[1]:
```
# Import libarys

library(dplyr)
library(ggplot2)
library(gridExtra)
library(forcats)
library(lubridate)
library(caret)
library(ROCR)
library(FSelector)
```

[2]:
```
# Import DataFrame

df <- read.csv("C:/Users/Pablo/MUIIA/CienciaDeDatos/data.csv", header = T)

df$action_type <- as.factor(df$action_type)
df$combined_shot_type <- as.factor(df$combined_shot_type)
df$season <- as.factor(df$season)
df$shot_type <- as.factor(df$shot_type)
df$shot_zone_area <- as.factor(df$shot_zone_area)
df$shot_zone_basic <- as.factor(df$shot_zone_basic)
df$shot_zone_range <- as.factor(df$shot_zone_range)
df$team_name <- as.factor(df$team_name)
df$game_date <- as.factor(df$game_date)
df$matchup <- as.factor(df$matchup)
df$opponent <- as.factor(df$opponent)
df$shot_made_flag <- as.factor(df$shot_made_flag)
```

[3]:
```
str(df)
```

```
'data.frame':   30697 obs. of  25 variables:
 $ action_type       : Factor w/ 57 levels "Alley Oop Dunk Shot",..: 27 27 27 27
6 27 28 27 27 42 ...
 $ combined_shot_type: Factor w/ 6 levels "Bank Shot","Dunk",..: 4 4 4 4 2 4 5 4
4 4 ...
 $ game_event_id     : int  10 12 35 43 155 244 251 254 265 294 ...
 $ game_id           : int  20000012 20000012 20000012 20000012 20000012
20000012 20000012 20000012 20000012 20000012 ...
 $ lat               : num  34 34 33.9 33.9 34 ...
 $ loc_x             : int  167 -157 -101 138 0 -145 0 1 -65 -33 ...
 $ loc_y             : int  72 0 135 175 0 -11 0 28 108 125 ...
 $ lon               : num  -118 -118 -118 -118 -118 ...
 $ minutes_remaining : int  10 10 7 6 6 9 8 8 6 3 ...
 $ period            : int  1 1 1 1 2 3 3 3 3 3 ...
 $ playoffs          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ season            : Factor w/ 20 levels "1996-97","1997-98",..: 5 5 5 5 5 5 5
5 5 5 ...
```

```
 $ seconds_remaining : int  27 22 45 52 19 32 52 5 12 36 ...
 $ shot_distance     : int  18 15 16 22 0 14 0 2 12 12 ...
 $ shot_made_flag    : Factor w/ 2 levels "0","1": NA 1 2 1 2 1 2 NA 2 1 ...
 $ shot_type         : Factor w/ 2 levels "2PT Field Goal",..: 1 1 1 1 1 1 1 1 1
1 ...
 $ shot_zone_area    : Factor w/ 6 levels "Back Court(BC)",..: 6 4 3 5 2 4 2 2 4
2 ...
 $ shot_zone_basic   : Factor w/ 7 levels "Above the Break 3",..: 5 5 5 5 6 5 6
6 3 3 ...
 $ shot_zone_range   : Factor w/ 5 levels "16-24 ft.","24+ ft.",..: 1 3 1 1 5 3
5 5 3 3 ...
 $ team_id           : int  1610612747 1610612747 1610612747 1610612747
1610612747 1610612747 1610612747 1610612747 1610612747 1610612747 ...
 $ team_name         : Factor w/ 1 level "Los Angeles Lakers": 1 1 1 1 1 1 1 1 1
1 ...
 $ game_date         : Factor w/ 1559 levels "1996-11-03","1996-11-05",..: 311
311 311 311 311 311 311 311 311 311 ...
 $ matchup           : Factor w/ 74 levels "LAL @ ATL","LAL @ BKN",..: 29 29 29
29 29 29 29 29 29 29 ...
 $ opponent          : Factor w/ 33 levels "ATL","BKN","BOS",..: 26 26 26 26 26
26 26 26 26 26 ...
 $ shot_id           : int  1 2 3 4 5 6 7 8 9 10 ...
```

