

Football Match Prediction Using Various Algorithms

Data Analytics (ECS648U/ ECS784P) Group Project

Group Members

Aizaz Shahid 180535507

Muhammad Arsam Kamran 180924192

Priyanka Puttaswamy 180758106

Shuang Wu 180345870

Queen Mary University of London

April 2019

Table of contents

Table of contents

1. Introduction

1.1 Background

1.2 Project Outline - Problem Statement

2. Literature Review

3. Dataset

3.1 Sources of dataset

3.2 Description of dataset

4. Data Analysis

4.1 Description of data pre-processing steps

4.2 Methodology

4.3 Data selection

4.4 Things learned from exploring the data, including visualisations

4.5 Justification for features used

5. Implementation

5.1 Tools, libraries and programming language

5.2 Implementation

5.2.1 KNN Classifier

5.2.2 Regression

5.2.3 Naive Bayes

6. Testing/Results

6.1 Validation approaches

6.1.1 RMSE(Root Mean Square Error)

6.1.2 MPE(Mean Percentage Error)

6.2 Results

7. Conclusions

7.1 Possible extensions or business applications of your project

7.2 Future improvements

7.3 Challenges and successes

7.4 Summary

8. Appendices

8.1 Code

8.2 Data

8.3 Sources

9. References

Student	Effort
Muhammad Arsam Kamran	25%
Shuang Wu	25%
Priyanka Puttaswamy	25%
Aizaz Shahid	25%

1. Introduction

Result analysis of game fits has been a controversial matter over time for both data analysts and standard public. Modern methods for processing statistics, in aggregate with strong computational aids of computers, enable us to predict the effects of the subsequent suits by applying various algorithms on related gathered data. Football is one such crew game that draws high-quality mass target audience. Considering the detailed description of football matches data over time, matches assemble an enormous and profitable database to test forecast of match results. Our present work deals with data-driven analyses of football match performance using three distinct algorithms (KNN Classifier, Regression and Naive Bayes) and comparing the accuracy to predict the outcome of further matches.

1.1 Background

Football basically refers to a game played by two teams of eleven players each defending goals across each other where a point corresponds to kicking the ball into opponent's goal line. There are three conceivable outcomes of a football match: win, draw and lose. The winning team accomplishes the most elevated score in a span of 90 minutes. The outcome of a football game has been considered as subject of various logical exertion in the undertaking to improve the team features and game tactics. Considerably less effort has been dedicated, to the comprehension of football from the point of view of the foreseeing results in light of the way that various components which must be considered may not constitute quantitatively. The result of the football matches may rely upon different highlights like an immaterial factor of haphazardness and different angles, for instance, mind-set of players, weather or effect of media and fans than just relying on the team performance which is the reason why the games forecasts have dependably been testing.

Predictive analysis is an established approach that is intently connected with sports. Many individuals in the scholarly world and industry have handled the issue of football coordinate expectation, attributable to its fascinating nature as well as its economic significance. Numerous endeavours has been made in order to foresee football matches result and choosing critical

variables in football. The mode of exhibiting football data has emerged to being increasingly noticeable in the latest years and distinctive calculations have been implemented to envision football matches results in various investigations. Prediction is exceptionally valuable in assisting supervisors and clubs settle on the right choice to win leagues and tournaments. Gauging football is involved aftereffect of a match and score, which can be utilized by the bookmakers to place bets. Predictions can also be utilised by team members to acknowledge and beat the odds from inculcating different strategies. Team performance analysis is valuable from a strategic point of view; giving the chance to improve one's understanding about exhibitions with fruitful results.

The data upheaval has cleared through professional game and a progress from customary, subjective testing strategies to the latest information driven investigations of games execution has pursued. Our present methodology analyses data in order to anticipate the outcomes utilizing the information recorded from the previous football matches for over a decade.

1.2 Project Outline

In this report we have demonstrated the the comparison of three machine learning algorithms i.e KNN, Linear Regression and Naive Bayes, to predict football match outcome.

The project includes the following steps:

1. Perform statistical analysis of the past data and develop a statistical model.
2. On the basis of statistical model we have developed a predictive model which is predicting the outcome of a match.
3. It will Predict the outcome as draw win or lose and help devise strategies that would increase the probability of winning and what will be the league tables like etc.
4. Compare the predictive accuracy of different machine learning algorithms.

2. Literature Review

In Literature, we find various studies over prediction Analysis using different approaches to obtain maximum optimisation. Using disparate algorithms for analysing a dataset to acquire maximum accuracy looks to be expanding. Fenton, Constantinou and Neil, 2013 [14] study the forecasts of football matches where the outcome can be win, draw or lose. Darwin and Dea. Harlili, 2016 determined the game prediction collecting the dataset from the video game and using Logistic regression to increase the prediction accuracy[6]. The studies involve using different techniques like Gaussian Naive Bayes, random forest(RF), Support vector machine, hidden Markov model, etc by analysing the goals scored at different seasons.(Ulmer and Fernandez(2013)[12]. With a lot more noteworthy measure of data, Parinaz and Sadat (2013) utilized information identified with physiology and football systems so as to break down the Football Club Barcelona in the Spanish Championship. In that particular work, the creators depicted a Bayesian network approach for football results forecast with the NETICA programming. The model considered just a single team to foresee the outcome of football matches.[13]

