

# CSX415\_Project\_flight\_delay\_cancellation\_analysis

Ajeey Patil

5/7/2018

## R Markdown

### Data Cleaning done in load.project() data munging

```
#currdir <- getwd()
#install.packages('ProjectTemplate')
library(ProjectTemplate)
load.project()
```

```
# {r setup, include=FALSE, echo=FALSE} #knitr::opts_chunk$set(echo=TRUE) #
```

## Install Packages

```
#Example package for RMSE calculation in Regression Analysis
#devtools::install_github("ajeypatil/rmse")
library(rmse)
```

## Perform Exploratory Data Analysis

graphs stored in graph directory

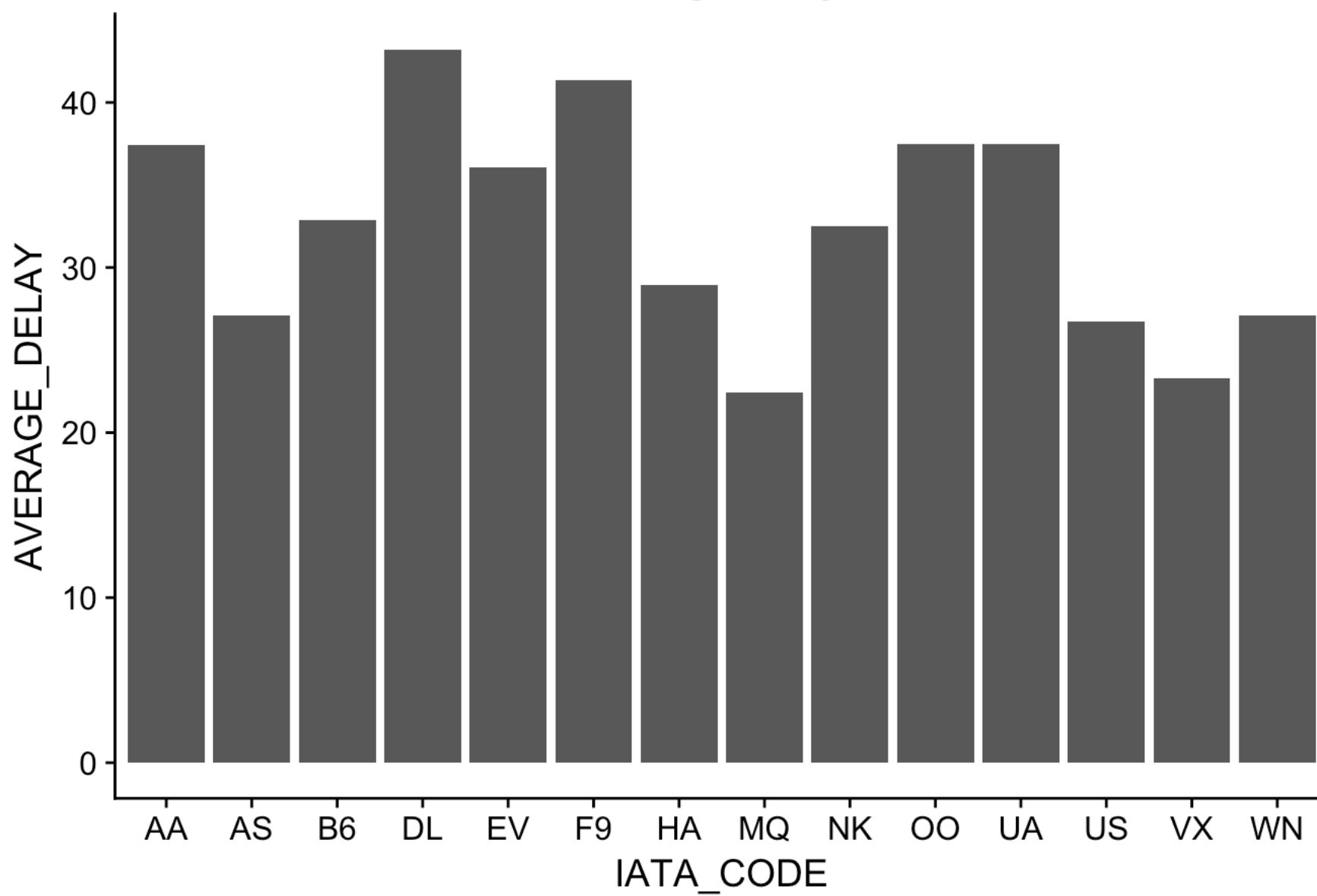
```
source('src/01-EDA/CSX415_Project_ExploratoryDataAnalysis.R')
```

```
## Saving 7 x 5 in image
## Saving 7 x 5 in image
## Saving 7 x 5 in image
## Saving 7 x 5 in image
## Saving 7 x 5 in image
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

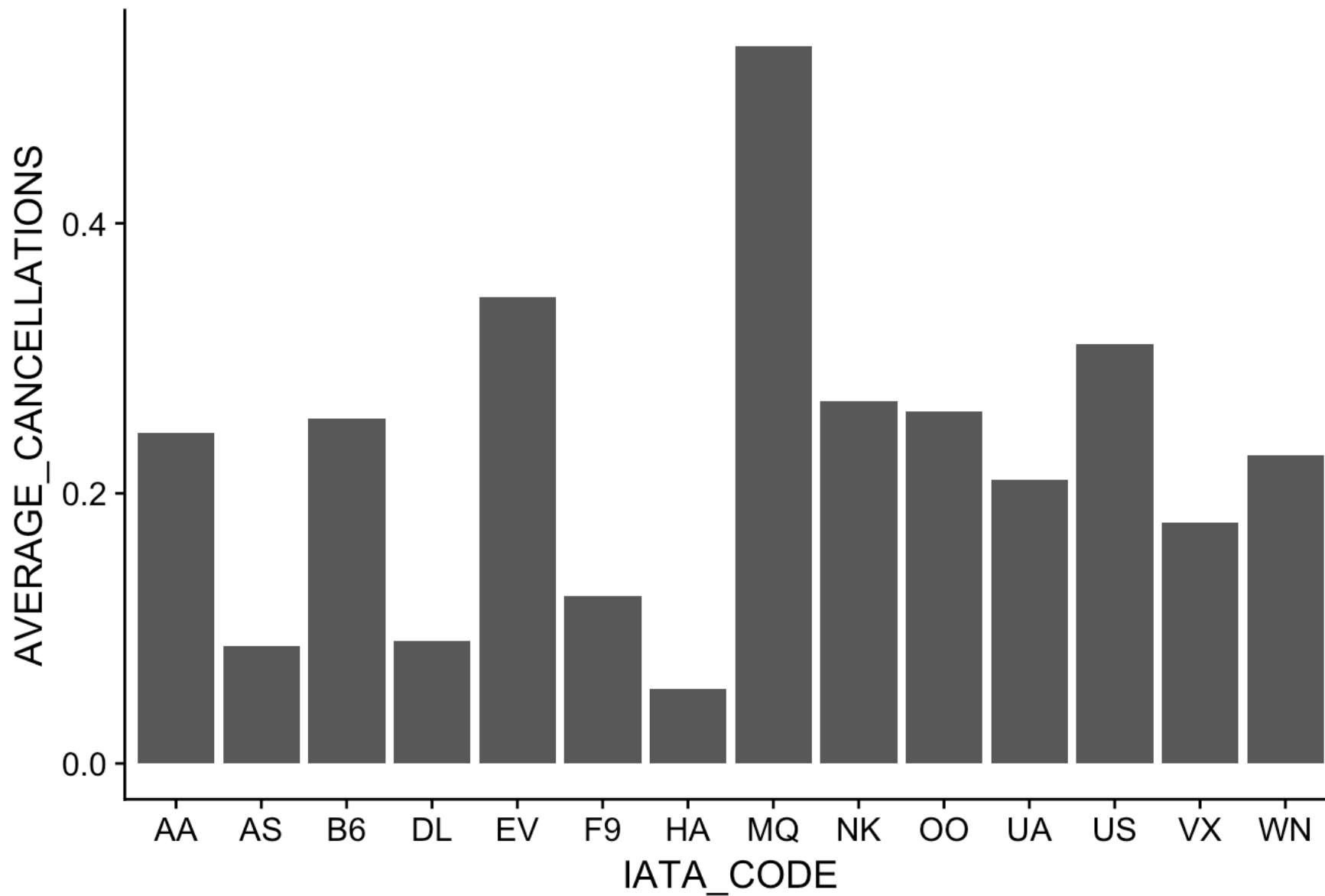
```
ggplot(alldelays, aes(x=IATA_CODE, y=AVERAGE_DELAY)) + geom_bar(stat='identity') + ggtitle("Average Delays")