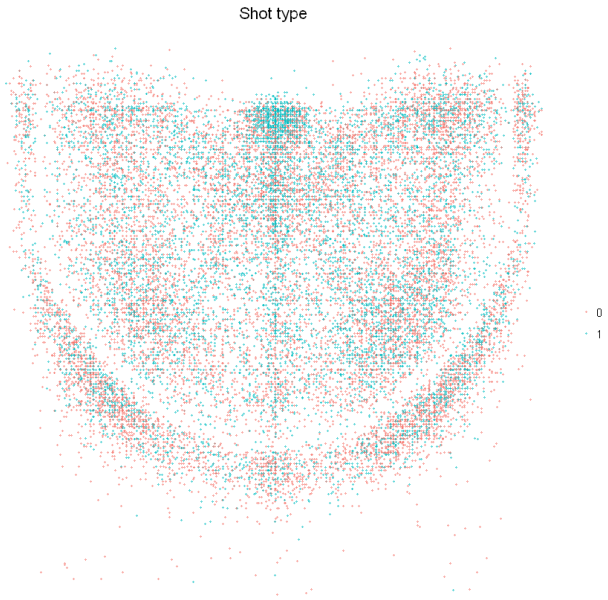
[4]:
```
summary(df)
```

```
              action_type    combined_shot_type game_event_id
 Jump Shot           :18880   Bank Shot:  141    Min.   :  2.0
 Layup Shot          : 2567   Dunk     : 1286    1st Qu.:110.0
 Driving Layup Shot  : 1978   Hook Shot:  153    Median :253.0
 Turnaround Jump Shot: 1057   Jump Shot:23485    Mean   :249.2
 Fadeaway Jump Shot  : 1048   Layup    : 5448    3rd Qu.:368.0
 Running Jump Shot   :  926   Tip Shot :  184    Max.   :659.0
 (Other)             : 4241
    game_id              lat            loc_x              loc_y
 Min.   :20000012   Min.   :33.25   Min.   :-250.000   Min.   :-44.00
 1st Qu.:20500077   1st Qu.:33.88   1st Qu.: -68.000   1st Qu.:  4.00
 Median :20900354   Median :33.97   Median :   0.000   Median : 74.00
 Mean   :24764066   Mean   :33.95   Mean   :   7.111   Mean   : 91.11
 3rd Qu.:29600474   3rd Qu.:34.04   3rd Qu.:  95.000   3rd Qu.:160.00
 Max.   :49900088   Max.   :34.09   Max.   : 248.000   Max.   :791.00

      lon         minutes_remaining     period         playoffs
 Min.   :-118.5   Min.   : 0.000    Min.   :1.000   Min.   :0.0000
 1st Qu.:-118.3   1st Qu.: 2.000    1st Qu.:1.000   1st Qu.:0.0000
 Median :-118.3   Median : 5.000    Median :3.000   Median :0.0000
 Mean   :-118.3   Mean   : 4.886    Mean   :2.519   Mean   :0.1466
 3rd Qu.:-118.2   3rd Qu.: 8.000    3rd Qu.:3.000   3rd Qu.:0.0000
 Max.   :-118.0   Max.   :11.000    Max.   :7.000   Max.   :1.0000
```

```
    season      seconds_remaining shot_distance   shot_made_flag
2005-06: 2318   Min.   : 0.00     Min.   : 0.00   0   :14232
2008-09: 2242   1st Qu.:13.00     1st Qu.: 5.00   1   :11465
2002-03: 2241   Median :28.00     Median :15.00   NA's: 5000
2007-08: 2153   Mean   :28.37     Mean   :13.44
2009-10: 2080   3rd Qu.:43.00     3rd Qu.:21.00
2001-02: 2028   Max.   :59.00     Max.   :79.00
(Other):17635
         shot_type                    shot_zone_area
2PT Field Goal:24271   Back Court(BC)       :   83
3PT Field Goal: 6426   Center(C)            :13455
                       Left Side Center(LC) : 4044
                       Left Side(L)         : 3751
                       Right Side Center(RC): 4776
                       Right Side(R)        : 4588

             shot_zone_basic         shot_zone_range    team_id
Above the Break 3    : 5620   16-24 ft.      :8315   Min.   :1.611e+09
Backcourt            :   71   24+ ft.        :6275   1st Qu.:1.611e+09
In The Paint (Non-RA): 4578   8-16 ft.       :6626   Median :1.611e+09
Left Corner 3        :  280   Back Court Shot:  83   Mean   :1.611e+09
Mid-Range            :12625   Less Than 8 ft.:9398   3rd Qu.:1.611e+09
Restricted Area      : 7136                          Max.   :1.611e+09
Right Corner 3       :  387
             team_name          game_date            matchup
Los Angeles Lakers:30697   2016-04-13:   50   LAL @ SAS  : 1020
                           2002-11-07:   47   LAL vs. SAS:  936
                           2006-01-22:   46   LAL @ SAC  :  889
                           2006-12-29:   45   LAL vs. HOU:  878
                           2007-03-30:   44   LAL @ DEN  :  873
                           2008-01-14:   44   LAL @ PHX  :  859
                           (Other)   :30421   (Other)    :25242
   opponent        shot_id
SAS    : 1978   Min.   :    1
PHX    : 1781   1st Qu.: 7675
HOU    : 1666   Median :15349
SAC    : 1643   Mean   :15349
DEN    : 1642   3rd Qu.:23023
POR    : 1539   Max.   :30697
(Other):20448
```

