Football Match Prediction

- Analytics system technology(ALY6140)
- Capstone Project
- Date: 03/28/2022
- Submitted by: Abhinav Jain
- Submitted To: Prof. Richard Zhi

Agenda

Introduction

Background Information

Research question

Finding/Approach

- Exploratory Data Analysis & Visualization
- Predicting/ forecasting

Conclusion




Introduction

Predicting match results is the most difficult task. Football enthusiasts try to predict how each match will end. Betting on the outcome of football matches is a tradition in the United Kingdom. Most of the bets are put on the half-time and full-time results. Probability aids in determining the likelihood of a good outcome. By using a statistical model to forecast the likely outcome in the match.

Background Information

• To assess the chance of a future event, data analytics, statistical algorithms, and machine learning approaches are used to compare data to previous events. Instead of only knowing what has happened, the goal is to make better predictions of what will happen in the future.
• Predictive techniques are used to analyze the team's performance during the match. In this study, different models will be employed to estimate the results of a football match, including Random Forest Classifier, K Nearest Neighbor, and Logistics Regression.
• The purpose of a data scientist is to collect data, do analysis, interpret the data in a meaningful way, and apply prediction models to the data.

Methods:
I: Predict the matches by getting historical data of the team with the home team coach and opponent team coach.
II. Predict the accuracy by knowing the history match dates of the home and opponent team
III. Predict the motivation of the team by evaluating the team history rating.

Research Question

Q1. What was the team performance history at home play?

Q2. What was the rating of the opponent team?

Q3. In the past, when did a match help you?

Q4. Which coach had a better track record in previous leagues?

Q5. What were the results of a team's prior leagues?

Finding/Approach

Exploratory Data Analysis & Visualization

Predicting/ forecasting

Import: Raw Dataset

Dataset: 110938 rows and 190 col

Clean: Dataset after cleaning

Dataset: 110938 rows and 17 col

Football Raw Dataset

```
df.head()
```

	id	target	home_team_name	away_team_name	match_date	league_name	league_id	is_cup	home_team_coach_id	away_team_coach_id	...	away_te
0	11906497	away	Newell's Old Boys	River Plate	2019-12-01 00:45:00	Superliga	636	False	468196.00000	468200.00000	...	
1	11984383	home	Real Estelí	Deportivo Las Sabanas	2019-12-01 01:00:00	Primera Division	752	False	516788.00000	22169161.00000	...	
2	11983301	draw	UPNFM	Marathón	2019-12-01 01:00:00	Liga Nacional	734	False	2510608.00000	456313.00000	...	
3	11983471	away	León	Morelia	2019-12-01 01:00:00	Liga MX	743	False	1552508.00000	465797.00000	...	
4	11883005	home	Cobán Imperial	Iztapa	2019-12-01 01:00:00	Liga Nacional	705	False	429958.00000	426870.00000	...	

5 rows × 190 columns

```
df.shape
```

(110938, 190)

