Classification Case

Overview

The aim of this work is to predict the gender class and evaluate the algorithms which are used. First the libraries we are going to use are installed to the environment.

```
library(ggplot2)
library(caret)
library(gridExtra)
library(randomForest)
```

The dataset is loaded and summary about the dataset is shown.

```
data <- read.csv("dataset.CSV", header = T)
str(data)
```

```
## 'data.frame':    10000 obs. of  3 variables:
##  $ Gender: Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Height: num  73.8 68.8 74.1 71.7 69.9 ...
##  $ Weight: num  242 162 213 220 206 ...
```

```
prop.table(table(data$Gender))
```

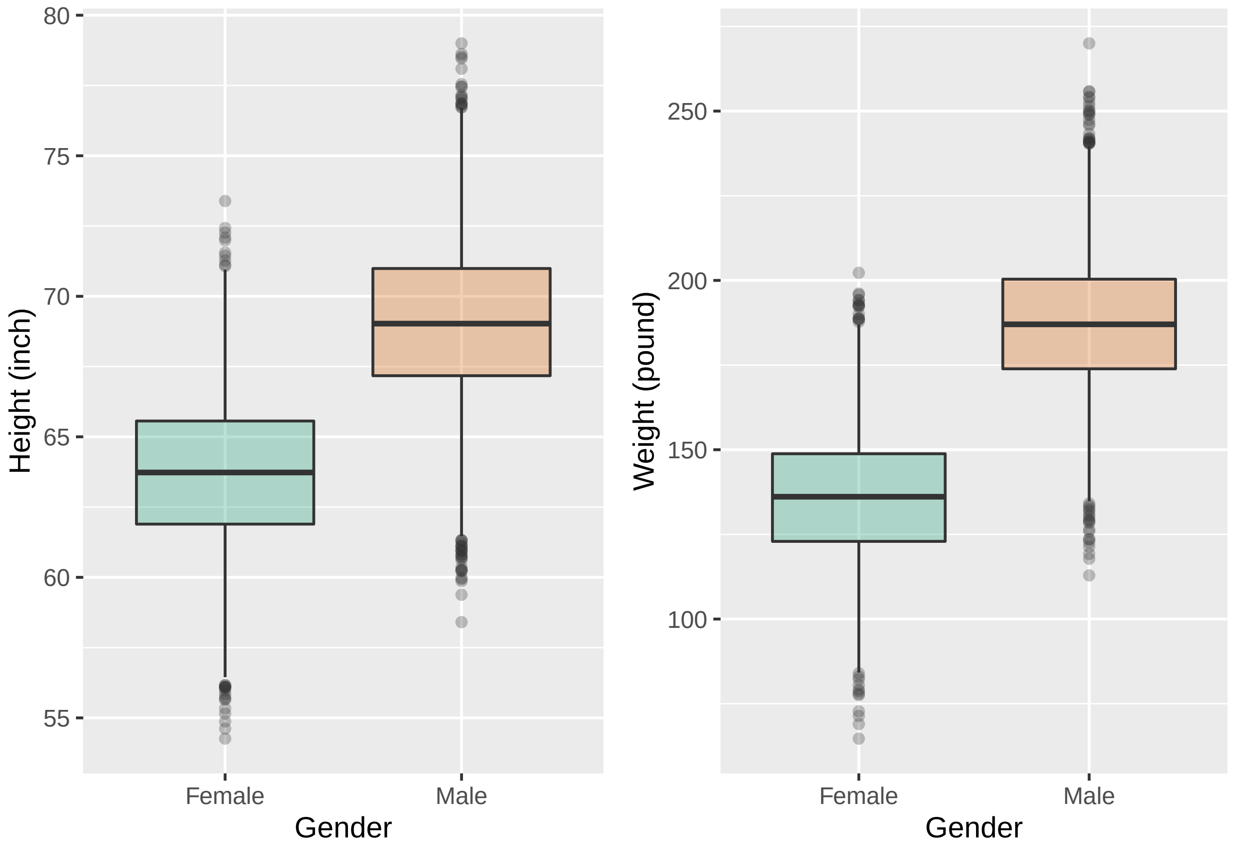
```
##
## Female   Male
##    0.5    0.5
```

