**STAT 691P: Project Seminar**

Final Project Report: December 15, 2022

Group 1

Benjamin Abijah, Edwin Amponsah, Qi Chen, Jake Naughton

**Analysis of Flights Data from Newark Liberty International Airport, Simulation Study, R Function, Package, and Shiny App**

# Abstract

This project sought to gain insight on some dynamics in flight delays using evidence from flights departing from Newark Liberty International Airport from the time of June through September 2021. The primary goal was to predict whether any particular flight departing from Newark will be delayed given some information about the flight. To do this, we explored different forms of binary associations between predictors and the outcome, and then fitted a binary logistic regression to these data. We employed advanced statistical modeling techniques such as stepwise regression, ridge regression and lasso regression in an effort to further refine our predictions. The primary method used to evaluate models were ROC curves and the resulting AUC. Here, we also introduce an R Shiny app to handle data visualization of these data. We then perform a simulation exercise to understand how data quality could affect statistical models and their performance. Finally, we develop R functions and a corresponding R package for making inference about linear transformations of linear regression parameters.

# Table of Contents

[**Abstract**](#_lx28f89jwzue) **2**

[**Table of Contents**](#_aattczfdr9id) **3**

[**1. Introduction**](#_ugghkm6nwj2q) **4**

[1.0 Introduction](#_ly790b3mam9m) 4

[1.1 The Newark Liberty International Airport](#_sv9rnbcmdmmi) 5

[1.2 Simulation](#_q4xpb7msswju) 6

[1.3 R Function and R Shiny App](#_dszkh4psndfk) 7

[**2. Methods**](#_qpmszdy6tefa) **8**

[2.1 Exploratory Data Analysis](#_30j0zll) 8

[2.2 The Logistic Regression Model](#_g4h2u6jvf7cc) 8

[2.2.1 Assumptions and Method of Parameter Estimation](#_dib5utgcjghw) 9

[2.2.2 Interpretation of Coefficients](#_3dy6vkm) 10

[2.2.3 Model Evaluation Using ROC and AUC](#_1t3h5sf) 11

[2.3 Stepwise Logistic Regression Modeling](#_cdh7nu432gxf) 11

[2.3.1 Limitation of Stepwise Regression](#_7v815z7s7frh) 13

[2.4 Regularization: Ridge Regression and Lasso](#_1ksv4uv) 13

[2.4.1 Ridge Regression](#_k1vk9vm437lp) 13

[2.4.2 Lasso Regression](#_z337ya) 15

[2.5 Confusion Matrix](#_pgtxzlphgffc) 15

[2.6 Simulation Data Set with Binary Outcome](#_r7yv6s9htbc) 16

[2.6.1 Creating High AUC Simulation Data Set](#_iekw2jhfnok5) 16

[2.6.2 Creating Low AUC Simulation Data Set](#_3as4poj) 18

[2.7 Hypothesis Test for Regression Parameters](#_qu6cqkzh9uyy) 20

[**3. Results**](#_zfzxbo6g4dhv) **22**

[3.1 Data Description](#_ehw1oweuqazy) 22

[3.2 Exploratory Data Analysis](#_g4msiv3t16ch) 22

[3.3 Fitting Logistic Regression Model](#_blre26w3pfcp) 26

[3.4 Stepwise Logistic Regression Model](#_ptpmbqhgk3pq) 30

[3.5 Lasso Logistic Regression Model](#_83f05v5gtzlo) 35

[3.6 Simulation](#_r929yv2tout6) 38

[3.6.1 Results from Simulation](#_bs7pawwxdbtq) 38

[3.6.2 Analysis of Simulated Dataset](#_3urk05clu6u6) 39

[3.6.3 Assessment of Fitted Model to Simulated Dataset](#_5x94b72wzk5a) 44

[3.7 Illustration of R Functions and Package](#_b7bcd7igwl43) 50

[3.7.1 Description of Arguments of Function](#_4kfw34v8jcvj) 50

[3.7.2 Testing with Linear Regression](#_yzx4yulw4au3) 51

[3.7.3 R Package](#_88qj796vzw0k) 53

[3.8 Illustration of R Shiny App](#_b088n3i2ptg7) 53

[3.8.1 Description of R Shiny App](#_9spp12y2ohnn) 54

[**4. Conclusion**](#_4wmcsxdgc4zt) **57**

[**References**](#_w831tlt9fmrf) **60**

[**Contributions**](#_vvtg75xa58pp) **62**

# 1. Introduction

## 1.0 Introduction

It will soon be *Thanksgiving* or *Christmas*! Hurray!! Certainly, it is that time of the year where friends and families come together to share food and drinks, while they reminisce about past events and experiences throughout the year. Have you acquired your flight ticket? A few questions to consider: Which airports do you plan on departing from or arriving at, or even the specific airline to fly with? Have you considered what month or day of the week you would want to take this flight? What time in the day would that flight be? Well, if you have ever had to go through the frustration that comes with flight delays, then you will probably try to get the “best” options that optimize your chances of departing and/or arriving as scheduled.

According to the Bureau of Transportation statistics (2022), 631,933 out of 3,793,114 flights (16.66%) were delayed in 2021 while the percentage of delayed flights in 2022 is currently pegged at 21.52%. Also, an FAA–sponsored study issued in 2020 estimates the total cost of airline delay in the U.S. at 33 billion dollars for the year 2019 only. Clearly, flight delay is not only bad news for passengers but has a significant impact on the country’s economy. Hence, the analysis of flight delays becomes crucial in improving understanding of the causes of these delays, and may inform measures that ought to be taken to improve the performance of airlines and airports (Xu et al., 2008). In this project, we explore some dynamics in flight delays with evidence from flight data obtained from the *Newark Liberty International Airport*.

To do this, we consider a brief exploration of all variables of interest and how they affect the response variable. We further explore these relationships using a binary logistic regression. Of course, one goal is to obtain a model that could adequately generalize patterns and provide better predictions on flight delays. However, this current study of flight delays involves several predictors, some of which are multi-leveled. This increases the overall number of explanatory variables, leading to a needlessly complex logistic regression model. Like in most regression modeling procedures, we attempt to obtain a set of predictors that adequately account for the apparent variation in flight delays—for the purposes of interpretation and prediction. To achieve this, we consider and compare variants of performance measures on the prediction of flight delays obtained from three main techniques: stepwise regression, ridge regression, and Lasso.

## 1.1 The Newark Liberty International Airport

The Newark Liberty International Airport (Code: EWR), hereafter referred to as Newark Airport, was the first major airport in the United States—first opened in October 1928—within the city limits of both Newark and Elizabeth, New Jersey. As an international airport located about 15 miles (24km) southwest of Midtown Manhattan and 60 miles (97 km) northeast of Philadelphia, serving both metropolitan areas, it is owned and operated by the Port Authority of New York and New Jersey (Port Authority of New York and New Jersey, 2015). The airport, along with JFK Airport and LaGuardia Airport, combine to create the largest airport system in the United States, the second largest in the world in terms of passenger traffic, and largest in the world in terms of total flight operations. In just 2019, the airport was patronized by a whopping 46,336,452 passengers, and, subsequently, 15,892,892 and 29,049,552 passengers in 2020 and 2021, respectively.

*Figure 1.1: Newark International Airport*



***Source***: [Informational Guide to EWR](https://www.airport-ewr.com/newark)

## 1.2 Simulation

Often, in most scientific research, one may be interested in exploring how certain parameters of the study vary with several other fixed or varying parameters. Simulation exercises have been a useful tool for which many researchers have achieved this, as it provides the space for them to have control over certain parameters being considered in their studies, while studying the properties of others. The usefulness of simulation is non-denumerable. For example, in the study of flight delays, it provides a useful tool to better understand the underlying data structure that can lead to a good or bad performance in fitted logistic regression models. In the next part of this project, we look at a simulation exercise that attempts to obtain two sets of data on a set of “predictors”. The overarching goal is to obtain and analyze one set which, when modeled through various logistic modeling techniques, yields a better classification of the outcome variable (i.e., high AUC). The second set, on the other hand, is expected to yield relatively poorer predictions of the outcome variable (i.e., low AUC). While we conduct an analysis with the data from one of the other groups, we also assess how that other group did in modeling the data that we generated.

## 1.3 R Function and R Shiny App

In most regression modeling techniques, the researcher may be interested in making inferences about coefficients. Fortunately, this is quite easy to do because outputs from regression fits in most statistical software packages usually provide results of inferences about each predictor (eg., Wald's test) as well as inferences about all predictors in the overall model (eg., *F* test). Now, how about if we wanted to compare the effect of *duration* and, say, *departure time* on flight *delay*? Essentially, we want to know whether flight duration presents a similar effect as flight departure time – i.e., . In this study, we also attempt to develop a set of R functions—and a corresponding R package—to make such hypothesis testing easier. Lastly, we will build an interactive Shiny app that performs exploratory data analysis and visualization from the Newark Airport data. In addition to data visualization, the app also allows a user to predict the probability of a flight departing from the Newark airport being delayed or arriving on time.

# 2. Methods

## 2.1 Exploratory Data Analysis

To see whether the data can offer any insights, we perform some exploratory data analysis. While exploratory data analysis can take many forms, for this report, we relied primarily on graphs and plots, such as bar and mosaic plots, histograms, and correlation plots. These plots allow us to form certain hypotheses about our data such as: what distribution certain variables may follow, whether the desired outcome event is balanced or unbalanced, whether there exists any potential correlation between our variables, among others. Once we have a general understanding of the data, we can proceed to attempt to model it based on those insights. There are multiple ways to model our data with a binary response variable, which we discuss in the following sections.

## 2.2 The Logistic Regression Model

Let be the flight delay status with if the flight got delayed, and if otherwise; be a set of predictors (or flight parameters) for flight , and be the observed value of the -th predictor and the -th flight; denote the expected probability that whenever flight parameter is observed. Thus, . Then the logistic regression model is given by

A quick algebraic manipulation of this equation expresses the log of the *odds* of a flight being delayed (also called *logit*) as a linear combination of the set of predictors as

The expression represents the *odds*, which is the ratio of the probability of a flight being delayed to the probability that it arrives on time. Of course, since it is a ratio between odds, it can take values between 0 and , where values close to 0 indicate lower chance of a flight arriving late whereas values close to indicate a higher chance (Sperandei, 2014).

