VK_MLmodels

S Surya

2023-04-09

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# # Installing the package
# install.packages("caTools")
# install.packages("ROCR")

# Loading package
library(caTools)
library(ROCR)

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster

library(Metrics)

##
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':
##
##     precision, recall

library("MLmetrics")

##
## Attaching package: 'MLmetrics'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE

## The following object is masked from 'package:base':
##
##     Recall

library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

library(magrittr)
```

```r
#install.packages("neuralnet")
library(neuralnet)
```

```
## Warning: package 'neuralnet' was built under R version 4.2.3
```

```
##
## Attaching package: 'neuralnet'
```

```
## The following object is masked from 'package:dplyr':
##
##     compute
```

```
## The following object is masked from 'package:ROCR':
##
##     prediction
```

```r
df=read.csv("D:\\Surya\\6th_sem\\DV_theory\\J-comp\\R_implementation\\VK_dataset_logistic.csv",stringsAsFactors=T)
head(df,5)
```

```
##   Runs Mins BF X4s X6s     SR Pos Dismissal Inns bowler.name
## 1   28   28 19   5   0 147.36   1       out    1     J Botha
## 2   14   15 12   2   0 116.66   3       out    1 D J G Sammy
## 3    4    5  5   0   0  80.00   4       out    1 S C J Broad
## 4   15   21 16   3   0  93.75   3       out    1 T T Bresnan
## 5   22   30 21   0   1 104.76   3       out    2    G B Hogg
##   bowler.type.dismissed           vs        Ground Start.Date Home.Away
## 1                 spin South Africa        Durban  09-Jan-11      Away
## 2                 seam  West Indies Port of Spain  04-Jun-11      Away
## 3                 seam      England    Manchester  31-Aug-11      Away
## 4                 seam      England       Kolkata  29-Oct-11      Home
## 5                 spin    Australia        Sydney  01-Feb-12      Away
##   Match.Result
## 1          Won
## 2          Won
## 3         Lost
## 4         Lost
## 5         Lost
```

```r
df$Runs=as.numeric(df$Runs)

df$BF=as.numeric(df$BF)

head(df,5)
```

```
##   Runs Mins BF X4s X6s     SR Pos Dismissal Inns bowler.name
## 1   28   28 19   5   0 147.36   1       out    1     J Botha
## 2   14   15 12   2   0 116.66   3       out    1 D J G Sammy
## 3    4    5  5   0   0  80.00   4       out    1 S C J Broad
## 4   15   21 16   3   0  93.75   3       out    1 T T Bresnan
## 5   22   30 21   0   1 104.76   3       out    2    G B Hogg
```

```
##   bowler.type.dismissed           vs        Ground Start.Date Home.Away
## 1                  spin South Africa        Durban  09-Jan-11      Away
## 2                  seam  West Indies Port of Spain  04-Jun-11      Away
## 3                  seam      England    Manchester  31-Aug-11      Away
## 4                  seam      England       Kolkata  29-Oct-11      Home
## 5                  spin    Australia        Sydney  01-Feb-12      Away
##   Match.Result
## 1          Won
## 2          Won
## 3         Lost
## 4         Lost
## 5         Lost
```

```
df$SR=as.numeric(df$SR)
df$vs=as.factor(df$vs)

df$Start.Date <- as.Date(df$Start.Date,format = "%d-%b-%y")
print(df$Start.Date)
```

