Machine Learning Model for Mall Customers

Suliman Alkhalifa

15 April 2019

1. Introcution

This report includes the structure, analysis and techniques used to present my 'Choose Your Own Project' as part of the capstone course.

1.1 Dataset

For this project, I will be using the mall customer segmentation data which is provided publicly from Kaggle.com. This includes data for 200 customers collecting their information like gender, age, annual income and spending score.

1.2 Project Overview

For this project, I will build a machine learning model that predicts the gender of the customers who visit the mall. This can help businesses to modify their production strategies based on majority of males or females visiting their stores. I will use different models and compare their results to select the best one.

2. Prepare Data

This section includes libraries used, downloaded dataset and creating validation set.

2.1 Load Libraries

The libraries I will use are the following:

```
library(tidyverse)
library(ggplot2)
library(caret)
library(e1071)
library(randomForest)
library(RCurl)
```

2.2 Load dataset

I downloaded the dataset from kaggle.com and saved it to my Github repository. To download it from there, I will use the following code:

```
dl <- getURL(
  "https://raw.githubusercontent.com/sulimankhalifa/Choose-Your-Own-Project/master/Mall_Customers.csv")
```

I will read the dataset into R using the following code:

```
my_dataset <- read.csv(textConnection(dl))
```

I will change the columns' names to simplify them:

```
colnames(my_dataset) <- c("customerID", "gender", "age", "annual_income", "spending_score")
```

2.3 Create validation set

I will split the dataset into 80% to use in the training set and 20% to use in the testing set:

```
set.seed(1)
test_index <- createDataPartition(my_dataset$gender, p=0.80, list = FALSE)
test_set <- my_dataset[-test_index, ]
train_set <- my_dataset[test_index, ]
```

Training set has 161 objects while testing set has 39 objects.

3. Explore Data

This section includes dimensions of our data, types of attributes and summary.

3.1 Dimensions of Dataset

We can look at how many rows and columns in our dataset:

```
dim(train_set)
```

```
## [1] 161   5
```

We have 161 rows and 5 columns in our data.

3.2 Types of Attributes

To know more about our dataset, we can look at the class of each attribute:

```
sapply(train_set, class)
```

```
##     customerID         gender            age  annual_income spending_score
##      "integer"       "factor"      "integer"      "integer"      "integer"
```

All columns have integer values except for gender which is a factor.

3.3 Peek at the Dataset

We can take a peek at the first 6 rows of dataset:

```
head(train_set)
```

```
##   customerID gender age annual_income spending_score
## 1          1   Male  19            15             39
## 2          2   Male  21            15             81
## 4          4 Female  23            16             77
## 6          6 Female  22            17             76
## 7          7 Female  35            18              6
## 8          8 Female  23            18             94
```

3.4 Class Distribution

Now, we will look at the percentage of gender:

```
percentage <- prop.table(table(train_set$gender)) * 100
cbind(freq = table(train_set$gender), percentage = percentage)
```

```
##        freq percentage
## Female   90   55.90062
## Male     71   44.09938
```

We have around 56% females and 44% males.

3.5 Statistical Summary

Finally, we can look at the summary of our dataset:

```
summary(train_set)
```

```
##    customerID       gender        age        annual_income
##  Min.   :  1.0   Female:90   Min.   :18.00   Min.   : 15.00
##  1st Qu.: 54.0   Male  :71   1st Qu.:28.00   1st Qu.: 43.00
##  Median : 99.0               Median :36.00   Median : 61.00
##  Mean   : 99.8               Mean   :38.89   Mean   : 60.34
##  3rd Qu.:148.0               3rd Qu.:49.00   3rd Qu.: 77.00
##  Max.   :200.0               Max.   :70.00   Max.   :137.00
##  spending_score
##  Min.   : 1.00
##  1st Qu.:36.00
##  Median :51.00
##  Mean   :51.45
##  3rd Qu.:73.00
##  Max.   :98.00
```

