Classification

Linus Fackler

Fernando Colman and Linus Fackler CS 4375.003 Linear Models Project

How do linear models work for classification and their strengths and weaknesses

Models for classifications include Logistic Regression, Deep Learning, as well as Naive Bayes. Logistic Regression gives the user interpretable results and is extendable to multiclass. On the other hand, overfitting can be controlled but still occurs quickly, once there are less observations than predictors. Deep Learning can classify other sources of data, such as audio and image data. A weakness, though, is the need for large amounts of data to fully train these algorithms. Naive Bayes algorithms are easy to implement and work better with data containing independent features. It requires less training data, compared to other algorithms A disadvantage of it is the chance of getting unrealistic estimations, once there is data in a set that isn't available. Advantages and disadvantages of Logistic Regression and Naive Bayes is further explained in part g. This program will work with both algorithms.

Dataset Citation: FIFA World Cup 2022, International soccer matches and team strengths (1993-2022), https://www.kaggle.com/datasets/brenda89/fifa-world-cup-2022

Load the data

```
matches <- read.csv("international_matches.csv", header=TRUE)
str(matches)
```

```
## 'data.frame':    23921 obs. of  25 variables:
##  $ date                          : chr  "1993-08-08" "1993-08-08" "1993-08-08" "1993-08-08" ...
##  $ home_team                     : chr  "Bolivia" "Brazil" "Ecuador" "Guinea" ...
##  $ away_team                     : chr  "Uruguay" "Mexico" "Venezuela" "Sierra Leone" ...
##  $ home_team_continent           : chr  "South America" "South America" "South America" "Africa" ...
##  $ away_team_continent           : chr  "South America" "North America" "South America" "Africa" ...
##  $ home_team_fifa_rank           : int  59 8 35 65 67 70 50 65 111 4 ...
##  $ away_team_fifa_rank           : int  22 14 94 86 5 19 102 86 9 3 ...
##  $ home_team_total_fifa_points   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ away_team_total_fifa_points   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ home_team_score               : int  3 1 5 1 1 0 2 4 0 1 ...
##  $ away_team_score               : int  1 1 0 0 3 1 0 0 7 2 ...
##  $ tournament                    : chr  "FIFA World Cup qualification" "Friendly" "FIFA World Cup qual
##  $ city                          : chr  "La Paz" "Maceió" "Quito" "Conakry" ...
##  $ country                       : chr  "Bolivia" "Brazil" "Ecuador" "Guinea" ...
##  $ neutral_location              : chr  "False" "False" "False" "False" ...
##  $ shoot_out                     : chr  "No" "No" "No" "No" ...
##  $ home_team_result              : chr  "Win" "Draw" "Win" "Win" ...
##  $ home_team_goalkeeper_score    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ away_team_goalkeeper_score    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ home_team_mean_defense_score : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ home_team_mean_offense_score : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ home_team_mean_midfield_score: num  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ away_team_mean_defense_score : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ away_team_mean_offense_score : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ away_team_mean_midfield_score: num  NA NA NA NA NA NA NA NA NA NA ...
```

Data cleaning

This is just to make the column names more readable and easier to type.

```
names(matches)[names(matches) == "home_team"] <- "home"
names(matches)[names(matches) == "away_team"] <- "away"
names(matches)[names(matches) == "home_team_continent"] <- "h_continent"
names(matches)[names(matches) == "away_team_continent"] <- "a_continent"
names(matches)[names(matches) == "home_team_fifa_rank"] <- "h_fifa"
names(matches)[names(matches) == "away_team_fifa_rank"] <- "a_fifa"
names(matches)[names(matches) == "home_team_total_fifa_points"] <- "h_fifa_points"
names(matches)[names(matches) == "away_team_total_fifa_points"] <- "a_fifa_points"
names(matches)[names(matches) == "home_team_score"] <- "h_score"
names(matches)[names(matches) == "away_team_score"] <- "a_score"
names(matches)[names(matches) == "home_team_result"] <- "h_result"
names(matches)[names(matches) == "home_team_goalkeeper_score"] <- "h_keeper_score"
names(matches)[names(matches) == "away_team_goalkeeper_score"] <- "a_keeper_score"
names(matches)[names(matches) == "home_team_mean_defense_score"] <- "h_def_score"
names(matches)[names(matches) == "away_team_mean_defense_score"] <- "a_def_score"
names(matches)[names(matches) == "home_team_mean_offense_score"] <- "h_off_score"
names(matches)[names(matches) == "away_team_mean_offense_score"] <- "a_off_score"
names(matches)[names(matches) == "home_team_mean_midfield_score"] <- "h_midfield_score"
names(matches)[names(matches) == "away_team_mean_midfield_score"] <- "a_midfield_score"
str(matches)
```

```
## 'data.frame':    23921 obs. of  25 variables:
##  $ date            : chr  "1993-08-08" "1993-08-08" "1993-08-08" "1993-08-08" ...
##  $ home            : chr  "Bolivia" "Brazil" "Ecuador" "Guinea" ...
##  $ away            : chr  "Uruguay" "Mexico" "Venezuela" "Sierra Leone" ...
##  $ h_continent     : chr  "South America" "South America" "South America" "Africa" ...
##  $ a_continent     : chr  "South America" "North America" "South America" "Africa" ...
##  $ h_fifa          : int  59 8 35 65 67 70 50 65 111 4 ...
##  $ a_fifa          : int  22 14 94 86 5 19 102 86 9 3 ...
##  $ h_fifa_points   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ a_fifa_points   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ h_score         : int  3 1 5 1 1 0 2 4 0 1 ...
##  $ a_score         : int  1 1 0 0 3 1 0 0 7 2 ...
##  $ tournament      : chr  "FIFA World Cup qualification" "Friendly" "FIFA World Cup qualification" "F
##  $ city            : chr  "La Paz" "Maceió" "Quito" "Conakry" ...
##  $ country         : chr  "Bolivia" "Brazil" "Ecuador" "Guinea" ...
##  $ neutral_location: chr  "False" "False" "False" "False" ...
##  $ shoot_out       : chr  "No" "No" "No" "No" ...
##  $ h_result        : chr  "Win" "Draw" "Win" "Win" ...
##  $ h_keeper_score  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ a_keeper_score  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ h_def_score     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ h_off_score     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ h_midfield_score: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ a_def_score     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ a_off_score     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ a_midfield_score: num  NA NA NA NA NA NA NA NA NA NA ...
```

Here, we are deleting all rows, where the game result is "draw", as we just care about win or lose

```
matches <- matches[!(matches$h_result=="Draw"),]
```

Handle missing values

This shows us how many NA's there are in each column, so we can prepare our data better before splitting it into train/test

```
sapply(matches, function(x) sum(is.na(x)==TRUE))
```

```
##             date             home             away      h_continent
##                0                0                0                0
##      a_continent           h_fifa           a_fifa    h_fifa_points
##                0                0                0                0
##    a_fifa_points          h_score          a_score       tournament
##                0                0                0                0
##             city          country neutral_location        shoot_out
##                0                0                0                0
##         h_result   h_keeper_score   a_keeper_score      h_def_score
##                0            12034            12388            12480
##      h_off_score h_midfield_score      a_def_score      a_off_score
##            11892            12173            12807            12268
## a_midfield_score
##            12514
```

We see that there is over 15000 rows with missing data in certain columns. Instead of deleting those, we will replace the NA's with the median.

```
matches$h_keeper_score[is.na(matches$h_keeper_score)] <- median(matches$h_keeper_score,na.rm=T)
matches$a_keeper_score[is.na(matches$a_keeper_score)] <- median(matches$a_keeper_score,na.rm=T)

matches$h_def_score[is.na(matches$h_def_score)] <- median(matches$h_def_score,na.rm=T)
matches$h_off_score[is.na(matches$h_off_score)] <- median(matches$h_off_score,na.rm=T)
matches$h_midfield_score[is.na(matches$h_midfield_score)] <- median(matches$h_midfield_score,na.rm=T)

matches$a_def_score[is.na(matches$a_def_score)] <- median(matches$a_def_score,na.rm=T)
matches$a_off_score[is.na(matches$a_off_score)] <- median(matches$a_off_score,na.rm=T)
matches$a_midfield_score[is.na(matches$a_midfield_score)] <- median(matches$a_midfield_score,na.rm=T)

sapply(matches, function(x) sum(is.na(x)==TRUE))
```

```
##             date             home             away      h_continent
##                0                0                0                0
##      a_continent           h_fifa           a_fifa    h_fifa_points
##                0                0                0                0
##    a_fifa_points          h_score          a_score       tournament
##                0                0                0                0
##             city          country neutral_location        shoot_out
##                0                0                0                0
##         h_result   h_keeper_score   a_keeper_score      h_def_score
##                0                0                0                0
##      h_off_score h_midfield_score      a_def_score      a_off_score
##                0                0                0                0
## a_midfield_score
##                0
```