### 2.2.1 Assumptions and Method of Parameter Estimation

***Basic Assumptions of the Logistic Regression Model***

A typical logistic regression model makes assumptions on the following (Sperandei, 2014):

1. The flight delay status s are independent and follow a binomial distribution with probability of success : .
2. The *logit* of the delay status is a linear function of the set of predictor variables. Thus, .
3. There are no extreme values or outliers in the continuous predictors, like duration of flights.
4. There is no multicollinearity among predictor variables.

***Parameter Estimation***

Unlike the Ordinary Least Squares (OLS) method employed in linear regression, logistic regression adopts the method of *Maximum Likelihood* (James et al, 2013). Let be the total number of flights and be the observed flight delay status. Then, the maximum likelihood estimator for the set of 's is obtained by finding 's that maximizes the likelihood function,

Essentially, we want to select a set of 's such that the predicted value of the probability of flights that were delayed is as close to 1 and those that arrived on time is as close to 0. Unlike the ordinary least squares fitting, there exists no closed form expressions or solutions for the estimate of the parameters. Determining such estimates involves partial derivatives and becomes more cumbersome, especially for higher dimensional research problems like in this current study (James et al, 2013). The estimates are usually obtained numerically, by using rigorous iterative procedures and algorithms, which are outside the scope of this study.

### 2.2.2 Interpretation of Coefficients

Once the coefficients are estimated, their individual significance in estimating chance of flight delay is conventionally judged using the normally distributed *Wald* statistic—the ratio of each estimate and its corresponding standard error, i.e., , where a large value indicates evidence against the hypothesis that the coefficient is zero. The *odds ratio* of the -th predictor variable, usually given by , represents the change in the odds for a unit change in that variable—if continuous, eg., flight duration—or the actual odds of a particular group against a specified reference category for an -level categorical variable (where ). confidence intervals (CI) are obtained for these odds ratios (James et al, 2013).

### 2.2.3 Model Evaluation Using ROC and AUC

An ROC (receiver operating characteristic) curve is a graphical plot which is used for evaluating the performance of binary classification algorithms. In an ROC curve, the true positive rate (Sensitivity) is plotted on the *y*-axis whilst the false positive rate (1 – Specificity) is plotted on the *x*-axis. A random classifier will produce an ROC point that slides back and forth on the diagonal, . Any classifier on the diagonal may be said to have no information about the class (Fawcett, 2006). For the true positive rate, the larger the better and, for the false positive rate, the smaller the better. Therefore, the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test. Likewise, the closer the ROC curve is to the diagonal, the less accurate the test.

Area under the ROC curve (AUC), just as its name implies, measures the entire area under the entire ROC curve. It gives a summary measure of how well the model can distinguish between two binary outcomes. AUC takes values from 0 to 1, where the higher the AUC, the better the model. An area of 1 represents a perfect classifier whilst an area of 0.5 represents a worthless one. When the AUC is between 0.7 and 0.8, it is considered acceptable, whereas an AUC between 0.8 and 0.9 is considered excellent, and, finally, an AUC more than 0.9 is considered outstanding. However, an AUC of 0 indicates a perfectly inaccurate classifier, but this is very rare (Hosmer and Lemeshow, 2000).

## 2.3 Stepwise Logistic Regression Modeling

Including all variables in a regression model may usually come with some tradeoffs. According to Sperandei (2014), it is likely to find one or more variables that are “significant” in the logistic regression model, yet we may be unable to link this apparent significance to flight delays in theory (or even in practice). Second, if our model is saturated with many variables, then it is highly likely to possess less statistical power, as it can miss detecting associations between some predictors and flight delay. Also, when the set of predictors is relatively large, there is the tendency of obtaining overfitted models (models that follow the errors or noise too closely) that seem to work “perfectly” with the training data yet performs poorly in predicting future flight delay statuses (James et al, 2013).

Of course, with predictors, there are exactly possible combinations that could be chosen to estimate the model, and all these models can be compared to obtain the optimum one. However, with large fitting so many models might be a herculean task to achieve, although not impossible. One common practice is to start off with a base model—with no predictors—and iteratively add other predictors (i.e., *forward stepwise regression*). An alternative is to begin with the so-called “saturated” model—which contains all predictors—and drop predictors in turns, as desired (i.e., *backward stepwise regression*). When all remaining variables in the model reach a threshold, we stop the selection procedure and obtain our final model. This threshold could be a fixed *p*-value, an AIC (Akaike Information Criterion), a BIC (Bayesian Information Criterion), or the cross validated prediction error, (James, et al 2013). It is worth mentioning that when the number of predictors, , is relatively larger than the sample size, , it may be desirable to test for associations between each of the predictors and the flight delay status (univariate models). Then all of the “good” predictors can be included in a multivariate model before applying any of the two stepwise selection techniques (Sperandei, 2014).

### 2.3.1 Limitation of Stepwise Regression

Although they give useful results, stepwise regression methods are not guaranteed to yield the most optimal subset of predictors, regardless of the criterion used in the selection. Here, we discuss a few drawbacks of the stepwise regression technique from Harrell (2001).

* It does not consider all possible combinations of potential predictors of flight delay, thus, it is not guaranteed to select the “best” possible combination of predictors.
* It overstates statistical significance by giving biased regression coefficients, lower standard errors—leading to narrower confidence intervals, and smaller p-values.
* When there are more predictors relative to the sample size, the selection of subsets is highly unstable, as many variable combinations can fit the data in a similar way.
* It does not consider the causal relationship between variables, which can make it difficult to make definite statements about the impact of certain variables on the response.

## 2.4 Regularization: Ridge Regression and Lasso

Here, we explore some *regularization* (or *shrinkage*) alternatives to the stepwise regression. These regularization techniques—which involve fitting the logistic regression model using all predictors—yield estimated coefficients that are shrunken towards zero relative to their standard estimates (James et al, 2013). As a result, they have the effect of reducing estimation variance and are very useful for variable selection, thus better predictions of flight delay.

### 2.4.1 Ridge Regression

First, it is important to recall how least squares chooses the coefficient estimates we have used in the past. Least squares methodology seeks to choose parameter coefficients that minimize the residual sum of squares (RSS).

As we can see, RSS is calculated using both the intercept and our parameters . With ridge regression, we will still utilize RSS but with an additional component for our parameter estimation, as shown in the equation below

Our coefficient for ridge regression will be determined by which values minimize equation 5. As we can see, the difference between (2.4) and (2.5) is the addition of , which is called a shrinkage penalty with a tuning parameter of (James et al, 2013).

Naturally, we want to minimize this shrinkage penalty in order to minimize the equation as a whole. When are close to zero then the shrinkage penalty is also small, thus having the effect of shrinking parameter estimates towards zero. The tuning parameter serves as a control on the importance of the shrinkage penalty on parameter estimation. If = 0, then the shrinkage penalty is of no importance and we are only minimizing RSS as with least squares. However, as , the shrinkage penalty grows thus giving more weight to its role in estimating our parameter coefficients. Unlike least squares, ridge regression generates a different set of coefficients depending on what our tuning parameter is set to. Ridge regression decreases the variance of the parameter estimates (James et al, 2013). However, due to the bias-variance trade-off, this is at the expense of increased bias. The final model contains all predictors, which means ridge regression yields non-sparse models and it can be difficult to interpret the model.

### 2.4.2 Lasso Regression

The Lasso, an alternative shrinkage method to ridge regression, estimates using the values that minimize

Comparing (2.6) with (2.5), it is easy to find that the Lasso regression and ridge regression have similar formulations. The only difference is that the Lasso uses penalty which corresponds to the absolute value of . Like ridge regression, the Lasso shrinks all the coefficient estimates towards zero. Equation 6 trades off two terms: the larger the value of , the more the coefficient estimates go toward zero. The value of is equivalently important in the Lasso. Unlike ridge regression, the penalty term in the Lasso has the effect of forcing some of the coefficient estimates to be exactly equal to zero. It performs like subset selection, and it yields sparse models. Therefore, the final model generated by the Lasso can be interpreted more easily than that generated by ridge regression.

## 2.5 Confusion Matrix

The confusion matrix, also known as the error matrix, is a matrix describing the performance of a classification model on a set of test data. The name confusion matrix emanates from the fact that it makes it easy to visualize whether the classification model is confusing two classes. Table 2.1 shows a template for any binary confusion matrix using four kinds of results.

*Table 2.1: Confusion Matrix*

| Predicted Class | True Class | | |
| --- | --- | --- | --- |
|  | Yes | No |
| Yes | True Positives (TP) | False Positives (FP) |
| No | False Negatives (FN) | True Negatives (TN) |

Each row of the confusion matrix represents the instances in a predicted class while each column represents the instances in an actual class. The cell “True Positives (TP)” is the number of positive examples that have been classified accurately by the classification model. Similarly, the cell “True Negatives (TN)” represents the number of negative examples that have been classified correctly. The cell “False Positive (FP)” shows the number of actual negative cases classified incorrectly as positive; while the cell “False Negatives (TN)” represents the number of actual positive cases that have been inaccurately classified as negative.

## 2.6 Simulation Data Set with Binary Outcome

In this new part of the project, we perform a brief simulation exercise with the goal of using some of the methods discussed, so far, to make desirable predictions. To do this efficiently, we obtain two sets of data, as detailed in sections 2.6.1 and 2.6.2. After, we apply stepwise regression and Lasso regression in an attempt to fit the simulated data received from other project groups.

### 2.6.1 Creating High AUC Simulation Data Set

For model 1 (easy model), we are required to simulate data from the model that we expect another group would find with a high AUC value (as near 1.0 as possible).

* ***Size of Simulated Data set***

A total of 14,000 observations were simulated for this logistic regression model. Two-thirds of these were split into a training set. Within the training set, all the “*y*” outcome values are known, so that another group can train the model. The remaining one-third goes into the testing set with the “*y*” outcome values unknown.

* ***Main Effects Predictors***

A total of six main effect predictors () were used as main predictors in our simulation. Five of them are continuous variables, obtained from different specifications of the normal distribution, while the remaining is a binary variable. Among the five continuous variables, three of them () were made to be correlated with each other. The last main effect variable, , was simulated from a binomial distribution.