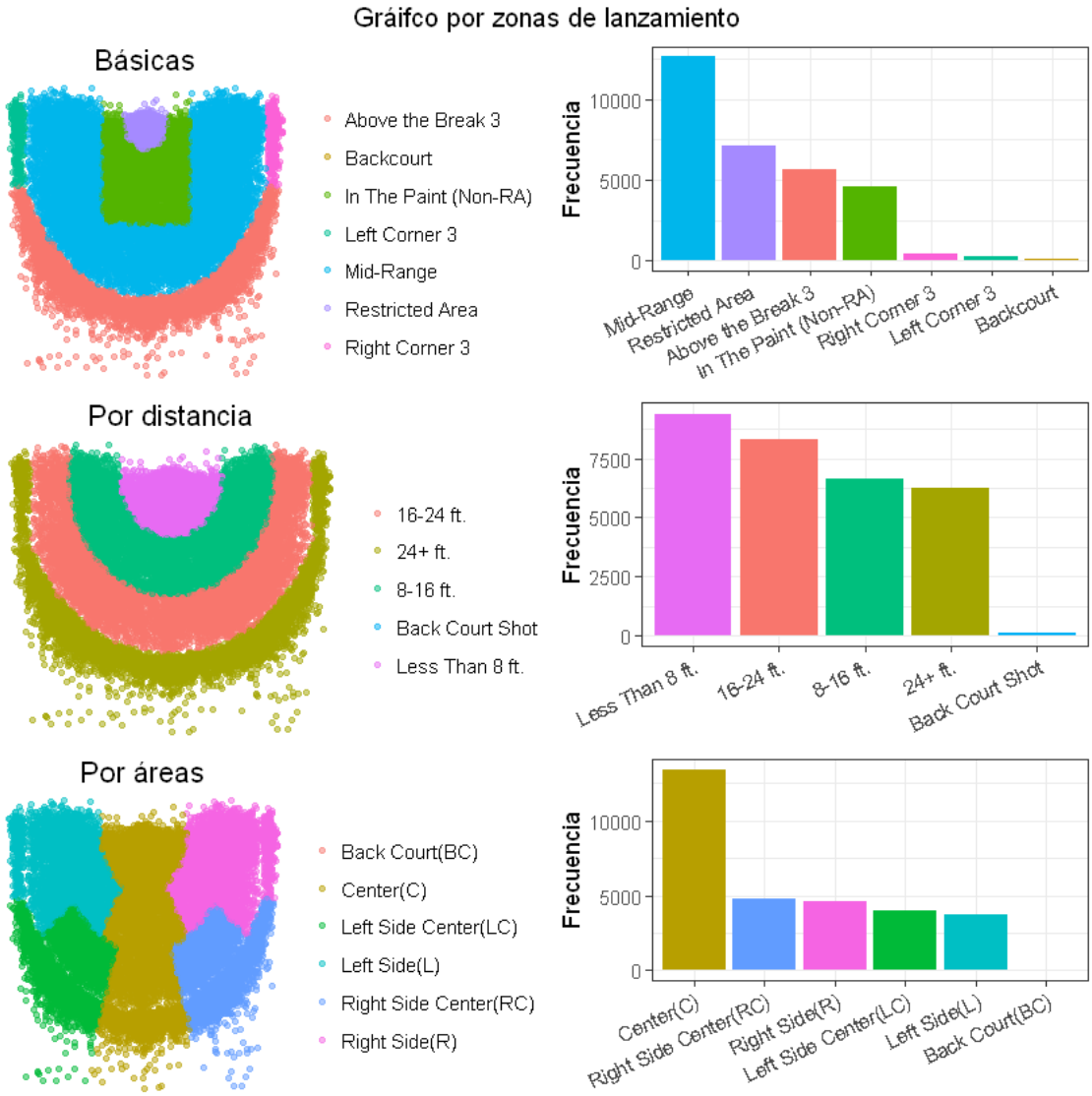
[5]:
```
lanz_graf <- ggplot(df[!is.na(df$shot_made_flag),], aes(x=lon, y=lat)) +
  geom_point(size = 0.5, alpha = 0.5, aes(color=shot_made_flag)) +
  labs(title="Shot type") +
  ylim(c(33.7, 34.0883)) +
  theme_void() +
```

```
  theme(legend.title=element_blank(),
        plot.title=element_text(hjust=0.5))
```