Let's change "Win" to 2, "Draw" to 1, and "Lose" to 0

```
matches <- matches[,c(2,3, 6, 7, 8, 9, 10, 11, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25)]
matches$home <- factor(matches$home)
matches$away <- factor(matches$away)
matches$h_result <- factor(matches$h_result)
matches$neutral_location <- factor(matches$neutral_location)
matches$shoot_out <- factor(matches$shoot_out)
matches$h_fifa_points <- factor(matches$h_fifa_points)

str(matches)
```

```
## 'data.frame':    18532 obs. of  19 variables:
##  $ home            : Factor w/ 211 levels "Afghanistan",..: 25 60 83 149 150 211 83 69 183 12 ...
##  $ away            : Factor w/ 211 levels "Afghanistan",..: 201 206 168 9 44 67 168 143 184 37 ...
##  $ h_fifa          : int  59 35 65 67 70 50 65 111 4 52 ...
##  $ a_fifa          : int  22 94 86 5 19 102 86 9 3 46 ...
##  $ h_fifa_points   : Factor w/ 1652 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ a_fifa_points   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ h_score         : int  3 5 1 1 0 2 4 0 1 2 ...
##  $ a_score         : int  1 0 0 3 1 0 0 7 2 1 ...
##  $ neutral_location: Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
##  $ shoot_out       : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...
##  $ h_result        : Factor w/ 2 levels "Lose","Win": 2 2 2 1 1 2 2 1 1 2 ...
##  $ h_keeper_score  : num  75 75 75 75 75 75 75 75 75 75 ...
##  $ a_keeper_score  : num  74 74 74 74 74 74 74 74 74 74 ...
##  $ h_def_score     : num  75.2 75.2 75.2 75.2 75.2 75.2 75.2 75.2 75.2 75.2 ...
##  $ h_off_score     : num  75.7 75.7 75.7 75.7 75.7 75.7 75.7 75.7 75.7 75.7 ...
##  $ h_midfield_score: num  76.2 76.2 76.2 76.2 76.2 76.2 76.2 76.2 76.2 76.2 ...
##  $ a_def_score     : num  74.5 74.5 74.5 74.5 74.5 74.5 74.5 74.5 74.5 74.5 ...
##  $ a_off_score     : num  75.3 75.3 75.3 75.3 75.3 75.3 75.3 75.3 75.3 75.3 ...
##  $ a_midfield_score: num  75.5 75.5 75.5 75.5 75.5 75.5 75.5 75.5 75.5 75.5 ...
```

a. Divide into 80/20 train/test

```
set.seed(1234)
i <- sample(1:nrow(matches), 0.8*nrow(matches), replace=FALSE)
train <- matches[i,]
test <- matches[-i,]
```

b. Use at least 5 R functions for data exploration, using the training data

```
summary(train)
```

```
##            home               away           h_fifa           a_fifa
##  USA         :  206   Zambia     :  140   Min.   :  1.00   Min.   :  1.00
##  Mexico      :  199   Costa Rica:  136   1st Qu.: 32.00   1st Qu.: 37.00
##  Saudi Arabia:  185   Brazil     :  130   Median : 71.00   Median : 76.00
##  Japan       :  167   Mexico     :  130   Mean   : 78.36   Mean   : 83.15
##  Brazil      :  165   Sweden     :  127   3rd Qu.:117.00   3rd Qu.:123.00
##  Qatar       :  161   Jamaica    :  126   Max.   :211.00   Max.   :211.00
##  (Other)     :13742   (Other)    :14036
##  h_fifa_points  a_fifa_points       h_score          a_score
##  0      :8827   Min.   :   0.0   Min.   : 0.000   Min.   : 0.000
```

```
##  924    :  18   1st Qu.:   0.0   1st Qu.: 1.000   1st Qu.: 0.000
##  260    :  16   Median :   0.0   Median : 2.000   Median : 1.000
##  1174   :  15   Mean   : 315.2   Mean   : 1.833   Mean   : 1.138
##  389    :  14   3rd Qu.: 525.0   3rd Qu.: 3.000   3rd Qu.: 2.000
##  427    :  14   Max.   :2164.0   Max.   :31.000   Max.   :17.000
##  (Other):5921
##  neutral_location shoot_out   h_result    h_keeper_score  a_keeper_score
##  False:11106      No :14575   Lose:5448   Min.   :47.00   Min.   :47.00
##  True : 3719      Yes:  250   Win :9377   1st Qu.:75.00   1st Qu.:74.00
##                                           Median :75.00   Median :74.00
##                                           Mean   :75.04   Mean   :74.07
##                                           3rd Qu.:75.00   3rd Qu.:74.00
##                                           Max.   :97.00   Max.   :97.00
##
##   h_def_score     h_off_score    h_midfield_score  a_def_score
##  Min.   :52.80   Min.   :53.30   Min.   :54.20    Min.   :52.80
##  1st Qu.:75.20   1st Qu.:75.70   1st Qu.:76.20    1st Qu.:74.50
##  Median :75.20   Median :75.70   Median :76.20    Median :74.50
##  Mean   :75.15   Mean   :75.78   Mean   :76.13    Mean   :74.45
##  3rd Qu.:75.20   3rd Qu.:75.70   3rd Qu.:76.20    3rd Qu.:74.50
##  Max.   :91.80   Max.   :93.00   Max.   :93.20    Max.   :91.80
##
##   a_off_score    a_midfield_score
##  Min.   :53.30   Min.   :54.20
##  1st Qu.:75.30   1st Qu.:75.50
##  Median :75.30   Median :75.50
##  Mean   :75.33   Mean   :75.42
##  3rd Qu.:75.30   3rd Qu.:75.50
##  Max.   :93.00   Max.   :93.20
##
```

```
names(train)
```

```
##  [1] "home"             "away"             "h_fifa"           "a_fifa"
##  [5] "h_fifa_points"    "a_fifa_points"    "h_score"          "a_score"
##  [9] "neutral_location" "shoot_out"        "h_result"         "h_keeper_score"
## [13] "a_keeper_score"   "h_def_score"      "h_off_score"      "h_midfield_score"
## [17] "a_def_score"      "a_off_score"      "a_midfield_score"
```

```
head(train)
```

```
##               home    away h_fifa a_fifa h_fifa_points a_fifa_points h_score
## 9603       Morocco    Mali     36     66             0             0       0
## 10320      Germany Denmark      5     22             0             0       0
## 9225       Albania  Turkey     86     12             0             0       0
## 10410 South Africa    Chad     59    133             0             0       4
## 11844      Belgium Armenia     51    101             0             0       2
## 821         Poland  Israel     34     44             0             0       4
##       a_score neutral_location shoot_out h_result h_keeper_score a_keeper_score
## 9603        1             True        No     Lose             63             74
## 10320       1            False        No     Lose             88             78
## 9225        1            False        No     Lose             75             76
## 10410       0            False        No      Win             59             74
## 11844       0            False        No      Win             73             64
## 821         3            False        No      Win             75             74
##       h_def_score h_off_score h_midfield_score a_def_score a_off_score
```

```
## 9603         73.5        76.0             71.5        76.5        74.0
## 10320        85.5        85.3             86.0        79.2        80.0
## 9225         75.2        63.3             75.0        74.0        79.0
## 10410        66.0        77.0             73.8        74.5        75.3
## 11844        77.0        73.0             76.8        74.5        75.3
## 821          75.2        75.7             76.2        74.5        75.3
##       a_midfield_score
## 9603              77.0
## 10320             81.8
## 9225              79.8
## 10410             75.5
## 11844             75.5
## 821               75.5
```

```
tail(train)
```

```
##                  home       away h_fifa a_fifa h_fifa_points a_fifa_points
## 1021            Japan Costa Rica     36     71             0             0
## 2276            Kenya     Guinea    105     59             0             0
## 20724        Tanzania Cabo Verde    140     67          1089          1350
## 16136        Zimbabwe     Malawi    114    108           304           312
## 13859          Panama    Bolivia     68     98             0             0
## 16678 North Macedonia     Latvia     83    111           402           281
##       h_score a_score neutral_location shoot_out h_result h_keeper_score
## 1021        3       0            False        No      Win             75
## 2276        1       0            False        No      Win             75
## 20724       2       0            False        No      Win             75
## 16136       1       1             True       Yes      Win             75
## 13859       2       0            False        No      Win             68
## 16678       2       1            False        No      Win             66
##       a_keeper_score h_def_score h_off_score h_midfield_score a_def_score
## 1021              74        75.2        75.7             76.2        74.5
## 2276              74        75.2        75.7             76.2        74.5
## 20724             74        75.2        75.7             76.2        71.5
## 16136             74        62.8        69.7             64.5        74.5
## 13859             74        75.2        75.7             76.2        74.5
## 16678             71        65.8        71.0             67.2        74.5
##       a_off_score a_midfield_score
## 1021         75.3             75.5
## 2276         75.3             75.5
## 20724        77.0             74.8
## 16136        75.3             75.5
## 13859        75.3             75.5
## 16678        66.7             75.5
```

```
nrow(train)
```

```
## [1] 14825
```