* ***Non-main Effects Predictors***

Based on the six main effects predictors, we simulated six more terms consisting of two pairwise interactions, two three-way interactions, and two polynomial terms up to third-order. Table 2.2 (a) shows how these terms were combined.

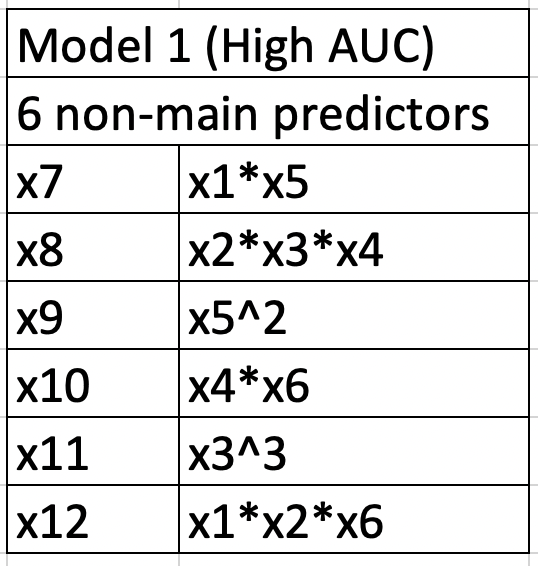
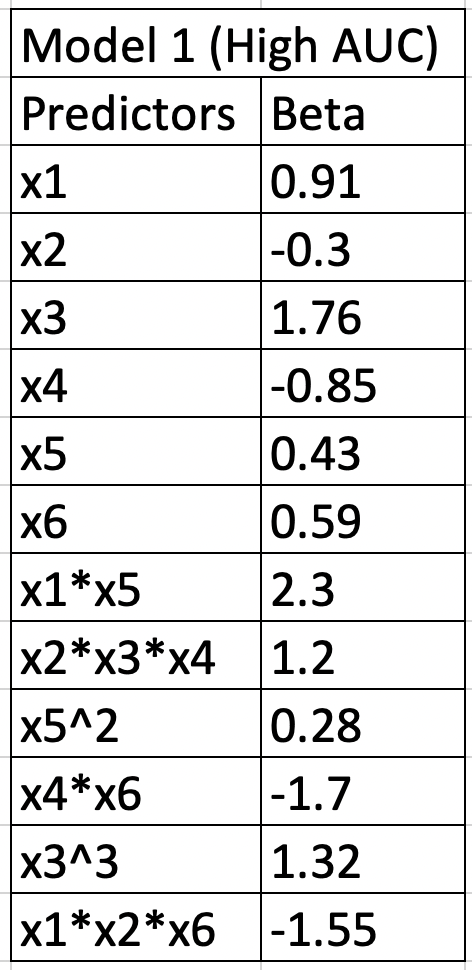
* ***Covariates (Betas)***

The Beta coefficients of the logistic regression model we used for this simulation are displayed in Table 2.2 (b).

* ***Outcome values***

After defining the twelve predictors and corresponding coefficients, we simulated the binary outcome, based on the logistic regression model.

*Table 2.2: Non-Main Effects and Covariates for high AUC model*

** 

(a) (b)

### 2.6.2 Creating Low AUC Simulation Data Set

Here, data is simulated for a binary logistic regression model where we expect the AUC of a fitted model to be as low as possible (i.e., near 0.5).

* ***Size of Simulated Data set***

A total of 300 observations was simulated for this logistic regression model. This leaves 200 data points (i.e., two-thirds) to be used by the other group for training and validation purposes.

* ***Main Effects Predictors***

A total of six main predictors, consisting of five continuous predictors and one binary predictor, were used as main effects predictors in our simulation. Four of the continuous predictors () were taken to be correlated, and simulated from a multivariate normal distribution while was simulated independently from a normal distribution. The binary predictor (i.e., ) was simulated using a binomial distribution.

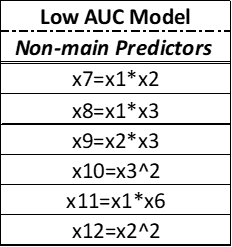
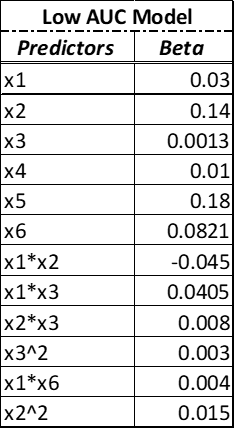
* ***Non-main Effects Predictors***

There are a total of six non-main effect predictors consisting of four pairwise interactions and two second-degree polynomial terms. The exact nature of these non-main effect predictors are described below in Table 2.3 (a).

* ***Covariates (Betas)***

The covariate values for the simulated terms used in this simulation are shown in Table 2.3 (b).

*Table 2.3: Non-Main Effects and Covariates for low AUC model*

(a) (b)

* ***Other Parameters***

Generally, we found that using a lower variance in simulating the main effect predictors led to a lower AUC. Hence, a variance of 0.1 was used for simulating the 4 correlated predictors and a variance 0.0025 was used for simulating that of the independent predictor variable.

## 2.7 Hypothesis Test for Regression Parameters

Tests for linear regression coefficients are usually based on the distribution of the estimate , which is approximately normally distributed as . Generally, is unknown but can be estimated. We note that the form of may change for other variants of regression methods like logistic regression. However, the results of testing still apply **(**Weisberg, 2005). Consider the linear regression model

and suppose that we are interested in testing the hypothesis or . Here, we want to test whether and have same effect on the response, . Doing this test directly requires the standard error of the linear combination , which is not readily available in a standard regression output. On another hand, one could simply refit the model in (2.7) as

and test the hypothesis , so that if the alternative is true, then and have the same effect on the response. However, when several of such hypotheses are of interest, it may be computationally inefficient to refit the model as many times as needed. So, can we have an easier way of doing this test?

Suppose that **a** is a vector of constants of the same length as . Then the linear combination has an estimate and the standard error is **(**Weisberg, 2005). To test , where is a scalar, the statistic is obtained as

For a sample size of and , the test statistic is compared with the *t*-distribution with .

Now, let **L** be a matrix of constants with full rank and be a column vector. If we wish to make inference about the linear combinations , i.e., , the test statistic is

where is the mean squared residuals (Theil, 1971). Here, the test statistic is compared with the *F* distribution with and degrees of freedom.

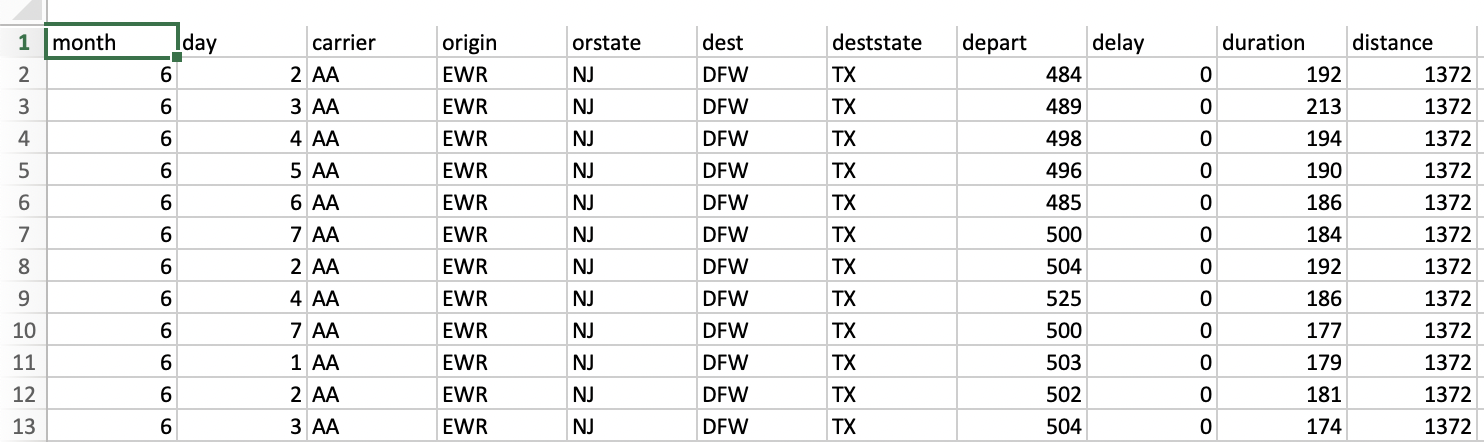
## 

# 3. Results

## 3.1 Data Description

The dataset used for this study contains flights departing from Newark Airport in New Jersey from June to September in 2021. There were 30,226 flights across 12 airlines that operated during this period. These flights took passengers to 73 different destination airports which are located in 36 states. Overall, eleven variables are included in the dataset. Eight of them are categorical – month, day, carrier, origin, orstate (state of flight origin), dest (airport destination), and destate (state of flight destination), delay (whether flight was delayed). Of course, three of the variables are numeric – depart (time of scheduled flight departure), duration, and distance. The data contains no entry with missing values. Also, we were unable to examine whether any of the flight records were duplicated as there are no unique identifiers for each flight in the data. We provide a snapshot for the flight data set for Newark Airport in Table 3.1.