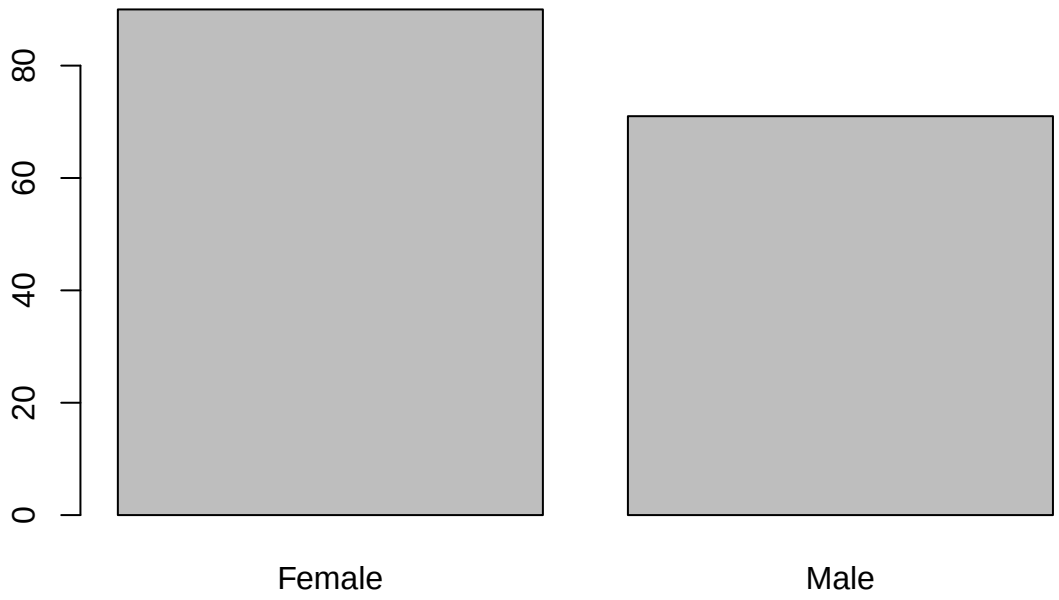
4. Visualize Dataset

In this section, we will use different plots/graphs to visualize our dataset.

4.1 Distribution of Gender

We can see that there are more females visiting the mall than males.

