FOOTBALL MATCH PREDICTION

Submitted by

RAJA HARIKESH N V : 16BLC1021

PRANAV. N : 16BLC1092

PRAVEEN KUMAR .R : 16BLC1135

J Component - Report

ECM2002 – Machine Learning Algorithms

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMPUTER ENGINEERING

October 2018

TABLE OF CONTENTS

1. ABSTRACT:

1. Every country in Europe has a club based football league. Each league has 20 teams which play two matches with every other team in their league - one at their home stadium and the opponent's home stadium. Each such match has three possible outcomes - the home team wins, the match ends in a draw or the visiting team wins.
2. English Premier League is the most popular football league, being watched by an estimated figure of 5 billion people across the globe. In a season, since each team plays two games against every other team, there are a total of 380 games.
3. Given such a format, it is natural that there are several online fantasy leagues, betting agencies and pundits who try to predict the outcome of each match.
4. In this project, an attempt has been made to find out the factors that affect the outcome of a match and also to predict the results of any fixture by using these factors.

2.1 DATASET DESCRIPTION:

Source of Data: http://www.football-data.co.uk/englandm.php.
16 seasons of data comprising 380 matches each season were manually downloaded in form of csv sheets

	Div	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Referee	HS	AS	HST
2	E0	08-08-2015	Bournemc	Aston Villa	0	1	A	0	0	D	M Clattenl	11	7	2
3	E0	08-08-2015	Chelsea	Swansea	2	2	D	2	1	H	M Oliver	11	18	3
4	E0	08-08-2015	Everton	Watford	2	2	D	0	1	A	M Jones	10	11	5
5	E0	08-08-2015	Leicester	Sunderlan	4	2	H	3	0	H	L Mason	19	10	8
6	E0	08-08-2015	Man Unite	Tottenhan	1	0	H	1	0	H	J Moss	9	9	1
7	E0	08-08-2015	Norwich	Crystal Pal	1	3	A	0	1	A	S Hooper	17	11	6
8	E0	09-08-2015	Arsenal	West Ham	0	2	A	0	1	A	M Atkinso	22	8	6
9	E0	09-08-2015	Newcastle	Southamp	2	2	D	1	1	D	C Pawson	9	15	4
10	E0	09-08-2015	Stoke	Liverpool	0	1	A	0	0	D	A Taylor	7	8	1
11	E0	10-08-2015	West Bron	Man City	0	3	A	0	2	A	M Dean	9	19	2
12	E0	14-08-2015	Aston Villa	Man Unite	0	1	A	0	1	A	M Dean	5	9	1
13	E0	15-08-2015	Southamp	Everton	0	3	A	0	2	A	M Oliver	17	10	4
14	E0	15-08-2015	Sunderlan	Norwich	1	3	A	0	2	A	K Friend	6	19	2
15	E0	15-08-2015	Swansea	Newcastle	2	0	H	1	0	H	M Jones	19	4	6
16	E0	15-08-2015	Tottenhan	Stoke	2	2	D	2	0	H	R Madley	13	16	7
17	E0	15-08-2015	Watford	West Bron	0	0	D	0	0	D	P Tierney	16	6	5
18	E0	15-08-2015	West Ham	Leicester	1	2	A	0	2	A	A Taylor	10	11	3
19	E0	16-08-2015	Crystal Pal	Arsenal	1	2	A	1	1	D	L Mason	11	20	4
20	E0	16-08-2015	Man City	Chelsea	3	0	H	1	0	H	M Atkinso	18	10	8
21	E0	17-08-2015	Liverpool	Bournemc	1	0	H	1	0	H	C Pawson	18	13	2
22	E0	22-08-2015	Crystal Pal	Aston Villa	2	1	H	0	0	D	K Stroud	16	11	6
23	E0	22-08-2015	Leicester	Tottenhan	1	1	D	0	0	D	M Atkinso	13	19	2
24	E0	22-08-2015	Man Unite	Newcastle	0	0	D	0	0	D	C Pawson	20	7	8
25	E0	22-08-2015	Norwich	Stoke	1	1	D	1	1	D	M Dean	21	6	7
26	E0	22-08-2015	Sunderlan	Swansea	1	1	D	0	1	A	N Swarbri	10	20	2
27	E0	22-08-2015	West Ham	Bournemc	3	4	A	0	2	A	J Moss	10	15	4
28	E0	23-08-2015	Everton	Man City	0	2	A	0	0	D	A Taylor	10	16	1
29	E0	23-08-2015	Watford	Southamp	0	0	D	0	0	D	A Marrine	13	14	0
30	E0	23-08-2015	West Bron	Chelsea	2	3	A	1	3	A	M Clattenl	15	15	6
31	E0	24-08-2015	Arsenal	Liverpool	0	0	D	0	0	D	M Oliver	19	15	5
32	E0	29-08-2015	Aston Villa	Sunderlan	2	2	D	2	1	H	R Madley	21	7	6
33	E0	29-08-2015	Bournemc	Leicester	1	1	D	1	0	H	N Swarbri	5	4	2
34	E0	29-08-2015	Chelsea	Crystal Pal	1	2	A	0	0	D	C Pawson	26	13	9
35	E0	29-08-2015	Liverpool	West Ham	0	3	A	0	2	A	K Friend	13	12	1
36	E0	29-08-2015	Man City	Watford	2	0	H	0	0	D	M Clattenl	18	7	5