```

**Average Delays**



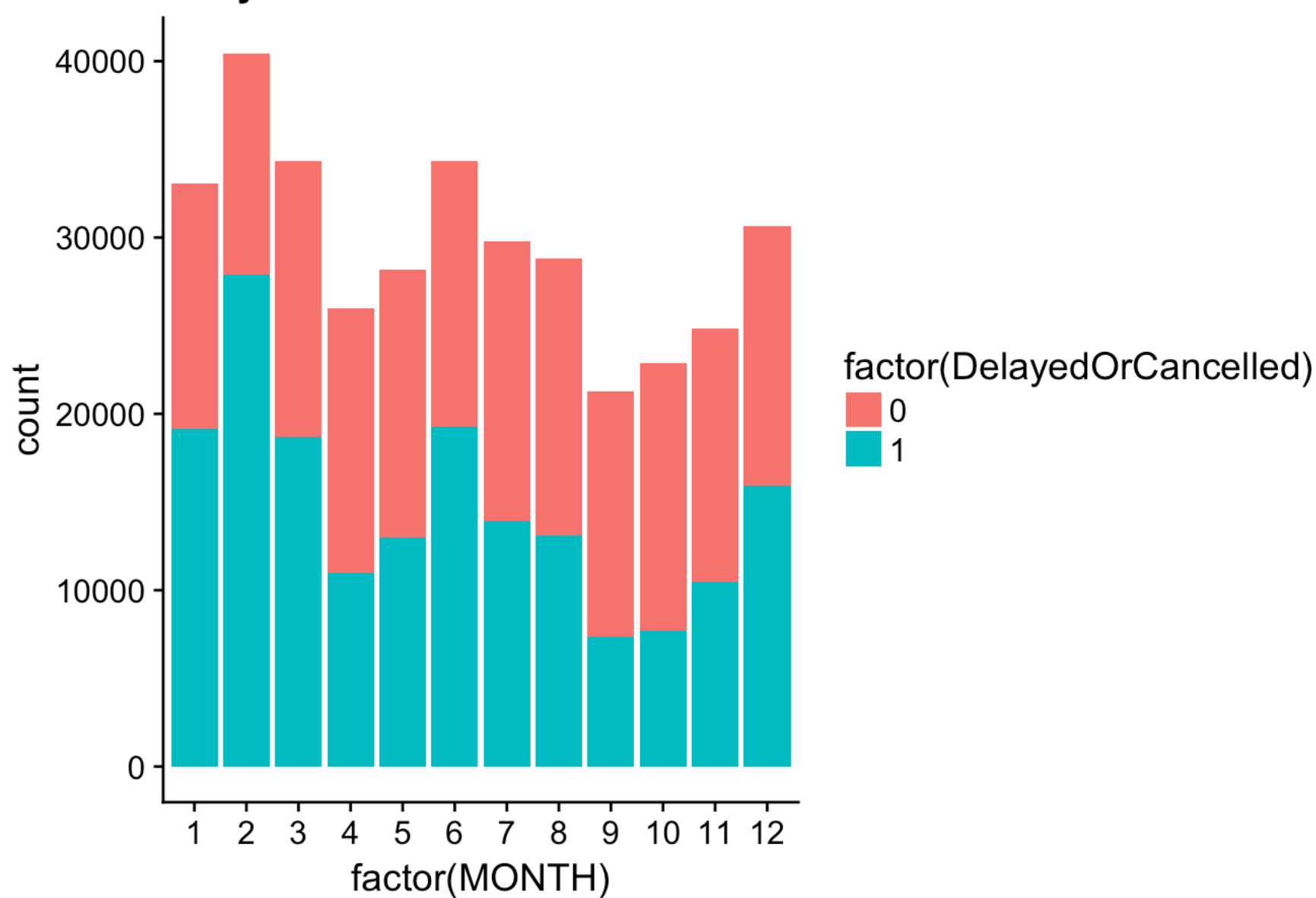
```
ggplot(allcancellations, aes(x=IATA_CODE, y=AVERAGE_CANCELLATIONS)) + geom_bar(stat='identity') + ggtitle("Average Cancellations")
```

**Average Cancellations**



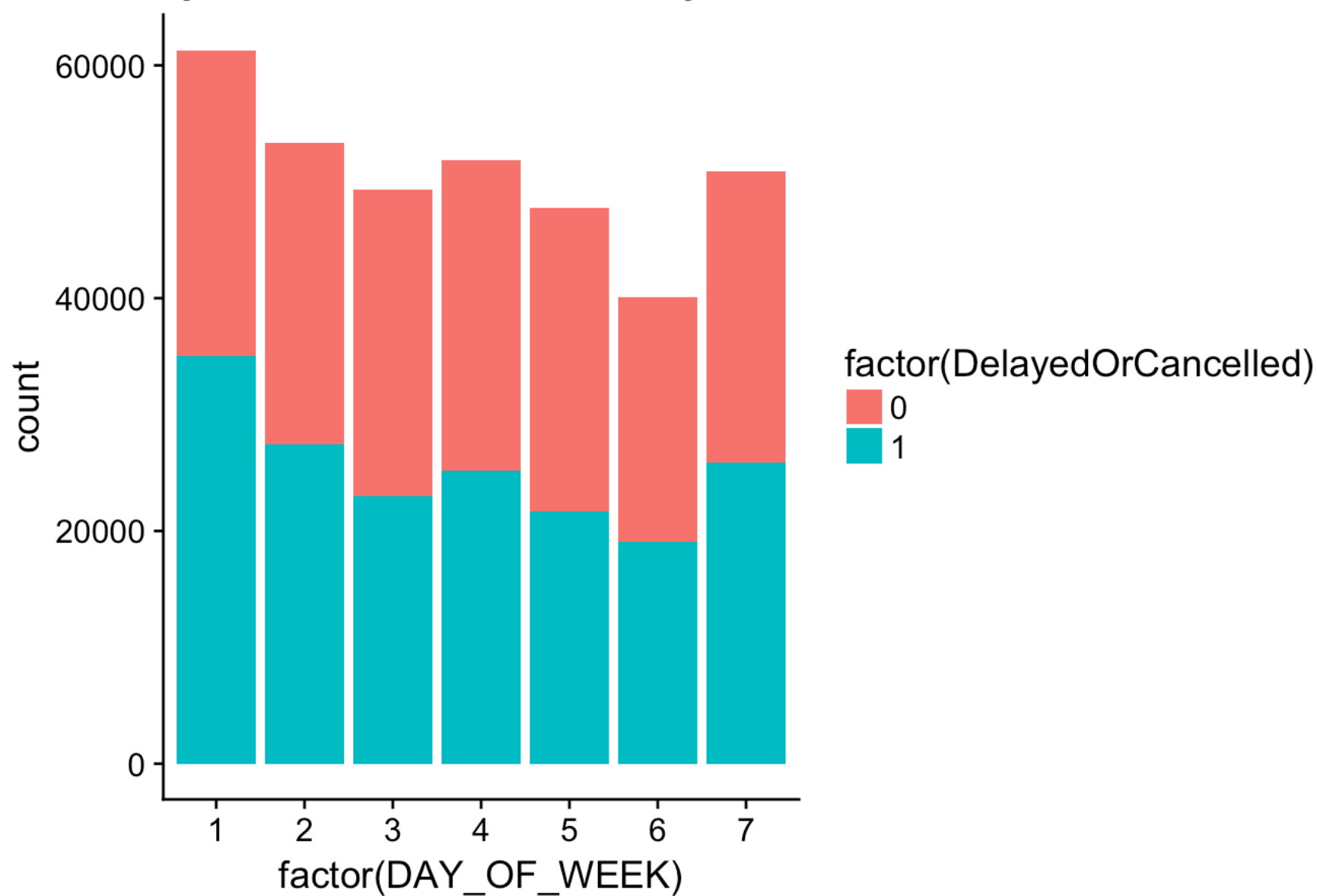
```
ggplot(DelaysAndCancellations, aes(factor(MONTH), group=DelayedOrCancelled, fill=factor(DelayedOrCancelled))) + geom_bar() + ggtitle("Delayed or Cancellations Per Month")
```

**Delayed or Cancellations Per Month**



