ITWS - 6600 Data Analytics

Assignment 7

Submitted by: Shrey Jain (RIN: 662046332)

Dataset1: Absenteeism at Work

Q.1 For the first dataset, I first looked at the dataset using the view function in R. Then I found that the target column is Absenteeism time (hrs). I used IQR method to remove outliers from this dataset. Using IQR, the **lower bound was -7 and higher bound was 17**. There was a total of **44 outlier entries** in the dataset.

Figures 1 and 2 below are the histograms of dataset with and without these 44 outliers. There are a few values that are more than 20. Since the higher bound is 17, all the values in the right seem to be in that range.

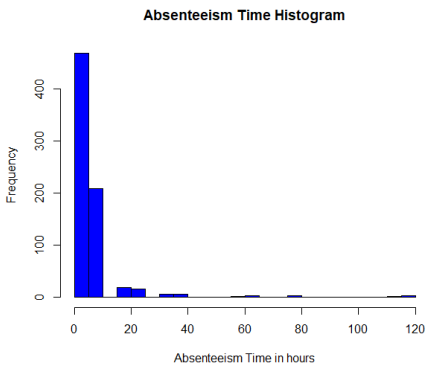

Figure 1: Histogram with outliers

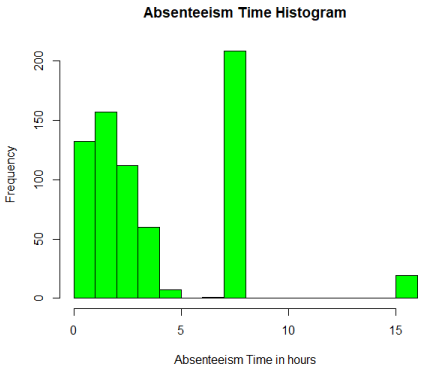

Figure 2: Histogram without outliers

Figures 3 and 4 are the boxplots of the same with and without outliers. It can again be seen that how important it is to identify and remove outliers, and in addition what histograms can not provide to us which boxplots do.

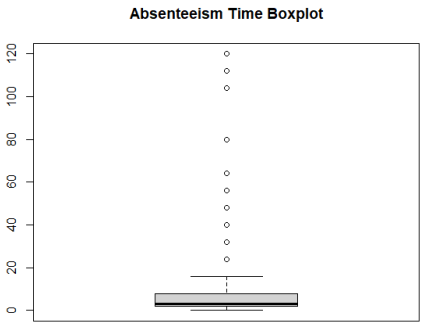

Figure 3: Boxplot with outliers

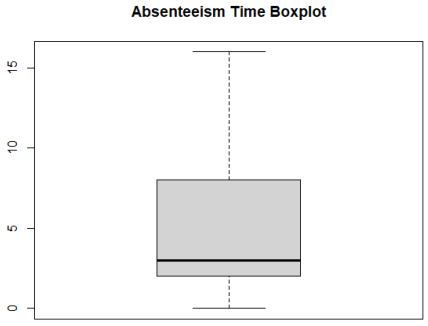

Figure 4: Boxplot without outliers

The code for this part of the question is attached below for reference.

```
# Question 1: EDA
#IQR
range <- IQR(dataset1$Absenteeism.time.in.hours)
range
Tmin = fivenum(dataset1$Absenteeism.time.in.hours)[2] - (1.5*range)
Tmax = fivenum(dataset1$Absenteeism.time.in.hours)[4] + (1.5*range)
Tmin
Tmax
# Finding outlier
length(dataset1$Absenteeism.time.in.hours[which(dataset1$Absenteeism.time.in.hours <= Tmin
| dataset1$Absenteeism.time.in.hours >= Tmax)])
# Removing outlier
dataset1_without_outlier <- dataset1[dataset1$Absenteeism.time.in.hours >= Tmin &
dataset1$Absenteeism.time.in.hours <= Tmax,]
# Histograms
hist(dataset1$Absenteeism.time.in.hours, main="Absenteeism Time Histogram",
xlab="Absenteeism Time in hours", col="blue", breaks = 20)
hist(dataset1_without_outlier$Absenteeism.time.in.hours, main="Absenteeism Time Histogram",
xlab="Absenteeism Time in hours", col="green", breaks = 20)
# Boxplots
boxplot(dataset1$Absenteeism.time.in.hours, main='Absenteeism Time Boxplot')
boxplot(dataset1_without_outlier$Absenteeism.time.in.hours, main='Absenteeism Time
Boxplot')
```