A. Joseph, Fenton and Neil work show the prediction analysis of football matches using Bayesian networks and various machine learning algorithms including Naive Bayes, decision tree, k-nearest neighbour implementation. The point was to perceive how the expert developed BN compares as far as both predictive accuracy and explanatory clearance for the features affecting the result of the matches under scrutiny.[8] Other works include the application of polynomial algorithm to analyse and define the outcomes of football matches. Rodrigo, Alessandro, Leandro, etc associated the machine learning algorithms and polynomial algorithm to receive a higher accuracy of upto 96% by distinguishing the relevant features.[4] Igiri(2015) studied the data in the English Championship related to football scores using support vector machine(SVM) [11]. There has been various other works on predicting football match results which involves the use of performance indicators[5] and using player attributes to determine the analysis[7]. Our report deals with comparison of various machine learning algorithms to predict football game outcomes.

3. Dataset

3.1 Sources of dataset

Dataset used in this report was downloaded from website: <http://www.football-data.co.uk>. [3] Football-Data is a free football betting portal providing historical results & odds to help football betting enthusiasts analyse many years of data quickly and efficiently to gain an edge over the bookmaker, unique in making available computer-ready data in Excel and CSV format for quantitative analysis, brings together all the latest results, tables and team stats, best football and betting news, wires, articles under one roof for the latest information about teams, players and transfers.

3.2 Description of dataset

A dataset is a collection of related sets of information that is composed of separate elements but can be manipulated as a unit by a computer. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows.

Data set is the unit to measure the information released in a public open data repository.

Several characteristics define a data set's structure and properties. These include the number and types of the attributes or variables, and various statistical measures applicable to them, such as standard deviation and kurtosis. [1] The values may be numbers, such as real numbers or integers, for example representing a person's height in centimeters, but may also be nominal data (i.e., not consisting of numerical values), for example representing a person's ethnicity. More generally, values may be of any of the kinds described as a level of measurement. For each variable, the values are normally all of the same kind. However, there may also be missing values, which must be indicated in some way.

In statistics, data sets usually come from actual observations obtained by sampling a statistical population, and each row corresponds to the observations on one element of that population. Data sets may further be generated by algorithms for the purpose of testing certain kinds of software. Some modern statistical analysis software such as SPSS still present their data in the classical data set fashion. If data is missing or suspicious an imputation method may be used to complete a data set.[2]

Dataset in our report here is used to refer to the result data of Spanish La Liga Primera Division , corresponding to football games, seasons from 2005 to 2019.

4. Data Analysis

4.1 Description of data pre-processing steps

Data preprocessing can include a number of steps which include cleaning the data set, instance selection, normalization and feature and attribute selection from a number of attributes. The final set that we have after preprocessing of data is the final training set that you use for the training of data and then perform the testing on it.

We filtered the data, there were a lot of attributes in the dataset we chose to perform the analysis on, but we only selected a number of attributes which are,

HS = Home Team Shots

AS = Away Team Shots

HST = Home Team Shots on Target

AST = Away Team Shots on Target

HHW = Home Team Hit Woodwork

AHW = Away Team Hit Woodwork

HC = Home Team Corners

AC = Away Team Corners

HF = Home Team Fouls Committed

AF = Away Team Fouls Committed

HFKC = Home Team Free Kicks Conceded

AFKC = Away Team Free Kicks Conceded

HO = Home Team Offsides

AO = Away Team Offsides

HY = Home Team Yellow Cards

AY = Away Team Yellow Cards

HR = Home Team Red Cards

AR = Away Team Red Cards

We have used Sklearn's preprocessing library to scale the independent variables and to encode the dependant variable into numeric values and figures. The code that we used for the preprocessing mentioned above is as follows,

- 1) For transforming the HomeTeam Variable,

```
#creating labelEncoder
```

```
le = preprocessing.LabelEncoder()
```

```
# Converting string labels into numbers.
```

```
team_encoded=le.fit_transform(dataset.loc[:, "HomeTeam"])
```

```
print(team_encoded)
```

- 2) For preprocessing of independent variable

```
X = preprocessing.scale(x_prime)
```

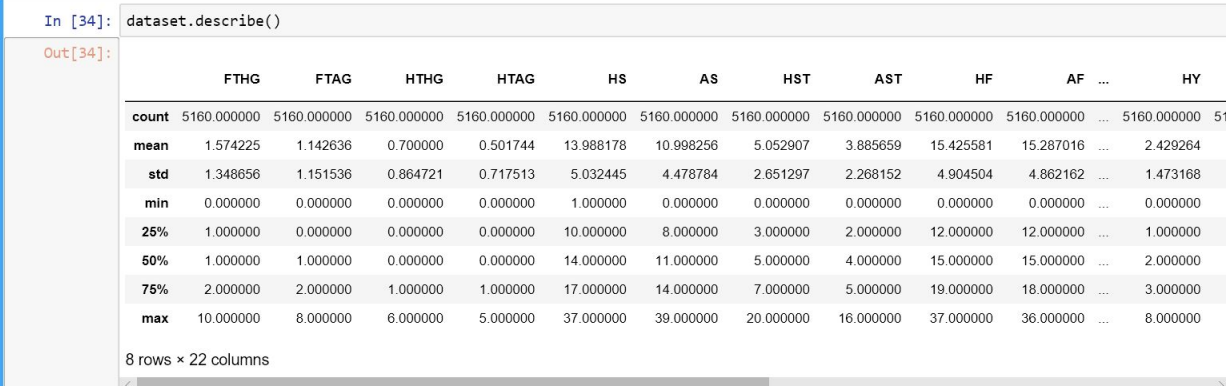
4.2 Methodology

We used the Matplot lib of python to visualize the data and analyze the pattern in it. We used two methods of data analysis:

(i) Quantitative

(ii) Qualitative

4.2.1 Quantitative:



In [34]: dataset.describe()

Out[34]:

	FTHG	FTAG	HTHG	HTAG	HS	AS	HST	AST	HF	AF	...	HY
count	5160.000000	5160.000000	5160.000000	5160.000000	5160.000000	5160.000000	5160.000000	5160.000000	5160.000000	5160.000000	...	5160.000000
mean	1.574225	1.142636	0.700000	0.501744	13.988178	10.998256	5.052907	3.885659	15.425581	15.287016	...	2.429264
std	1.348656	1.151536	0.864721	0.717513	5.032445	4.478784	2.651297	2.268152	4.904504	4.862162	...	1.473168
min	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000
25%	1.000000	0.000000	0.000000	0.000000	10.000000	8.000000	3.000000	2.000000	12.000000	12.000000	...	1.000000
50%	1.000000	1.000000	0.000000	0.000000	14.000000	11.000000	5.000000	4.000000	15.000000	15.000000	...	2.000000
75%	2.000000	2.000000	1.000000	1.000000	17.000000	14.000000	7.000000	5.000000	19.000000	18.000000	...	3.000000
max	10.000000	8.000000	6.000000	5.000000	37.000000	39.000000	20.000000	16.000000	37.000000	36.000000	...	8.000000

8 rows x 22 columns

As shown in the screenshot, it describes the important statistics of the data through which we can know the useful statistical attributes of the dataset. Count, mean, min and max are some of the statistical features that are represented in the describe function of the python's pandas dataframe.

4.2.2 Qualitative:

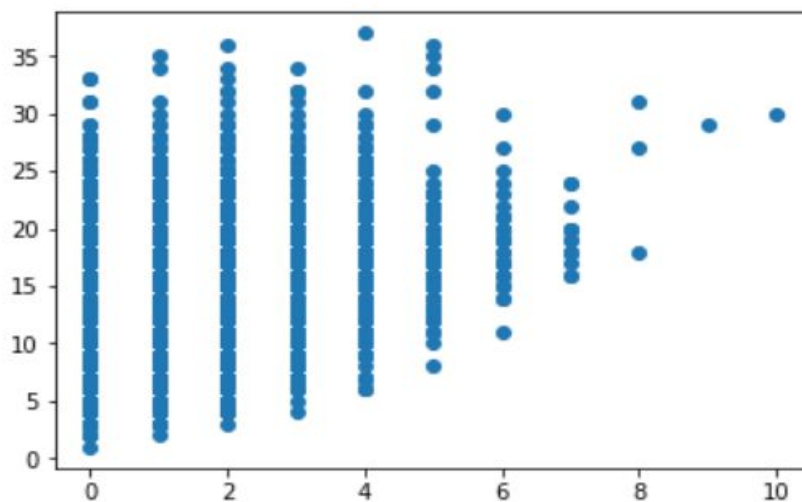
We used the qualitative analysis to become familiar with the dataset we used. It helps us categorize the dataset in terms of "code" as well, so it's easier to retrieve and access it. We tried to see the patterns in dataset and add significance to the data.

4.3 Data selection

The dataset we used is La Liga season from 2005 to 2019 (<http://www.football-data.co.uk>). We used this dataset because it contains a number of meaningful attributes, which even if a human is making a prediction about a football game, would use as well to predict the outcome of the game, so the dataset had very important and useful attributes.

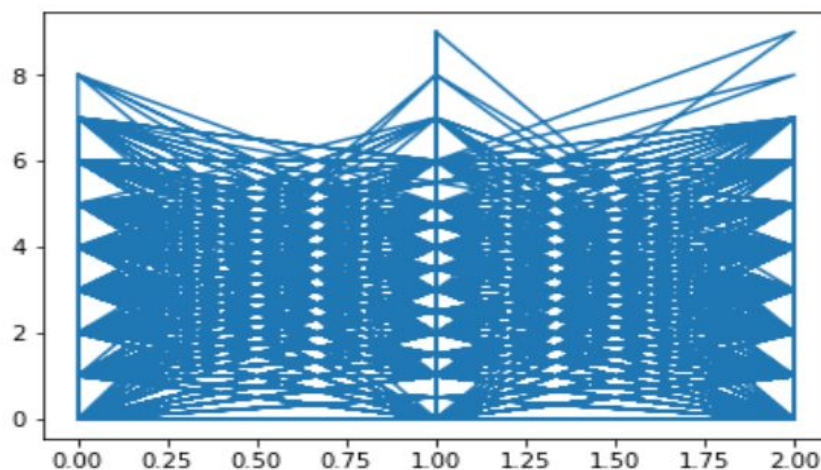
4.4 Things learned from exploring the data, including visualisations

```
In [35]: plt.scatter(dataset.iloc[:,3:4],dataset.iloc[:, 9:10])  
plt.show()
```