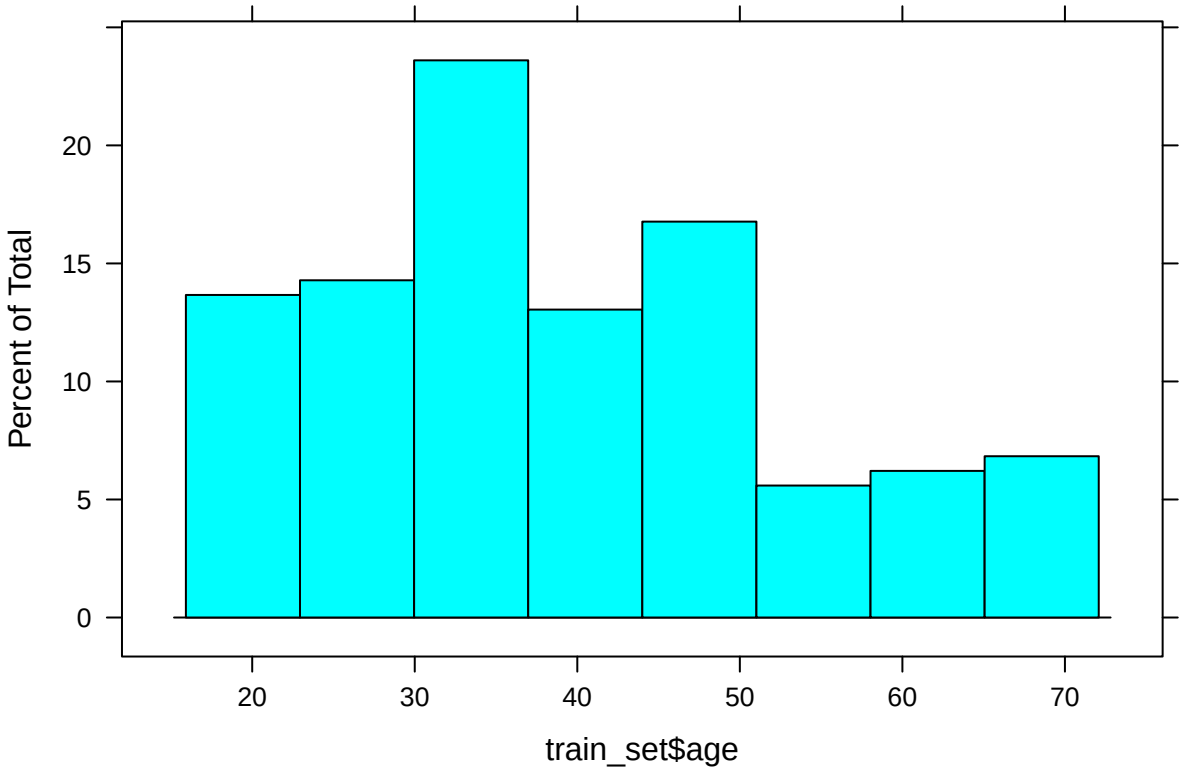
```
plot(train_set$gender)
```



4.2 Distribution of Age

Majority of customers are less than 50 years-old. Top customers are between 30 and 40 years-old.