Stastics of the dataset's features are shown on the boxplots below.

```
plot1 <- ggplot(data, aes(x=Gender, y=Height, fill=Gender)) +
  ylab("Height (inch)") +
  geom_boxplot(alpha=0.3) +
  theme(legend.position="none") +
  scale_fill_brewer(palette="Dark2")

plot2 <- ggplot(data, aes(x=Gender, y=Weight, fill=Gender)) +
  ylab("Weight (pound)") +
  geom_boxplot(alpha=0.3) +
  theme(legend.position="none") +
  scale_fill_brewer(palette="Dark2")

grid.arrange(plot1, plot2, ncol=2)
```



The dataset is split into train and test sets by half and half.

```
intrain <- createDataPartition(y=data$Gender,p=0.5,list=FALSE)
train <- data[intrain,]
test <- data[-intrain,]
```

Proportions of train and test sets' gender are found to be same.

```
prop.table(table(train$Gender))
```

```
##
## Female   Male
##    0.5    0.5
```

```
prop.table(table(test$Gender))
```

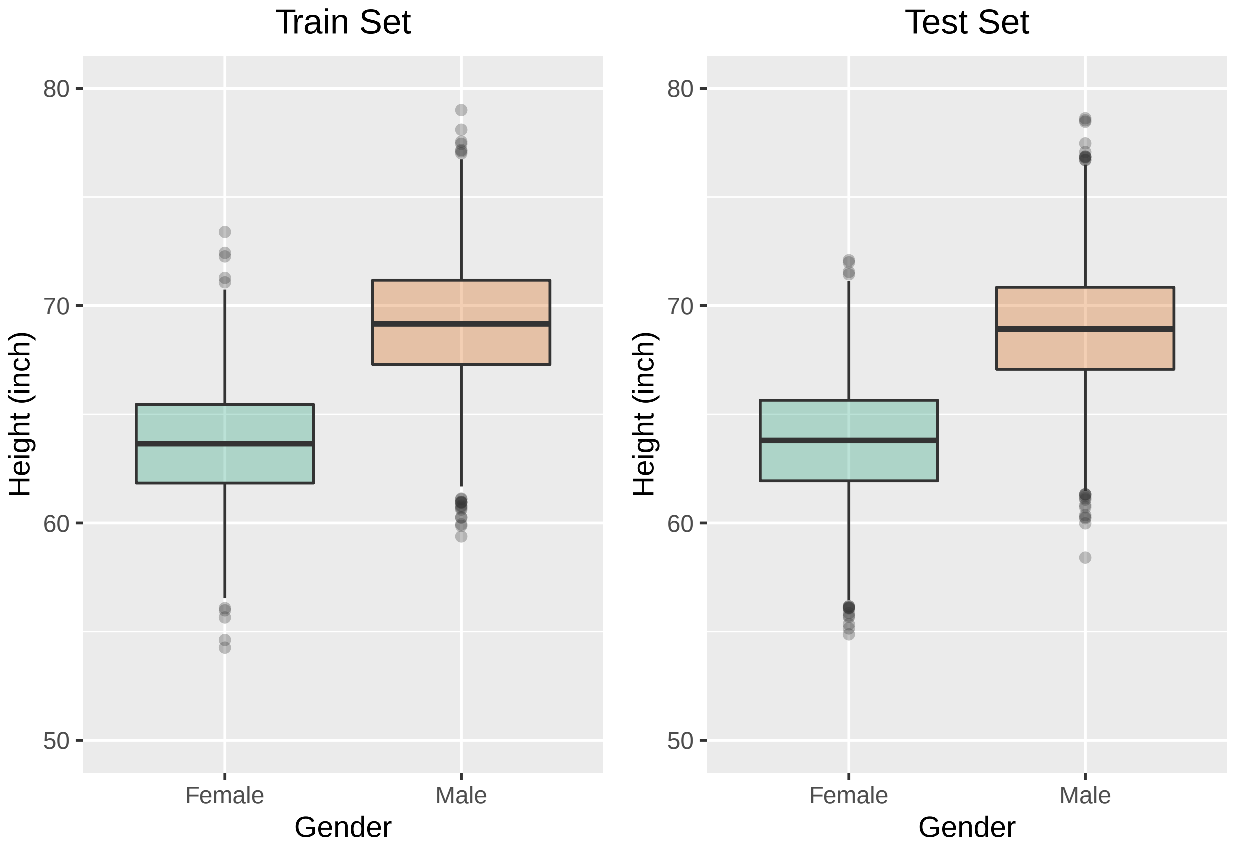
```
##
## Female   Male
##    0.5    0.5
```