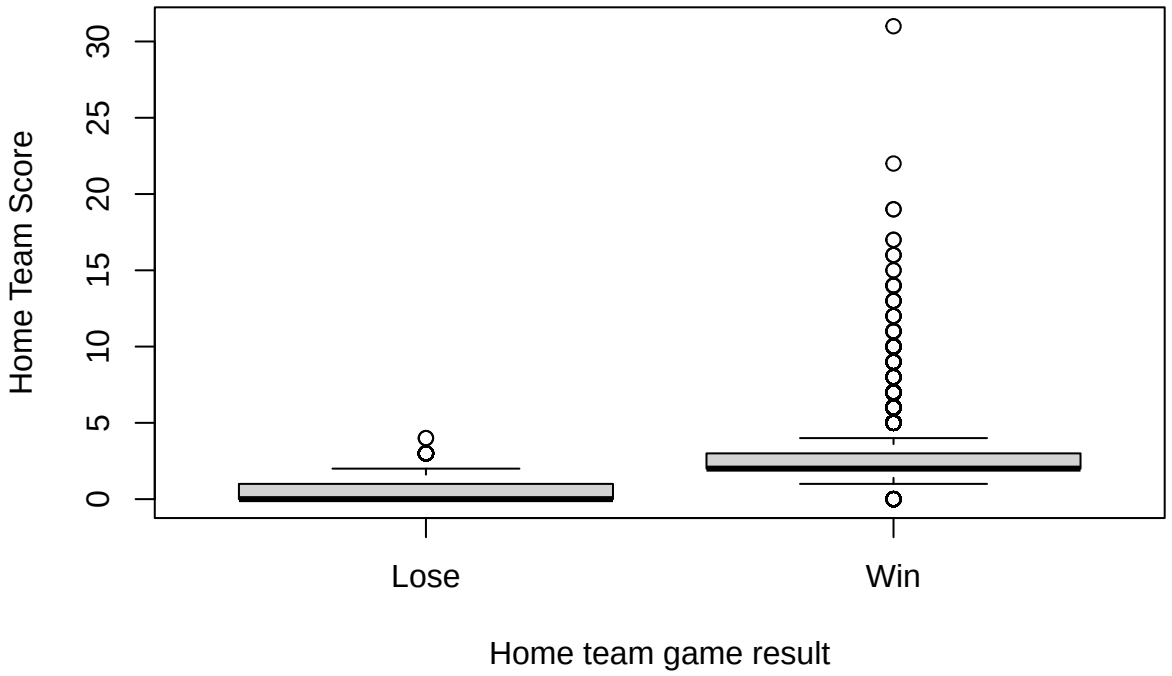
```
ncol(train)
```

```
## [1] 19
```

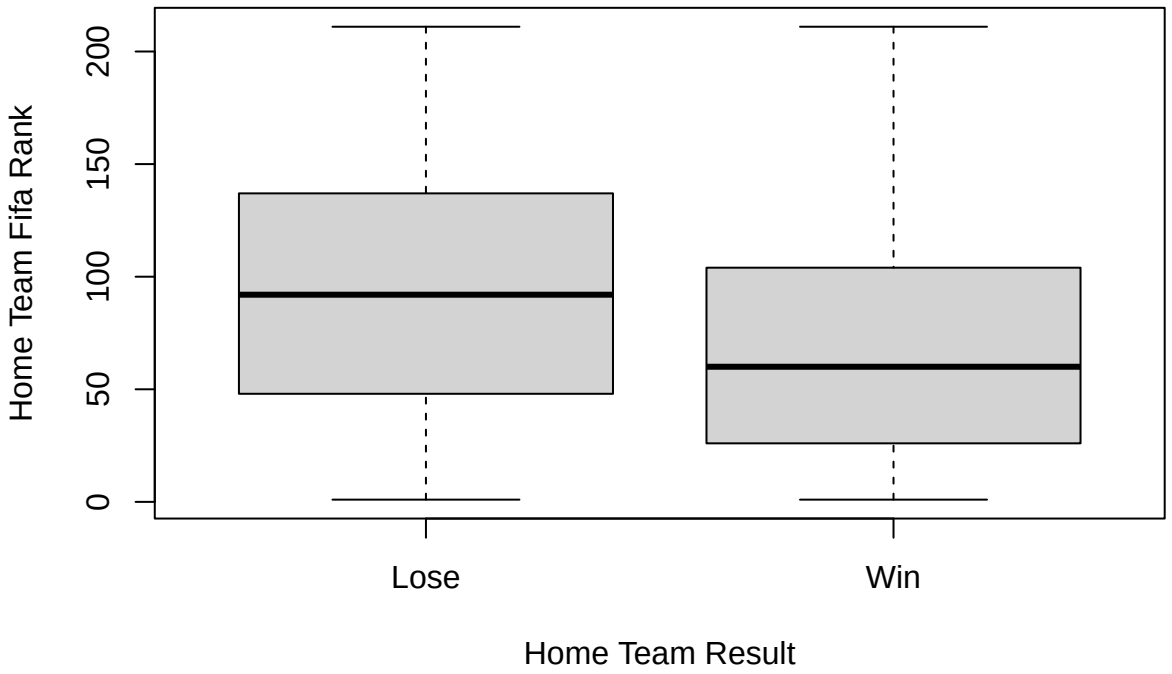
This gives us a better insight of what data we're dealing with

c. Create at least 2 informative graphs, using the training data

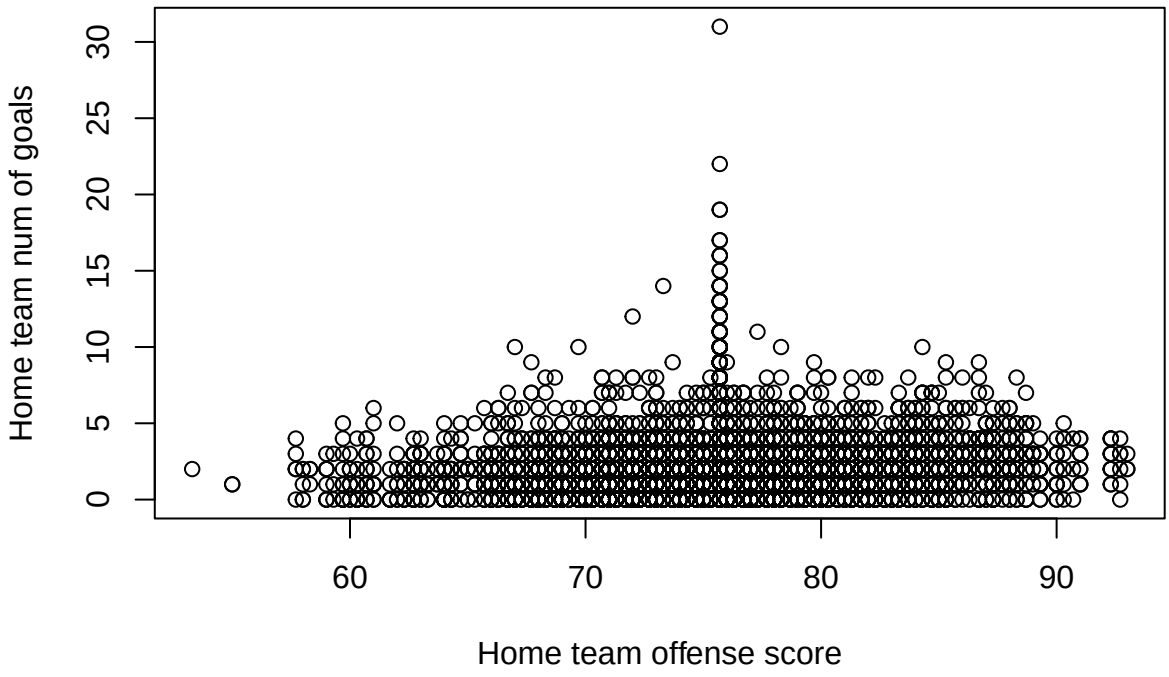
```
plot(train$h_result, train$h_score, xlab = "Home team game result", ylab = "Home Team Score")
```




```
plot(train$h_result, train$h_fifa, xlab = "Home Team Result", ylab = "Home Team Fifa Rank")
```




```
plot(train$h_off_score, train$h_score, xlab = "Home team offense score", ylab = "Home team num of goals
```



d. Build a logistic regression model

```
glm1 <- glm(h_fifa_points~h_result, data=train, family="binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = h_fifa_points ~ h_result, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.025  -1.014  -1.014   1.350   1.350
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.36904    0.02756 -13.391   <2e-16 ***
## h_resultWin -0.02746    0.03469  -0.792    0.429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20009  on 14824  degrees of freedom
## Residual deviance: 20008  on 14823  degrees of freedom
## AIC: 20012
##
## Number of Fisher Scoring iterations: 4
```