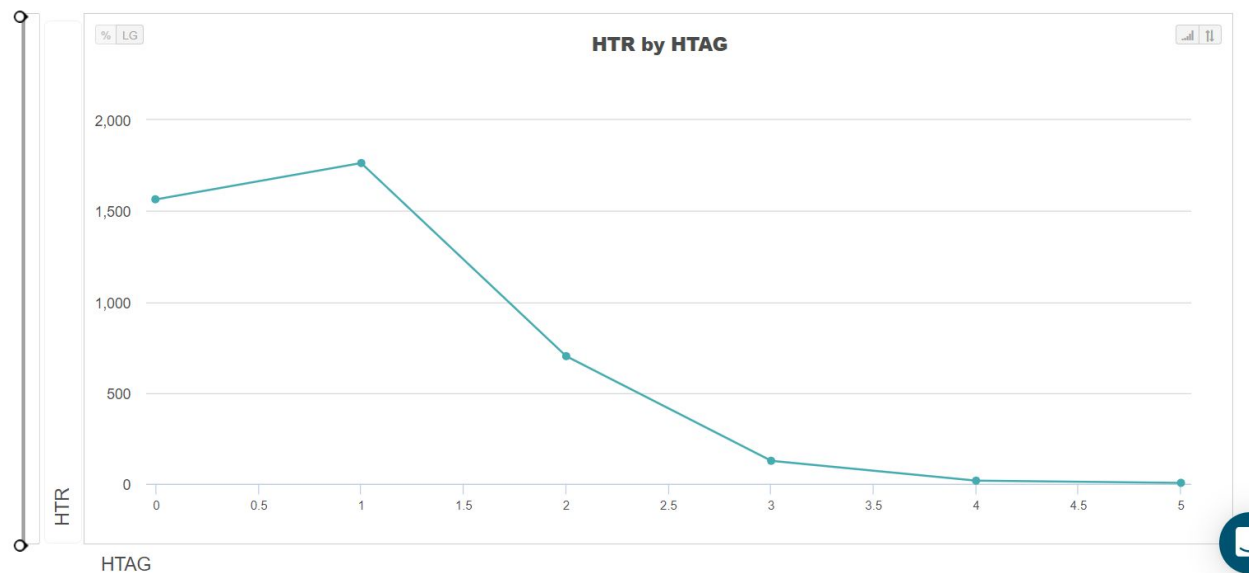


We used the full time home team goals and the home team's shot on target to make a scatter plot to show the relation between them, as the number of goals depend on the number of shots taken on target.

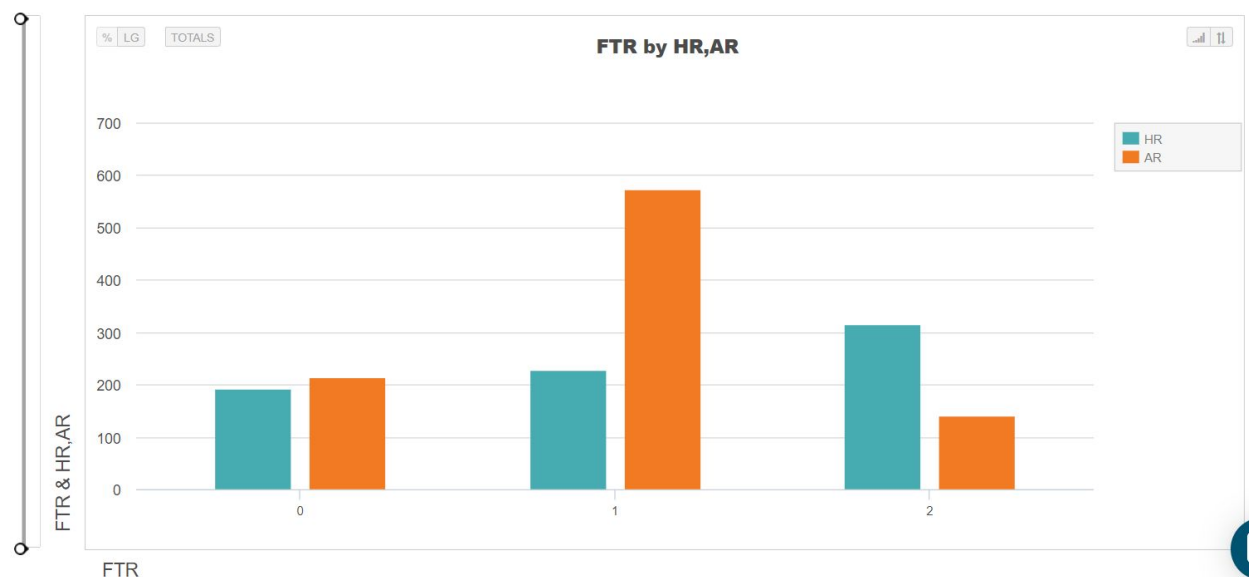
```
In [74]: plt.plot(dataset.iloc[:,5:6],dataset.iloc[:,18:19])  
plt.show()
```



0 is draw, 1 is home win, 2 is away win. This shows the number of red cards in relation to the chances of winning of home or away or a draw.