Cleanup

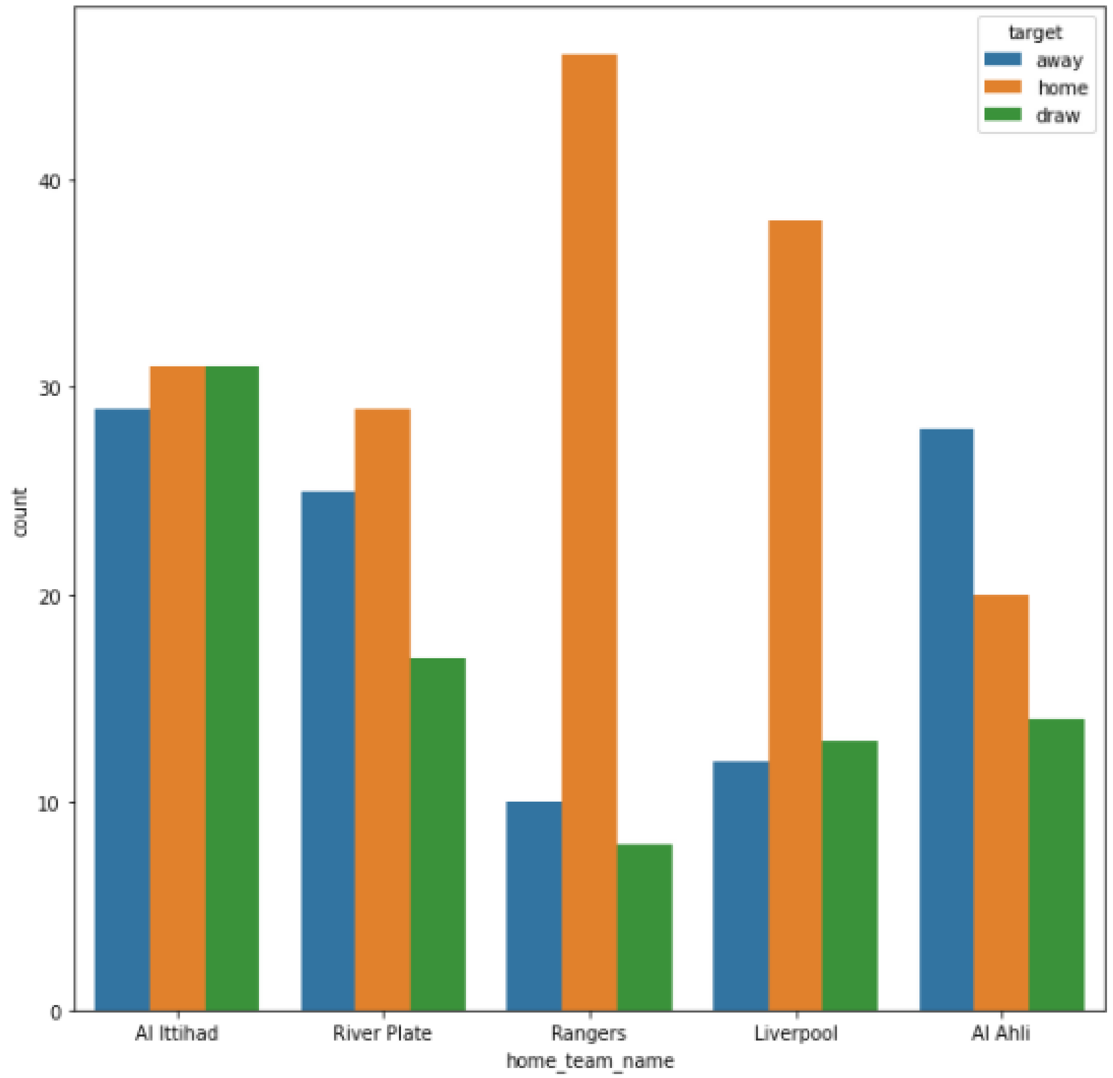
	target	home_team_name	away_team_name	league_name	home_team_coach_id	away_team_coach_id	home_
0	away	Newell's Old Boys	River Plate	Superliga	468196.00000	468200.00000	
1	home	Real Estelí	Deportivo Las Sabanas	Primera Division	516788.00000	22169161.00000	
2	draw	UPNFM	Marathón	Liga Nacional	2510608.00000	456313.00000	
3	away	León	Morelia	Liga MX	1552508.00000	465797.00000	
4	home	Cobán Imperial	Iztapa	Liga Nacional	429958.00000	426870.00000	

```
data.shape
```

```
(110938, 17)
```