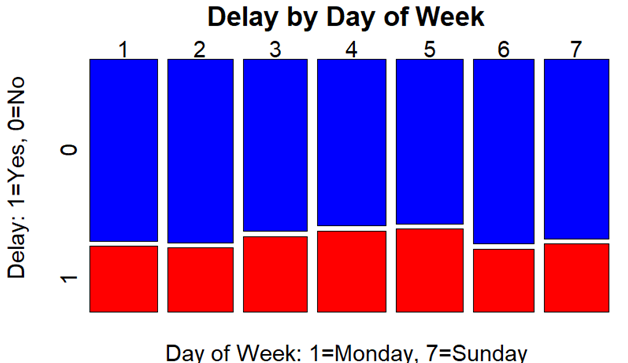
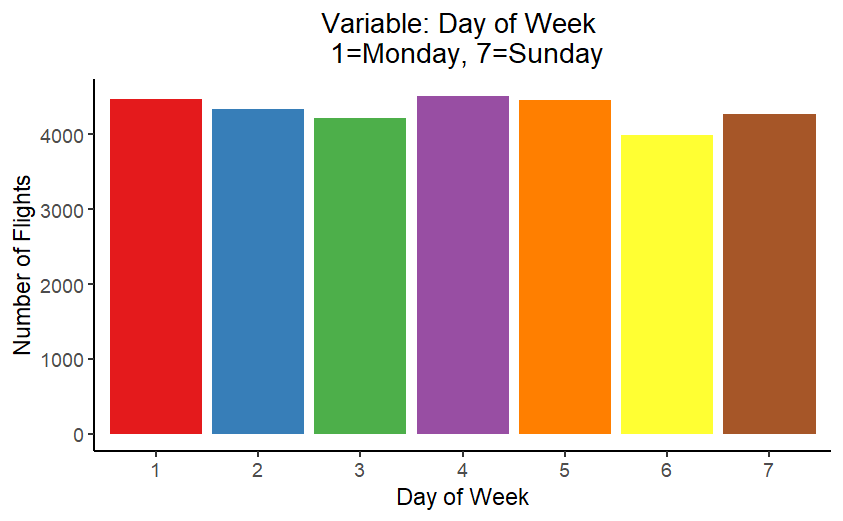
*Table 3.1: Snapshot of Flight Data Set for Newark Airport*



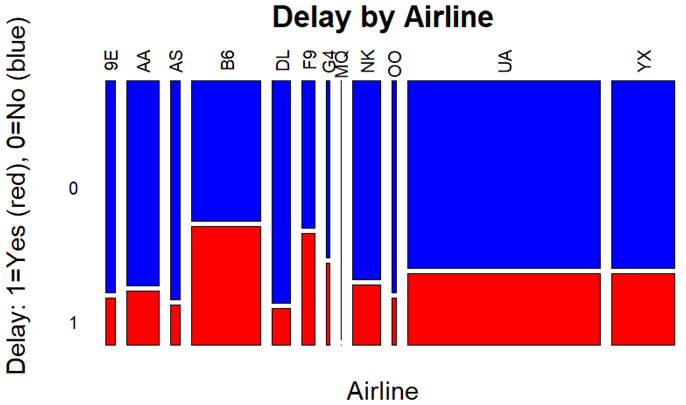
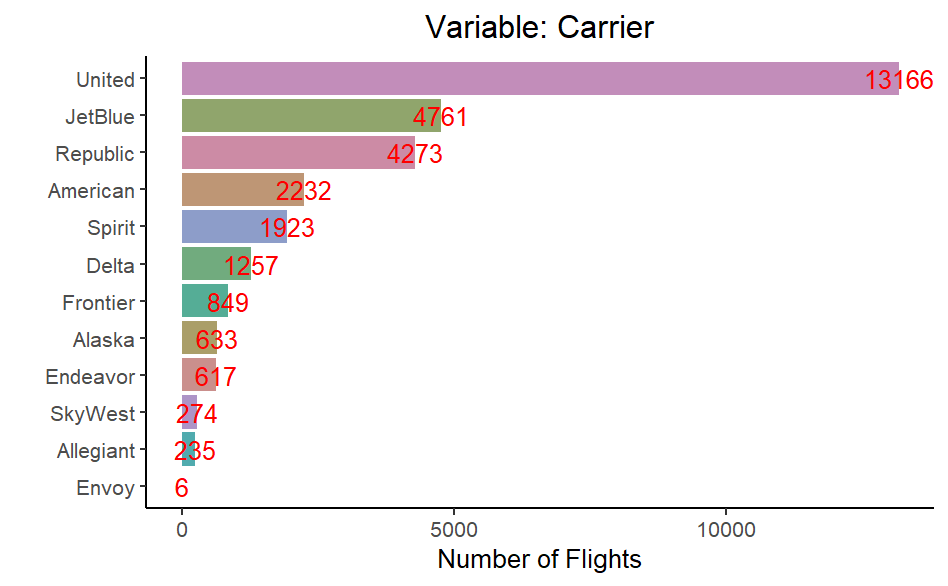
## 3.2 Exploratory Data Analysis

Next, we perform an exploratory analysis of the data and the results are presented in Figure 3.1 and Figure 3.2.

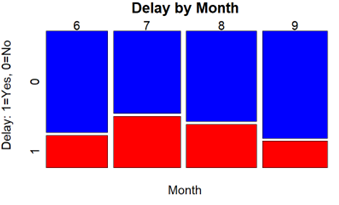
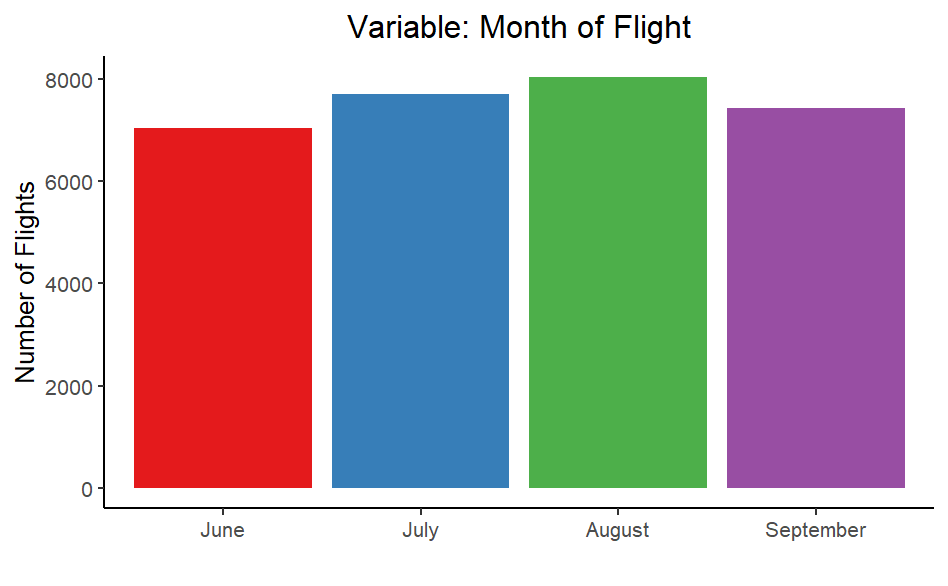
**(a) (b)**



**(c) (d)**

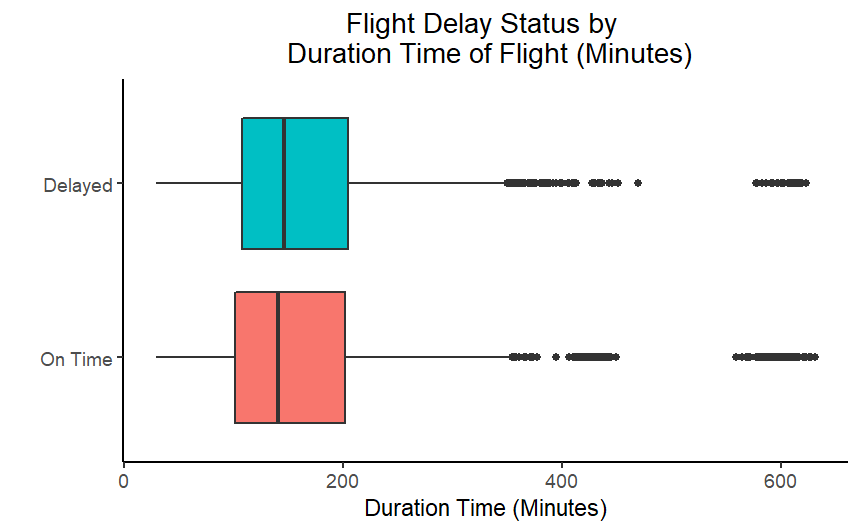
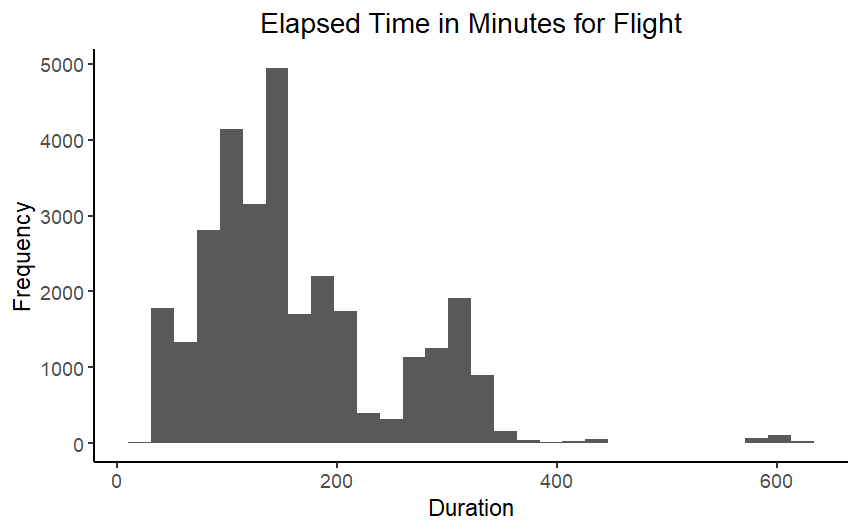