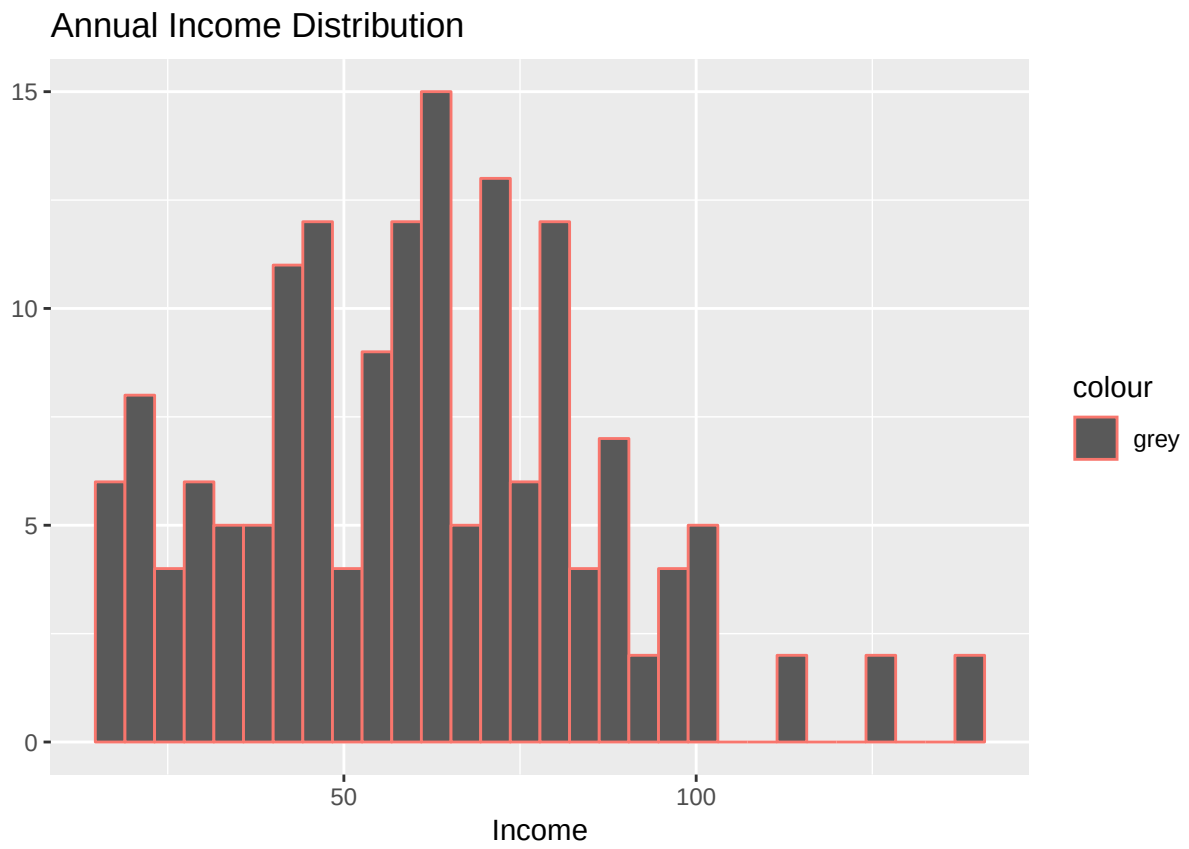
```
histogram(train_set$age)
```



4.3 Distribution of Annual Income

Most customers who visit the mall have annual income of $50,000 to $100,000.

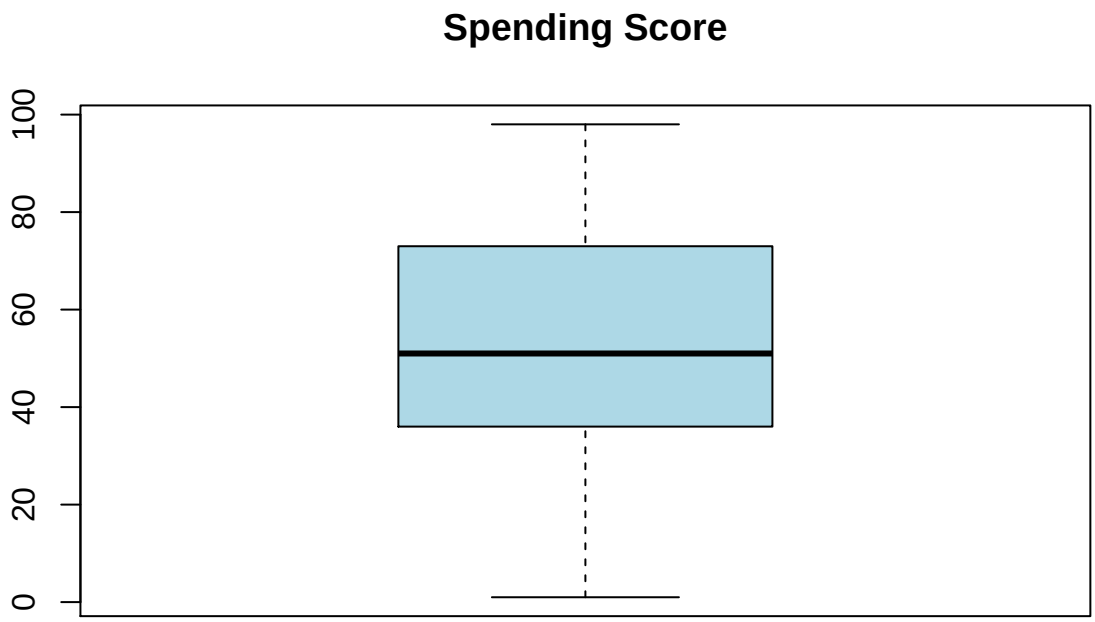
```
qplot(train_set$annual_income, bins = 30, col = "grey",
      main = "Annual Income Distribution", xlab = "Income")
```



4.4 Spending Score Visual

The average spending score is about 50. Maximum spending score is 98 and minimum is 1. Range of spending score is between 40 and 70.

```
boxplot(as.numeric(train_set$spending_score), main = "Spending Score", col = "light blue")
```



4.5 Top Spending Customers

This shows a list of top spending customers from the dataset.

```
train_set %>% group_by(customerID) %>%
  summarize(spending_score) %>%
  arrange(desc(spending_score))
```

```
## # A tibble: 161 x 2
##    customerID spending_score
##         <int>          <int>
##  1         20             98
##  2        186             97
##  3        128             95
##  4        168             95
##  5          8             94
##  6        164             93
##  7         34             92
##  8        174             92
##  9        194             91
## 10        150             90
## # ... with 151 more rows
```

5. Evaluate some Algorithms

In this section, we will create some models and select the best results.