Shows the relation between the half time away goals and half time result.



The relation of Full time result (FTR) in correspondence to Away red cards (AR) and home red cards (HR).

What we learned from exploring the data is how it reacts to certain attributes in relation to the other. It also tells us how many records we have in our dataset, how many variables, what's the structure of the variables and how they form a relation between each other.

4.5 Justification for features used in analysis

The features that we used in the data analysis are as respected,

Home Team Shots (HS), Away Team Shots (AS), Home Team Shots on Target (HST),

Away Team Shots on Target (AST) the attributes shots and shots on target are one of the most important features that you need to predict the score or the outcome of winning a match, as the more number of shots on target the more chances of scoring a goal are there.

Home Team Hit Woodwork (HMW) and Away Team Hit Woodwork (AMW) are also a part of the shots that the player was able to take and it could've been converted into a goal but it hit the woodwork and the player missed, but it still depicts that a shot was taken on the target.

Home Team Corners (HC) and Away Team Corners (AC). Corner are an essential part of a football game, it has become a crucial way of creating the chances of scoring, and the coaches let the players practise different type of skilled corner kicks so that it can be converted into goal.

Home Team Fouls Committed (HF), Away Team Fouls Committed (AF), Home Team Yellow Cards (HY), Away Team Yellow Cards (AY), Home Team Red Cards (HR), Away Team Red Cards (AR). The attributes fouls committed are important in the sense that all the fouls committed are in accordance with the referees interpretation of the rule, and the angle the referee or the assistant referee had when the foul was committed, it affects the game in the sense that if a foul is committed by some team then the team loses possession and the other team has the ball, also the other team might get a free kick or a penalty in which case, the chances to score a goal increases. Same goes for the yellow and red cards as they can increase the chance of injuries to other players and give fouls away as well as the number of players of the other team can decrease if you get a red card.

Home Team Free Kicks Conceded (HKFC), Away Team Free Kicks Conceded (AKFC). Free kicks are important as they also allow goal scoring opportunities.

Home Team Offsides (HO) and Away Team Offsides (AO). It terminates the attacking runs of the player and it also protects the defenders by not giving attackers an unfair advantage.

5. Implementation

5.1 Tools, Libraries, Language

The techstack, libraries and tools that used in this project are mentioned below:

- Python (Language)[10]
- Jupyter (Editor)
- Pandas
- Numpy
- Matplotlib
- WEKA (Analytics Tool)
- Scikit-learn

5.2 Implementation

The algorithms we have used are KNN Classifier, Regression and Naive Bayes.

5.2.1 KNN Classifier:

If we assume a match between two teams and we want to predict win, lose or draw based on the score of the match using KNN classifier then that's how we will go about it. What KNN does is classify a new datapoint between one of the classes in consideration based on k(can be any integer value) nearest neighbours to the new data point. So basically we are using already classified instances as training data and certain attributes to plot the new datapoint on the graph and then we will calculate the distance to k nearest data points and the majority vote will decide which class is this new datapoint belongs to. The dataset contains a column named FTR(Full Time Result) which has three values:

0 - Draw, 1- Home win, 2- Away Win.

Now KNN can use these three as classes and easily can classify among them. Now, the deciding factor could be the distance calculation technique. Here I want to use Minkowski Distance (p-norm).

Minkowski Distance Formula:

$$D(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

The reason for this selection is that it is really flexible when we have different types of attributes. It is basically a generalization of Euclidean distance, so when we have numeric attributes we can set the value of p as p=2 and for categorical attributes the value can be set to p=0. When the value is set to 2, it will act as euclidean(for numeric attributes) and for value 0 it will work as Hamming(for categorical attributes).

5.2.2 Regression:

Regression can also be used for making score predictions. Although regression does not give discrete output, its continuous predicted values for score attribute can still be used to evaluate the prediction to be a home win, away win or draw. But for that we would need to define some limits on the continuous values as explained in the paper provided in reference.

Basically what regression does is to find the best fitted line that passes as closely as possible to all the data points on the graph. That is measured by the squared distance of each data point from the line i.e deviation. The less the sum of deviation of all the points is the better the line is fitted, but then we need to take care that in pursuit of accuracy we do not end up creating an overfitted model.

The type of regression, either linear or polynomial, will then be decided based on the fitting of the line i.e what value of polynomial is giving us the best result. Which is increased test accuracy rather than just the train accuracy to avoid overfitting.

Regression will show us the impact of attributes on the prediction. So from the attributes which has the most impact we can predict the dependant variable or attribute using the equation of that regression line we have fitted.

5.2.3 Naive Bayes:

Naive Bayes takes the probability of occurrence of an event(dependent variable) into account based on the probability of occurrence of another event(independent variable). Now for a football match, using training data or past data we can calculate the probability values of certain attributes like messi playing etc, and then multiply all the probabilities to predict the probability of win or lose.

Formula:

$$P(X|Y) = P(Y|X) \cdot P(X) / P(Y)$$

where:

- $P(X)$ is the prior probability or marginal probability of X.
- $P(X|Y)$ is the posterior probability or conditional probability of X given Y .
- $P(Y |X)$ is the conditional probability of Y given X (the likelihood of data Y).
- $P(Y)$ is the prior probability or marginal probability of data Y (the evidence).