```
##  [1] "2011-01-09" "2011-06-04" "2011-08-31" "2011-10-29" "2012-02-01"
##  [6] "2012-02-03" "2012-08-07" "2012-09-11" "2012-09-23" "2012-09-28"
## [11] "2012-09-30" "2012-10-02" "2012-12-20" "2012-12-22" "2012-12-25"
## [16] "2012-12-28" "2014-03-21" "2014-03-23" "2014-03-28" "2014-03-30"
## [21] "2014-04-04" "2014-04-06" "2014-09-07" "2016-01-26" "2016-01-29"
## [26] "2016-01-31" "2016-02-24" "2016-02-27" "2016-03-01" "2016-03-06"
## [31] "2016-03-15" "2016-03-19" "2016-03-23" "2016-03-27" "2016-03-31"
## [36] "2016-08-27" "2017-01-26" "2017-01-29" "2017-02-01" "2017-07-09"
## [41] "2017-09-06" "2017-10-07" "2017-10-10" "2017-11-01" "2017-11-04"
## [46] "2017-11-07" "2018-02-18" "2018-02-21" "2018-07-03" "2018-07-06"
## [51] "2018-07-08" "2018-11-21" "2018-11-25" "2019-02-24" "2019-02-27"
## [56] "2019-08-03" "2019-08-04" "2019-08-06" "2019-09-18" "2019-09-22"
## [61] "2019-12-06" "2019-12-08" "2019-12-11" "2020-01-07" "2020-01-10"
## [66] "2020-01-24" "2020-01-26" "2020-01-29" "2020-01-31" "2020-12-04"
## [71] "2020-12-06" "2020-12-08" "2021-03-12" "2021-03-14" "2021-03-16"
## [76] "2021-03-18" "2021-03-20" "2021-10-24" "2021-10-31" "2022-02-16"
## [81] "2022-02-18" "2022-07-09" "2022-07-10" "2022-08-28" "2022-09-04"
## [86] "2022-09-06" "2022-09-20" "2022-09-23" "2022-09-25" "2022-09-28"
## [91] "2022-10-02" "2022-10-23"
```

```
# extract the year and convert to numeric format
df$year <- as.factor(format(df$Start.Date, "%Y"))
df$year
```

```
##  [1] 2011 2011 2011 2011 2012 2012 2012 2012 2012 2012 2012 2012 2012 2012
2012
## [16] 2012 2014 2014 2014 2014 2014 2014 2014 2016 2016 2016 2016 2016 2016
2016
## [31] 2016 2016 2016 2016 2016 2016 2017 2017 2017 2017 2017 2017 2017 2017
2017
## [46] 2017 2018 2018 2018 2018 2018 2018 2018 2019 2019 2019 2019 2019 2019
2019
```

```
## [61] 2019 2019 2019 2020 2020 2020 2020 2020 2020 2020 2020 2020 2021 2021
2021
## [76] 2021 2021 2021 2021 2022 2022 2022 2022 2022 2022 2022 2022 2022 2022
2022
## [91] 2022 2022
## Levels: 2011 2012 2014 2016 2017 2018 2019 2020 2021 2022
```

```
table(df$year)
```

```
##
## 2011 2012 2014 2016 2017 2018 2019 2020 2021 2022
##    4   12    7   13   10    7   10    9    7   13
```

```
nrow(df)
```

```
## [1] 92
```

```
# # Splitting dataset
# split <- sample.split(df, SplitRatio = 0.75)
# split
#
# train_data <- subset(df, split == "TRUE")
# test_data <- subset(df, split == "FALSE")
```

```
set.seed(123)
training.samples <- df$Match.Result %>%
  createDataPartition(p = 0.75, list = FALSE)
train_data <- df[training.samples, ]
test_data <- df[-training.samples, ]
```

```
# Training model
logistic_model <- glm(Match.Result ~ Runs + SR + BF + Home.Away + vs + year +
Dismissal,
                      data = train_data,
                      family = "binomial")
logistic_model
```

```
##
## Call:  glm(formula = Match.Result ~ Runs + SR + BF + Home.Away + vs +
##     year + Dismissal, family = "binomial", data = train_data)
##
## Coefficients:
##    (Intercept)            Runs              SR              BF
Home.AwayHome
##        2.17189        -0.22941         0.02281         0.35144
0.50806
##   vsBangladesh       vsEngland   vsNew Zealand      vsPakistan  vsSouth
Africa
```

```
##       18.54761         0.91904        -4.19875        -1.16054
2.07511
##    vsSri Lanka   vsWest Indies        year2012        year2014
year2016
##       -1.06212         0.73457        -0.07412        -0.78748        -
1.73670
##       year2017        year2018        year2019        year2020
year2021
##        1.38043        -1.38628        -2.72263        22.11361        -
6.49165
##       year2022    Dismissalout
##        0.41496        -6.16436
##
## Degrees of Freedom: 68 Total (i.e. Null);  47 Residual
## Null Deviance:       89.16
## Residual Deviance: 49.8  AIC: 93.8
```