**(e) (f)**

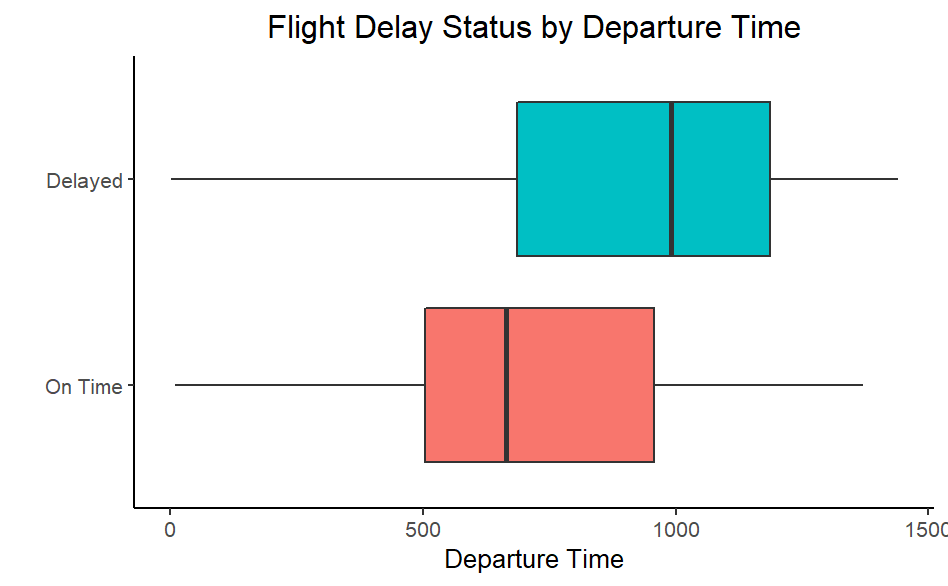
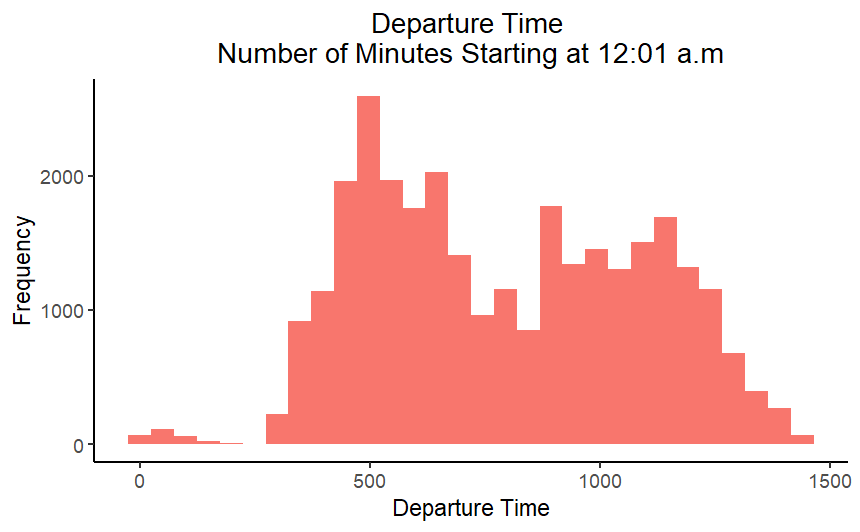


***Fig 3.1*: *Exploratory Data Analysis for Categorical variables*** – **(a)**. Number of Flights by Day. (**b)**. Flight Delay Rate by Day of Week. (**c)**. Number of flights by Airline (**d)**. Delay by Airline. (**e)**. Number of flights by Month. (**f)**. Flight Delay Rate by Month.

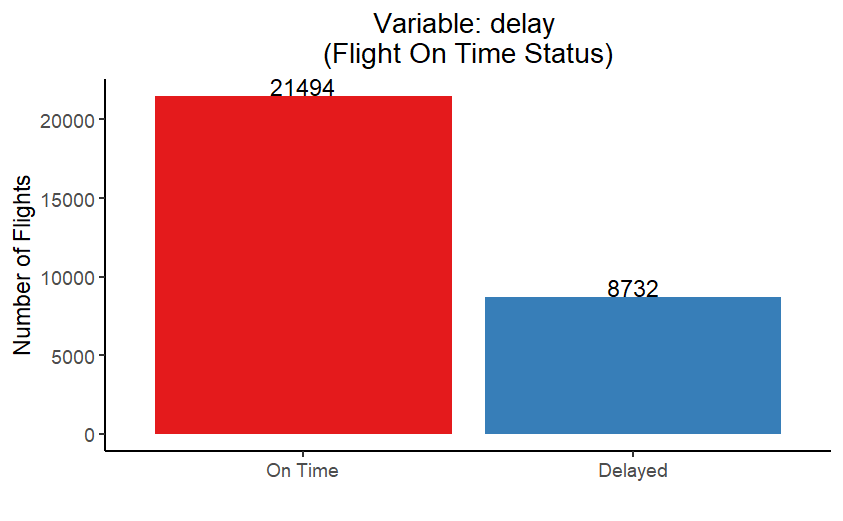
**(a) (b)**



**(c) (d)**



**(e) (f)**



***Fig 3.2*: *Exploratory Data Analysis for Continuous and Response******Variables* — (a)**. Distribution of Flight Duration. (**b)**. Flight Delay Status by Duration Time. (**c)**. Distribution of Flight Departure Time. (**d)**. Flight Delay by Departure Time. (**e)**. Flight On time Status. (**f)**. Correlation for Delay, Depart and Duration.

From Fig 3.1a, Thursday has the highest number of flights followed closely by Monday and then Friday with Saturday having the least number of flights. It can also be seen from Fig 1b that the delay rates for Monday, Tuesday, and Saturday are almost the same. The delay rates for the remaining days are slightly different although a bit higher with the highest delay rate observed on Friday. The day of the week may not be a good predictor for flight delay as the delay rates do not vary as much over the day of the week.

It can be seen from Fig. 3.1c that United Airline (UA), which is the largest tenant at Newark, had the most flights departing from Newark during June to September 2021. This comprises a total of 13,166 flights representing 44% of all flights departing from Newark during that period.. Also it can be seen from Fig. 3.1d that the delay rates vary by the carrier hence the carrier is likely to be a good predictor of delay rate.

It can be seen from Fig 3.1e that most of the flights departed in August, closely followed by July and then September. There does not appear to be a significant difference among these four months, as all of them had a relatively large number of flights. From Fig 3.1f, we can deduce that the month of the flight may be a good predictor of whether flights are delayed as there appears to be observable differences in delay rates, particularly when looking at September.

From Fig 3.2a most of the flights from Newark travel for shorter durations while the distribution of duration times for both groups (delayed and on time) are similar as there is no noticeable difference between the medians or first and third quartiles for delayed versus on-time flights in Fig 3.2b. Therefore, it cannot be concluded that longer duration flights are less likely to delay.

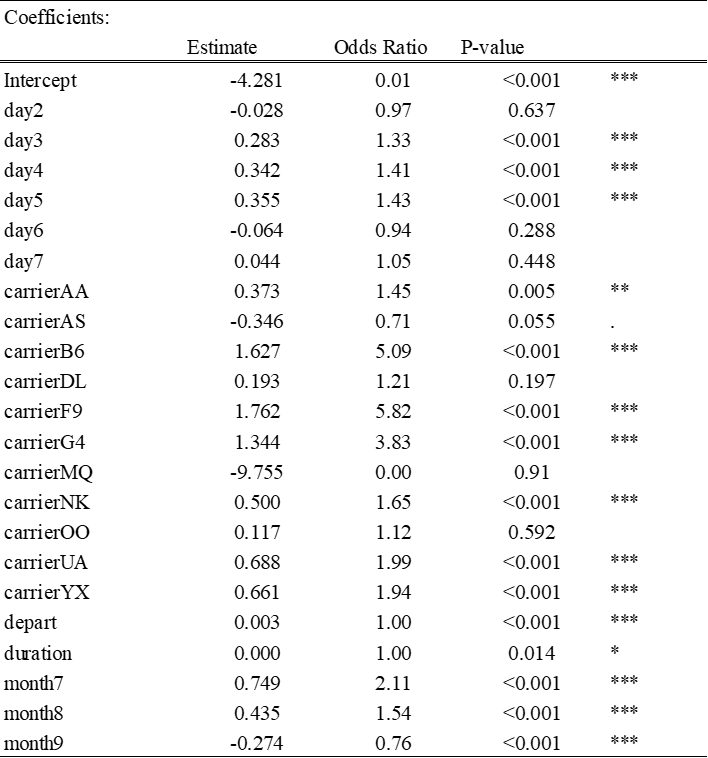
Fig 3.2e shows that more flights tend to depart from Newark during the daytime while peak times for flight departures around 8am in the morning. It can be seen from Fig 3.2f that flights that depart later in the day are more likely to get delayed while flights that depart earlier in the day tend to arrive on time more frequently. This may be due to the “ripple effects” that affect flight departure. This means that departure time of the flight explains whether the flight arrives on time, hence may be a good predictor of flight delay.

Finally, it can be noticed from Fig. 3.2e that 21,494 of the total flights (representing about 71%) arrived on time while the remaining 8,732 flights (representing about 29%) were delayed. The most common class (on time flights) is about 2.5 times the minority class (delayed) hence the data set may be considered as marginally imbalanced (Weiss, 2013).

## 3.3 Fitting Logistic Regression Model

The data set is partitioned into two parts: 80% of the data is used as the training set while the remaining 20% is reserved as the test set. The selection of the training and test data is done using a simple random sampling. Next, we build a logistic regression model to predict whether a flight gets delayed using the following variables as features: *day of the week*, *carrier*, *departure time*, *duration*, and *month of the flight*. The logistic regression model is fitted only to the training data set using the *glm* R function. The results of the fitted logistic regression are presented in Table 3.2. After training the model, the performance of the model is evaluated using the test data set and the ROC curve with the corresponding AUC is presented in Figure 3.3.

*Table 3.2 Results form Fitted Logistic Regression model*

**

***Signif. codes****: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

From Table 3.2, we can make the following conclusions at a 1% significance level:

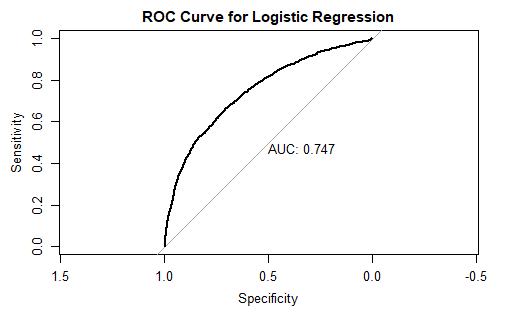
* About half of the days of the week (Wednesday, Thursday, and Friday) are significantly different from zero, showing that they are good predictors of the flight's on time status.
* Only the coefficients of the carriers *American* (AA), *JetBlue* (B6), *Frontier* (F9), *Allegiant Air* (G4), *Spirit Airline* (NK), *United Airlines* (UA), and *Republic* (YX) were found to be significant.
* Also, the coefficients of all the months (*July*, *August*, and *September*) considered in this study were significantly different from zero.
* Finally, the *departure time* was found to be significant while the *duration of the flight* was not found to be significant.

Overall, this suggests that all the predictors used in the model are good predictors of a flight’s on-time status except for the duration of the flight.