6. Testing/Results

6.1 Validation Approach

6.1.1 RMSE(Root Mean Square Error):

It is a measure of the difference in values predicted by a model and the actual values observed. It is essentially the square root of the average squared errors as can seen by the formula.

Formula:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (P_i - O_i)^2}{n}}$$

Where P = predicted value,

O = observed value.

6.1.2 MPE(Mean Percentage Error):

As the name suggests, it averages the percentage error; the difference in predicted and observed values.

Formula:

$$\text{MPE} = \frac{100\%}{n} \sum_{t=1}^n \frac{a_t - f_t}{a_t}$$

Where a_t is the actual value of the quantity being forecast, f_t is the forecast and n is the number of different times for which the variable is forecast. Because actual rather than absolute values of the forecast errors are used in the formula, positive and negative forecast errors can offset each other; as a result the formula can be used as a measure of the bias in the forecast.

A disadvantage of this measure is that it is undefined whenever a single actual value is zero.

6.2 Results

For the results we got confusion matrices and percentage accuracies which are as follows:

Confusion Matrices:

KNN:

```
In [44]: from sklearn.metrics import confusion_matrix  
print(confusion_matrix(y_test,y_pred))
```

```
[[162  32  97]  
 [ 60  35 145]  
 [ 56  34 411]]
```

Linear Regression:

```
In [84]: print(confusion_matrix(reg_y_test,rounded_pred))
```

```
[[ 0 237   3]  
 [ 2 497   2]  
 [ 0 264  27]]
```

Naive Bayes:

```
In [85]: print(confusion_matrix(y_test,nb_y_pred))
```

```
[[ 59   0 232]  
 [ 26   0 214]  
 [ 22   0 479]]
```

Accuracies in percentage:

1)KNN = 58.9%

2)Linear regression = 50.7%

3)Naive Bayes = 52.1%

By looking at confusion matrices and percentage accuracies, it's pretty evident that KNN outperformed the other two algorithms.

7. Conclusions

7.1 Possible extensions or business applications of your project

When the input data to an algorithm is too large to be processed and it is suspected to be redundant then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen, it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input. Feature extraction is performed on raw data prior to applying KNN algorithm on the transformed data in feature space.

The way similarity measurement is by creating a vector representation of the items, and then compare the vectors using an appropriate distance metric. KNN algorithm used in this report can be used in search applications where you are looking for “similar” items; that is, when your task is some form of “find items similar to this one”. It can also used for KNN classification, in the right kind of application. It’s easy to train, easy to use, and it’s easy to understand the results. KNN outputs the K nearest neighbours of the query from a dataset. KNN is “a non-parametric method used in classification or regression”. So industrial applications would be broadly based in these two areas. KNN is desirable in areas where there is even less knowledge of the data set than there would otherwise be. For example, a typical computer vision computation pipeline for face recognition using KNN including feature extraction and dimension reduction pre-processing steps”. If say you have a face detection algorithm and somebody just got the police sketch artist to sketch out what the perp looks like, you obviously DON’T want to

- show one picture for identification, or even
- show say 8 pictures all different except for one that is very much like the perp.

or even fingerprint detection. On TV, witnesses are fond of saying that “that is a one in a trillion chance of so-and-so having the same fingerprint (feature points)”. A KNN search will show the differences.

7.2 Future Improvements

We can consider more type of attributes and make the model more complex than it is right now e.g the formation of the team, if a specific player or the key players are playing and it affects the team’s performance, or e.g the no of penalties in the game and then try to improve its accuracy

further down the attributes. A generalized model can also be made such that it calculates the accuracy of any sports whose similar attributes are given to the model.

7.3 Challenges and Successes

The biggest challenge we had was to find a good dataset with meaningful attributes and preprocess it to make it suitable for modelling. Improving prediction accuracy was another tough task. We only managed to get 58.9% at most.

The success is that we overcame the hurdles and improved the accuracy a bit. Linear regression gave 40% accuracy earlier but now we have improved upto 50%.