```
ggplot(DelaysAndCancellations,aes(factor(DAY_OF_WEEK), group=DelayedOrCancelled, fill=factor(DelayedOrCancelled))) + geom_bar() + ggtitle("Delayed or Cancellations Per Day of Week")
```

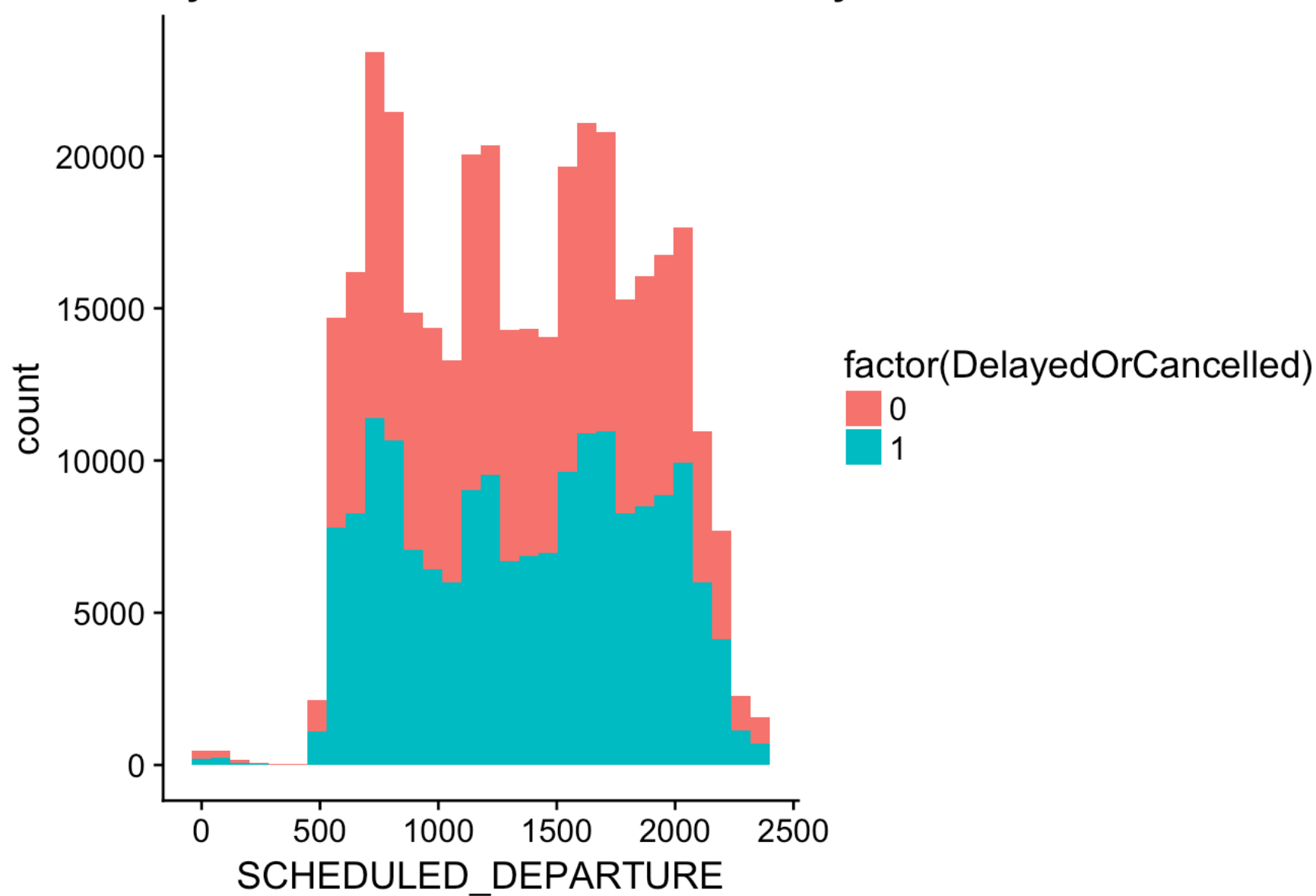
**Delayed or Cancellations Per Day of Week**