[6]:
```
g1 <- ggplot(df, aes(x=lon, y=lat)) +
  geom_point(size = 1, alpha = 0.5, aes(color=shot_zone_basic)) +
  labs(title="Básicas") +
  ylim(c(33.7, 34.0883)) +
  theme_void() +
  theme(legend.title=element_blank(),
        plot.title=element_text(hjust=0.5))
g2 <- ggplot(df, aes(x=lon, y=lat)) +
  geom_point(size = 1, alpha = 0.5, aes(color=shot_zone_range)) +
  labs(title="Por distancia") +
  ylim(c(33.7, 34.0883)) +
  theme_void() +
  theme(legend.title=element_blank(),
        plot.title=element_text(hjust=0.5))

g3 <- ggplot(df, aes(x=lon, y=lat)) +
  geom_point(size = 1, alpha = 0.5, aes(color=shot_zone_area)) +
  labs(title="Por áreas") +
  ylim(c(33.7, 34.0883)) +
  theme_void() +
  theme(legend.title=element_blank(),
        plot.title=element_text(hjust=0.5))

g4 <- ggplot(df, aes(x=fct_infreq(shot_zone_basic))) +
```

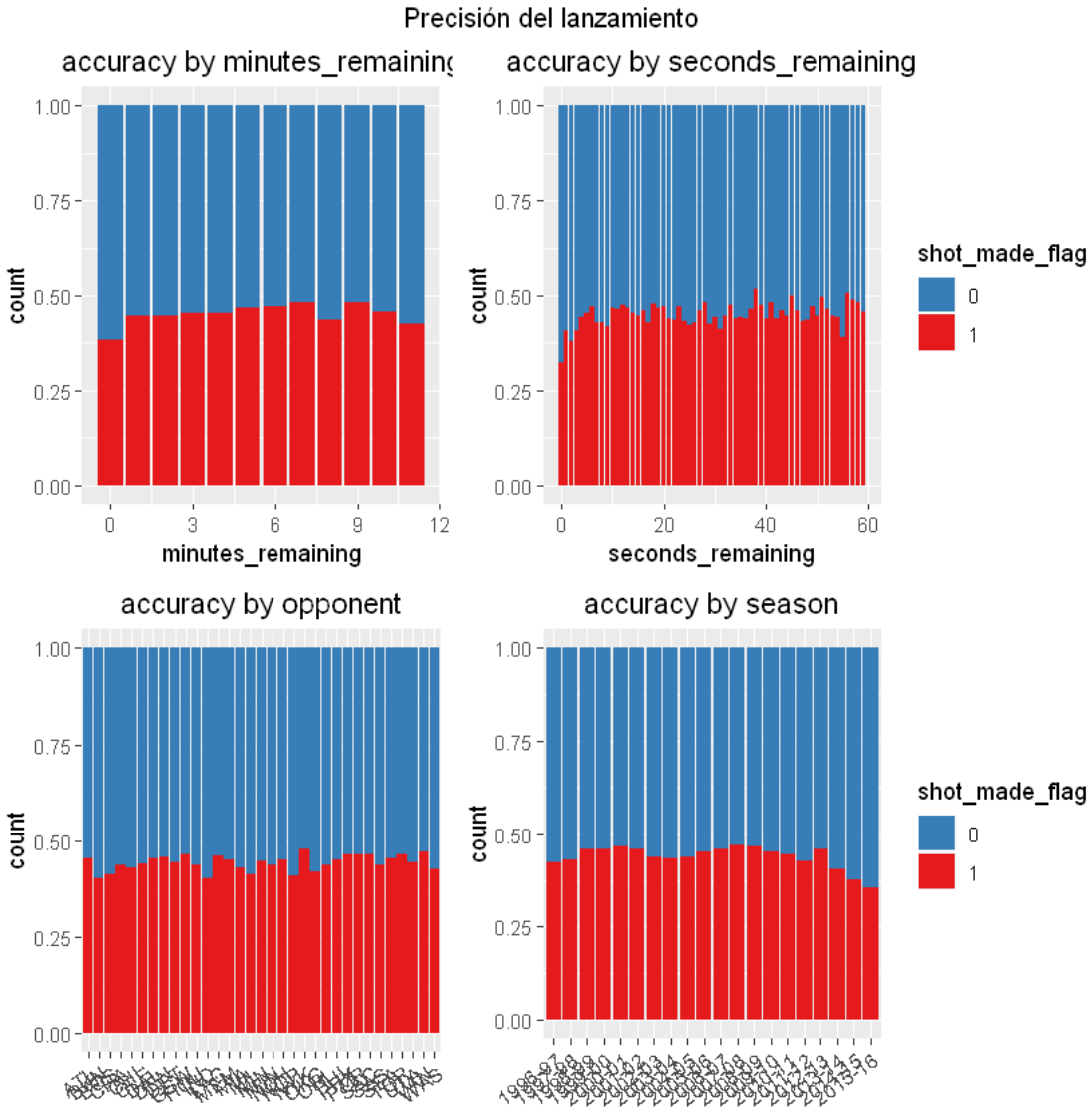
```
  geom_bar(aes(fill=shot_zone_basic)) +
  labs(y="Frecuencia") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1),
        axis.title.x=element_blank(),
        legend.position="none")

g5 <- ggplot(df, aes(x=fct_infreq(shot_zone_range))) +
  geom_bar(aes(fill=shot_zone_range)) +
  labs(y="Frecuencia") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1),
        axis.title.x=element_blank(),
        legend.position="none")

g6 <- ggplot(df, aes(x=fct_infreq(shot_zone_area))) +
  geom_bar(aes(fill=shot_zone_area)) +
  labs(y="Frecuencia") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1),
        axis.title.x=element_blank(),
        legend.position="none")

grid.arrange(g1, g4, g2, g5, g3, g6,
             top = "Gráifco por zonas de lanzamiento",
             layout_matrix=cbind(rbind(c(1, 2), c(3, 4), c(5,6))))
```