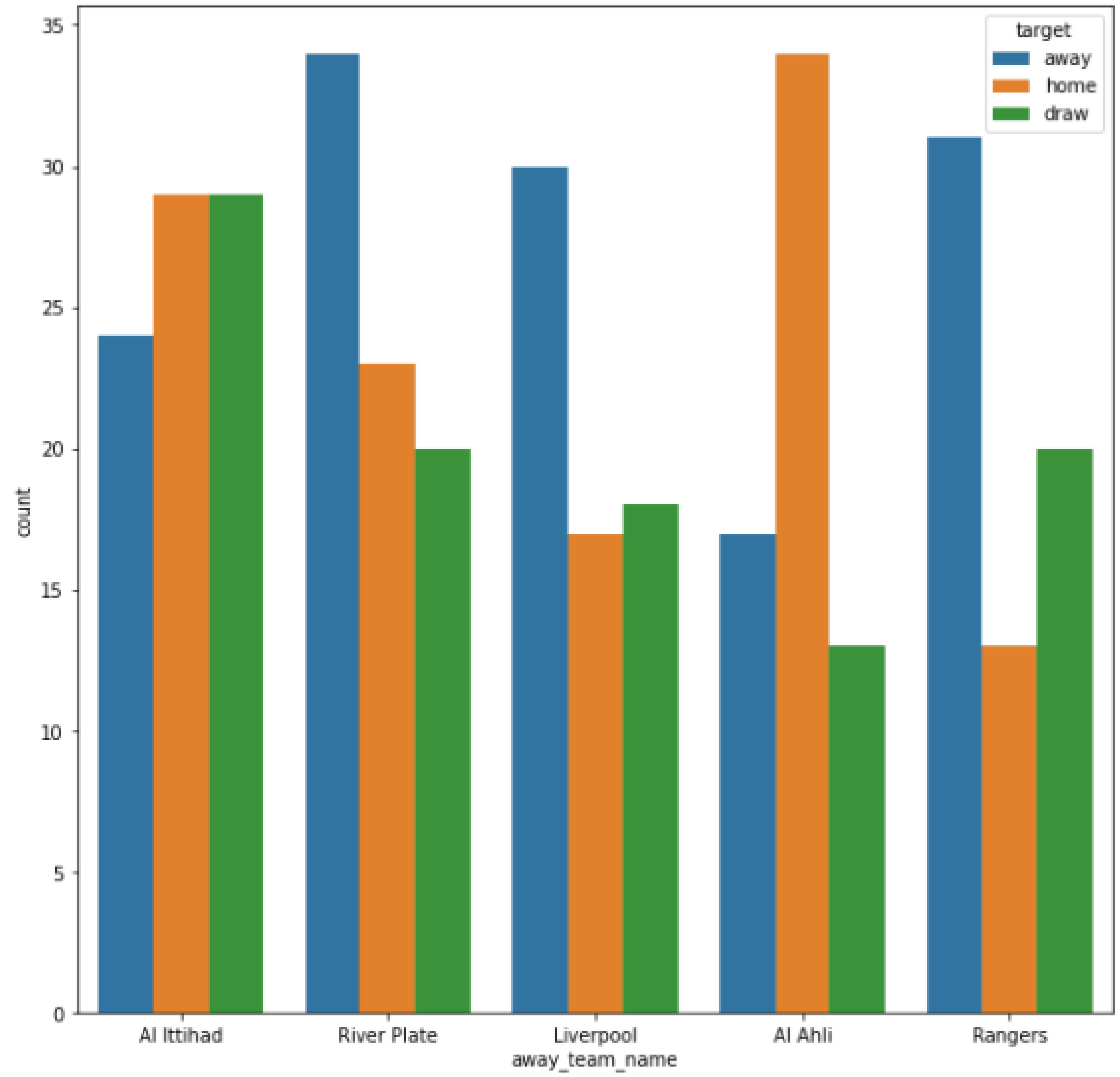
Home Team Name

```
plt.figure(figsize=(10,10))
sns.countplot(x="home_team_name",hue="target",data=df,order=df.home_team_name.value_counts().iloc[:5].i
plt.show()
```