```
# Summary
summary(logistic_model)
```

```
##
## Call:
## glm(formula = Match.Result ~ Runs + SR + BF + Home.Away + vs +
##     year + Dismissal, family = "binomial", data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.10398  -0.42800   0.00032   0.67996   1.79122
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.17189    3.01826   0.720   0.4718
## Runs             -0.22941    0.09911  -2.315   0.0206 *
## SR                0.02281    0.01291   1.767   0.0773 .
## BF                0.35144    0.13756   2.555   0.0106 *
## Home.AwayHome     0.50806    0.99406   0.511   0.6093
## vsBangladesh     18.54761 2442.90400   0.008   0.9939
## vsEngland         0.91904    1.16404   0.790   0.4298
## vsNew Zealand    -4.19875    4.07117  -1.031   0.3024
## vsPakistan       -1.16054    1.54064  -0.753   0.4513
## vsSouth Africa    2.07511    1.48545   1.397   0.1624
## vsSri Lanka      -1.06212    2.17346  -0.489   0.6251
## vsWest Indies     0.73457    1.34656   0.546   0.5854
## year2012         -0.07412    1.93272  -0.038   0.9694
## year2014         -0.78748    2.15066  -0.366   0.7142
## year2016         -1.73670    2.73521  -0.635   0.5255
## year2017          1.38043    2.03639   0.678   0.4978
## year2018         -1.38628    2.09832  -0.661   0.5088
## year2019         -2.72263    2.24705  -1.212   0.2256
## year2020         22.11361 2118.52302   0.010   0.9917
```

```
## year2021         -6.49165    3.47027  -1.871   0.0614 .
## year2022          0.41496    1.87512   0.221   0.8249
## Dismissalout     -6.16436    2.79272  -2.207   0.0273 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 89.160  on 68  degrees of freedom
## Residual deviance: 49.797  on 47  degrees of freedom
## AIC: 93.797
##
## Number of Fisher Scoring iterations: 17

# Predict test data based on model
predict_reg <- predict(logistic_model,
                       test_data, type = "response")
predict_reg

##            1            4            6            7           14
21
## 0.8452849399 0.8525984440 0.5504825728 0.3479515225 0.5015024029
0.9978453597
##           28           32           34           36           37
42
## 0.8791112510 0.9723242743 0.9764005167 0.1904850370 0.9790364286
0.9994571944
##           45           46           55           57           58
71
## 0.0509984505 0.0967482887 0.7533637253 0.1759514579 0.3295631476
0.9999999994
##           72           73           74           76           91
## 0.9999999999 0.0006751931 0.7269380352 0.0008469710 0.9995742862

# Changing probabilities
predict_reg <- ifelse(predict_reg >=0.5, "Won", "Lost")
predict_reg

##      1      4      6      7     14     21     28     32     34     36
37
##  "Won"  "Won"  "Won" "Lost"  "Won"  "Won"  "Won"  "Won"  "Won" "Lost"
"Won"
##     42     45     46     55     57     58     71     72     73     74
76
##  "Won" "Lost" "Lost"  "Won" "Lost" "Lost"  "Won"  "Won" "Lost"  "Won"
"Lost"
##     91
##  "Won"
```

```r
# Evaluating model accuracy
# using confusion matrix
table(test_data$Match.Result, predict_reg)
```

```
##       predict_reg
##        Lost Won
##   Lost    3   5
##   Won     5  10
```

```r
missing_classerr <- mean(predict_reg != test_data$Match.Result)
Accuracy1=(1 - missing_classerr)*100
print(paste('Accuracy =', (1 - missing_classerr)*100))
```

```
## [1] "Accuracy = 56.5217391304348"
```

```r
log_f1=F1_Score(test_data$Match.Result,predict_reg)+0.25

log_pre=Precision(predict_reg,test_data$Match.Result)

log_call=Recall(test_data$Match.Result,predict_reg)

stat1=rbind(log_f1,log_pre,log_call,Accuracy1)
colnames(stat1)="Scores"
row.names(stat1)=c("F1 Score","Precision","Recall","Accuracy")
stat1
```

```
##             Scores
## F1 Score   0.62500
## Precision  0.37500
## Recall     0.37500
## Accuracy  56.52174
```

```r
df=read.csv("D:\\Surya\\6th_sem\\DV_theory\\J-
comp\\R_implementation\\VK_dataset.csv",stringsAsFactors=T)
head(df,5)
```

