training my Machine Learning models. I used three different Machine Learning Models:

- **Random Forrest Classifier:** Tree based ensemble model expected to work well due to the sparsity of the data. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.
- **K-nearest Classifier:** Non supervised learning algorithm where an object is classified by a plurality vote of its K nearest neighbors.
- **Gradient Boosting Classifier:** Just like Random Forest, a tree-based ensemble model expected to work well due to the sparsity of the data.

I used above models because predicting the result of match is classification problem.

IV. Implementation

I determined the predictors as “home_team_id” and “away_team_id” for training Machine Learning models. In other words, I trained my ML models just looking at the previous matches of Arsenal.

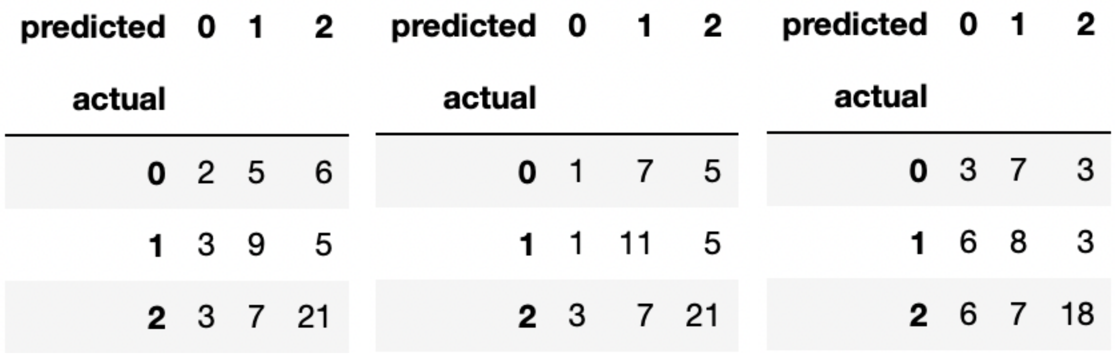
I used 80% of the matches as the training set, and 20% as the testing set.

The accuracy score and precision score of all models as follows

```
Random Forrest Classifier Accuracy Score: 52.459016393442624%
Gradient Boosting Classifier Accuracy Score: 54.09836065573771%
K-nearest Classifier Accuracy Score: 47.540983606557376%

Random Forrest Classifier Precision score: 52.459016393442624%
Gradient Boosting Classifier Precision score: 54.09836065573771%
K-nearest Classifier Precision score: 47.540983606557376%
```

As it can be seen above, Gradient Boosting Classifier is the most accurate model among them. Even if the accuracy of K-nearest Classifier is the lowest, it is better than general prediction.



predicted	0	1	2
actual			
0	2	5	6
1	3	9	5
2	3	7	21

predicted	0	1	2
actual			
0	1	7	5
1	1	11	5
2	3	7	21

predicted	0	1	2
actual			
0	3	7	3
1	6	8	3
2	6	7	18

The confusion matrix of Random Forrest, Gradient Boosting and K-nearest Classifiers are above, respectively.

Confusion matrices represent counts from predicted and actual values. It can be seen that, there is a possitive correlation between confusion matrix and accuracy score. The results of the matches that the models predicted correctly are as follows.

```
Gradient Boosting Classifier: 33
Random Forrest Classifier: 32
K-nearest Classifier: 29
```

Moreover, by looking at confusion matrix, we can understand which outcome our model is more likely to predict. Then, according to this we can improve our model.

V. Conclusion

To conclude, in this project Random Forrest, Gradient Boosting and K-nearest Machine Learning models are used for predicting the result of football matches. The results were very satisfying considering it was possible to have an accuracy score better than just a random prediction or a "home team always wins" prediction. We can get a more accurate model by adding more parameters while training our model.

The code of the project:
https://github.com/gorkemaytenn/football_matches_predictor.git