Q.2 As the question asked to choose between 3 models for the 6000 level students, I chose the following three:

- **Linear Regression:** Firstly, I chose the training and testing split of 70-30%. Then I used the lm function to build the model where the target column was absenteeism time in hours. To have a better comparison I chose to apply linear regression to both the dataset, one that is without outlier and the one that is with outlier. And much to my surprise, the results were astounding. The root mean square error of the linear regression model for dataset with outliers was about **9.697248** while that of the model for data without outliers was about **2.863149.**
- **Random Forest:** Second choice of model was random forest regression. The dataset was again split for 70% training and 30% testing. I used the RMS value to better understand the model that was built. The RMS value came out to be **2.756696.** It seems that random forest did perform well than linear regression. One thing to note is that I only used the dataset without outlier to build this model.
- **KNN:** This is my first-time hands on with this model. Since this was my first usage of the model, I had no much expectation from the results. I followed the same steps as the other two models. First split the data into training and testing and then performed model building. The RMS value for this model came out to be significantly lower that the other two, which is **0.7905544**

The code for this part of the question is attached below for reference.

```
# Question 2: Model Development
# Linear regression
```

```
training_dataset1 <- sample(dim(dataset1)[1], 0.7*dim(dataset1)[1])
training_dataset2 <- sample(dim(dataset1_without_outlier)[1],
0.7*dim(dataset1_without_outlier)[1])
linear_model1 <- lm(dataset1$Absenteeism.time.in.hours~., data=dataset1, subset =
training_dataset1)
linear_model2 <- lm(dataset1_without_outlier$Absenteeism.time.in.hours~.,
data=dataset1_without_outlier, subset = training_dataset2)
summary(linear_model1)
summary(linear_model2)
coef(linear_model1)
coef(linear_model2)
sqrt(mean((dataset1$Absenteeism.time.in.hours-predict(linear_model1, dataset1))[-
training_dataset1]^2))
sqrt(mean((dataset1_without_outlier$Absenteeism.time.in.hours-predict(linear_model2,
dataset1_without_outlier))[-training_dataset2]^2))

# Random Forest
training_dataset3 <- dataset1_without_outlier[training_dataset2,]
random_forest_model <- randomForest(Absenteeism.time.in.hours~., data=training_dataset3)
test_x <- dataset1_without_outlier[-training_dataset2, -dim(dataset1_without_outlier)[2]]
test_y <- dataset1_without_outlier[-training_dataset2, dim(dataset1_without_outlier)[2]]
random_forest_rms <- sqrt(mean((test_y - predict(random_forest_model, test_x))^2))
random_forest_rms
plot(sqrt(random_forest_model$mse), main = "Root mean square error plot using absenteeism
dataset for random forest", xlab = "Number of trees", ylab = "RMS")

# KNN regression model
training_dataset4 <- dataset1_without_outlier[training_dataset2,]
train_1 <- dataset1_without_outlier[training_dataset2,]
test_1 <- dataset1_without_outlier[-training_dataset2,]
knn_model <- knn(train=train_1, test=test_1, k=sqrt(nrow(dataset1_without_outlier)), cl =
train_1$Absenteeism.time.in.hours)
knn_rms <- mean(train_1$Absenteeism.time.in.hours != knn_model)
knn_rms
```

Q.3 The following can be assumed after performing analysis and model development on the dataset:

- The summary function and histograms were helpful, especially for identifying patterns or trends and to get the general idea about the data I am working on.
- Removal of NA rows might not be the correct way for cleaning the data, a better approach could be to fill those missing values using various techniques.
- Linear regression may not be the best model choice to predict absenteeism time. That is because the data is scattered in a non-linear manner. Multivariate regression could be a better option for this case.
- Looking at all the three models, linear regression performed the worst.
- Outlier removal is very important to perform accurate prediction, as we saw in the case of linear regression.
- Random forest and KNN both performed well, however, KNN performed better in this case as it had the lowest root mean square error.