	AST	HF	AF	HC	AC	HY	AY	HR	AR	B365H	B365D	B365A	BWH	BWD
2	3	13	13	6	3	3	4	0	0	2	3.6	4	2	3.3
3	10	15	16	4	8	1	3	1	0	1.36	5	11	1.4	4.75
4	5	7	13	8	2	1	2	0	0	1.7	3.9	5.5	1.7	3.5
5	5	13	17	6	3	2	4	0	0	1.95	3.5	4.33	2	3.3
6	4	12	12	1	2	2	3	0	0	1.65	4	6	1.65	4
7	7	14	20	1	4	1	0	0	0	2.55	3.3	3	2.6	3.2
8	4	12	9	5	4	1	3	0	0	1.29	6	12	1.28	5.75
9	5	9	12	6	6	2	4	0	0	2.88	3.3	2.7	2.8	3.1
10	3	9	16	3	5	2	4	0	0	3.4	3.4	2.3	3.2	3.4
11	7	12	9	6	6	4	1	0	0	5.75	4	1.67	4.75	4
12	2	14	10	3	5	2	2	0	0	5.5	3.8	1.73	5	3.75
13	4	11	10	9	9	4	2	0	0	1.95	3.6	4.33	1.95	3.5
14	6	7	7	6	6	1	2	0	0	2.5	3.3	3.1	2.5	3.1
15	2	11	8	4	4	2	1	0	1	1.95	3.6	4.33	1.95	3.5
16	7	15	11	4	3	2	2	0	0	1.8	3.75	5	1.8	3.7
17	0	13	10	2	4	1	2	0	0	2.3	3.4	3.4	2.3	3.2
18	6	11	12	8	4	1	3	1	0	2.25	3.5	3.4	2.25	3.2
19	7	14	12	6	6	1	1	0	0	5.25	3.9	1.73	5	3.75
20	3	19	13	5	1	4	2	0	0	2.1	3.5	3.75	2.1	3.4
21	2	11	18	6	8	1	4	0	0	1.45	4.75	8	1.48	4.75
22	2	12	15	6	4	3	3	0	0	1.85	3.6	4.75	1.85	3.5
23	6	7	14	2	7	1	4	0	0	2.63	3.6	2.75	2.65	3.3
24	0	15	11	11	5	2	2	0	0	1.36	5.25	10	1.36	5
25	1	16	5	9	0	4	1	0	0	2.3	3.3	3.5	2.35	3.2
26	9	16	9	3	4	4	2	0	0	4	3.4	2.1	3.5	3.25
27	7	8	11	5	4	0	0	1	0	2.38	3.4	3.2	2.35	3.4
28	9	5	7	8	7	0	3	0	0	5	4	1.73	4.4	3.75
29	5	10	11	5	2	0	2	0	0	2.8	3.3	2.75	2.65	3.25
30	5	9	12	8	7	2	1	0	1	6.5	4	1.62	5.25	3.8
31	8	2	13	7	8	1	4	0	0	1.75	4	5	1.75	3.75
32	3	10	14	5	4	2	1	0	0	1.91	3.5	4.75	1.91	3.6
33	2	9	19	4	4	2	4	0	0	2.2	3.6	3.4	2.2	3.3
34	6	9	15	7	7	1	1	0	0	1.36	5	11	1.36	4.75
35	5	17	11	5	7	3	1	1	1	1.36	5.5	9	1.36	5
36	0	9	12	9	3	1	2	0	0	1.2	8	15	1.2	7

2015-16

```
Notes for Football Data
```

All data is in csv format, ready for use within standard spreadsheet applications.