```
ggplot(DelaysAndCancellations,aes(SCHEDULED_DEPARTURE, group=DelayedOrCancelled, fill=factor(DelayedOrCancelled))) + geom_histogram() + ggtitle("Delayed or Cancellations Per Time of Day")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

## Delayed or Cancellations Per Time of Day



## PreProcess Data

Remove zero variance columns

Test-Train split

```
source('src/02-PREPROCESS/CSX415_Project_process.R')
```

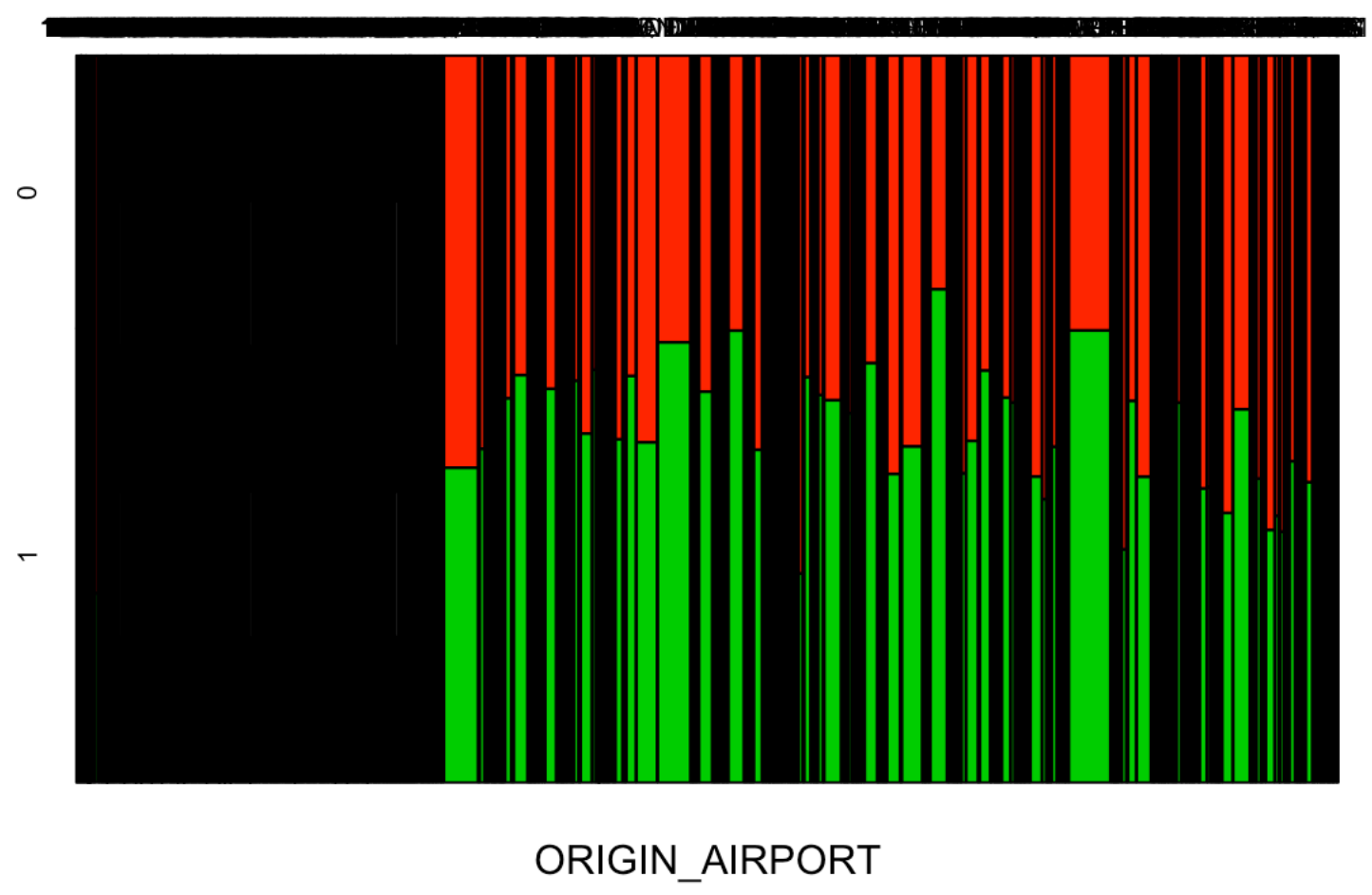
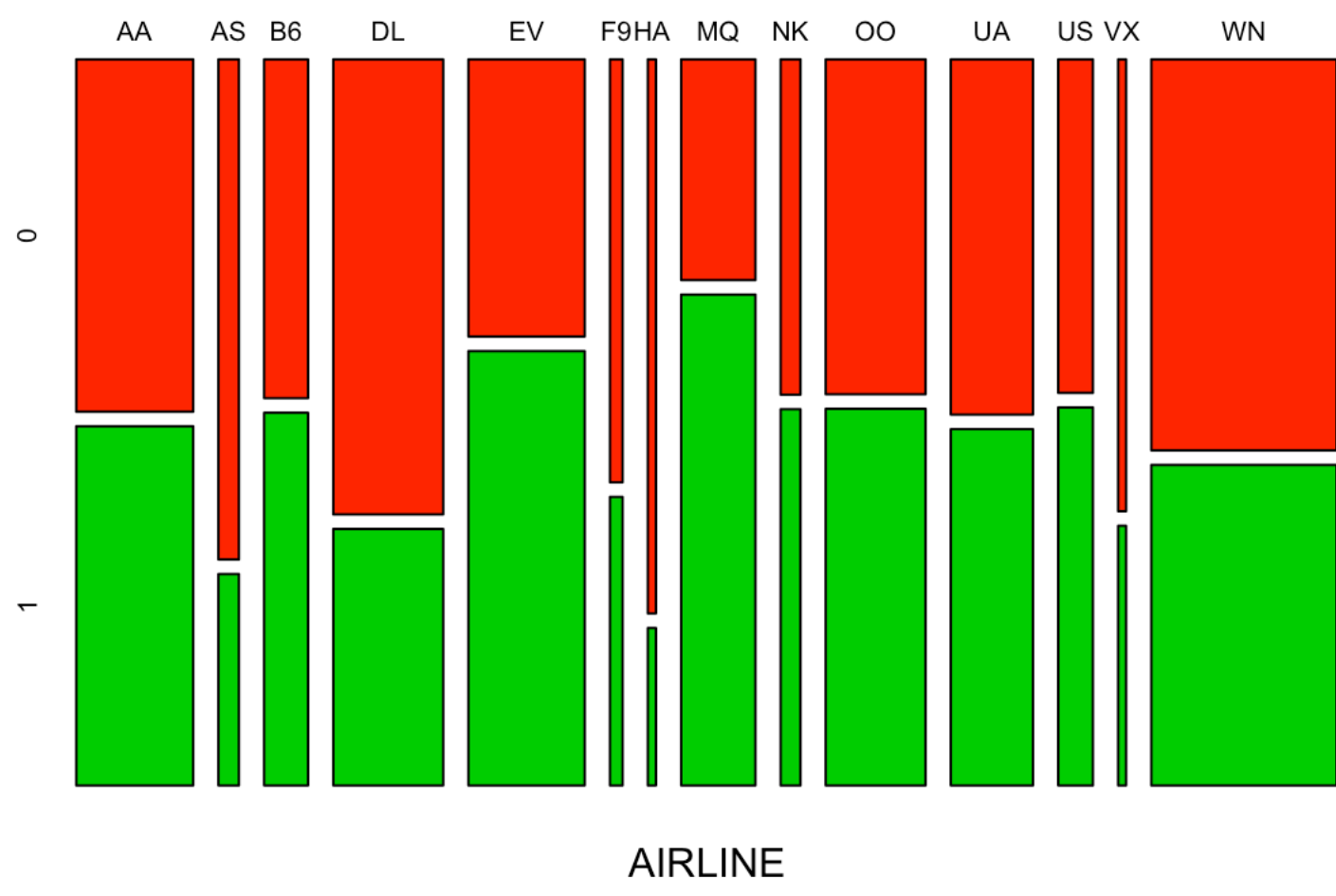
```
## Warning in pre_process_options(method, column_types): The following pre-  
## processing methods were eliminated: 'medianImpute'
```