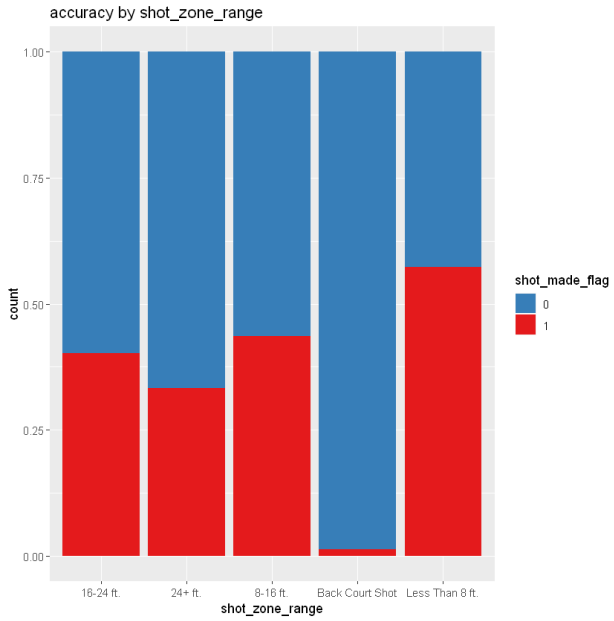
[7]:
```
pplot <- function(feat) {
    feat <- substitute(feat)
    ggplot(data = df[!is.na(df$shot_made_flag),], aes_q(x = feat)) +
    geom_bar(aes(fill = shot_made_flag), stat = "count", position = "fill") +
    scale_fill_brewer(palette = "Set1", direction = -1) +
    ggtitle(paste("accuracy by", feat))
    }
# Función editada de la función de Alexandru Papiu (https://www.kaggle.com/
 ↪apapiu/exploring-kobe-s-shots) con una
```

[8]:
```
g7 <- pplot(minutes_remaining) +
    theme(legend.position="none",
          plot.title=element_text(hjust=0.5))
```

```
g8 <-pplot(seconds_remaining)+
    theme(plot.title=element_text(hjust=0.5))

g9 <- pplot(opponent) +
    theme(axis.text.x = element_text(angle = 35, hjust = 1),
          axis.title.x=element_blank(),
          legend.position="none",
          plot.title=element_text(hjust=0.5))

g10 <- pplot(season) +
    theme(axis.text.x = element_text(angle = 35, hjust = 1),
          axis.title.x=element_blank(),
          plot.title=element_text(hjust=0.5))

grid.arrange(g7, g8, g9, g10,
             top = "Precisión del lanzamiento",
             layout_matrix=cbind(rbind(c(1, 2), c(3, 4))), widths =c(14,20))
```



[9]:
```
pplot(shot_zone_range)
```



[10]:
```
# eliminamos "team_id" y "team_name"
df <- select(df, -team_id, -team_name)

# eliminamos "matchup"
df <- select(df, -matchup)

# eliminamos "lon" y "lat"
df <- select(df, -lon, -lat)
```

[11]:
```
# Generamos una nueva variable de tiempos restante
df$time_remaining <- hms(with(df, paste(0, minutes_remaining,␣
 ↪seconds_remaining)))
df <- select(df, -minutes_remaining, -seconds_remaining)
```

[12]:
```
# Split DataFrame
df_train <- df[!is.na(df$shot_made_flag),]
df_test <- df[is.na(df$shot_made_flag),]

#Eliminamos la variable "shot_id" del conjunto de entrenamiento dado que no␣
 ↪proporcionara información para los modelos.
df_train <- select(df_train, -shot_id)
```

[13]:
```
# Método ranking
weights <- gain.ratio(shot_made_flag~., df_train)
weights=weights[order(weights$attr_importance,decreasing=TRUE),drop=F]
print(weights)
```

```
                   attr_importance
combined_shot_type      0.0457252803
action_type             0.0443522697
shot_distance           0.0184669916
shot_zone_basic         0.0154790433
loc_y                   0.0153972319
shot_type               0.0146291317
shot_zone_range         0.0129614062
loc_x                   0.0105297116
shot_zone_area          0.0077762085
game_date               0.0042831606
time_remaining          0.0037744865
game_event_id           0.0011515189
period                  0.0011403233
season                  0.0004332244
opponent                0.0001905921
game_id                 0.0000000000
playoffs                0.0000000000
```

[14]:
```
df_train <- select(df_train, -game_id, -playoffs)
```

[15]:
```
# Cross-validation 5 kflods
fitControl <- trainControl(method = "cv",
                           number = 5)

# Naive Bayes
set.seed(123)
naive_bayesFit <- train(shot_made_flag ~ ., data = df_train,
                        method = "naive_bayes",
                        trControl = fitControl,
                        verbose = FALSE)

# Árbol de decisión
set.seed(123)
DTFit <- train(shot_made_flag ~ ., data = df_train,
                        method = "C5.0",
                        trControl = fitControl,
                        verbose = FALSE)

# Vecinos más cercanos
set.seed(123)
kknnFit <- train(shot_made_flag ~ ., data = df_train,
                 method = "kknn",
                 trControl = fitControl,
                 verbose = FALSE)

# Support Vecto Machina (linear kernel)
```

```
set.seed(123)
svmLinearFit <- train(shot_made_flag ~ ., data = df_train,
                      method = "svmLinear",
                      trControl = fitControl,
                      verbose = FALSE)
```