Opponent Team Name

```python
plt.figure(figsize=(10,10))
sns.countplot(x="away_team_name",hue="target",data=df,order=df.away_team_name.value_counts().iloc[:5].i
plt.show()
```



History home team with features(using groupby())

```
df.groupby('home_team_name')[play_home_features].mean()
```

home_team_name	home_team_history_is_play_home_1	home_team_history_is_play_home_2	home_team_history_is_play_home_3
07 Vestur	0.53846	0.38462	0.53846
1. FC M'gladbach	0.00000	1.00000	0.00000
1. FC Merseburg	0.00000	0.66667	0.50000
1. Maj Ruma	0.20000	0.50000	0.50000
12 de Octubre	0.32000	0.64000	0.44000
...	...	...	...
Žilina	0.34483	0.48276	0.55172
Žilina II	0.30000	0.65000	0.50000
Žilina U19	0.14286	0.71429	0.28571
Župa	0.00000	0.50000	0.00000
Žďár nad Sázavou	0.25000	0.50000	0.25000

9813 rows × 20 columns

History opponent team with features(using groupby())

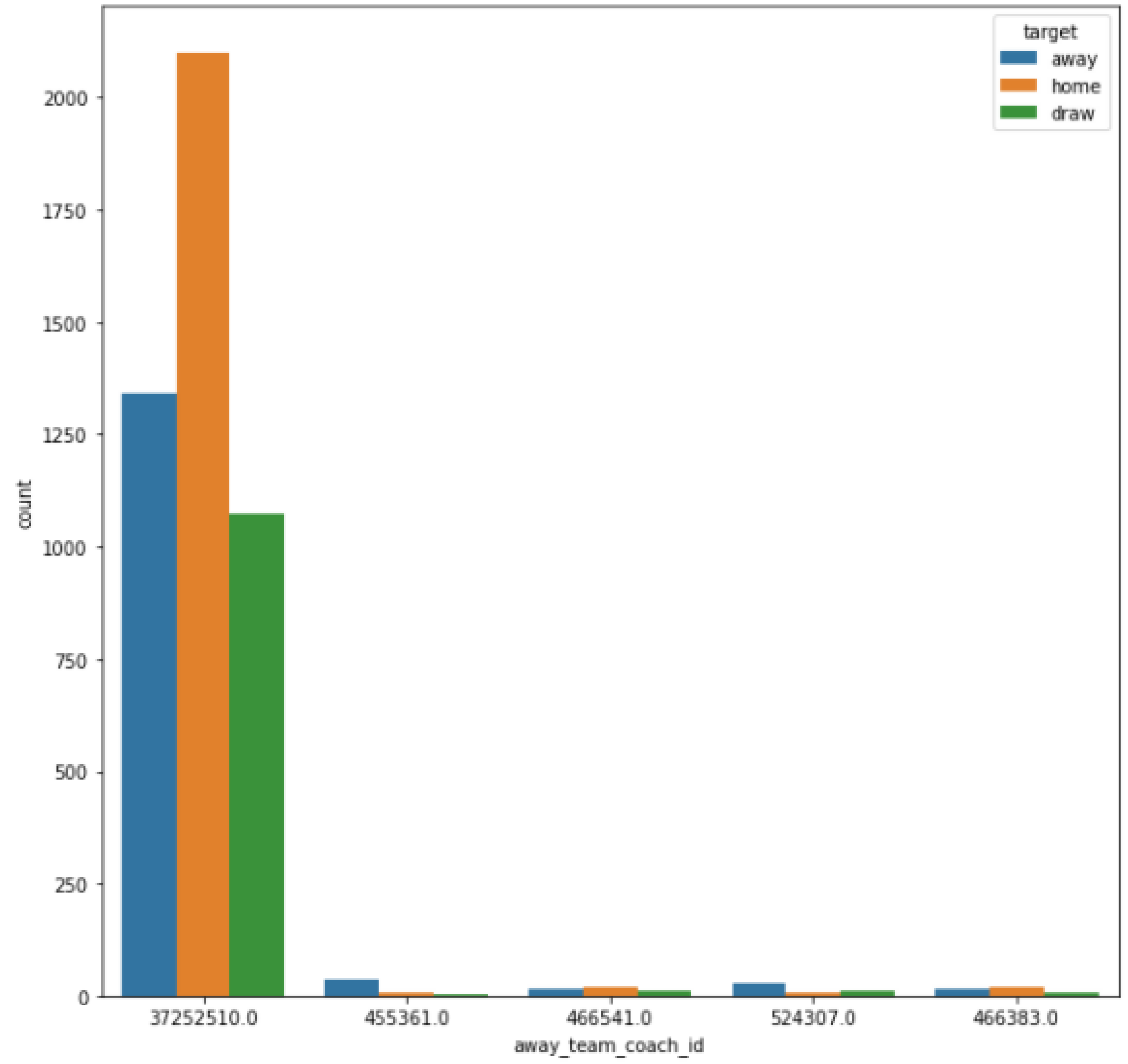
```
df.groupby('home_team_name')[opponent_rating_features].mean()
```