Dataset2: Obesity levels dataset

Q.1 For the second dataset, I first looked at the dataset using the view function in R. The dataset has 2111 rows and 17 columns. Then I found that the column of interest was **NObeyesdad**. I used the unique function to identify different possible values of this column. IT turns out there are 7 different possible values of this variable which are as mentioned in Figure 5 below:

```
> unique(dataset2$NObeyesdad)
[1] "Normal_Weight"       "Overweight_Level_I"  "Overweight_Level_II"
[4] "Obesity_Type_I"      "Insufficient_Weight" "Obesity_Type_II"
[7] "Obesity_Type_III"
```

Figure 5: Unique values of target column

Since this is a categorical column, the models would be classification and not regression.

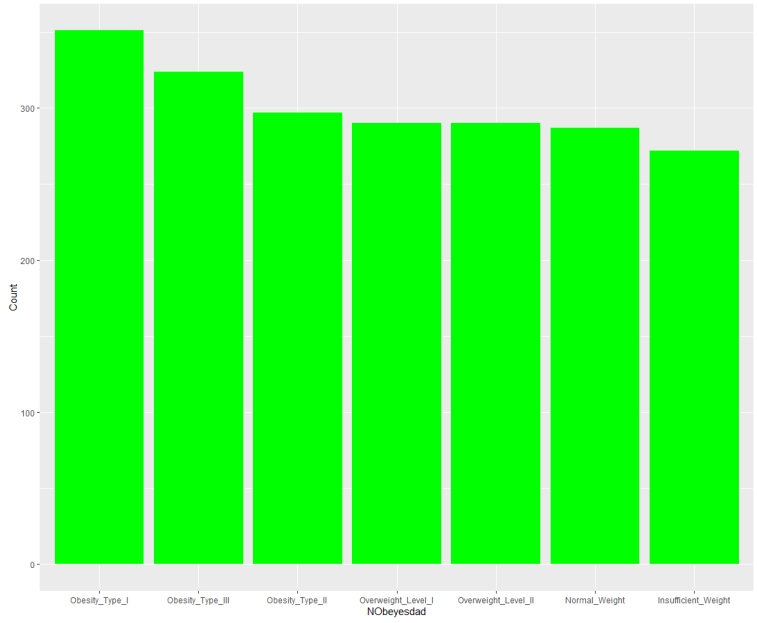

Figure 6: Count of different NObeyesdad

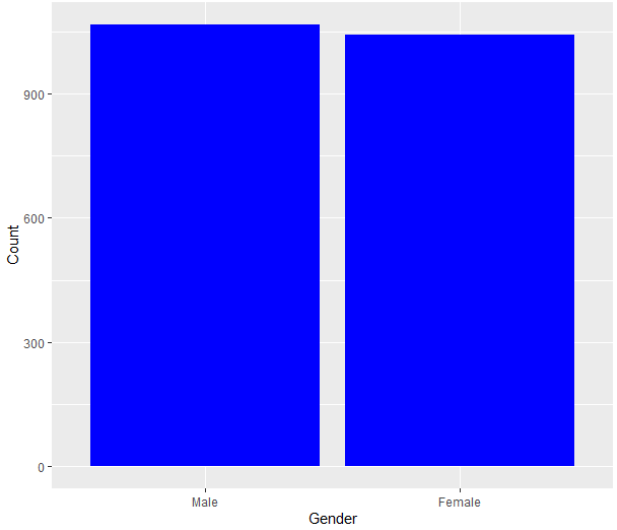

Figure 7: Count of male and female in the dataset

The code for this part of the question is attached below for reference.

```
# Data set 2: Obesity level dataset
# Set directory and read obesity levels dataset
setwd("C:/Users/Shrey Jain/Documents/Study/Data
Analytics/DataAnalyticsFall2022_SHREY_JAIN/Lab/DataAnalytics_A7_SHREY_JAIN/")
dataset2 <- read.csv('ObesityDataSet_raw_and_data_sinthetic.csv')
View(dataset2)


# Question 1: EDA
summary(dataset2$NObeyesdad)
unique(dataset2$NObeyesdad)
ggplot(dataset2, aes(x=reorder(dataset2$NObeyesdad, dataset2$NObeyesdad, function(x)-
length(x)))) + geom_bar(fill='green') +  labs(x='NObeyesdad', y='Count')
ggplot(dataset2, aes(x=reorder(dataset2$Gender, dataset2$Gender, function(x)-length(x)))) +
geom_bar(fill='blue') +  labs(x='Gender', y='Count')
```

Q.2 As the question asked to choose between 3 models for the 6000 level students, I chose the following three:

- Logistic Regression
- Decision Tree
- K nearest neigbours

The code for this part of the question is attached below for reference.

```
# Question 2: Model Development
# Logistic regression
training_dataset1 <- sample(dim(dataset1)[1], 0.7*dim(dataset1)[1])
training_dataset2 <- sample(dim(dataset1_without_outlier)[1],
0.7*dim(dataset1_without_outlier)[1])
linear_model1 <- lm(dataset1$Absenteeism.time.in.hours~., data=dataset1, subset =