Div = League Division
Date = Match Date (dd/mm/yy)
HomeTeam = Home Team
AwayTeam = Away Team
FTHG and HG = Full Time Home Team Goals
FTAG and AG = Full Time Away Team Goals
FTR and Res = Full Time Result (H=Home Win, D=Draw, A=Away Win)
HTHG = Half Time Home Team Goals
HTAG = Half Time Away Team Goals
HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win)

Match Statistics (where available)
Attendance = Crowd Attendance
Referee = Match Referee
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
HBP = Home Team Bookings Points (10 = yellow, 25 = red)
ABP = Away Team Bookings Points (10 = yellow, 25 = red)

Key to 1X2 (match) betting odds data:

B365H = Bet365 home win odds
B365D = Bet365 draw odds
B365A = Bet365 away win odds
BSH = Blue Square home win odds
BSD = Blue Square draw odds
BSA = Blue Square away win odds
BWH = Bet&Win home win odds
BWD = Bet&Win draw odds
BWA = Bet&Win away win odds
GBH = Gamebookers home win odds
GBD = Gamebookers draw odds
GBA = Gamebookers away win odds
IWH = Interwetten home win odds
IWD = Interwetten draw odds
IWA = Interwetten away win odds
LBH = Ladbrokes home win odds
LBD = Ladbrokes draw odds
LBA = Ladbrokes away win odds
PSH and PH = Pinnacle home win odds
PSD and PD = Pinnacle draw odds

PSA and PA = Pinnacle away win odds
SOH = Sporting Odds home win odds
SOD = Sporting Odds draw odds
SOA = Sporting Odds away win odds
SBH = Sportingbet home win odds
SBD = Sportingbet draw odds
SBA = Sportingbet away win odds
SJH = Stan James home win odds
SJD = Stan James draw odds
SJA = Stan James away win odds
SYH = Stanleybet home win odds
SYD = Stanleybet draw odds
SYA = Stanleybet 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

Bb1X2 = Number of BetBrain bookmakers used to calculate match odds averages and maximums
BbMxH = Betbrain maximum home win odds
BbAvH = Betbrain average home win odds
BbMxD = Betbrain maximum draw odds
BbAvD = Betbrain average draw win odds
BbMxA = Betbrain maximum away win odds
BbAvA = Betbrain average away win odds

MaxH = Oddsportal maximum home win odds
MaxD = Oddsportal maximum draw win odds
MaxA = Oddsportal maximum away win odds
AvgH = Oddsportal average home win odds
AvgD = Oddsportal average draw win odds
AvgA = Oddsportal average away win odds

Key to total goals betting odds:

BbOU = Number of BetBrain bookmakers used to calculate over/under 2.5 goals (total goals) averages and maximums
BbMx>2.5 = Betbrain maximum over 2.5 goals
BbAv>2.5 = Betbrain average over 2.5 goals
BbMx<2.5 = Betbrain maximum under 2.5 goals
BbAv<2.5 = Betbrain average under 2.5 goals

GB>2.5 = Gamebookers over 2.5 goals
GB<2.5 = Gamebookers under 2.5 goals
B365>2.5 = Bet365 over 2.5 goals
B365<2.5 = Bet365 under 2.5 goals

Key to Asian handicap betting odds:

BbAH = Number of BetBrain bookmakers used to Asian handicap averages and maximums
BbAHh = Betbrain size of handicap (home team)
BbMxAHH = Betbrain maximum Asian handicap home team odds
BbAvAHH = Betbrain average Asian handicap home team odds
BbMxAHA = Betbrain maximum Asian handicap away team odds

```
BbAvAHA = Betbrain average Asian handicap away team odds

GBAHH = Gamebookers Asian handicap home team odds
GBAHA = Gamebookers Asian handicap away team odds
GBAH = Gamebookers size of handicap (home team)
LBAHH = Ladbrokes Asian handicap home team odds
LBAHA = Ladbrokes Asian handicap away team odds
LBAH = Ladbrokes size of handicap (home team)
B365AHH = Bet365 Asian handicap home team odds
B365AHA = Bet365 Asian handicap away team odds
B365AH = Bet365 size of handicap (home team)


Closing odds (last odds before match starts)

PSCH = Pinnacle closing home win odds
PSCD = Pinnacle closing draw odds
PSCA = Pinnacle closing away win odds
```

2.2 PREPROCESSING OF DATA:

Preprocessing of data was carried using various tools in Python.

TOOLS USED:

Pandas: Loading the data, data wrangling and manipulation, feature engineering.
Scikitlearn: Libraries for classifiers, model evaluation, metrics, cross-validation
Matplotlib: Data visualization

The main objectives in pre-processing are to extract the meaningful features relating to on-field football and to separate whole data into 2 for model training and testing purposes.

Training data: 15 seasons = 5700 matches
Test data: 1 season = 380 matches.