Boxplots below also demonstrate that distribution of 'Height' is similar in both sets.

```
plot3 <- ggplot(train, aes(x=Gender, y=Height, fill=Gender)) +
  ggtitle("Train Set") +
  ylab("Height (inch)") +
  ylim(50,80) +
  geom_boxplot(alpha=0.3) +
  theme(legend.position="none", plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette="Dark2")

plot4 <- ggplot(test, aes(x=Gender, y=Height, fill=Gender)) +
  ggtitle("Test Set") +
  ylab("Height (inch)") +
  ylim(50,80) +
  geom_boxplot(alpha=0.3) +
  theme(legend.position="none", plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette="Dark2")

grid.arrange(plot3, plot4, ncol=2)
```

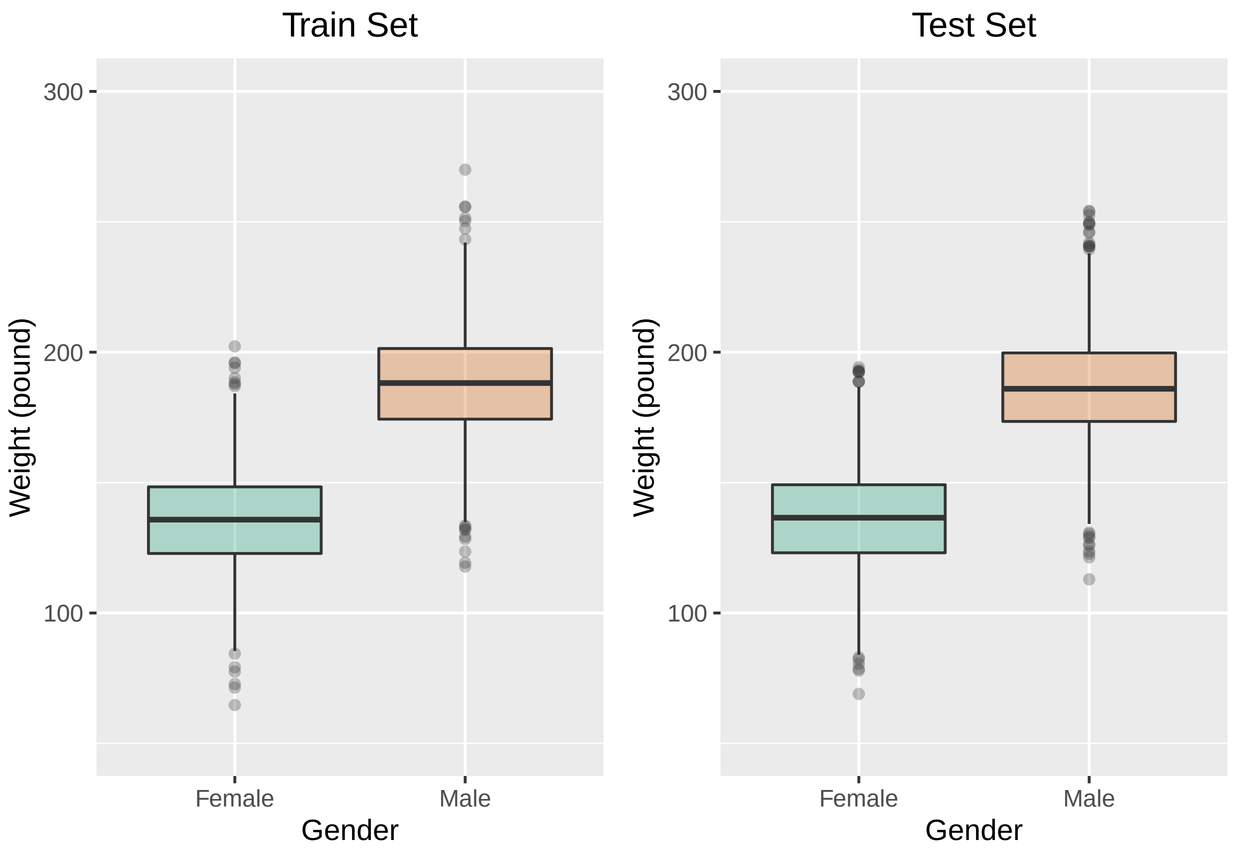


Boxplots below also demonstrate that distribution of ‘Weight’ is similar in both sets.

```
plot5 <- ggplot(train, aes(x=Gender, y=Weight, fill=Gender)) +
  ggtitle("Train Set") +
  ylab("Weight (pound)") +
  ylim(50,300) +
  geom_boxplot(alpha=0.3) +
  theme(legend.position="none", plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette="Dark2")

plot6 <- ggplot(test, aes(x=Gender, y=Weight, fill=Gender)) +
  ggtitle("Test Set") +
  ylab("Weight (pound)") +
  ylim(50,300) +