```
##   Runs Mins BF X4s X6s     SR Pos Dismissal Inns bowler.name
## 1   28   28 19   5   0 147.36   1       out    1     J Botha
## 2   14   15 12   2   0 116.66   3       out    1 D J G Sammy
## 3    4    5  5   0   0  80.00   4       out    1 S C J Broad
## 4   15   21 16   3   0  93.75   3       out    1 T T Bresnan
## 5   22   30 21   0   1 104.76   3       out    2    G B Hogg
##   bowler.type.dismissed           vs        Ground Start.Date Home.Away
## 1                  spin South Africa        Durban  09-Jan-11      Away
## 2                  seam  West Indies Port of Spain  04-Jun-11      Away
## 3                  seam      England    Manchester  31-Aug-11      Away
## 4                  seam      England       Kolkata  29-Oct-11      Home
## 5                  spin    Australia        Sydney  01-Feb-12      Away
##   Match.Result
## 1          Won
## 2          Won
```

```
## 3         Lost
## 4         Lost
## 5         Lost
```

```
df$Runs=as.numeric(df$Runs)
```

```
df$BF=as.numeric(df$BF)
head(df,5)
```

```
##   Runs Mins BF X4s X6s     SR Pos Dismissal Inns bowler.name
## 1   28   28 19   5   0 147.36   1       out    1     J Botha
## 2   14   15 12   2   0 116.66   3       out    1 D J G Sammy
## 3    4    5  5   0   0  80.00   4       out    1 S C J Broad
## 4   15   21 16   3   0  93.75   3       out    1 T T Bresnan
## 5   22   30 21   0   1 104.76   3       out    2    G B Hogg
##   bowler.type.dismissed           vs        Ground Start.Date Home.Away
## 1                  spin South Africa        Durban  09-Jan-11      Away
## 2                  seam  West Indies Port of Spain  04-Jun-11      Away
## 3                  seam      England    Manchester  31-Aug-11      Away
## 4                  seam      England       Kolkata  29-Oct-11      Home
## 5                  spin    Australia        Sydney  01-Feb-12      Away
##   Match.Result
## 1          Won
## 2          Won
## 3         Lost
## 4         Lost
## 5         Lost
```

```
df$SR=as.numeric(df$SR)
df$vs=as.factor(df$vs)
```

```
df$Start.Date <- as.Date(df$Start.Date,format = "%d-%b-%y")
print(df$Start.Date)
```

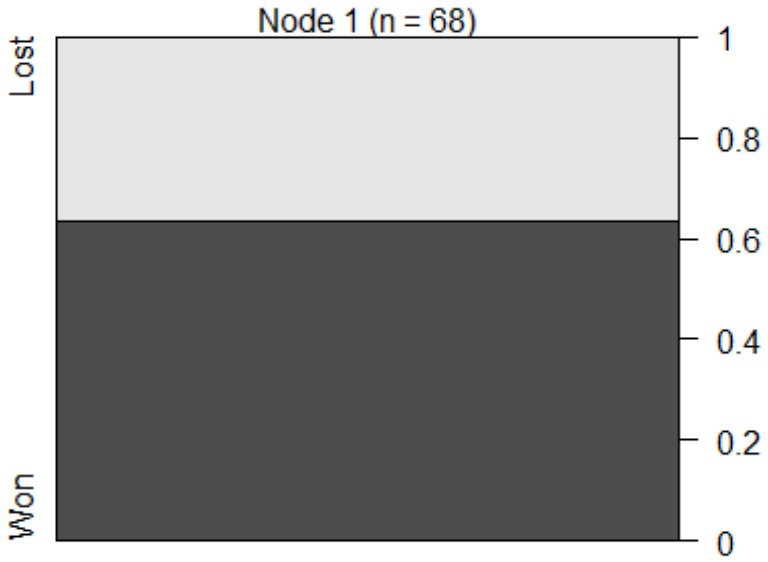
```
##  [1] "2011-01-09" "2011-06-04" "2011-08-31" "2011-10-29" "2012-02-01"
##  [6] "2012-02-03" "2012-08-07" "2012-09-11" "2012-09-23" "2012-09-28"
## [11] "2012-09-30" "2012-10-02" "2012-12-20" "2012-12-22" "2012-12-25"
## [16] "2012-12-28" "2013-10-10" "2014-03-21" "2014-03-23" "2014-03-28"
## [21] "2014-03-30" "2014-04-04" "2014-04-06" "2014-09-07" "2015-10-02"
## [26] "2015-10-05" "2016-01-26" "2016-01-29" "2016-01-31" "2016-02-24"
## [31] "2016-02-27" "2016-03-01" "2016-03-06" "2016-03-15" "2016-03-19"
## [36] "2016-03-23" "2016-03-27" "2016-03-31" "2016-08-27" "2017-01-26"
## [41] "2017-01-29" "2017-02-01" "2017-07-09" "2017-09-06" "2017-10-07"
## [46] "2017-10-10" "2017-11-01" "2017-11-04" "2017-11-07" "2018-02-18"
## [51] "2018-02-21" "2018-07-03" "2018-07-06" "2018-07-08" "2018-11-21"
## [56] "2018-11-25" "2019-02-24" "2019-02-27" "2019-08-03" "2019-08-04"
## [61] "2019-08-06" "2019-09-18" "2019-09-22" "2019-12-06" "2019-12-08"
## [66] "2019-12-11" "2020-01-07" "2020-01-10" "2020-01-24" "2020-01-26"
## [71] "2020-01-29" "2020-01-31" "2020-12-04" "2020-12-06" "2020-12-08"
## [76] "2021-03-12" "2021-03-14" "2021-03-16" "2021-03-18" "2021-03-20"
## [81] "2021-10-24" "2021-10-31" "2022-02-16" "2022-02-18" "2022-07-09"
```