3.1 LOGIT:

```
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv", header = TRUE)
> dim(dataset)
[1] 6080   43
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv", header = TRUE)
> dim(test)
[1] 380  43
> library(clusterSim)
> dataset = data.frame(lapply(dataset, function(x) as.numeric(x)))
> dataset = data.Normalization (dataset,type="n4",normalization="column")
> test = data.frame(lapply(test, function(x) as.numeric(x)))
> test = data.Normalization (test,type="n4",normalization="column")
> glm.fits=glm(FTR ~ .,data=dataset,family=binomial)
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
> #glm.fits=glm(FTR ~ HomeTeam + AwayTeam + DiffPts+DiffLP+ DiffFormPts,da
ta=dataset,family=binomial)
> summary(glm.fits)

Call:
glm(formula = FTR ~ ., family = binomial, data = dataset)

Deviance Residuals: 
            Min                 1Q             Median                 3Q  
Max  
-0.0000134258884  -0.0000000210734   0.0000000210734   0.0000074003043   0
.0000114591770  

Coefficients: (2 not defined because of singularities)
                    Estimate       Std. Error  z value Pr(>|z|)
(Intercept)       24.777835096  41140.470790097  0.00060  0.99952
X                  2.832266444 583824.709939697  0.00000  1.00000
Date              -2.740236949 581339.323508215  0.00000  1.00000
HomeTeam          -0.113207924   5940.976712012 -0.00002  0.99998
AwayTeam          -0.101499952   5954.867029370 -0.00002  0.99999
FTHG            -430.091202898  32670.733741953 -0.01316  0.98950
FTAG             286.241820871  22348.509378651  0.01281  0.98978
HTGS              -0.134851137  39014.916733706  0.00000  1.00000
ATGS               0.142451016  37900.521012846  0.00000  1.00000
HTGC               0.255795467  34719.176369211  0.00001  0.99999
ATGC               0.246261645  33631.291961249  0.00001  0.99999
HTP                0.350616531  29109.485458441  0.00001  0.99999
ATP                0.187161381  28679.114968246  0.00001  0.99999
HM1                0.336088210  29845.014706247  0.00001  0.99999
HM2                0.292520241  12078.595036077  0.00002  0.99998
HM3                0.139292378   7096.816970097  0.00002  0.99998
HM4                0.028816661   5963.409513349  0.00000  1.00000
HM5               -0.040035893   5686.760236158 -0.00001  0.99999
AM1                1.077214689  29490.873608620  0.00004  0.99997
AM2                0.378240118  11943.243082862  0.00003  0.99997
AM3                0.098414596   6892.930782557  0.00001  0.99999
AM4                0.095207671   5984.354733471  0.00002  0.99999
AM5               -0.007577885   5772.348839886  0.00000  1.00000
HomeTeamLP         0.100898393   7202.061698739  0.00001  0.99999
AwayTeamLP        -0.017877180   7308.578551302  0.00000  1.00000
```

```
MW                  -0.391787353  31448.781734635 -0.00001  0.99999
HTFormPtsStr        -0.890941115  42546.813305330 -0.00002  0.99998
ATFormPtsStr        -1.813896947  42384.900918111 -0.00004  0.99997
HTFormPts           -0.022660524  22486.617562129  0.00000  1.00000
ATFormPts           -0.209716234  21733.339220800 -0.00001  0.99999
HTWinStreak3        -0.016772026  10169.383971325  0.00000  1.00000
HTWinStreak5         0.130261856  16493.923311396  0.00001  0.99999
HTLossStreak3        0.009183538   9406.991368763  0.00000  1.00000
HTLossStreak5        0.216805324  19411.604241708  0.00001  0.99999
ATWinStreak3         0.098532600  10023.389944058  0.00001  0.99999
ATWinStreak5        -0.293865560  17866.714842300 -0.00002  0.99999
ATLossStreak3       -0.067678888   9624.037581689 -0.00001  0.99999
ATLossStreak5        0.095418803  18946.938573252  0.00001  1.00000
HTGD                -0.020878120  53909.889815444  0.00000  1.00000
ATGD                -0.546292650  52997.739892298 -0.00001  0.99999
DiffPts                       NA               NA       NA       NA
DiffFormPts         -0.569780309  54118.336747709 -0.00001  0.99999
DiffLP                        NA               NA       NA       NA

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8395.6292532105308  on 6079  degrees of freedom
Residual deviance:    0.0000002440788  on 6039  degrees of freedom
AIC: 82

Number of Fisher Scoring iterations: 25

> glm.probs=predict(glm.fits,test,type="response")
Warning message:
In predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type ==  :
  prediction from a rank-deficient fit may be misleading
> glm.probs = ifelse(glm.probs>0.5,1,0)
> table(glm.probs,test[,7])

glm.probs   0   1
        0 157  49
        1   0 174
> mean(glm.probs==test[,7])
[1] 0.8710526316
```