home_team_name	home_team_history_opponent_rating_1	home_team_history_opponent_rating_2	home_team_history_opponent_r
07 Vestur	8.61168	8.75077	
1. FC M'gladbach	11.37313	9.09024	
1. FC Merseburg	9.23196	7.66353	
1. Maj Ruma	7.18673	5.48778	
12 de Octubre	7.52886	7.21602	
...	...	...	
Žilina	6.19983	6.05111	
Žilina II	8.96911	7.35407	
Žilina U19	7.32640	7.03376	
Župa	8.55600	6.40655	
Žďár nad Sázavou	11.56933	12.10130	

9813 rows × 20 columns

Opponent team coach_id

```
plt.figure(figsize=(10,10))
sns.countplot(x="away_team_coach_id",hue="target",data=df,order=df.away_team_coach_id.value_counts().il
plt.show()
```



Track record of history cup of opponent team

```
df.groupby('away_team_name')[away_is_cup_features].mean()
```

away_team_name	away_team_history_is_cup_1	away_team_history_is_cup_2	away_team_history_is_cup_3	away_team_history_is
07 Vestur	0.00000	0.00000	0.11111	
1. FC M'gladbach	0.00000	0.00000	0.00000	
1. FC Merseburg	0.00000	0.00000	0.00000	
1. Maj Ruma	0.00000	0.00000	0.00000	
12 de Octubre	0.11538	0.07692	0.03846	
...	...	...	...	
Žilina	0.10714	0.10714	0.21429	
Žilina II	0.00000	0.00000	0.00000	
Žilina U19	0.00000	0.00000	0.00000	
Župa	0.00000	0.00000	0.00000	
Žďár nad Sázavou	0.00000	0.00000	0.00000	