[16]:
```
# RESULTADOS DE LOS MODELOS
tabla <- resamples(list(Naive_Bayes = naive_bayesFit,
                        Decision_Tree = DTFit,
                        Nearest_Neighbour = kknnFit,
                        SVM_Linear= svmLinearFit))
summary(tabla)
```

```
Call:
summary.resamples(object = tabla)

Models: Naive_Bayes, Decision_Tree, Nearest_Neighbour, SVM_Linear
Number of resamples: 5

Accuracy
                      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
Naive_Bayes       0.5538042 0.5538042 0.5538042 0.5538390 0.5538911 0.5538911
Decision_Tree     0.6769800 0.6775637 0.6791205 0.6804293 0.6824903 0.6859922
Nearest_Neighbour 0.5448531 0.5461089 0.5517510 0.5504535 0.5538042 0.5557501
SVM_Linear        0.6528507 0.6629694 0.6636187 0.6630344 0.6667315 0.6690018
                  NA's
Naive_Bayes          0
Decision_Tree        0
Nearest_Neighbour    0
SVM_Linear           0

Kappa
                        Min.    1st Qu.     Median       Mean    3rd Qu.
Naive_Bayes       0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
Decision_Tree     0.32185678 0.32458511 0.32679115 0.33018269 0.33499970
Nearest_Neighbour 0.06127401 0.06898287 0.06967463 0.07369084 0.07709389
SVM_Linear        0.28174277 0.30092187 0.30194333 0.30056316 0.30725757
                        Max. NA's
Naive_Bayes       0.00000000    0
Decision_Tree     0.34268069    0
Nearest_Neighbour 0.09142882    0
SVM_Linear        0.31095023    0
```

[17]:
```
# pred_DTFit_train <- predict(DTFit, df_train)
# pred_svmLinearFit_train <- predict(svmLinearFit, df_train)
```

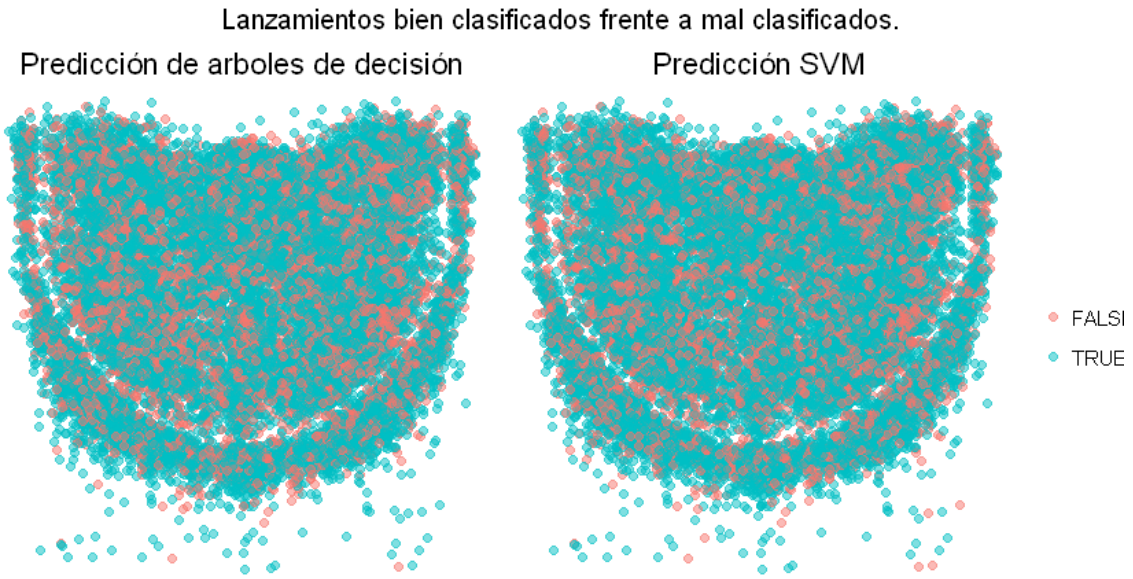
```r
print("Árbol de decisión")
print(confusionMatrix(as.factor(pred_DTFit_train), as.
 ↪factor(df_train$shot_made_flag)))
print("Support Vector Machine")
print(confusionMatrix(as.factor(pred_svmLinearFit_train), as.
 ↪factor(df_train$shot_made_flag)))
```

```
[1] "Árbol de decisión"
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 12110  6063
         1  2122  5402

               Accuracy : 0.6815
                 95% CI : (0.6757, 0.6872)
    No Information Rate : 0.5538
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3332

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8509
            Specificity : 0.4712
         Pos Pred Value : 0.6664
         Neg Pred Value : 0.7180
             Prevalence : 0.5538
         Detection Rate : 0.4713
   Detection Prevalence : 0.7072
      Balanced Accuracy : 0.6610

       'Positive' Class : 0

[1] "Support Vector Machine"
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 11866  5523
         1  2366  5942

               Accuracy : 0.693
                 95% CI : (0.6873, 0.6986)
    No Information Rate : 0.5538
```

```
   P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3617

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8338
            Specificity : 0.5183
         Pos Pred Value : 0.6824
         Neg Pred Value : 0.7152
             Prevalence : 0.5538
         Detection Rate : 0.4618
   Detection Prevalence : 0.6767
      Balanced Accuracy : 0.6760

       'Positive' Class : 0
```