3.2 LDA:

```
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv")
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv")
> library(MASS)
> #lda.fit = lda(FTR ~ .,data=dataset)
> lda.fit = lda(FTR ~ HomeTeam + AwayTeam + DiffPts+DiffLP+ DiffFormPts,da
ta=dataset)
> summary(lda.fit)
        Length Class  Mode
prior     2    -none- numeric
counts    2    -none- numeric
means   170    -none- numeric
scaling  85    -none- numeric
lev       2    -none- character
svd       1    -none- numeric
N         1    -none- numeric
call      3    -none- call
```

```
terms      3     terms   call
xlevels    2     -none- list
> lda.pred=predict(lda.fit, test[-7,])
> names(lda.pred)
[1] "class"     "posterior" "x"
> lda.class=predict(lda.fit,test)$class
> test[,7]
  [1] NH NH NH H  H  NH NH NH NH NH NH NH NH H  NH NH NH NH H  H  H  NH NH
NH NH NH NH NH NH NH NH NH NH NH H
 [36] NH NH NH H  H  H  NH H  H  H  H  NH H  NH H  NH H  H  NH NH NH NH NH
NH H  NH H  H  NH H  H  H  NH NH NH
 [71] NH NH NH H  H  NH NH H  NH NH H  NH NH H  NH NH NH H  H  NH H  NH H
NH NH H  NH NH NH H  NH NH H  NH NH
[106] H  NH H  H  H  NH H  H  H  H  NH NH NH NH NH H  H  NH NH NH NH NH H
H  NH NH NH H  NH H  H  H  NH NH NH
[141] H  NH NH NH H  NH H  NH H  NH H  H  H  NH NH NH NH NH NH H  H  NH NH
NH NH NH NH NH H  H  NH NH NH H  H
[176] NH H  H  H  H  H  NH NH NH H  NH H  H  NH NH H  NH H  H  H  NH H  H
NH NH H  NH NH NH NH NH H  H  NH NH
[211] NH H  NH H  H  H  H  NH NH H  NH H  NH NH NH H  NH NH NH NH NH NH H
H  NH NH NH H  H  NH H  NH NH H  H
[246] NH NH H  NH NH NH H  NH NH NH H  NH H  NH NH H  NH H  NH H  H  H  H
NH H  NH NH NH NH H  H  H  H  NH NH
[281] H  NH NH H  NH NH NH H  H  NH NH NH H  NH NH NH H  NH NH NH NH H  H
H  NH NH NH H  NH NH NH H  H  NH H
[316] H  H  H  NH NH H  NH H  NH NH NH H  H  NH NH NH NH NH NH NH H  H  H
H  NH NH NH H  H  NH NH H  H  H  NH
[351] H  NH NH H  H  NH NH NH H  H  NH H  NH H  NH NH H  NH H  H  H  NH H
H  H  H  NH NH NH H
Levels: H NH
> table(lda.class,test[,7])

lda.class   H  NH
       H   75  60
       NH  82 163
> mean(lda.class==test[,7])
[1] 0.6263157895
```

3.3 QDA:

```
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv")
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv")
> library(MASS)
> qda.fit = qda(FTR ~ HomeTeam + AwayTeam + DiffPts+DiffLP+ DiffFormPts,da
ta=dataset)
> summary(qda.fit)
        Length Class  Mode
prior       2  -none- numeric
counts      2  -none- numeric
means     170  -none- numeric
scaling 14450  -none- numeric
ldet        2  -none- numeric
lev         2  -none- character
N           1  -none- numeric
call        3  -none- call
terms       3  terms  call
```

```
xlevels     2  -none- list
> qda.class=predict(qda.fit,test)$class
> table(qda.class,test[,7])

qda.class   H  NH
       H   88  86
       NH  69 137
> mean(qda.class==test[,7])
[1] 0.5921052632
```

3.4 KNN:

```
> library(class)
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv")
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv")
> attach(dataset)

> library(clusterSim)
> dataset = data.frame(lapply(dataset, function(x) as.numeric(x)))
> dataset = data.Normalization (dataset,type="n4",normalization="column")
> #
> test = data.frame(lapply(test, function(x) as.numeric(x)))
> test = data.Normalization (test,type="n4",normalization="column")
> train.X=cbind(dataset[])
> attach(test)

> test.X=cbind(test[])
> train.FTR=dataset[,7]
> set.seed(1)
> knn.pred=knn(train.X,test.X,train.FTR,k=1)
> table(knn.pred,test[,7])

knn.pred   0   1
       0 157   0
       1   0 223
> mean(knn.pred==test[,7])
[1] 1
> knn.pred=knn(train.X,test.X,train.FTR,k=100)
> mean(knn.pred==test[,7])
[1] 0.9947368421
> knn.pred=knn(train.X,test.X,train.FTR,k=200)
> mean(knn.pred==test[,7])
[1] 0.9973684211
> knn.pred=knn(train.X,test.X,train.FTR,k=250)
> mean(knn.pred==test[,7])
[1] 1
> knn.pred=knn(train.X,test.X,train.FTR,k=300)
> mean(knn.pred==test[,7])
[1] 1
```