We can make the following conclusions:

* Flying on *Friday* has the highest likelihood of flights getting delayed as the odds of the flight getting delayed increased by 43% compared to if the flight had taken off on *Monday*. This is followed by *Thursday* with the odds of the flight being delayed increasing by 41%, and then *Wednesday* with the odds increasing by 33%. Although not statistically significant, flying on Tuesday and Saturday reduces the odds of getting delayed.
* Also, flying on the carrier *Frontier* (F9) had the highest odds of getting delayed with the odds increasing by 482% when compared to flying on *Endeavor* (9E). This is followed by *JetBlue* (B6) with the odds increasing by 409%, and then *Allegiant Air* (G4) with the odds increasing by 283% when compared to *Endeavor* (9E). *American Airline* (AA) had one of the lowest odds of getting delayed—increasing by only 45% when compared to *Endeavor* (9E). The odds ratio of the other carriers (eg. Alaska Airlines, Delta, etc.) are not significantly different from 1.
* Out of all the months considered in the study, flying in *July* has the highest odds of the flight getting delayed with the odds increasing by 111% when compared to *June*. This is followed by *August* with the odds increasing by only 54% when compared to June. Interestingly, flying in *September* has the lowest odds of flights getting delayed with the odds decreasing by 24% when compared to June – which may be because of reduction in summer travel.

*Figure 3.3: ROC and AUC for Logistic Regression model*



It can be seen from Fig. 3.3 that the AUC is 0.747, which indicates that the fitted model is performing better than if we had made random guesses.

## 3.4 Stepwise Logistic Regression Model

In this section, a total of three stepwise logistic regression models are fitted to the Newark airport dataset and the results are presented together with their interpretations using effect plots.

The first model (***Model 1***) attempts to predict the flight on time status (delay) using the predictor variables *day*, *carrier*, *depart*, *duration*, and month as main effects as well as their pairwise interaction terms for all the predictor variables except *carrier*. The second model (***Model 2***) includes all the predictor variables and pairwise interaction terms in *Model 1* as well as second and third degree polynomial terms for *depart* and *duration*. Lastly, the third model (***Model 3***) builds on the Model 2 by including three-term interactions terms for the predictors variables *day*, *depart*, *duration* and *month*. Also, we include a fourth and a fifth degree polynomial for the predictor variables, *depart* and *duration*, in addition to the second and third degree polynomial terms in Model 2.

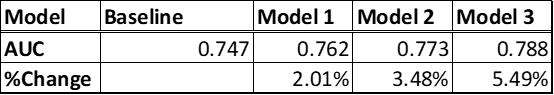
All the three models are evaluated on the test dataset and the results of the ROC curves with the corresponding AUC values are presented in Fig. 3.4. Also, we compare the AUC values for the three models in Table 3.3.

### 

*Figure 3.4: ROC and AUC for the three Models*



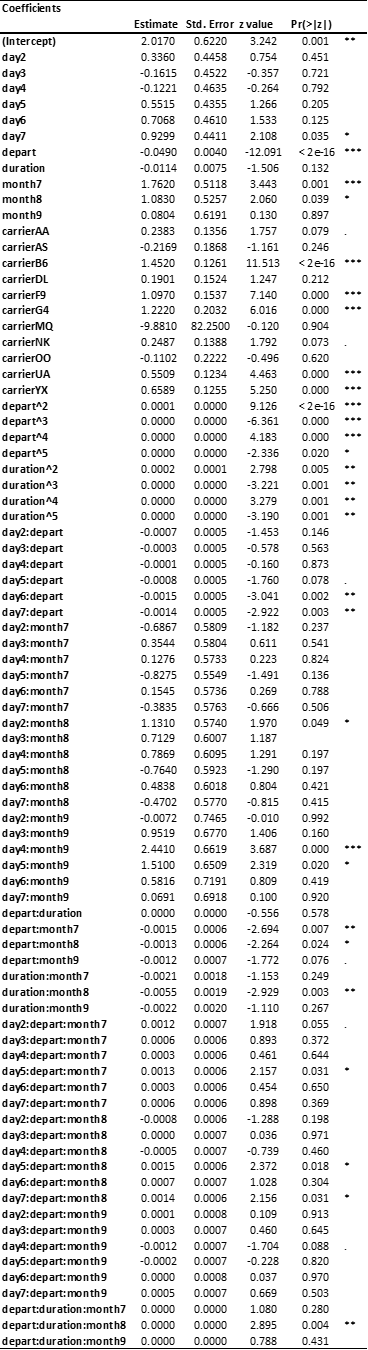
*Table 3.3: Comparing AUC for fitted Models*



From Fig. 3.4, the third model (Model 3) had the highest AUC followed by Model 2 and Model 1, respectively. It seems that adding more complexity to the model improves the AUC overall, however it is not clear if other metrics like accuracy or sensitivity would also improve. The increase in the AUC of each of the three models is compared to the base model (logistic regression model with only main effects and with no stepwise procedure applied).

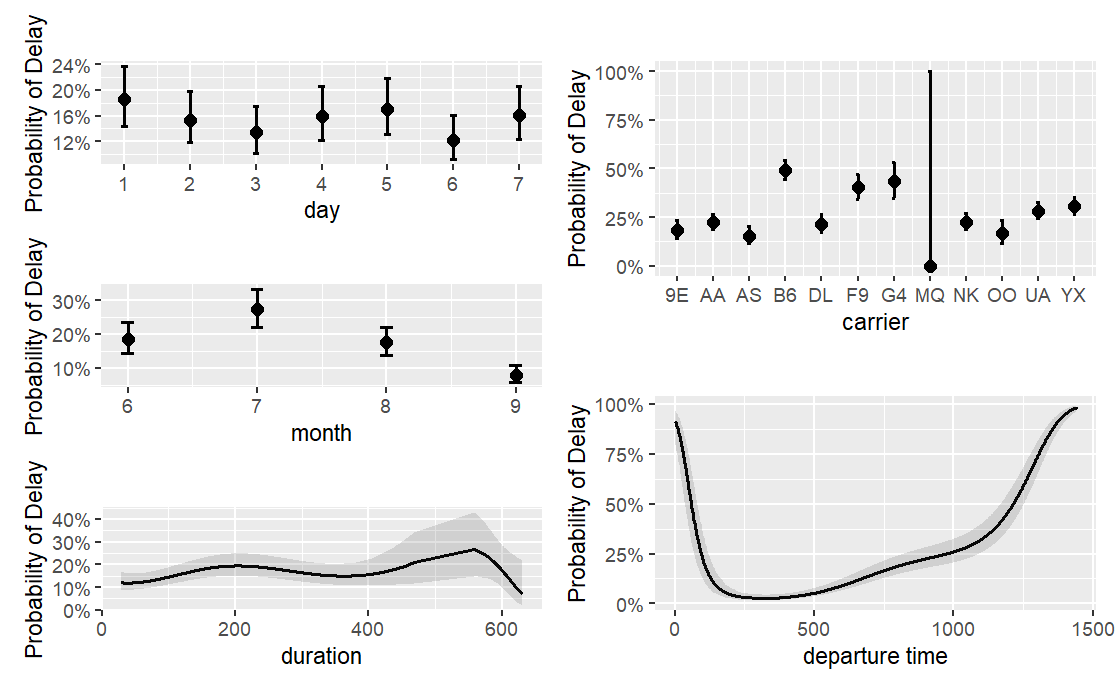
Also, the summary results of Model 3 are presented in Table 3.4. Also, we present effects plots for the main effects and two-way interactions in Fig 3.5 and Fig 3.6 respectively.

*Table 3.4: Summary Results for Model 3*



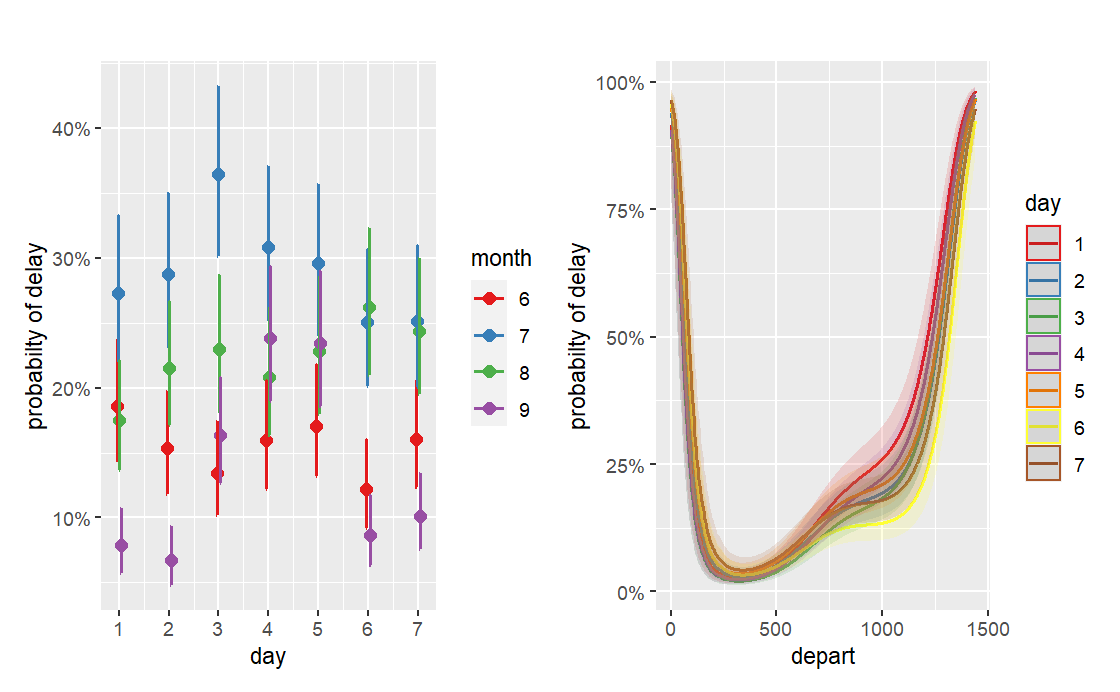
***Signif. codes:*** *0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Figure 3.5: Effect Plot of Main Effects (Model 3)*



From the main effect plot in Figure 3.5, Monday has the highest probability of the flight being delayed while Saturday has the lowest probability of the flight being delayed. Also, the month of July has the highest probability being delayed while September has the lowest. However, the probability of the flight getting delayed increases slightly for shorter duration flights but starts decreasing when the duration is over 200 minutes and then starts increasing again when the flight is over 400 minutes but eventually decreases for duration greater than about 500 minutes. Also, the probability of the flight getting delayed decreases more sharply when the departure time is less than 250 minutes after 12:00 am then starts increasing gradually after 500 mins from 12:00 am.