5.1 Test Harness

We will 10-fold crossvalidation to estimate accuracy. This will split our dataset into 10 parts, 9 in the training set and 1 in the testing set.

```
control <- trainControl(method = "cv", number = 10)
metric <- "Accuracy"
```

We are using metric "Accuracy" to evaluate our models and select the best one.

5.2 Build Models

We will evaluate 4 models:

- Linear Discriminant Analysis (LDA)
- k-Nearest Neighbors (kNN)
- Random Forest (RF)
- Classifaction and Regression Trees (CART)

This is a good mixture of a simple linear (LDA), nonlinear (CART, kNN) and complex nonlinear (RF) models.

We reset the seed number for each model to ensure the evaluation is performed using the same dataset.

5.2.1 Linear Discriminant Analysis (LDA)

```
set.seed(7)
fit.lda <- train(gender~., data = train_set, method = "lda", metric = metric, trControl = control)
```

5.2.2 k-Nearest Neighbors (kNN)

```
set.seed(7)
fit.knn <- train(gender~., data = train_set, method = "knn", metric = metric, trControl = control)
```

5.2.3 Random Forest (RF)

```
set.seed(7)
fit.rf <- train(gender~., data = train_set, method = "rf", metric = metric, trControl = control)
```

5.2.4 Classifaction and Regression Trees (CART)

```
set.seed(7)
fit.cart <- train(gender~., data = train_set, method = "rpart", metric = metric, trControl = control)
```

5.3 Select Best Model

After we have used our 4 models, we can now look at the results using the following code:

```
results <- resamples(list(lda = fit.lda, knn = fit.knn, rf = fit.rf, cart = fit.cart))
summary(results)
```