3.5 NAÏVE BAYES:

```
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv")
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv")
> library('varhandle')
> library('e1071',warn.conflicts=FALSE)
> naive_bayes_model<-naiveBayes(FTR ~ ., data = dataset)
> naive_bayes_predictions<-predict(naive_bayes_model, newdata=test)
> naive_bayes_predictions
  [1] NH NH NH H  NH NH NH NH NH NH NH NH NH H  NH NH NH NH H  H  H  H  H 
NH NH NH NH NH NH NH NH NH NH NH H 
 [36] NH NH NH H  NH H  NH H  H  H  NH NH H  NH H  NH H  NH H  NH NH NH H 
NH H  NH H  H  NH H  NH NH H  NH NH
 [71] NH NH NH H  H  NH NH H  NH NH H  NH NH H  H  H  NH H  H  NH H  NH H 
NH H  H  NH NH H  NH NH NH H  NH NH
[106] NH NH H  H  H  NH H  H  NH H  NH NH H  NH H  H  H  NH NH H  H  NH NH
H  H  NH NH H  NH H  NH H  NH H  NH
[141] H  NH H  H  NH NH H  NH NH NH NH H  H  NH NH NH NH H  H  H  H  NH H 
H  NH NH NH NH H  H  NH NH H  NH H 
[176] NH NH H  NH H  H  H  NH NH H  H  H  H  NH NH H  H  H  NH NH NH NH H 
NH NH NH NH NH H  NH H  H  H  NH H 
[211] NH H  H  H  NH H  H  NH NH H  NH H  H  NH NH H  H  NH H  NH H  NH H 
H  NH NH NH H  H  NH H  H  NH NH H 
[246] NH H  H  NH NH NH H  NH H  NH NH NH H  NH NH H  NH H  H  H  H  H  H 
NH NH H  NH NH H  H  H  NH NH H  NH
[281] H  NH H  H  H  NH NH NH H  NH NH NH H  H  NH NH H  NH H  NH NH NH H 
H  NH NH H  H  H  NH H  H  H  NH H 
[316] H  H  NH NH H  H  NH H  H  NH NH H  H  NH H  H  NH H  NH NH H  H  H 
H  NH NH H  H  H  NH H  H  H  NH H 
[351] H  NH NH NH NH NH NH NH H  H  NH NH NH H  NH NH NH H  H  H  H  NH H 
H  H  NH NH NH NH H 
Levels: H NH
> table(naive_bayes_predictions,test[,7])

naive_bayes_predictions   H  NH
                     H  121  56
                     NH  36 167
> mean(naive_bayes_predictions==test[,7])
[1] 0.7578947368
```