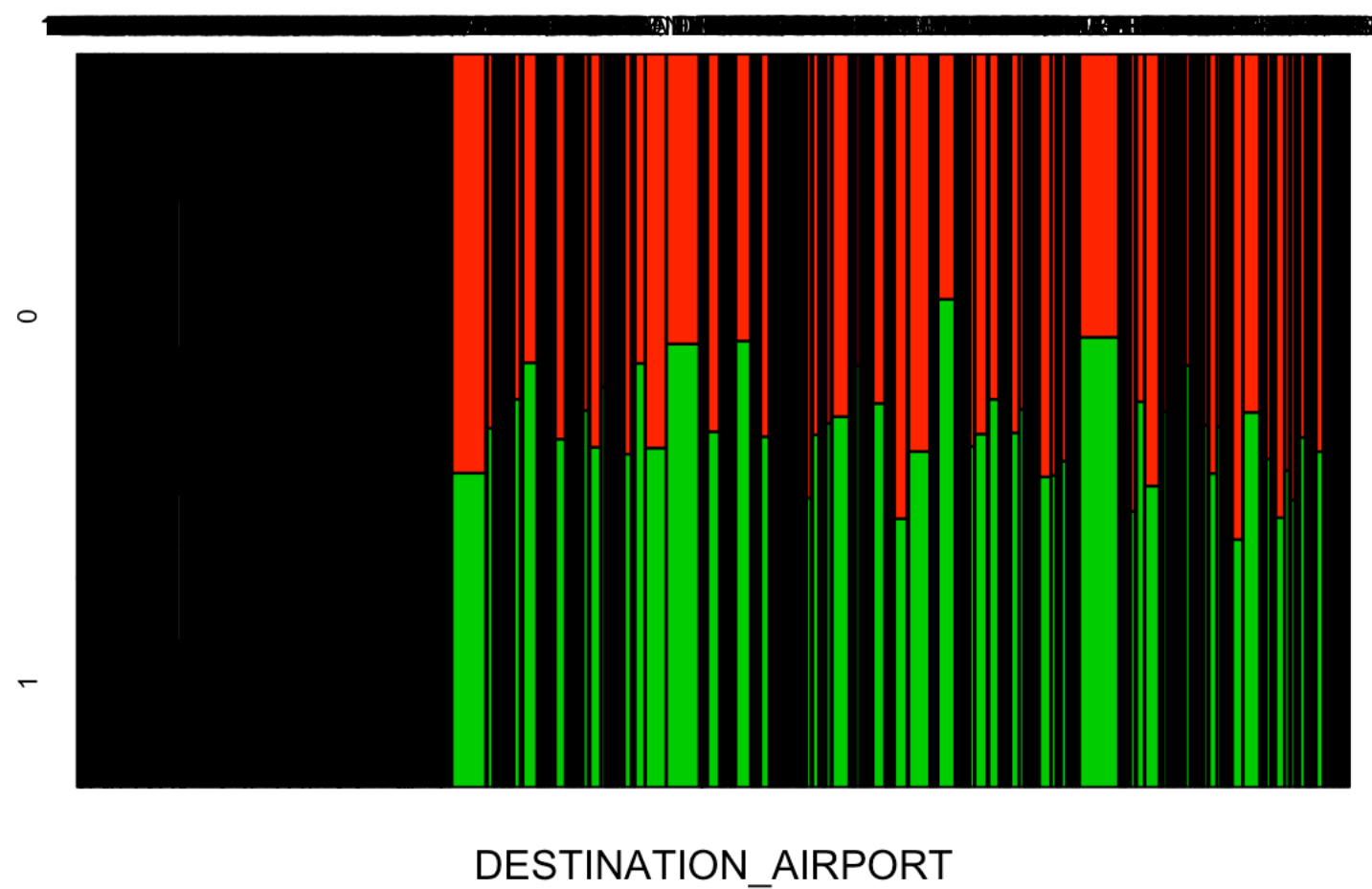
## Modelling

Apply Model

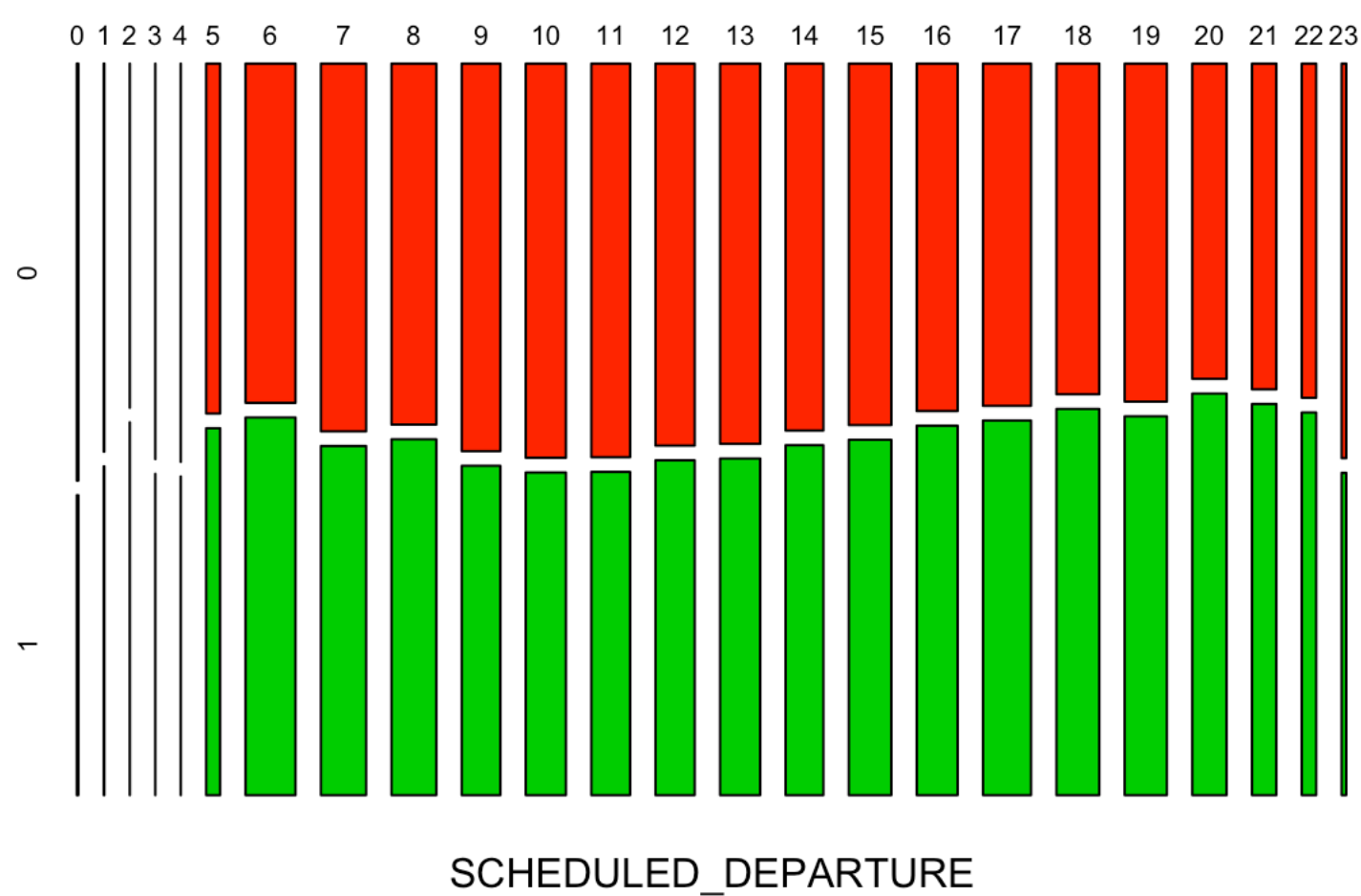
## Naive Bayes

```
source('src/03-MODELS/CSX415_Project_Data_Model_Naive.R')  
#nb.model  
#summary(nb.model)  
plot(nb.model)
```