training_dataset1)
linear_model2 <- lm(dataset1_without_outlier$Absenteeism.time.in.hours~.,
data=dataset1_without_outlier, subset = training_dataset2)
summary(linear_model1)
summary(linear_model2)
coef(linear_model1)
coef(linear_model2)
sqrt(mean((dataset1$Absenteeism.time.in.hours-predict(linear_model1, dataset1))[-
training_dataset1]^2))
sqrt(mean((dataset1_without_outlier$Absenteeism.time.in.hours-predict(linear_model2,
dataset1_without_outlier))[-training_dataset2]^2))

# Decision Tree
training_dataset3 <- dataset1_without_outlier[training_dataset2,]
random_forest_model <- randomForest(Absenteeism.time.in.hours~., data=training_dataset3)
test_x <- dataset1_without_outlier[-training_dataset2, -dim(dataset1_without_outlier)[2]]
test_y <- dataset1_without_outlier[-training_dataset2, dim(dataset1_without_outlier)[2]]
random_for_rms <- sqrt(mean((test_y - predict(random_forest_model, test_x))^2))
random_for_rms
```

```
plot(sqrt(random_forest_model$mse), main = "Root mean square error plot using absenteeism
dataset for random forest", xlab = "Number of trees", ylab = "RMS")

# K nearest neigbours
training_dataset4 <- dataset1_without_outlier[training_dataset2,]
train_1 <- dataset1_without_outlier[training_dataset2,]
test_1 <- dataset1_without_outlier[-training_dataset2,]
knn_model <- knn(train=train_1, test=test_1, k=sqrt(nrow(dataset1_without_outlier)), cl =
train_1$Absenteeism.time.in.hours)
knn_rms <- mean(train_1$Absenteeism.time.in.hours != knn_model)
knn_rms
```

Q.3 The following can be assumed after performing analysis and model development on the dataset: Obesity levels:

- The summary function and histograms were helpful, especially for identifying patterns or trends and to get the general idea about the data I am working on. However, since this is a classification problem, Bar charts are a better option to see the count of different categories.
- Removal of NA rows might not be the correct way for cleaning the data, a better approach could be to fill those missing values using various techniques.
- Since this is a categorical problem, we didn't worry about the outlier removal.
- The most common type of obesity level is Obesity level 1