4.1 SVM:

```
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv")
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv")
> library("e1071")
> library(caret)
> options(repos = c(CRAN = "http://cran.rstudio.com"))
> set.seed(319)
> ind <- sample(2, nrow(dataset), replace = TRUE, prob = c(0.90,0.10))
> training = dataset[ind==1,]
> testing = dataset[ind==2,]
> dim(training)
[1] 5449   43
> dim(testing)
[1] 631  43
> svm_model1 <- svm(FTR ~ .,kernel = "radial", data=training)
> summary(svm_model1)

Call:
svm(formula = FTR ~ ., data = training, kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.0004103406

Number of Support Vectors:  3265

 ( 1632 1633 )


Number of Classes:  2

Levels:
 H NH



> pred1 <- predict(svm_model1, newdata = testing)
> confusionMatrix(pred1, testing$FTR )
Confusion Matrix and Statistics

          Reference
Prediction   H  NH
        H  269   8
        NH  25 329

               Accuracy : 0.9477
                 95% CI : (0.9273, 0.9637)
    No Information Rate : 0.5341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8945
 Mcnemar's Test P-Value : 0.005349
```

```
            Sensitivity : 0.9150
            Specificity : 0.9763
         Pos Pred Value : 0.9711
         Neg Pred Value : 0.9294
             Prevalence : 0.4659
         Detection Rate : 0.4263
   Detection Prevalence : 0.4390
      Balanced Accuracy : 0.9456

       'Positive' Class : H

> svm_model2 <- svm(FTR ~ .,kernel = "linear", data=training)
> summary(svm_model2)

Call:
svm(formula = FTR ~ ., data = training, kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.0004103406

Number of Support Vectors:  1512

 ( 724 788 )


Number of Classes:  2

Levels:
 H NH



> pred2 <- predict(svm_model2, newdata = testing)
> confusionMatrix(pred2, testing$FTR )
Confusion Matrix and Statistics

          Reference
Prediction   H  NH
        H  294   0
        NH   0 337

               Accuracy : 1
                 95% CI : (0.9942, 1)
    No Information Rate : 0.5341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.4659
         Detection Rate : 0.4659
   Detection Prevalence : 0.4659
      Balanced Accuracy : 1.0000
```

```
       'Positive' Class : H

> svm_model3 <- svm(FTR ~ .,kernel = "polynomial",degree = 3, data=trainin
g)
> summary(svm_model3)

Call:
svm(formula = FTR ~ ., data = training, kernel = "polynomial", degree = 3)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  1
     degree:  3
      gamma:  0.0004103406
     coef.0:  0

Number of Support Vectors:  5044

 ( 2522 2522 )


Number of Classes:  2

Levels:
 H NH



> pred3 <- predict(svm_model3, newdata = testing)
> confusionMatrix(pred3, testing$FTR )
Confusion Matrix and Statistics

          Reference
Prediction   H  NH
        H    0   0
        NH 294 337

               Accuracy : 0.5341
                 95% CI : (0.4943, 0.5735)
    No Information Rate : 0.5341
    P-Value [Acc > NIR] : 0.5163

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.5341
             Prevalence : 0.4659
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : H

> svm_model4 <- svm(FTR ~ .,kernel ="polynomial",degree = 5, data=training
)
> summary(svm_model4)
```

```
Call:
svm(formula = FTR ~ ., data = training, kernel = "polynomial", degree = 5)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  1
     degree:  5
      gamma:  0.0004103406
     coef.0:  0

Number of Support Vectors:  5044

 ( 2522 2522 )


Number of Classes:  2

Levels:
 H NH



> pred4 <- predict(svm_model4, newdata = testing)
> confusionMatrix(pred4, testing$FTR )
Confusion Matrix and Statistics

          Reference
Prediction   H  NH
        H    0   0
        NH 294 337

               Accuracy : 0.5341
                 95% CI : (0.4943, 0.5735)
    No Information Rate : 0.5341
    P-Value [Acc > NIR] : 0.5163

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.5341
             Prevalence : 0.4659
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : H
```