*Figure 3.6: Effect Plots of Two-Way Interaction Effects*



From Figure 3.6, flying in the month of September on Tuesdays has the lowest probability of the flight being delayed while flying in July on Wednesday has the highest probability of the flight being delayed. The probability of delay increases over the days of the week for the month of August while there is not much variation in the probability of delay for the month of June.

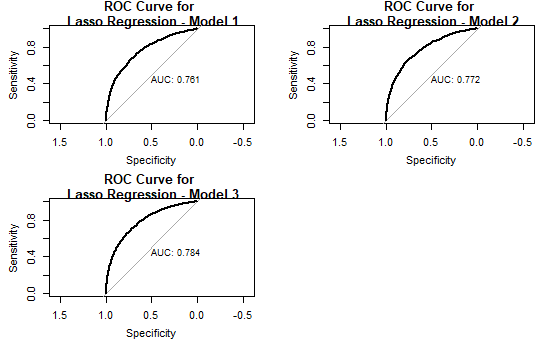
However, the effects plots for the other two-way interaction terms (*depart:month* and *duration:month*) get more wigglier (less smoother) making it harder to interpret.

## 3.5 Lasso Logistic Regression Model

In this section, a total of three lasso regression models are fitted to the Newark airport dataset and the results are presented together with their interpretations using effect plots. The model terms for all the three models are the same as the model terms used in Stepwise Regression.

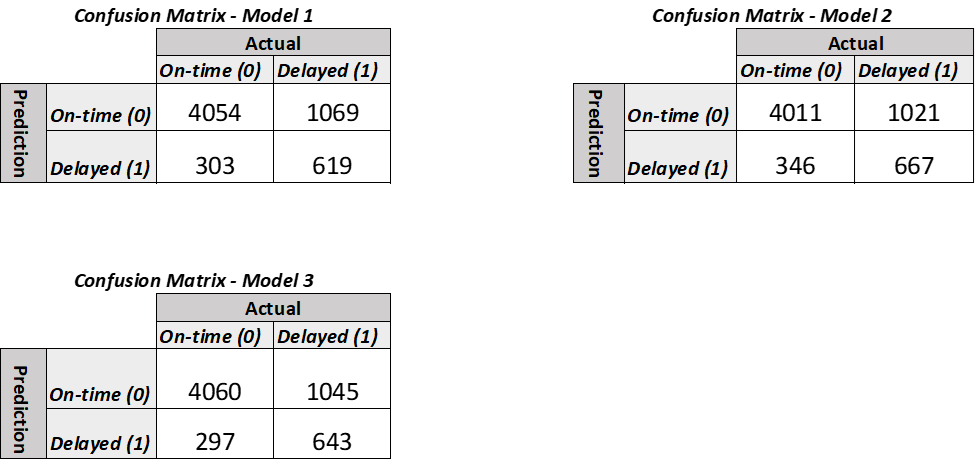
All the three models are evaluated on the test dataset and the results of the ROC curves with the corresponding AUC values are presented in Fig. 3.7. Also, the Confusion matrices and performance metrics for the three models are presented in Fig 3.8 and Table 3.5 respectively.

*Figure 3.7: ROC and AUC for the three Models*



From Fig. 3.7, the third model (Model 3) had the highest AUC followed by Model 2 and Model 1, respectively.

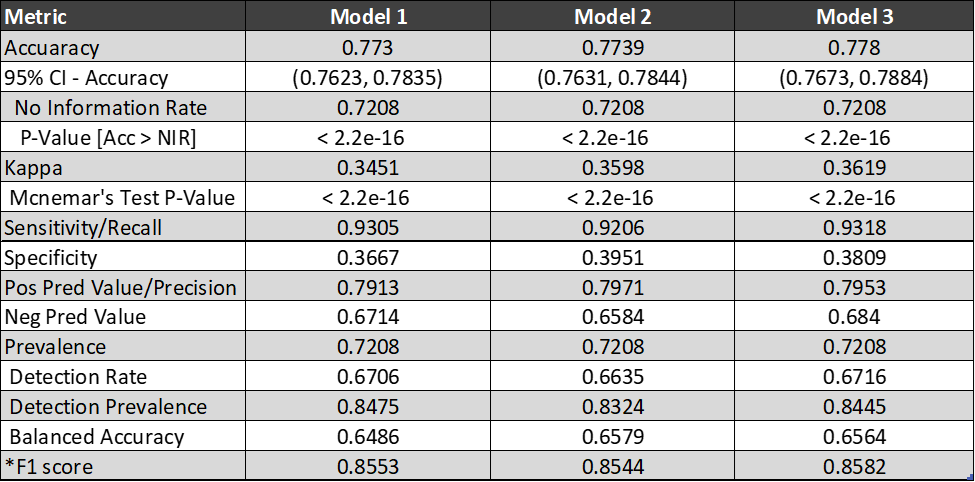
*Figure 3.8: Confusion matrices for all three models*



We can make the following observations from the confusion matrices for the three Models:

* Model 3 accurately classifies the most on-time flights with a total of 4060 accurate predictions, followed by Model 1 with a total of 4054 accurate predictions, and then Model 2 making a total of 4011 accurate predictions.
* Model 2 accurately predicted the most delayed flights with a total of 667 accurate predictions, followed by Model 3 with 643 accurate predictions, and then Model 1 with a total of 619 accurate predictions.
* Model 2 made the lowest incorrect prediction of on-time flights that actually delayed with a total of 1021 inaccurate predictions, followed by Model 3 with a total of 1045 inaccurate predictions of on-time flights, and then Model 1 with a total of 1069 inaccurate predictions of on-time flights.
* Model 3 made the least wrong predictions of delayed flights that actually arrived on-time with a total of 297 inaccurate predictions, followed by Model 1 with a total of 303 inaccurate predictions of delayed flights and then Model 2 performed worse in this regard with a total of 346 inaccurate predictions of delayed flights.

*Table 3.5. Performance Metric Table for the Confusion Matrices*



*\* The F1 score was not part of output from R but calculated separately*

*The positive class used for the construction of the performance metrics is on-time flights (0)*

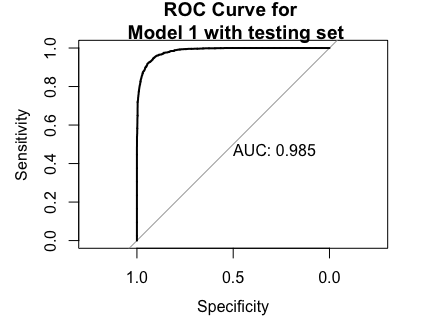
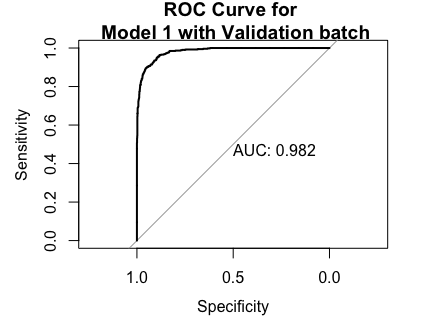
Here, we compare some important performance metrics across the three models. From Table 3.5, we can see that, in terms of *accuracy*, Model 3 performs better than Models 2 and Model 1 (ie. 77.8%, 77.39%, and 77.3%, respectively). However, this judgment should be taken with a pinch of salt, especially in this flight delay study where we have imbalanced classes. For instance, a model may perform poorly on the minority class, yet far better on the majority class, thus inflating the overall accuracy. In this case, an appropriate alternative to assess prediction accuracy is the *balanced accuracy*—which is desirable when the number of positive classes (on time) is relatively larger than the negative classes (delayed). Here, we see that Model 2 (65.79%) tends to perform better in terms of accurate predictions than Model 3 (65.64%), which also performs better than Model 1 (64.86%). Furthermore, there is more *precision* in the prediction made by Model 2 than in Models 3 and 1 (ie. 79.71%, 79.53%, and 79.13%, respectively). In terms of *sensitivity*, Model 3 predicts more positive classes (ie., *on time*) relative to the total number of flights that were actually on time (93.18%). This is followed by Models 1 and 2 with 93.05% and 92.06%, respectively. Considering *specificity*, on the other hand, Model 2 predicted more negative classes (ie., *delayed*) out of the total flights that were actually delayed (39.51%), followed by Models 3 and 1 with 38.09% and 36.67%, respectively.

It seems that adding more complexity to the model improves the AUC overall, however this performance is not the same for other metrics like the balanced accuracy where Model 2 performed better than Model 1 and 3. Here, considering other metrics like Balanced accuracy is important as the AUC is indifferent to class imbalance in the dataset which is the case for the Newark flight dataset we are analyzing.