In this logistic regression model, we can see how the number of Fifa points of the home team affects the result of the game. A one unit increase in the predictor variable h_result is associated with an average change of -0.02746 in the log odds of the response variable h_fifa_points taking on a value of 1. That means, a

"win" in h_results tends to be associated with a higher value in h_fifa_points, which represents the strength of a given team, based on the stats in the video game "Fifa". Since there is a very high value in degrees of freedom of either Null and Residual deviance, the model poorly fits the dataset.

e. Build a naïve Bayes model

```
library(naivebayes)
```

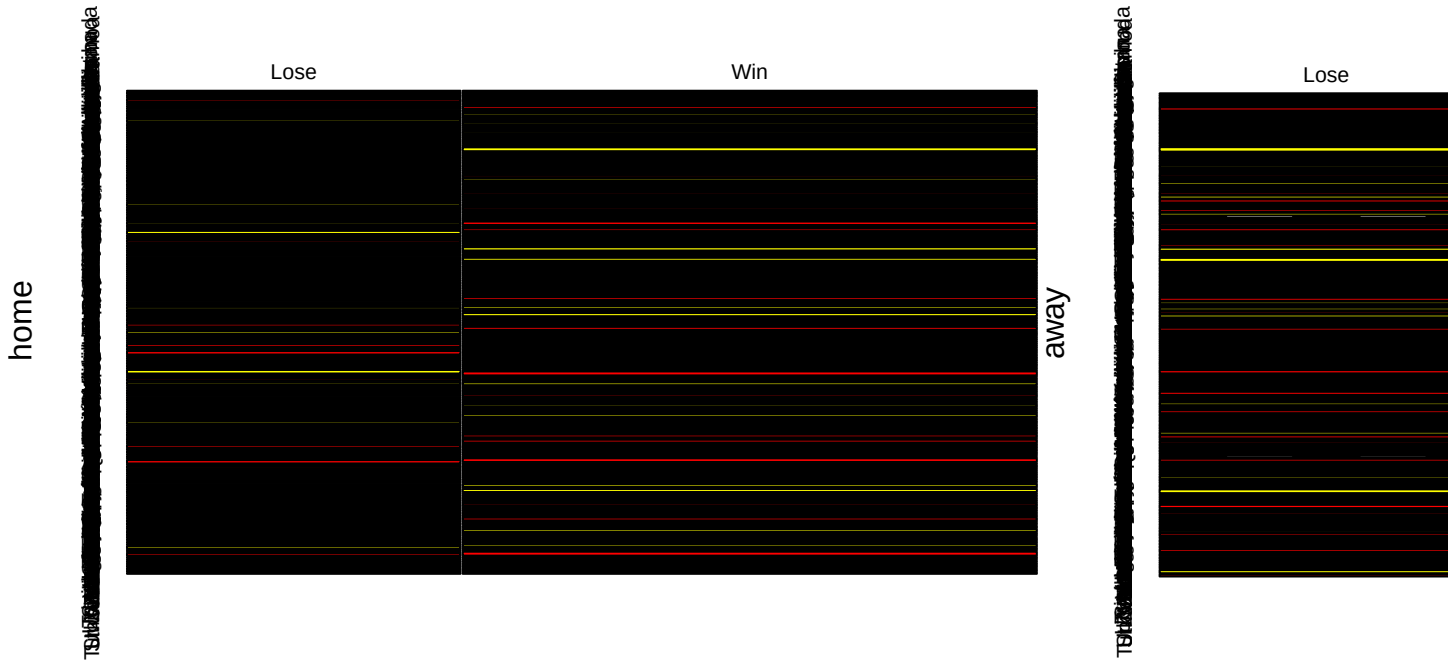
```
## naivebayes 0.9.7 loaded
```

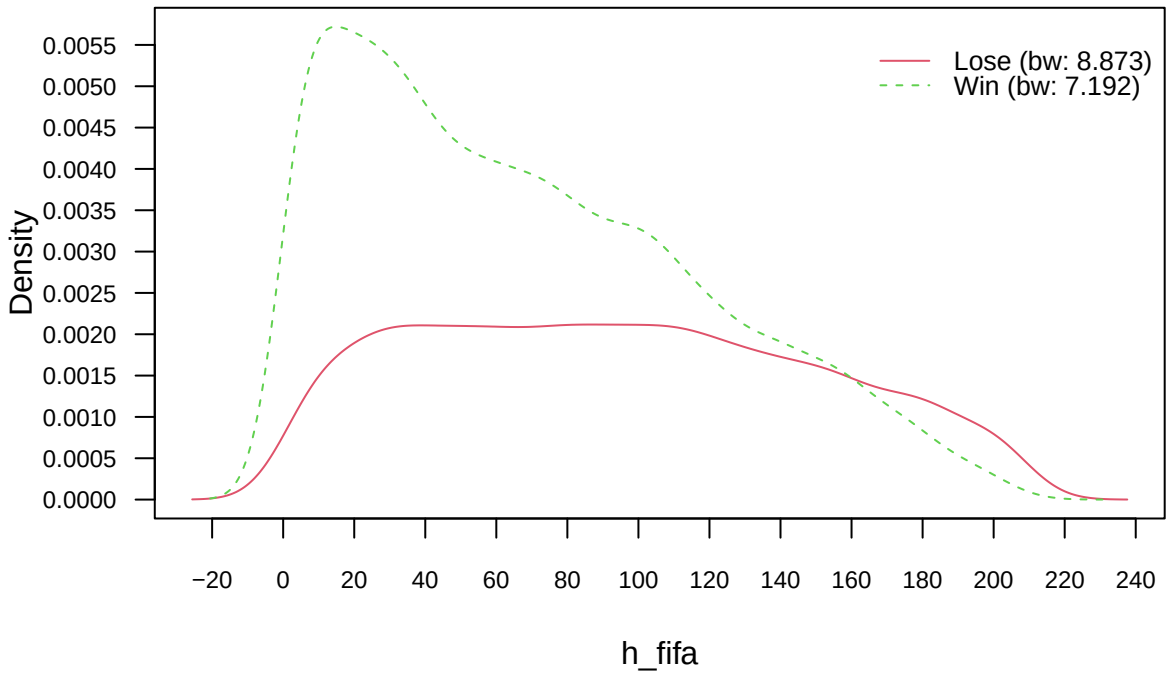
```
model <- naive_bayes(h_result~., data=train, usekernel = T)
```