9892 rows × 10 columns

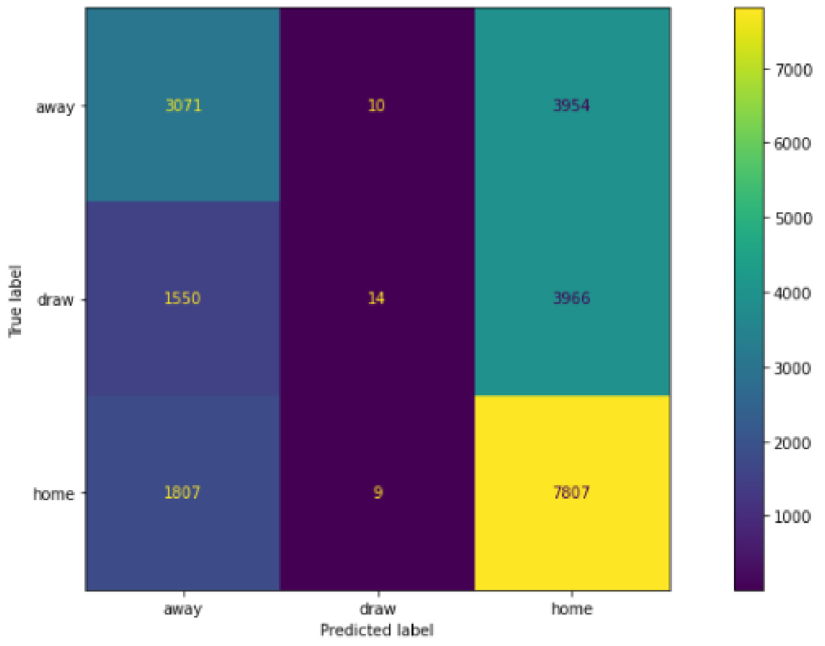
Timeline

Collecting Data	Data Cleanup/ Pre-processing	Data Extraction	Implementation	Testing	Train
collect the dataset From Kaggle	removing missing data drop variables Select variables	extracted Home team Opponent team data	analysis through Bar graphs Implementation of the prediction models Logistics Regression, Random forest classifier, KNN	created test train variable	training the models to get the optimum Prediction results

Prediction Models

LOGISTICS REGRESSION

```python
# Confusion matrix
cm = confusion_matrix(y_test, y_pred, labels=pipeline.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=pipeline.classes_)
fig, ax = plt.subplots(figsize=(15,7))
disp.plot(ax=ax)
plt.show()
```

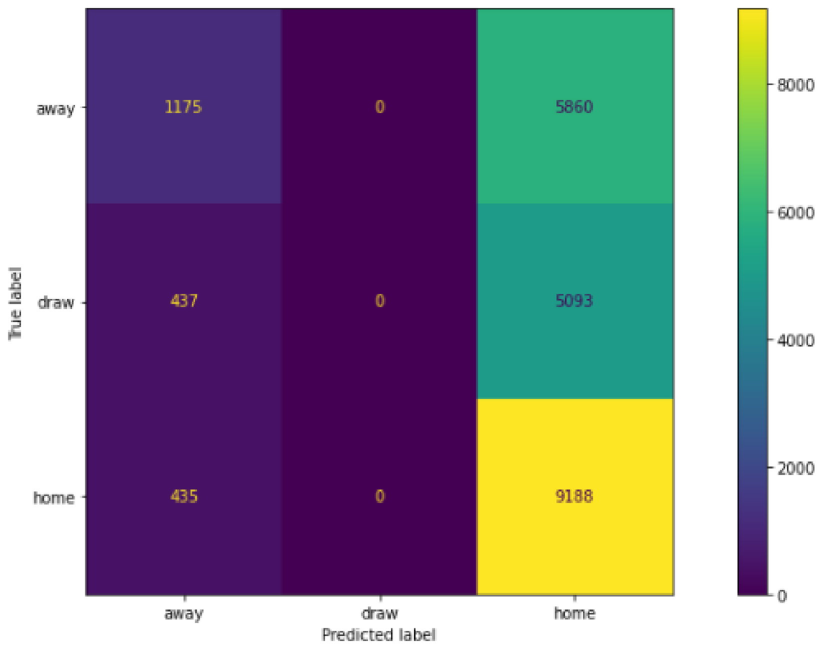


- Easier to Implement
- Efficient to train
- Fast while classifying unknown records
- Interpret the Model coefficient as indicators
- Accuracy score: 49%

```
Accuracy score: 0.4908959798089057

              precision    recall  f1-score   support

        away       0.48      0.44      0.46      7035
        draw       0.42      0.00      0.01      5530
        home       0.50      0.81      0.62      9623

    accuracy                           0.49     22188
   macro avg       0.47      0.42      0.36     22188
weighted avg       0.47      0.49      0.41     22188
```