4.2 NEURAL NETWORKS:

```
> dataset = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-an
d-Prediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/
final_dataset.csv")
> test = read.csv("D:/SEMESTER V/B1_ECM2002/J/Football-Data-Analysis-and-P
rediction-master/Football-Data-Analysis-and-Prediction-master/Datasets/tes
t.csv")
> #Pre-Processing
> str(dataset)
'data.frame':	6080 obs. of  43 variables:
 $ X             : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Date          : Factor w/ 1592 levels "2000-08-19","2000-08-20",..: 1 1
1 1 1 1 1 1 1 2 ...
 $ HomeTeam      : Factor w/ 42 levels "Arsenal","Aston Villa",..: 11 12 13
15 20 21 22 35 37 24 ...
 $ AwayTeam      : Factor w/ 42 levels "Arsenal","Aston Villa",..: 23 40 26
33 16 2 8 1 19 27 ...
 $ FTHG          : int  4 4 1 2 2 0 1 1 3 2 ...
 $ FTAG          : int  0 2 3 2 0 0 0 0 1 0 ...
 $ FTR           : Factor w/ 2 levels "H","NH": 1 1 2 2 1 2 1 1 1 1 ...
 $ HTGS          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ATGS          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTGC          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ATGC          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTP           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ATP           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ HM1           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ HM2           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ HM3           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ HM4           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ HM5           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ AM1           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ AM2           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ AM3           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ AM4           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ AM5           : Factor w/ 4 levels "D","L","M","W": 3 3 3 3 3 3 3 3 3 3
...
 $ HomeTeamLP    : num  18 5 14 16 3 8 4 7 10 1 ...
 $ AwayTeamLP    : num  18 9 12 15 13 6 17 2 18 11 ...
 $ MW            : num  1 1 1 1 1 1 1 1 1 1 ...
 $ HTFormPtsStr : Factor w/ 349 levels "DDDDD","DDDDL",..: 230 230 230 230
230 230 230 230 230 230 ...
 $ ATFormPtsStr : Factor w/ 359 levels "DDDDD","DDDDL",..: 241 241 241 241
241 241 241 241 241 241 ...
 $ HTFormPts     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ATFormPts     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTWinStreak3 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTWinStreak5 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTLossStreak3: int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTLossStreak5: int  0 0 0 0 0 0 0 0 0 0 ...
 $ ATWinStreak3 : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
 $ ATWinStreak5 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ATLossStreak3: int  0 0 0 0 0 0 0 0 0 0 ...
 $ ATLossStreak5: int  0 0 0 0 0 0 0 0 0 0 ...
 $ HTGD         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ATGD         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DiffPts      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DiffFormPts  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DiffLP       : num  0 -4 2 1 -10 2 -13 5 -8 -10 ...
> library(clusterSim)
> dataset = data.frame(lapply(dataset, function(x) as.numeric(x)))
> dataset = data.Normalization (dataset,type="n4",normalization="column")
> test = data.frame(lapply(test, function(x) as.numeric(x)))
> test = data.Normalization (test,type="n4",normalization="column")
> # Neural Networks
> library(neuralnet)
> set.seed(319)
> # 5 neurons hidden layer
> n = neuralnet(FTR ~ X + Date + HomeTeam + AwayTeam + FTHG + FTAG + HTGS
+ ATGS +
+                 HTGC + ATGC + HTP + ATP + HM1 + HM2 + HM3 + HM4 + HM5 +
AM1 +
+                 AM2 + AM3 + AM4 + AM5 + HomeTeamLP + AwayTeamLP + MW + H
TFormPtsStr +
+                 ATFormPtsStr + HTFormPts + ATFormPts + HTWinStreak3 + HT
WinStreak5 +
+                 HTLossStreak3 + HTLossStreak5 + ATWinStreak3 + ATWinStre
ak5 +
+                 ATLossStreak3 + ATLossStreak5 + HTGD + ATGD + DiffPts +
DiffFormPts +
+                 DiffLP,data = dataset,hidden = 5,err.fct = "ce",linear.o
utput = FALSE)
> # Prediction
> output <- compute(n, dataset[,-7])
> head(output$net.result)
                     [,1]
1 0.0000000000005496020571
2 0.0000000000094983300103
3 1.0000000000000000000000
4 0.9999998434028086746750
5 0.0000000000065638662296
6 0.9999997957432051887140
> head(dataset[1,])
  X Date     HomeTeam     AwayTeam          FTHG FTAG FTR HTGS ATGS HTGC ATG
C HTP ATP           HM1          HM2          HM3
1 0     0 0.243902439 0.5365853659 0.4444444444    0    0    0    0    0
0   0    0 0.6666666667 0.6666666667 0.6666666667
           HM4          HM5          AM1          AM2          AM3
AM4          AM5 HomeTeamLP AwayTeamLP MW
1 0.6666666667 0.6666666667 0.6666666667 0.6666666667 0.6666666667 0.66666
66667 0.6666666667          1          1  0
  HTFormPtsStr ATFormPtsStr HTFormPts ATFormPts HTWinStreak3 HTWinStreak5
HTLossStreak3 HTLossStreak5 ATWinStreak3
1  0.658045977 0.6703910615         0         0            0            0
0             0            0
  ATWinStreak5 ATLossStreak3 ATLossStreak5         HTGD         ATGD       D
iffPts DiffFormPts DiffLP
1            0             0             0 0.4285714286 0.487804878 0.5123
152709         0.5    0.5
> # Confusion Matrix & Misclassification Error - training data
> output <- compute(n, dataset[,-7])
> p1 <- output$net.result
> pred1 <- ifelse(p1>0.5, 1, 0)
```

```
> tab1 <- table(pred1, dataset$FTR)
> tab1

pred1    0    1
    0 2816    0
    1    0 3264
> sum(diag(tab1))/sum(tab1)
[1] 1
> # Confusion Matrix & Misclassification Error - testing data
> output <- compute(n, test[,-7])
> p2 <- output$net.result
> pred2 <- ifelse(p2>0.5, 1, 0)
> tab2 <- table(pred2, test$FTR)
> tab2

pred2   0   1
    0 157  51
    1   0 172
> sum(diag(tab2))/sum(tab2)
[1] 0.8657894737
```

5.1: COMPARISON OF ALL MODELS:

MODEL	TEST ACCURACY
LOGISTIC REGRESSION	87.10%
LDA	62.63%
QDA	59.21%
KNN	100%
NAÏVE BAYES	75.78%
SVM	100%
NEURAL NETWOKRS	86.57%

5.2: CONCLUSION:

- Home teams have a definite advantage over Away teams. On aggregate, home team win 46.65% matches compared to 27.72% matches won by the away teams.
- But Football is an unpredictable affair. The English Premier League is known for its famous upsets by bogey teams i.e. a consistently lower ranked team beating a higher ranked team.
- Despite all the facts, an attempt has been made by us to use statistical models to predict whether a home team will win or not given any match.
- Models like K Nearest Neighbours and Support Vector Machines with linear kernel have been able to predict correctly the 380 matches in test data, whereas other models have their test accuracy in the range of 60-90%.