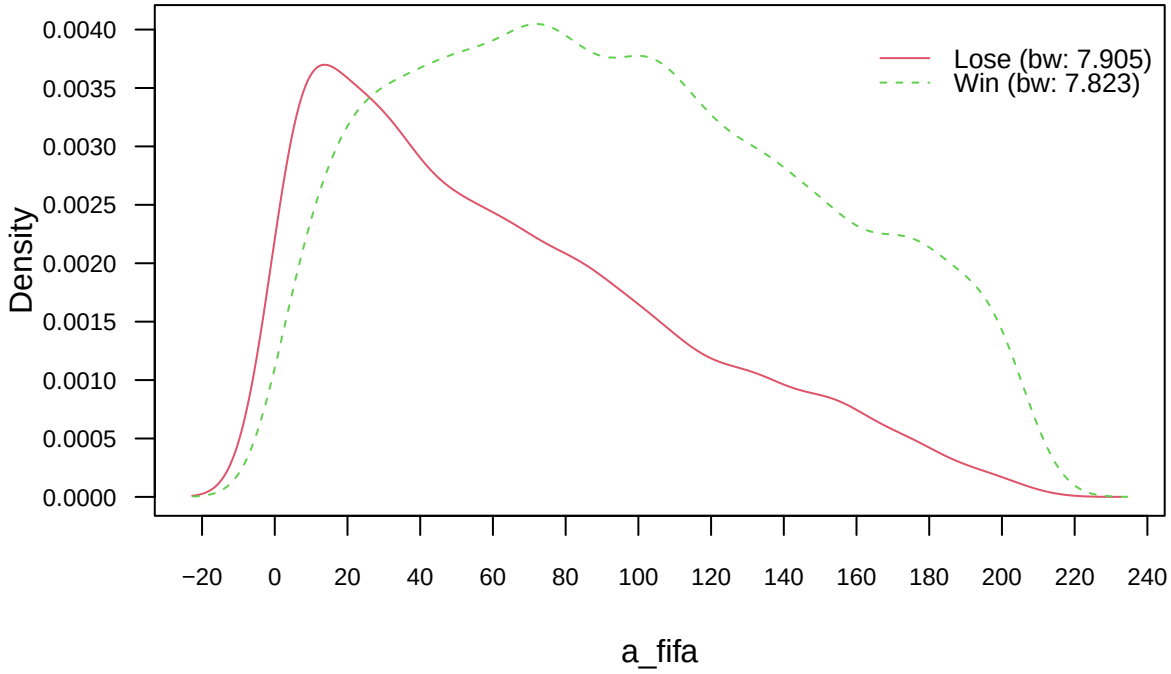
```
## Warning: naive_bayes(): Feature away - zero probabilities are present. Consider
## Laplace smoothing.
```

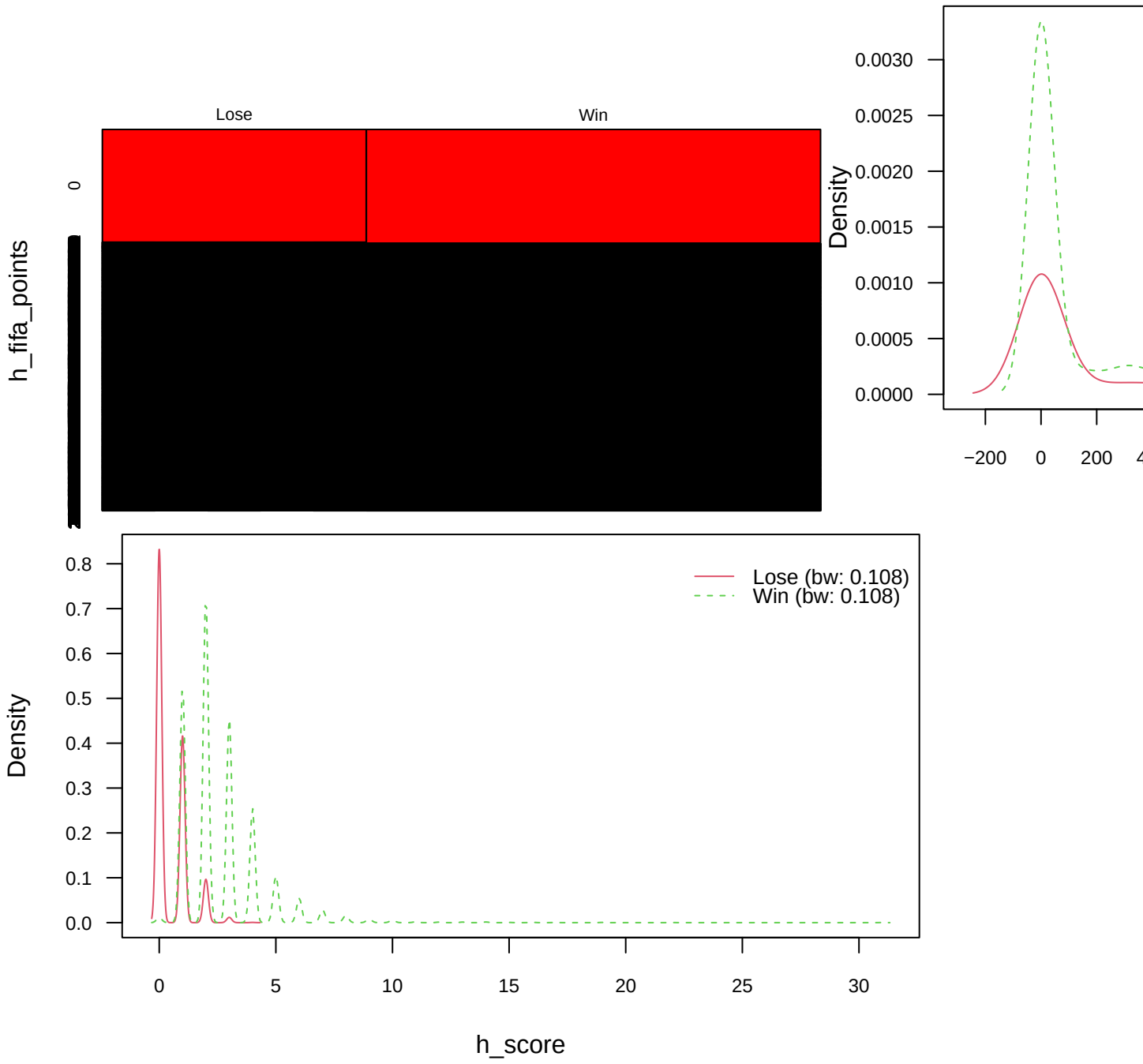
```
## Warning: naive_bayes(): Feature h_fifa_points - zero probabilities are present.
## Consider Laplace smoothing.
```