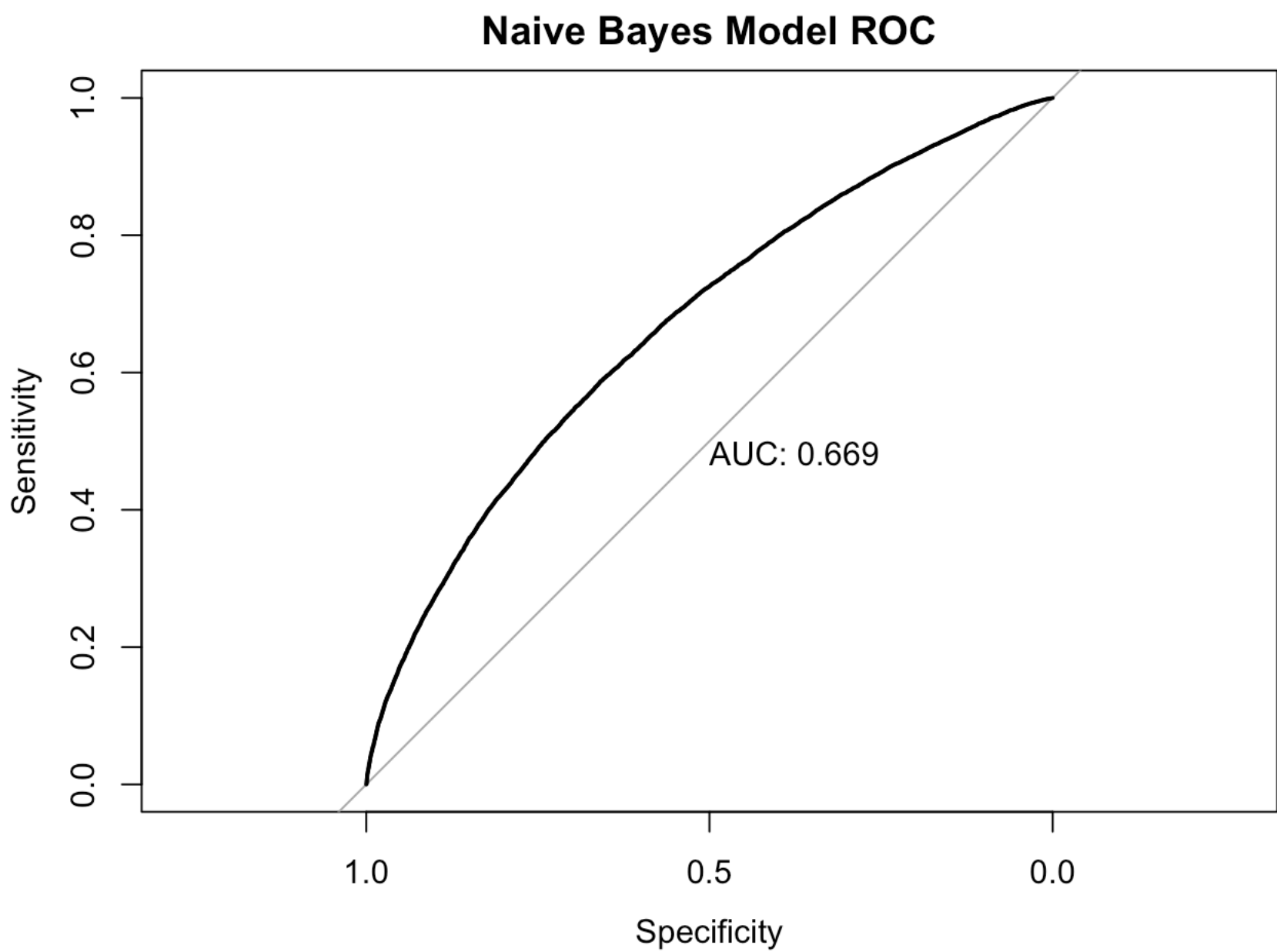






## Naive Bayes Model ROC plot

```
plot.roc(TestData$DelayedOrCancelled,nb_pred_prob[,2],print.auc=TRUE,main="Naive Bayes Model ROC")
```