  geom_boxplot(alpha=0.3) +
  theme(legend.position="none", plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette="Dark2")

grid.arrange(plot5, plot6, ncol=2)
```



Genders for the test dataset using different probabilities are predicted randomly. Predictions are stored in a dataframe.

```
mat <- matrix(nrow = 5000, ncol = 0)
for (p in (1:9)/10) {
  mat <- cbind(mat,sample(c("Male", "Female"), nrow(test), replace = TRUE, prob=c(p, 1-p)))
}

df <- as.data.frame(mat)
```

Accuracy and F1 score of each prediction set is stored in a matrix.

```
accuracy <- matrix(nrow = 0, ncol = 2)
for (i in 1:9) {
  tp <- table(test$Gender,df[,i])[2,2]
  tn <- table(test$Gender,df[,i])[1,1]
  fp <- table(test$Gender,df[,i])[1,2]
  fn <- table(test$Gender,df[,i])[2,1]
  accuracy <- rbind(accuracy, c(i/10,((tp+tn)/(tp+tn+fp+fn))))
}

f1_score <- matrix(nrow = 0, ncol = 1)
for (i in 1:9) {
  tp <- table(test$Gender,df[,i])[2,2]
  tn <- table(test$Gender,df[,i])[1,1]
  fp <- table(test$Gender,df[,i])[1,2]
  fn <- table(test$Gender,df[,i])[2,1]
  precision <- tp/(tp+fp)
  recall <- tp/(tp+fn)
  f1_score <- rbind(f1_score, 2*(precision*recall)/(precision+recall))
}
```