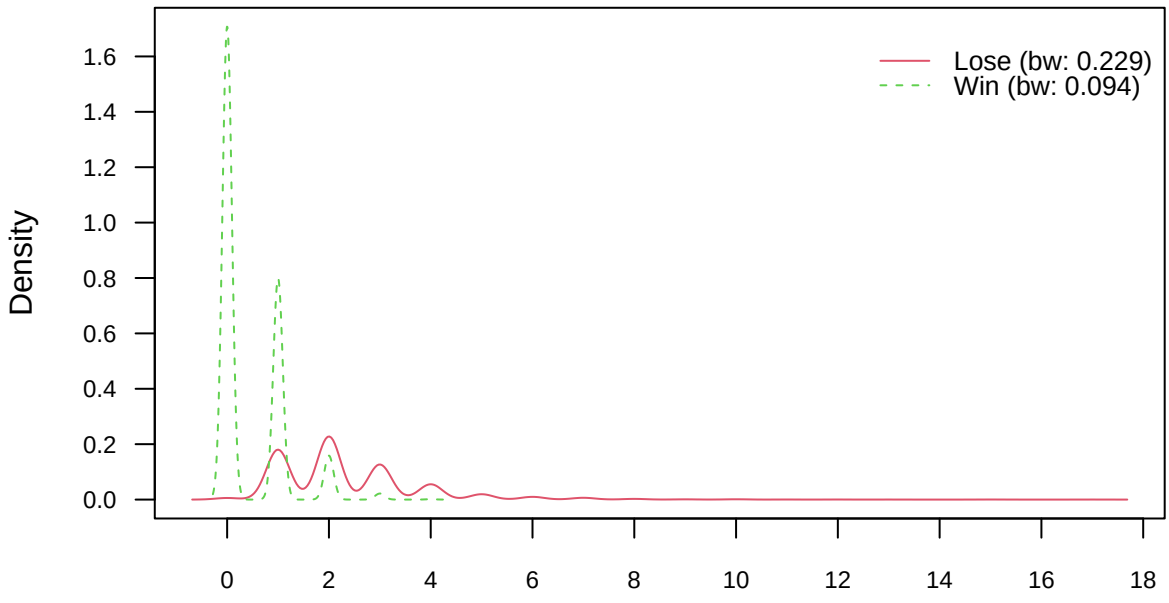
```
plot(model)
```

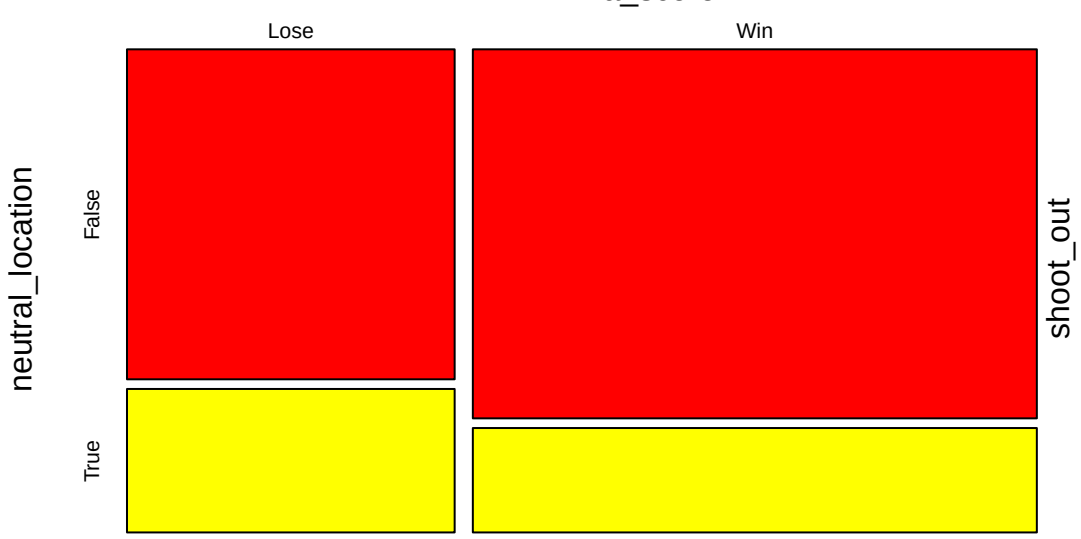


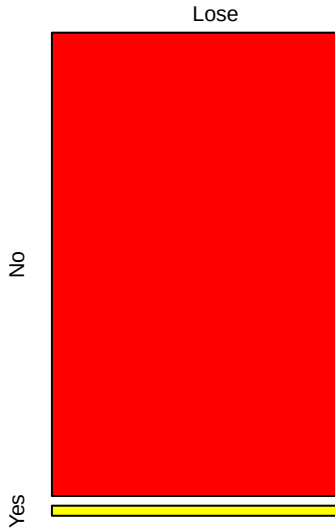

Lose (bw: 8.873)
Win (bw: 7.192)
Density
0.0055
0.0050
0.0045
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
−20
0
20
40
60
80
100
120
140
160
180
200
220
240
h_fifa

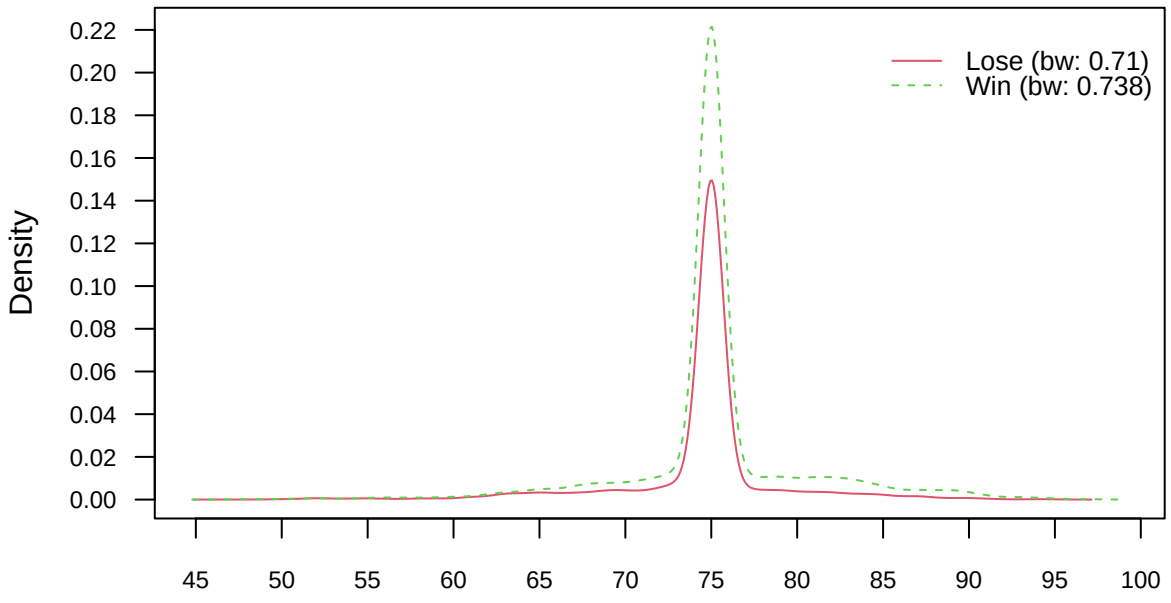

Lose (bw: 7.905)
Win (bw: 7.823)
Density
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
−20
0
20
40
60
80
100
120
140
160
180
200
220
240
a_fifa

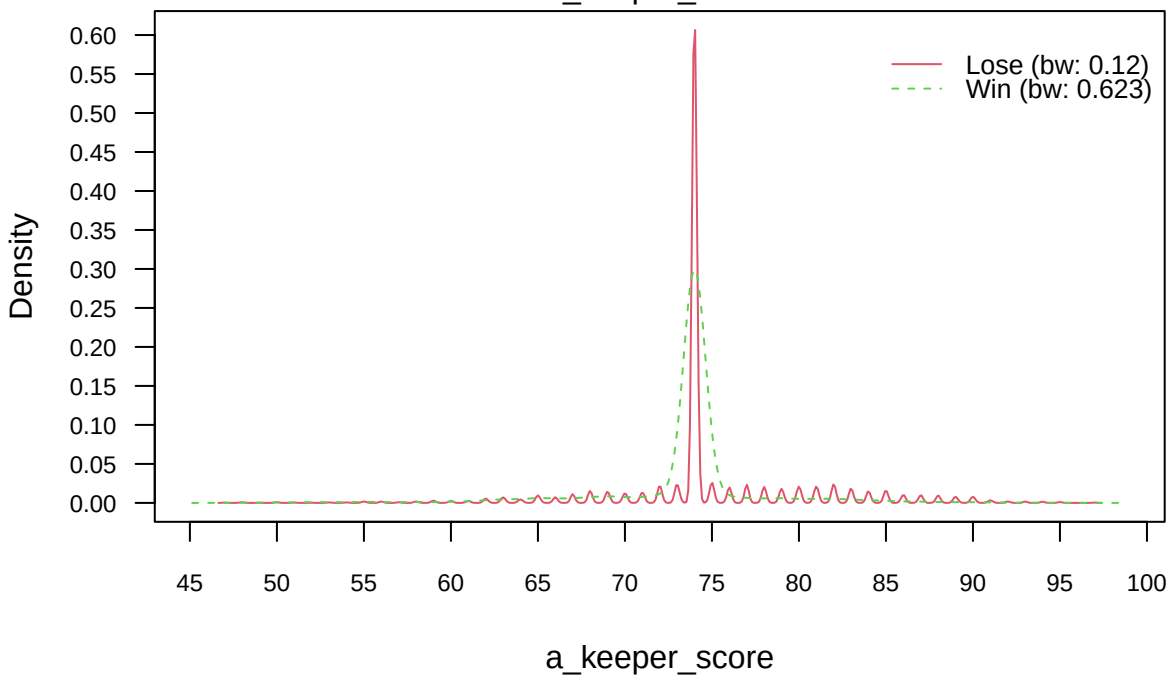

Lose
Win
h_fifa_points
0
Density
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
−200
0
200
Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Lose (bw: 0.108)
Win (bw: 0.108)
0
5
10
15
20
25
30
h_score