8. Appendices

8.1 Code - commented Python scripts used to develop project

CODE:

```
from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import preprocessing

from sklearn import linear_model

from sklearn.model_selection import train_test_split

from sklearn import neighbors

dataset = pd.read_csv("/home/aizaz/Downloads/seasons_2005_2019_new.csv")

dataset.head()
```

```
plt.scatter(dataset.iloc[:,3:4],dataset.iloc[:, 9:10])

plt.show()

#creating labelEncoder

le = preprocessing.LabelEncoder()

# Converting string labels into numbers.

team_encoded=le.fit_transform(dataset.loc[:,"HomeTeam"])

print(team_encoded)

x_prime = dataset.iloc[:,[9,10,11,12,13,14,15,16,17,18,19,20]].values

y = dataset.iloc[:,5:6].values

x_prime

X = preprocessing.scale(x_prime)

x_train,x_test,y_train,y_test = train_test_split(X,y,test_size=0.20, random_state=4)

clf = neighbors.KNeighborsClassifier(n_neighbors=25)

clf.fit(x_train,y_train)

print(clf)

y_expect = y_test

y_pred = clf.predict(x_test)

print(metrics.classification_report(y_expect,y_pred))

from sklearn.metrics import accuracy_score

print('Accuracy Score:',accuracy_score(y_test,y_pred)*100,'%')

from sklearn.metrics import confusion_matrix

print(confusion_matrix(y_test,y_pred))
```

```
dataset.describe()

# Linear regression

reg = linear_model.LinearRegression()

reg_x_prime = dataset.iloc[:,[9,10,11,12,13,14,15,16,17,18,19,20]].values

reg_y = dataset.iloc[:,3:4].values

reg_x_train,reg_x_test,reg_y_train,reg_y_test = train_test_split(X,reg_y,test_size=0.20,
random_state=4)

reg.fit(reg_x_train,reg_y_train)

reg_y_expect = reg_y_test

reg_y_pred = reg.predict(reg_x_test)

reg_y_expect

rounded_pred = np.around(reg_y_pred)

rounded_pred

import numpy as np

np.mean((rounded_pred-reg_y_expect)**2)

print('Accuracy Score:',accuracy_score(reg_y_expect,rounded_pred)*100,'%')

from sklearn.naive_bayes import MultinomialNB

MultiNB = MultinomialNB()

MultiNB.fit(np.absolute(x_train),y_train)

print(MultiNB)

nb_y_pred = MultiNB.predict(x_test)

print('Accuracy Score:',accuracy_score(y_test,nb_y_pred)*100,'%')
```

8.2 Data – a sample of the dataset used

```
In [75]: dataset.head()
```

```
Out[75]:
```

	Div	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	HS	...	HF	AF	HC	AC	HY	AY	HR	AR
0	SP1	Alaves	Barcelona	0	0	0	0	0	0	5	...	17	19	3	7	0	1	0	0
1	SP1	Ath Bilbao	Sociedad	3	0	1	0	0	0	10	...	13	19	3	4	0	1	0	0
2	SP1	Valencia	Betis	1	0	1	0	0	0	9	...	18	14	8	5	2	3	0	0
3	SP1	Ath Madrid	Zaragoza	0	0	0	0	0	0	16	...	16	22	8	4	2	7	0	0
4	SP1	Cadiz	Real Madrid	1	2	2	0	1	2	15	...	19	25	8	8	2	2	0	0

5 rows × 23 columns

Div	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR	B365H	B365D	B365A	BWH	BWD	BWA	IWH	IWD	IWA	PSH	PSD	PSA	WHH	WHD	WHA	VCH	VCD	VCA	Bb1X2	BbMxH	BbAvH
SP1	Betis	Levante	0	3	A	0	1	A	22	6	8	4	10	10	5	3	0	2	0	0	1.66	4	5	1.7	3.7	5.25	1.75	3.6	4.9	1.69	4.19	5.11	1.67	3.9	4.75	1.67	4.2	5.2	40	1.75	1.68
SP1	Girona	Valladolid	0	0	D	0	0	D	13	2	1	1	21	20	3	2	1	1	0	0	1.75	3.6	5	1.75	3.5	5.25	1.8	3.6	4.5	1.8	3.7	4.99	1.75	3.6	4.6	1.8	3.7	4.8	40	1.85	1.78
SP1	Barcelona	Alaves	3	0	H	0	0	D	25	3	9	0	6	13	7	1	0	2	0	0	1.11	10	21	1.11	10	20	1.12	9	20	1.11	11.27	25.4	1.08	9	29	1.1	10.5	34	40	1.13	1.1
SP1	Celta	Espanol	1	1	D	0	1	A	12	14	2	5	13	14	8	7	3	2	0	0	1.85	3.5	4.5	1.91	3.4	4.25	1.9	3.5	4.1	1.93	3.64	4.27	1.91	3.5	4	1.93	3.5	4.4	38	1.97	1.9
SP1	Villarreal	Sociedad	1	2	A	1	1	D	16	8	7	4	16	10	4	6	2	3	0	0	2.04	3.4	3.8	2.05	3.3	3.9	2	3.4	3.8	2.06	3.51	3.91	2.05	3.3	3.6	2.05	3.5	3.9	40	2.11	2.03
SP1	Eibar	Huesca	1	2	A	0	2	A	18	8	6	6	12	13	7	0	1	1	0	0	1.66	3.75	5.5	1.7	3.7	5.25	1.7	3.75	5	1.72	3.9	5.26	1.73	3.6	4.75	1.7	3.8	5	40	1.76	1.7
SP1	Real Madrid	Getafe	2	0	H	1	0	H	10	4	3	1	11	27	3	0	1	7	0	0	1.2	7	13	1.18	7.25	16	1.2	6.5	15	1.2	7.36	17.47	1.22	6	13	1.2	7	13	39	1.24	1.21
SP1	Vallecano	Sevilla	1	4	A	0	3	A	13	17	2	8	6	15	2	6	1	0	0	0	3.25	3.6	2.14	3.5	3.5	2.1	3.5	3.4	2.1	3.46	3.74	2.13	3.3	3.7	2.05	3.4	3.6	2.1	40	3.53	3.38
SP1	Ath Bilbao	Leganes	2	1	H	1	1	D	17	12	5	2	12	13	6	2	4	5	0	0	1.75	3.3	5.5	1.78	3.5	5	1.85	3.5	4.4	1.79	3.54	5.46	1.8	3.4	4.75	1.8	3.4	5	40	1.85	1.78
SP1	Valencia	Ath Madrid	1	1	D	0	1	A	13	9	4	3	10	15	4	10	2	3	0	0	3	3.2	2.5	2.85	3.25	2.55	2.85	3.2	2.55	3.12	3.18	2.57	3	3.2	2.4	3	3.2	2.45	39	3.12	2.99
SP1	Getafe	Eibar	2	0	H	1	0	H	7	9	4	0	14	11	5	5	2	1	0	0	1.95	3.2	4.33	1.95	3.3	4.2	2	3.2	4.2	2.01	3.35	4.43	1.95	3.1	4.33	2	3.25	4.5	40	2.06	1.98
SP1	Leganes	Sociedad	2	2	D	0	2	A	18	7	5	6	12	9	5	0	1	2	0	0	3.5	3.3	2.1	3.5	3.2	2.2	3.6	3.2	2.15	3.66	3.36	2.2	3.4	3.25	2.15	3.7	3.3	2.2	40	3.76	3.57
SP1	Alaves	Betis	0	0	D	0	0	D	16	13	4	5	17	15	5	6	3	1	0	0	2.8	3.25	2.6	2.65	3.3	2.6	2.8	3.2	2.6	2.87	3.36	2.63	2.8	3.25	2.5	2.88	3.3	2.63	40	2.92	2.77
SP1	Ath Madrid	Vallecano	1	0	H	0	0	D	9	13	2	3	7	8	8	5	1	1	0	0	1.16	7	21	1.17	7.25	18	1.17	7.2	18.5	1.16	8.09	24.21	1.14	7	21	1.17	7.5	23	41	1.19	1.16
SP1	Valladolid	Barcelona	0	1	A	0	0	D	7	10	4	5	13	11	6	10	1	1	0	0	17	7.5	1.16	15	7	1.19	13	7	1.2	19.07	8.18	1.18	17	7.5	1.15	21	7.5	1.18	41	21	16.55
SP1	Espanol	Valencia	2	0	H	0	0	D	19	16	9	4	16	12	6	10	1	1	0	0	3.25	3.2	2.37	2.95	3.3	2.45	3.1	3.3	2.35	3.19	3.48	2.35	3.1	3.4	2.25	3.1	3.4	2.3	39	3.25	3.08
SP1	Girona	Real Madrid	1	4	A	1	1	D	10	20	5	8	13	11	4	7	2	3	0	0	8	5.5	1.36	9.25	4.75	1.36	7	5.2	1.4	8.99	5.42	1.37	7	5.5	1.36	8.5	5.5	1.33	41	9.25	8.29
SP1	Sevilla	Villarreal	0	0	D	0	0	D	19	14	2	6	13	14	4	5	3	2	0	0	1.72	4.2	4.33	1.72	3.9	4.75	1.75	3.9	4.45	1.79	3.98	4.7	1.75	3.9	4.33	1.73	3.9	4.6	41	1.83	1.76