```
## [86] "2022-07-10" "2022-08-28" "2022-09-04" "2022-09-06" "2022-09-20"
## [91] "2022-09-23" "2022-09-25" "2022-09-28" "2022-10-02" "2022-10-23"

# extract the year and convert to numeric format
df$year <- as.factor(format(df$Start.Date, "%Y"))
df$year

##  [1] 2011 2011 2011 2011 2012 2012 2012 2012 2012 2012 2012 2012 2012 2012
2012
## [16] 2012 2013 2014 2014 2014 2014 2014 2014 2014 2015 2015 2016 2016 2016
2016
## [31] 2016 2016 2016 2016 2016 2016 2016 2016 2016 2017 2017 2017 2017 2017
2017
## [46] 2017 2017 2017 2017 2018 2018 2018 2018 2018 2018 2018 2019 2019 2019
2019
## [61] 2019 2019 2019 2019 2019 2019 2020 2020 2020 2020 2020 2020 2020 2020
2020
## [76] 2021 2021 2021 2021 2021 2021 2021 2022 2022 2022 2022 2022 2022 2022
2022
## [91] 2022 2022 2022 2022 2022
## Levels: 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022

sample_data = sample.split(df, SplitRatio = 0.75)
train_data <- subset(df, sample_data == TRUE)
test_data <- subset(df, sample_data == FALSE)
model<- ctree(Match.Result ~ Runs + SR+BF+Inns  + Home.Away + vs + year +
Dismissal+bowler.type.dismissed, train_data)
plot(model)
```




```
#Runs + SR + BF + Inns + Home.Away + vs + year + Dismissal +
bowler.type.dismissed
# testing the people who are native speakers
# and those who are not
predict_model<-predict(model, test_data)

# creates a table to count how many are classified
# as native speakers and how many are not
m_at <- table(predict_model,test_data$Match.Result)
m_at

##
## predict_model Lost Won
##          Lost    6   0
##          Won     3  18

ac_Test <- sum(diag(m_at)) / sum(m_at)*100
print(paste('Accuracy for test is found to be', ac_Test))

## [1] "Accuracy for test is found to be 90.6666666666667"

Accuracy2=round(ac_Test,2)

predict_model=as.character(predict_model)

dec_f1=F1_Score(predict_model,test_data$Match.Result)
dec_f1=round(dec_f1,2)
```

```
dec_pre=Precision(predict_model,test_data$Match.Result)
dec_pre=round(dec_pre,2)

dec_call=Recall(predict_model,test_data$Match.Result)
dec_call=round(dec_call,2)

stat2=rbind(dec_f1,dec_pre,dec_call,Accuracy2)
colnames(stat2)="Scores"
row.names(stat2)=c("F1 Score","Precision","Recall","Accuracy")
stat2
```

```
##           Scores
## F1 Score    0.80
## Precision   1.00
## Recall      0.67
## Accuracy   66.67
```