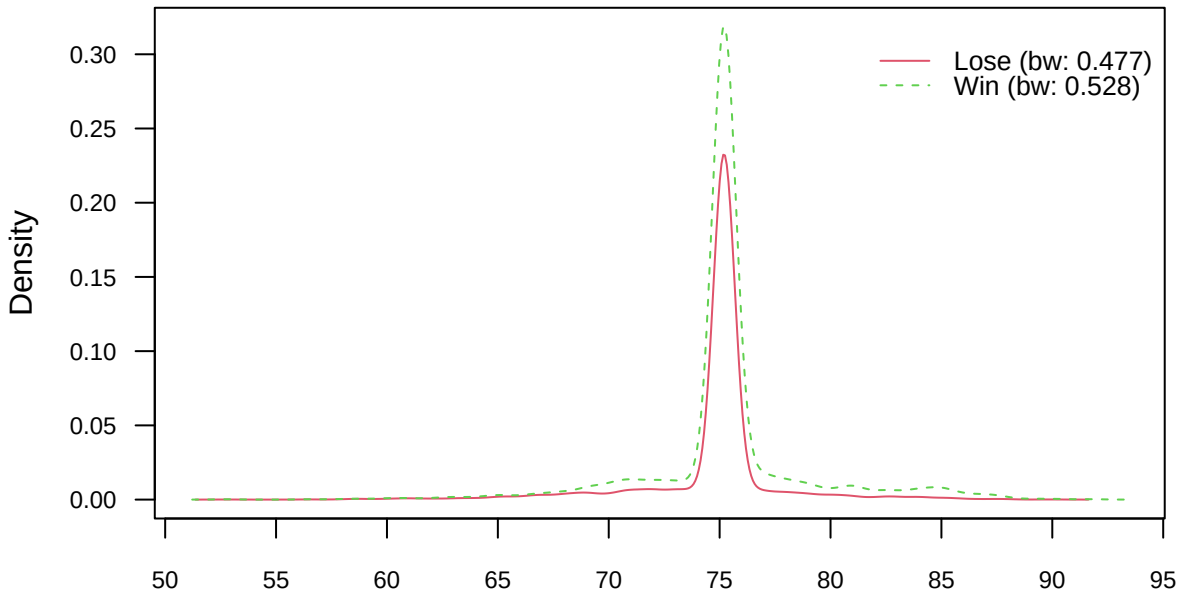

Density
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Lose (bw: 0.229)
Win (bw: 0.094)
0
2
4
6
8
10
12
14
16
18
a_score

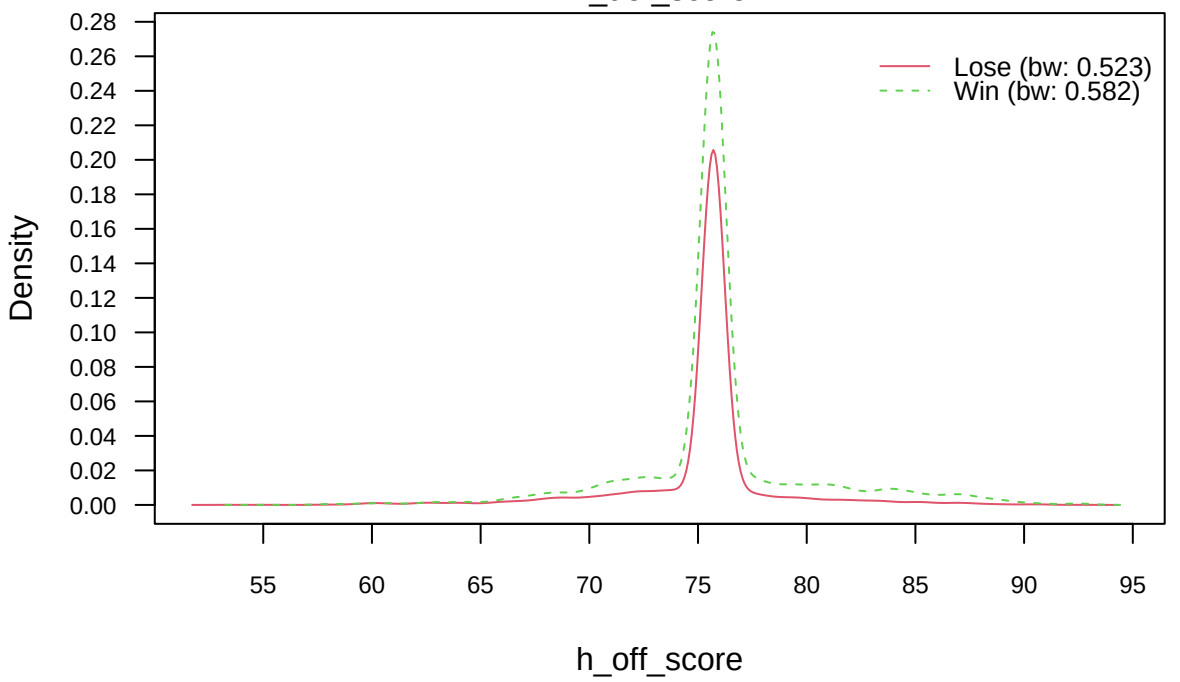

Lose
Win
neutral_location
False
True
shoot_out

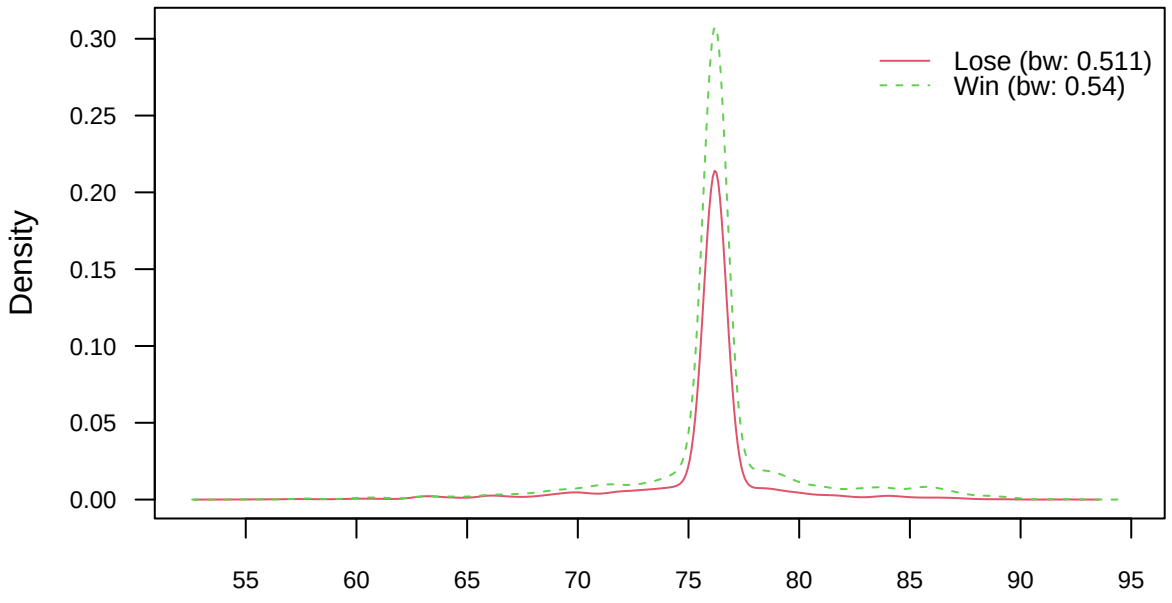

Lose
No
Yes

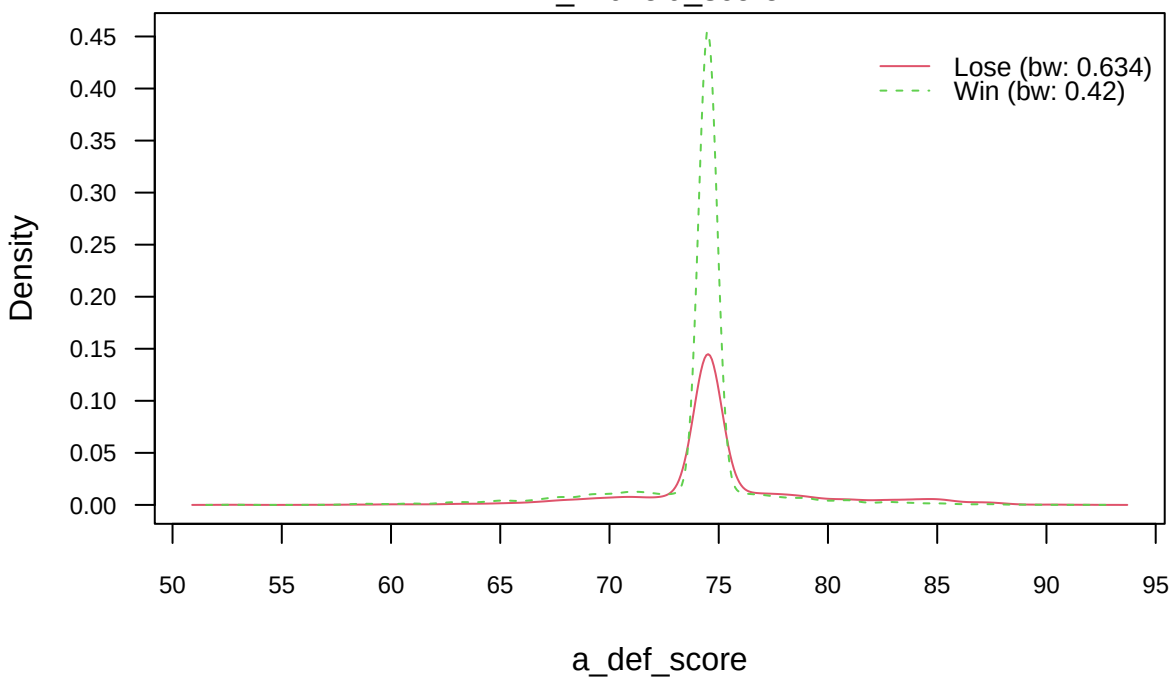

Density
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Lose (bw: 0.71)
Win (bw: 0.738)
45
50
55
60
65
70
75
80
85
90
95
100
h_keeper_score