[18]:
```r
# VISUALIZACIÓN DE LOS RESULTADOS
shot_made_flag <- df_train$shot_made_flag

pred_DTFit_train_GoodBad <-ifelse(pred_DTFit_train==LaMeteOno, TRUE, FALSE)
pred_svmLinearFit_train_GoodBad <-ifelse(pred_svmLinearFit_train==LaMeteOno,␣
↪TRUE, FALSE)
pred_svmRadialFit_train_GoodBad <-ifelse(pred_svmRadialFit_train==LaMeteOno,␣
↪TRUE, FALSE)

df_train <-cbind(df_train,pred_DTFit_train_GoodBad)
df_train <-cbind(df_train,pred_svmLinearFit_train_GoodBad)
df_train <-cbind(df_train,pred_svmRadialFit_train_GoodBad)

g11 <- ggplot(df_train, aes(x=lon, y=lat)) +
  geom_point(size = 1.5, alpha = 0.5, aes(color=pred_DTFit_train_GoodBad)) +
  labs(title="Predicción de arboles de decisión") +
  ylim(c(33.7, 34.0883)) +
  theme_void() +
  theme(legend.title=element_blank(),
        legend.position="none",
        plot.title=element_text(hjust=0.5))
g12 <- ggplot(df_train, aes(x=lon, y=lat)) +
  geom_point(size = 1.5, alpha = 0.5,␣
↪aes(color=pred_svmLinearFit_train_GoodBad)) +
  labs(title="Predicción SVM") +
  ylim(c(33.7, 34.0883)) +
  theme_void() +
  theme(legend.title=element_blank(),
        plot.title=element_text(hjust=0.5))
```

```
grid.arrange(g11, g12, ncol =2, widths = c(16, 20), heights = c(15,15),
             top = "Lanzamientos bien clasificados frente a mal clasificados.")
```



[19]:
```
pred_DTFit <- predict(DTFit, df_test)
samples <- data.frame(shot_id = df_test$ shot_id, shot_made_flag = pred_DTFit)
write.csv(samples,file = "sumbissionDTF.csv", row.names = F)

pred_svmLinearFit <- predict(svmLinearFit, df_test)
samples <- data.frame(shot_id = df_test$ shot_id, shot_made_flag =␣
 ↪pred_svmLinearFit)
write.csv(samples,file = "sumbissionSVMLinear.csv", row.names = F)
```