9. References

1. Jan M. Żytkow, Jan Rauch (1999). *Principles of data mining and knowledge discovery*. ISBN 978-3-540-66490-1.
2. United Nations Statistical Commission; United Nations Economic Commission for Europe (2007). *Statistical Data Editing: Impact on Data Quality: Volume 3 of Statistical Data Editing, Conference of European Statisticians Statistical standards and studies*. United Nations Publications. p. 20. ISBN 9211169526. Retrieved 19 July 2015.
3. Data Files: Spain. <http://www.football-data.co.uk/spainm.php>
4. Rodrigo G. Martins a , c , AlessandroS. Martins a , LeandroA. Neves b , LucianoV. Lima c , Edna L. Flores c , Marcelo Z. do Nascimento, “Exploring polynomial classifier to predict match results in football championships”, 2017.
5. Christopher M. Younga,*, Wei Luob, Paul Gastina, Jacqueline Trana,c, Dan B. Dwyera, ” The relationship between match performance indicators and outcome in Australian Football”, 2019.

-
6. Darwin Prasetio, DRa. Harlili, M.Sc., "Predicting Football Match Results with Logistic Regression", 2016.
 7. Norbert Danisik, Peter Lacko, Michal Farkas, "Football Match Prediction using Players Attributes", 2018.
 8. A. Joseph, N.E. Fenton, M. Neil, "Predicting football results using Bayesian nets and other machine learning techniques", 2006.
 9. Rahul Baboota, Harleen Kaur, "Predictive analysis and modelling football results using machine learning approach for English Premier League", 2019.
 10. Wes McKinney, Python for data Analysis, Agile tool for real-world data, first edition, 2012.
 11. Igiri, Chinwe Peace¹; Nwachukwu, Enoch Okechukwu², "An Improved Prediction System for Football a Match Result", December 2014.
 12. Ben Ulmer, Matthew Fernandez, "Predicting Soccer Match Results in the English Premier League", 2013.
 13. Farzin Owramipur, Parinaz Eskandarian, and Faezeh Sadat Mozneb, "Football Result Prediction with Bayesian Network in Spanish League-Barcelona Team", 2013.
 14. Constantinou, A. C., Fenton, N. E., & Neil, M. (2013). "Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks. Knowledge-Based Systems"