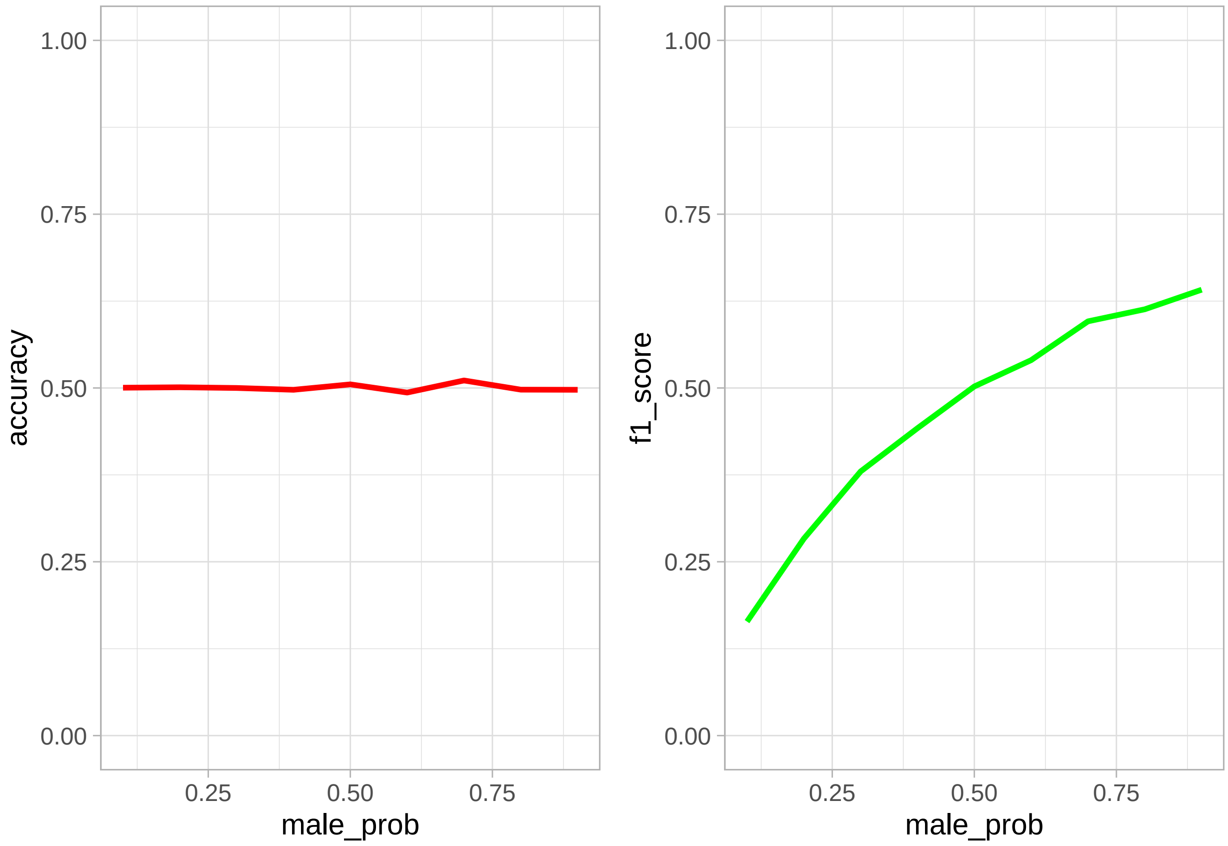
Aacuracy and F1 Score of each randomly predicted set is plotted below. Positive class is "Male". In some cases, true positive cases are more valuable than the true negative cases. Accuracy does not take that into account whereas F1 Score gives higher weight to positive cases. That' why i the plots, accuracy does not change throughout the different prediction sets. On the other hand, F1 score increases while probability of being male increases.

```
perf_df <- as.data.frame(cbind(accuracy,f1_score))
perf_df <- setNames(perf_df, c("male_prob","accuracy","f1_score"))

plot7 <- ggplot(perf_df, aes(x = male_prob, y = accuracy)) +
  geom_line(color = "red", size = 1) +
  ylim(0,1) +
  theme_light()

plot8 <- ggplot(perf_df, aes(x = male_prob, y = f1_score)) +
  geom_line(color = "green", size = 1) +
  ylim(0,1) +
  theme_light()

grid.arrange(plot7, plot8, ncol=2)
```


accuracy
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
male_prob
f1_score
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
male_prob