RANDOM FOREST CLASSIFIER

```
# Confusion matrix
cm = confusion_matrix(y_test, y_pred, labels=pipeline.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=pipeline.classes_)
fig, ax = plt.subplots(figsize=(15,7))
disp.plot(ax=ax)
plt.show()
```



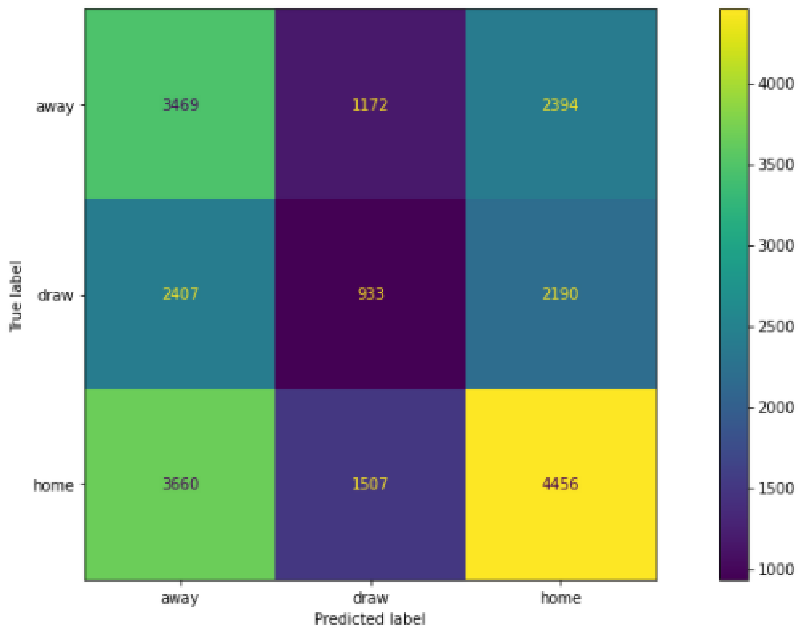
- Works well with large dimensional data
- Working with a subset
- Fast to train than the decision tree
- Easily work with hundreds of features
- Low correlation is the key
- Accuracy score: 46%

```
Accuracy score: 0.46705426356589147
```

	precision	recall	f1-score	support
away	0.57	0.17	0.26	7035
draw	0.00	0.00	0.00	5530
home	0.46	0.95	0.62	9623
accuracy			0.47	22188
macro avg	0.34	0.37	0.29	22188
weighted avg	0.38	0.47	0.35	22188

KNN-CLASSIFIER (K-NEAREST NEIGHBOR)

```
# Confusion matrix
cm = confusion_matrix(y_test, y_pred, labels=pipeline.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=pipeline.classes_)
fig, ax = plt.subplots(figsize=(15,7))
disp.plot(ax=ax)
plt.show()
```



- Highly accurate Predictions
- Solves both classification and regression problem statement
- KNN algorithm for multiclass classification
- Recommendation Systems
- Accuracy score: 39%

```
Accuracy score: 0.3992248062015504

              precision    recall  f1-score   support

        away       0.36      0.49      0.42      7035
        draw       0.26      0.17      0.20      5530
        home       0.49      0.46      0.48      9623

    accuracy                           0.40     22188
   macro avg       0.37      0.37      0.37     22188
weighted avg       0.39      0.40      0.39     22188
```

Conclusion

Finally, forecasting the outcome will contribute to assessing the psychological outcome of the match and will provide an opportunity. In the field of data analytics, many firms are employing these forecasting tools to prepare for outperformance. We need to train and test the model in such a manner that we can attain that accuracy to develop the model to the level of 90-95 percent correctness, and we need to provide a better outcome after running the model numerous times.

References

[1]Football Match Probability Prediction

https://www.kaggle.com/c/football-match-probability-prediction/data?select=test.csv

Thank You