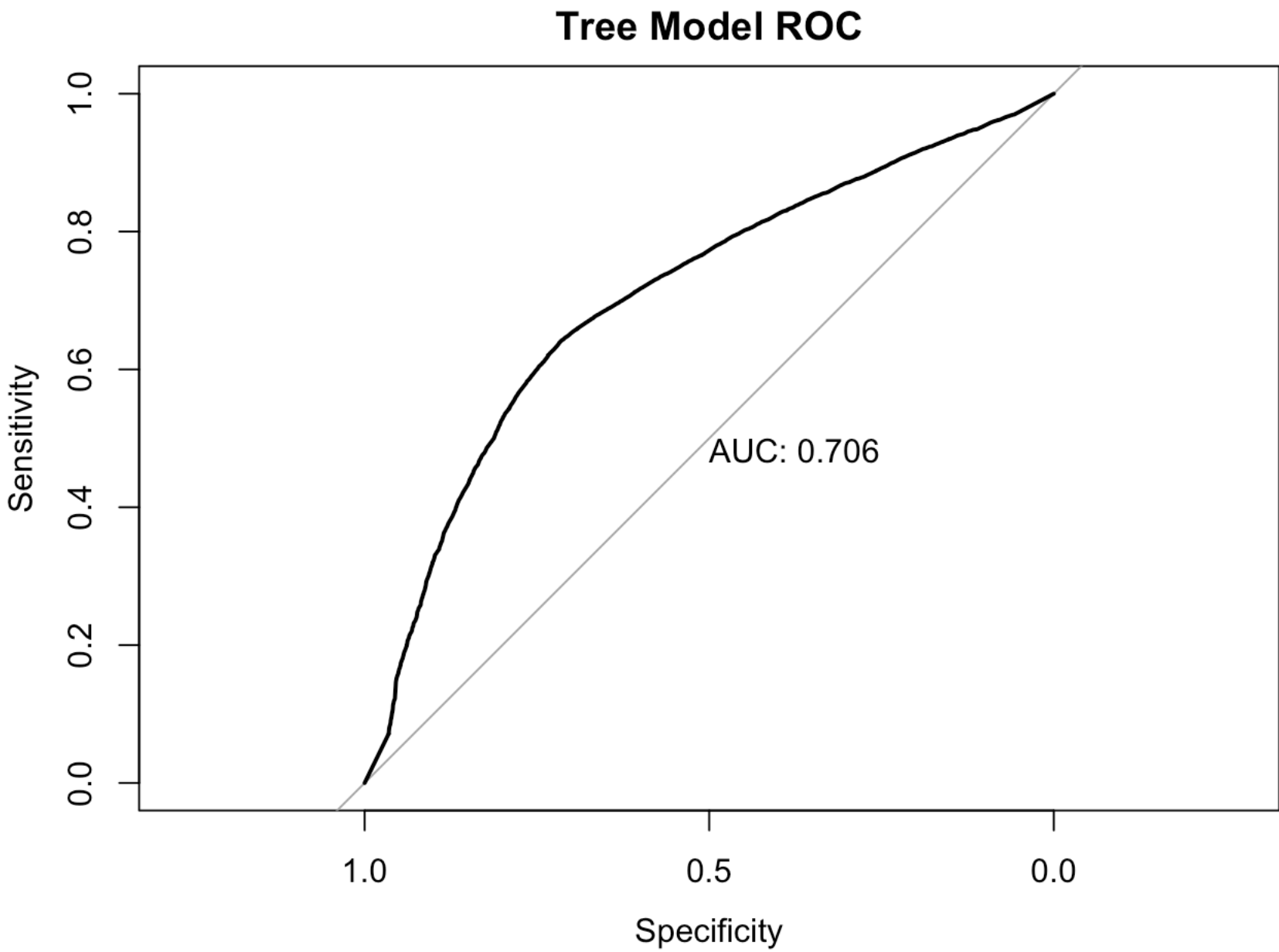
## Logistic Regression model



```
# Logistic Regression model for flights data takes more than 10 hours
# to train, the trained model is saved but is 3.9 Gb in size (.rds)
# Also the ROC calculated is only 0.592, hence not using this model
#source('src/03-MODELS/CSX415_Project_Data_Model_LogisticRegression.R')
#glm.model
#summary(glm.model)
#plot.roc(TestData$DelayedOrCancelled,glm_predictions,print.auc=TRUE,main="GLM Model ROC")
```

# Tree

```
source('src/03-MODELS/CSX415_Project_Data_Model_Tree.R')
#tree.model
#summary(tree.model)
#plot(tree.model)
plot(TestData$DelayedOrCancelled,tr_predictions[,2],print.auc=TRUE,main="Tree Model ROC")
```



# Model Evaluation

## Naive Bayes Model Evaluation

```
confusionMatrix(TestData$DelayedOrCancelled,nb_predictions)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction      0      1
##           0 21825 13632
##           1 13314 22143
##
##               Accuracy : 0.62
##               95% CI : (0.6164, 0.6236)
##       No Information Rate : 0.5045
##       P-Value [Acc > NIR] : < 2e-16
##
##               Kappa : 0.24
##  McNemar's Test P-Value : 0.05347
##
##       Sensitivity : 0.6211
##       Specificity : 0.6190
##       Pos Pred Value : 0.6155
##       Neg Pred Value : 0.6245
##       Prevalence : 0.4955
##       Detection Rate : 0.3078
##       Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.6200
##
##       'Positive' Class : 0
##
```

# Tree Model Evaluation

```
tr_pred <- ifelse((tr_predictions[,2]>0.8), 1,0)
confusionMatrix(TestData$DelayedOrCancelled,tr_pred)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction      0      1
##           0 29194  6263
##           1 18261 17196
##
##               Accuracy : 0.6542
##               95% CI : (0.6507, 0.6577)
##       No Information Rate : 0.6692
##       P-Value [Acc > NIR] : 1
##
##               Kappa : 0.3083
##  McNemar's Test P-Value : <2e-16
##
##       Sensitivity : 0.6152
##       Specificity : 0.7330
##       Pos Pred Value : 0.8234
##       Neg Pred Value : 0.4850
##       Prevalence : 0.6692
##       Detection Rate : 0.4117
##       Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.6741
##
##       'Positive' Class : 0
##
```

# Model Selection

Tree Model is more accurate and ROC is greater than Naive Bayes model.

▪

Logistic Regression model takes long time to train and ROC is less than 0.65 hence not selecting Logistic Regression model also because saved model .rds is 3.9 Gb in size hence not suitable for deployment

▪

Comparing the metrics, the accuracy and Kappa values of Tree Model are greater than Naive Bayes Model

▪

Conclusion: Tree Model satisfies the requirements criteria of accuracy greater than 60% and ROC(AUC) greater than 0.65 and hence used for deployment