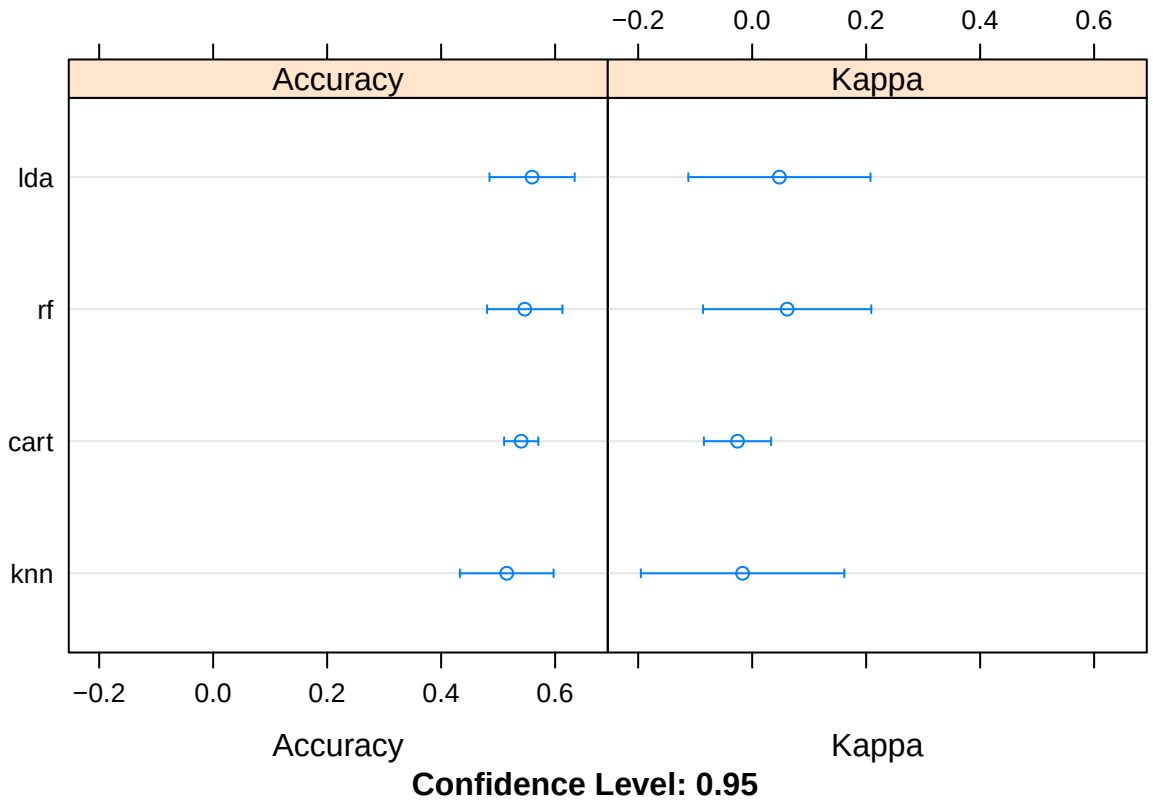
```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, knn, rf, cart
## Number of resamples: 10
##
## Accuracy
```

```
##        Min.   1st Qu.    Median      Mean   3rd Qu.   Max. NA's
## lda  0.4375 0.5000000 0.5000000 0.5595588 0.6250000 0.7500    0
## knn  0.3125 0.4531250 0.5312500 0.5150735 0.5818015 0.6875    0
## rf   0.4375 0.5000000 0.5147059 0.5466912 0.6093750 0.6875    0
## cart 0.4375 0.5376838 0.5625000 0.5404412 0.5625000 0.5625    0
##
## Kappa
##             Min.     1st Qu.       Median        Mean   3rd Qu.       Max.
## lda  -0.2413793 -0.08474576 -0.066963045  0.04773962 0.1993097 0.45762712
## knn  -0.4666667 -0.16229508  0.025396825 -0.01679162 0.1321072 0.35483871
## rf   -0.2413793 -0.01587302  0.006349206  0.06142836 0.2274038 0.35483871
## cart -0.2413793  0.00000000  0.000000000 -0.02571596 0.0000000 0.03448276
##      NA's
## lda     0
## knn     0
## rf      0
## cart    0
```

We can see the accuracy for each model and other metrics like Kappa.

I will create a plot of the results as follow:

```
dotplot(results)
```



We can see the most accurate model (i.e. LDA) from this plot. Although random forest model (RF) has very close result.

6. Make Predictions

As the LDA model has the highest accuracy, we want to test that using our testing (validation) set. We will also use a summary of our predictions using confusionMatrix function.

```
predictions <- predict(fit.lda , test_set)
confusionMatrix(predictions, test_set$gender)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Female Male
##     Female     14   14
##     Male        8    3
##
##                Accuracy : 0.4359
##                  95% CI : (0.2781, 0.6038)
##     No Information Rate : 0.5641
##     P-Value [Acc > NIR] : 0.9616
##
##                   Kappa : -0.195
##
##  Mcnemar's Test P-Value : 0.2864
##
##             Sensitivity : 0.6364
##             Specificity : 0.1765
##          Pos Pred Value : 0.5000
##          Neg Pred Value : 0.2727
##              Prevalence : 0.5641
##          Detection Rate : 0.3590
##    Detection Prevalence : 0.7179
##       Balanced Accuracy : 0.4064
##
##        'Positive' Class : Female
##
```

We can see that our accuracy is about 43% since this was a small testing set.

7. Conclusion

Using machine learning in marketing segmentation is quite popular and critical these days. Businesses are constantly trying to identify success factors to make more profits. The results from the models I used to predict the gender of customers showed different accuracy measures. The best model was the LDA where accuracy is 56% but this is low due to small set of data. As we collect more data and apply the models again we might see improvements.