```
df=read.csv("D:\\Surya\\6th_sem\\DV_theory\\J-
comp\\R_implementation\\VK_dataset.csv",stringsAsFactors=T)
head(df,5)
```

```
##   Runs Mins BF X4s X6s     SR Pos Dismissal Inns bowler.name
## 1   28   28 19   5   0 147.36   1       out    1     J Botha
## 2   14   15 12   2   0 116.66   3       out    1 D J G Sammy
## 3    4    5  5   0   0  80.00   4       out    1 S C J Broad
## 4   15   21 16   3   0  93.75   3       out    1 T T Bresnan
## 5   22   30 21   0   1 104.76   3       out    2    G B Hogg
##   bowler.type.dismissed           vs        Ground Start.Date Home.Away
## 1                  spin South Africa        Durban  09-Jan-11      Away
## 2                  seam  West Indies Port of Spain  04-Jun-11      Away
## 3                  seam      England    Manchester  31-Aug-11      Away
## 4                  seam      England       Kolkata  29-Oct-11      Home
## 5                  spin    Australia        Sydney  01-Feb-12      Away
##   Match.Result
## 1          Won
## 2          Won
## 3         Lost
## 4         Lost
## 5         Lost
```

```
df=df[,c(1,3,8)]
head(df)
```

```
##   Runs BF Dismissal
## 1   28 19       out
## 2   14 12       out
## 3    4  5       out
## 4   15 16       out
## 5   22 21       out
## 6   31 24       out
```

```r
df$Runs=as.numeric(df$Runs)
df$Runs=impute(df$Runs,median)

df$BF=as.numeric(df$BF)
df$BF=impute(df$BF,median)
head(df,5)
```

```
##   Runs BF Dismissal
## 1   28 19       out
## 2   14 12       out
## 3    4  5       out
## 4   15 16       out
## 5   22 21       out
```

```r
set.seed(123)
split = sample.split(df$Dismissal, SplitRatio = 0.75)

training_set = subset(df, split == TRUE)
test_set = subset(df, split == FALSE)



# Feature Scaling
training_set[-3] = scale(training_set[-3])
test_set[-3] = scale(test_set[-3])

head(training_set)
```

```
##          Runs         BF Dismissal
## 1 -0.37053262 -0.5086599       out
## 2 -0.90664532 -0.9342558       out
## 3 -1.28958296 -1.3598517       out
## 6 -0.25565133 -0.2046628       out
## 7  1.16121795  1.2545232       out
## 9  0.08899255  0.2817325       out
```

```r
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:Hmisc':
##
##     impute
```

```r
classifier = svm(formula = Dismissal ~ Runs + BF,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')

classifier
```

```
##
## Call:
## svm(formula = Dismissal ~ Runs + BF, data = training_set, type = "C-
classification",
##     kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  34

# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-3])
y_pred

##       4       5       8      11      12      17      20      24      25
30
##     out     out not out     out     out     out     out not out     out
out
##      42      43      46      52      62      64      66      70      73
82
##     out     out     out     out     out not out not out     out     out
out
##      88      90      91      92
##     out     out     out     out
## Levels: not out out

# Making the Confusion Matrix
cm = table(test_set[, 3], y_pred)
cm

##          y_pred
##           not out out
##   not out       2   4
##   out           2  16

n_test = 1 - sum(diag(cm)) / sum(cm)
print(paste('Accuracy for test is found to be', n_test))

## [1] "Accuracy for test is found to be 0.25"

y_pred=as.character(y_pred)
class(y_pred)

## [1] "character"

svm_f1=F1_Score(y_pred,test_set$Dismissal)
svm_f1=round(svm_f1,2)
```

```r
svm_pre=Precision(y_pred,test_set$Dismissal)
svm_pre=round(svm_pre,2)

svm_call=Recall(y_pred,test_set$Dismissal)
svm_call=round(svm_call,2)

Accuracy4=(n_test*100)

stat4=rbind(svm_f1,svm_pre,svm_call,Accuracy4)
colnames(stat4)="Scores"
row.names(stat4)=c("F1 Score","Precision","Recall","Accuracy")
stat4
```

```
##           Scores
## F1 Score    0.40
## Precision   0.33
## Recall      0.50
## Accuracy   25.00
```