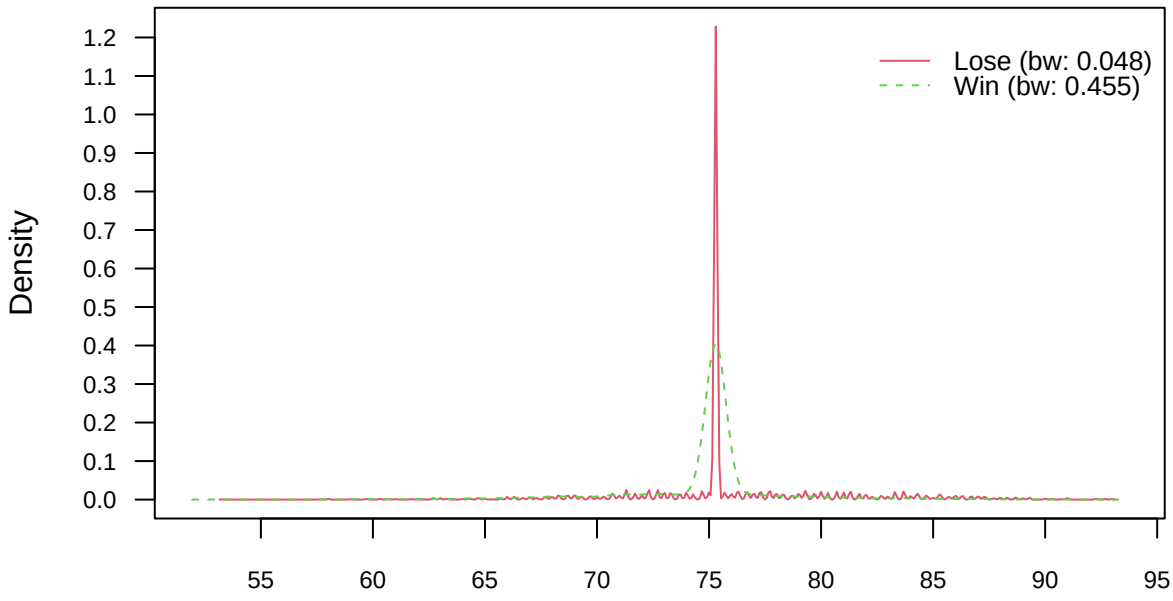

Density
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Lose (bw: 0.12)
Win (bw: 0.623)
45
50
55
60
65
70
75
80
85
90
95
100
a_keeper_score

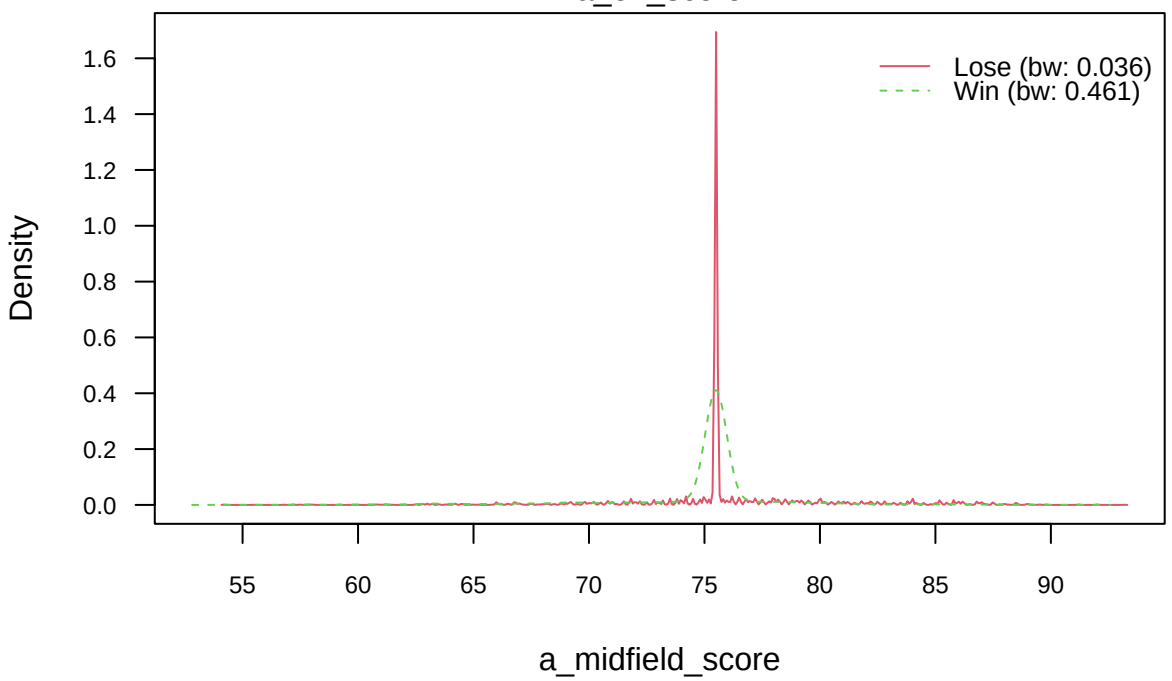

Lose (bw: 0.477)
Win (bw: 0.528)
Density
0.30
0.25
0.20
0.15
0.10
0.05
0.00
50
55
60
65
70
75
80
85
90
95
h_def_score


Lose (bw: 0.523)
Win (bw: 0.582)
Density
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
55
60
65
70
75
80
85
90
95
h_off_score


Density
0.30
0.25
0.20
0.15
0.10
0.05
0.00
55
60
65
70
75
80
85
90
95
Lose (bw: 0.511)
Win (bw: 0.54)
h_midfield_score


Density
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
50
55
60
65
70
75
80
85
90
95
Lose (bw: 0.634)
Win (bw: 0.42)
a_def_score





Here we can see that the attribute "neutral_location" barely has any influence on the outcome of the game. The home game loses slightly more when they played in a neutral location, other than that, there is not much correlation.

f. Using these two classification models models, predict and evaluate on the test data

```
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, 1, 0)
acc <- mean(pred==test$h_result)
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0"
```

```
table(pred, test$h_result)
```

```
##
## pred Lose  Win
##    0 1323 2384
```

```
probs2 <- predict(model, newdata=test, type="class")
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```
pred2 <- ifelse(probs2>0.5, 1, 0)
```

```
## Warning in Ops.factor(probs2, 0.5): '>' not meaningful for factors
```

```
acc2 <- mean(pred2==test$h_result)
print(paste("accuracy = ", acc2))
```

```
## [1] "accuracy =  NA"
```

```
table(pred2, test$h_result)
```

```
## < table of extent 0 x 2 >
```

I could not get an actual result out of the naive bayes model, which is why I am unable to compare the 2 models.

g. Strengths and Weaknesses of Naïve Bayes and Logistic Regression

Naïve Bayes

- It works better with data containing independent features, compared to other models, and therefore requires less training data.
- Another strength is that it works better with categorical input variables than numeric variables.
- It is less applicable for real world problems, as it assumes that all predictors are independent, which is very unlikely in real world data
- If data in the test set wasn't available in the training data set, it assigns zero probability to it, which can completely mess up your data and give unrealistic estimations

Logistic Regression

- Compared to other models, it is quite easy to implement and trains efficiently
- It classifies unknown data faster than other models
- It is extendable to multiple classes
- Overfitting occurs quickly, if there are less observations than predictors
- It assumes a linearity between dependent and independent variables

h. Benefits & drawbacks of each of the classification metrics used

Accuracy

It measures how often the classifier correctly predicts. It is easy to understand and gives the user a basic idea of the effectiveness of the model. It can be very misleading though when used on unbalanced data, as it will show high effectiveness if just wild guesses could give the same "accuracy". Since it is just a single value, it is not as interpretable as a Confusion Matrix.

Confusion Matrix

It is a matrix showing correctly classified instances, as well as errors. You can see what types of errors were made by the model. This is an advantage over Accuracy, as this does not show incorrect predictions. Confusions Matrices are not made for Multiclass and cannot give class probabilities.