## 3.6 Simulation

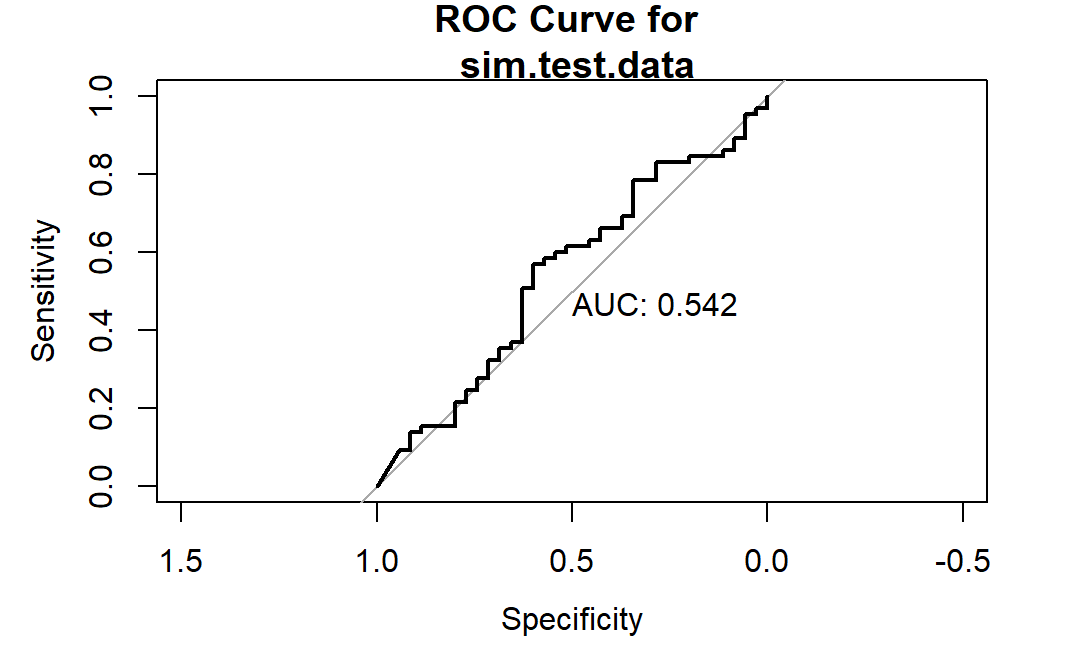
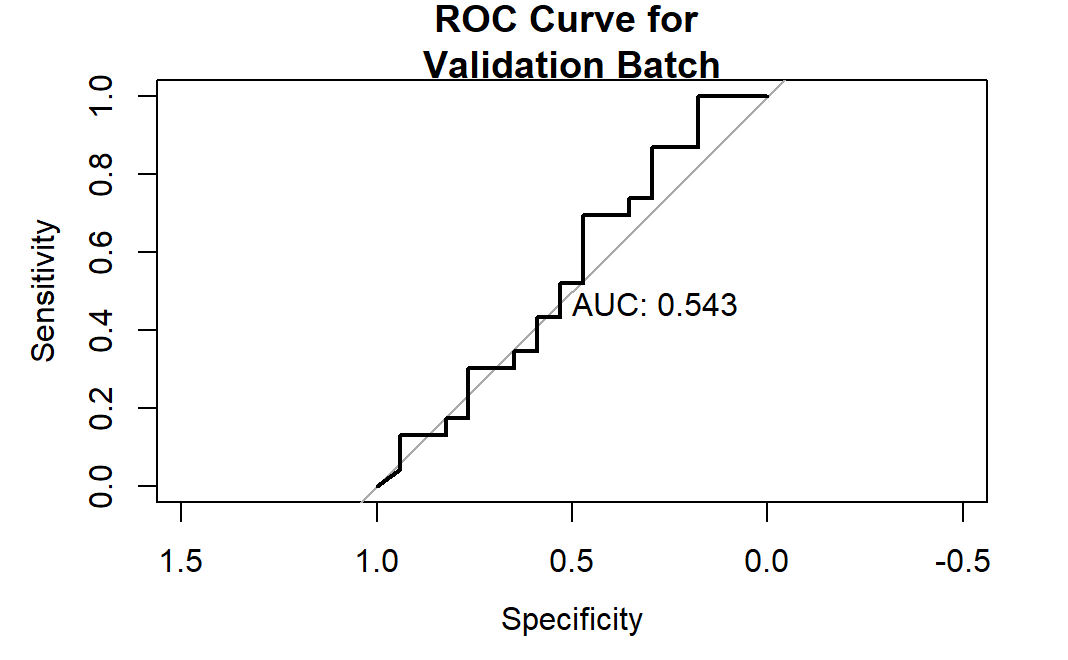
### 3.6.1 Results from Simulation

We first present the results of our simulated dataset and the targeted AUC for the high AUC model and low AUC model are presented in Fig. 3.9 and Fig. 3.10 respectively.

*Figure 3.9: Evaluation of High AUC Simulation Data* 

From Figure 3.9, we can see that the AUC values for both the validation batch and the testing set are close to 1, which means the simulated dataset produces a high AUC as expected.

*Figure 3.10: Evaluation of low AUC simulation data set*

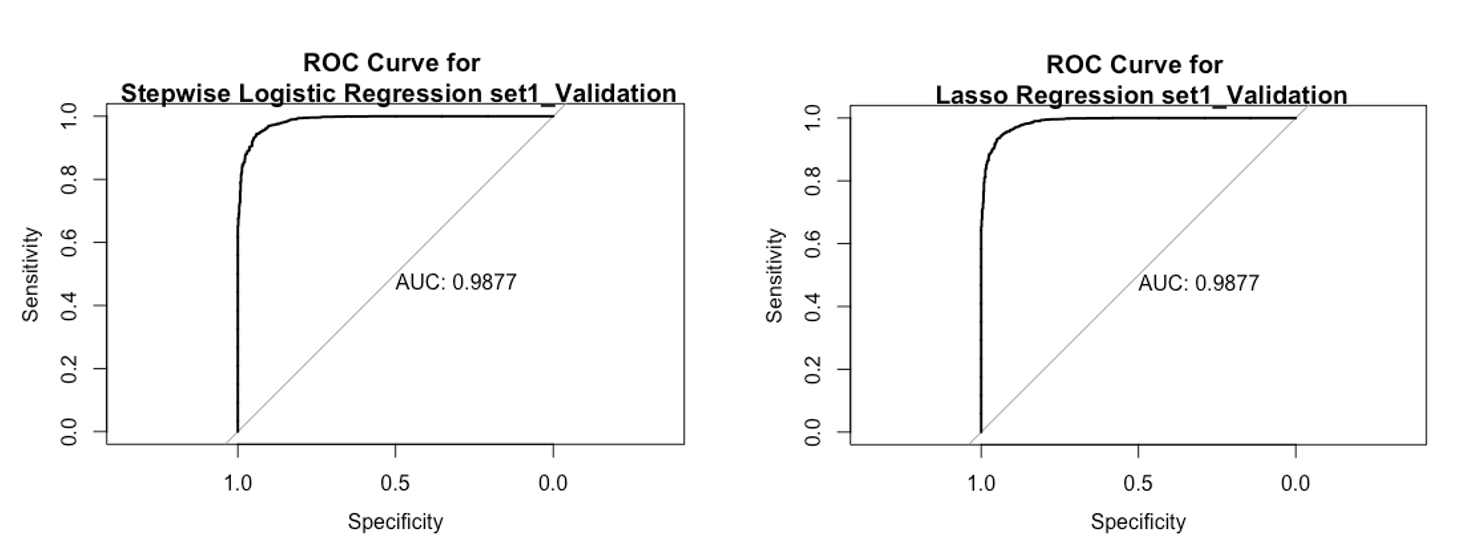


From Figure 3.10, the AUC for both the validation batch and the *sim.test.data* (test data) are both close to 0.5, which shows that the simulated dataset produces a low AUC as expected.

### 3.6.2 Analysis of Simulated Dataset

Next, we present the results of the analysis conducted on the simulated dataset received from the other group. Our final model for the high AUC dataset ended up with 29 terms. A total of two regression models including a stepwise regression and Lasso regression was fitted to the data and the model was evaluated using the validation dataset that was held out during the training. The ROC curves for both regression models and their corresponding AUC values can be found in Fig. 3.11. It can be seen from Fig. 3.11 that the AUC values for the two fitted models are similar with a value of 0.9877. Since there is no noticeable difference between these two models just by looking at the AUC values, we calculated the misclassification rate for the validation batch using these two models and the results are presented in Figure 3.12.

*Figure 3.11: ROC Curves for both regression models on Validation set*



*Figure 3.12:* *Misclassification Rate VS. Threshold Value for two models*

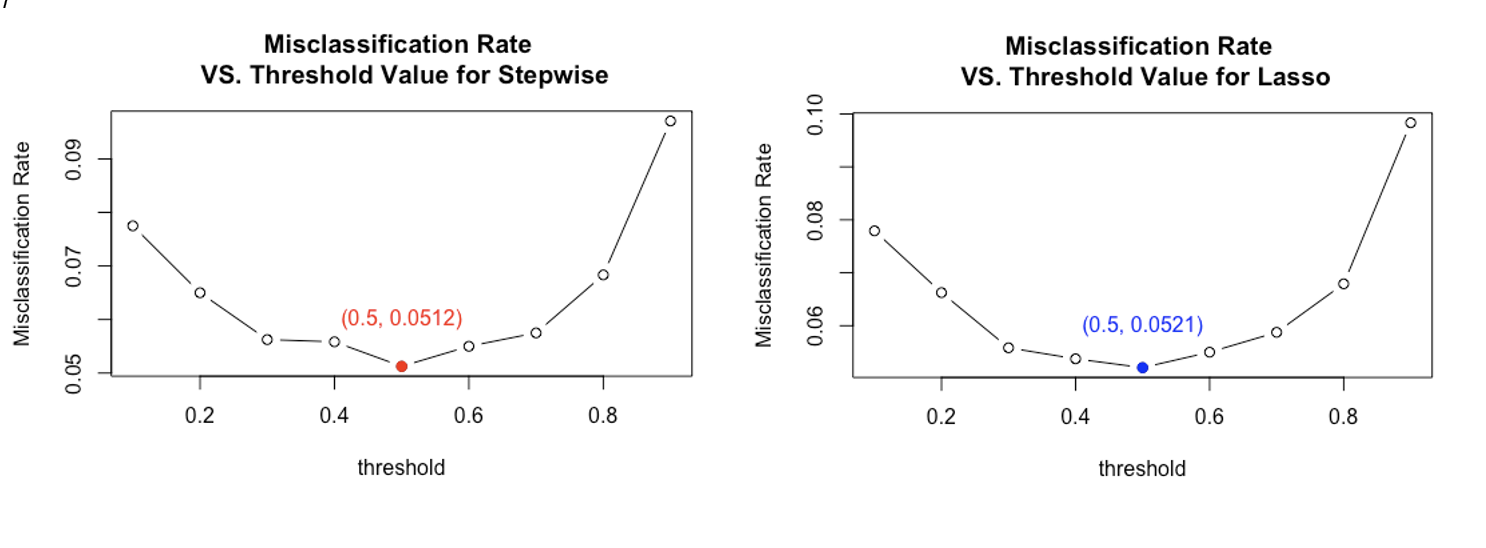
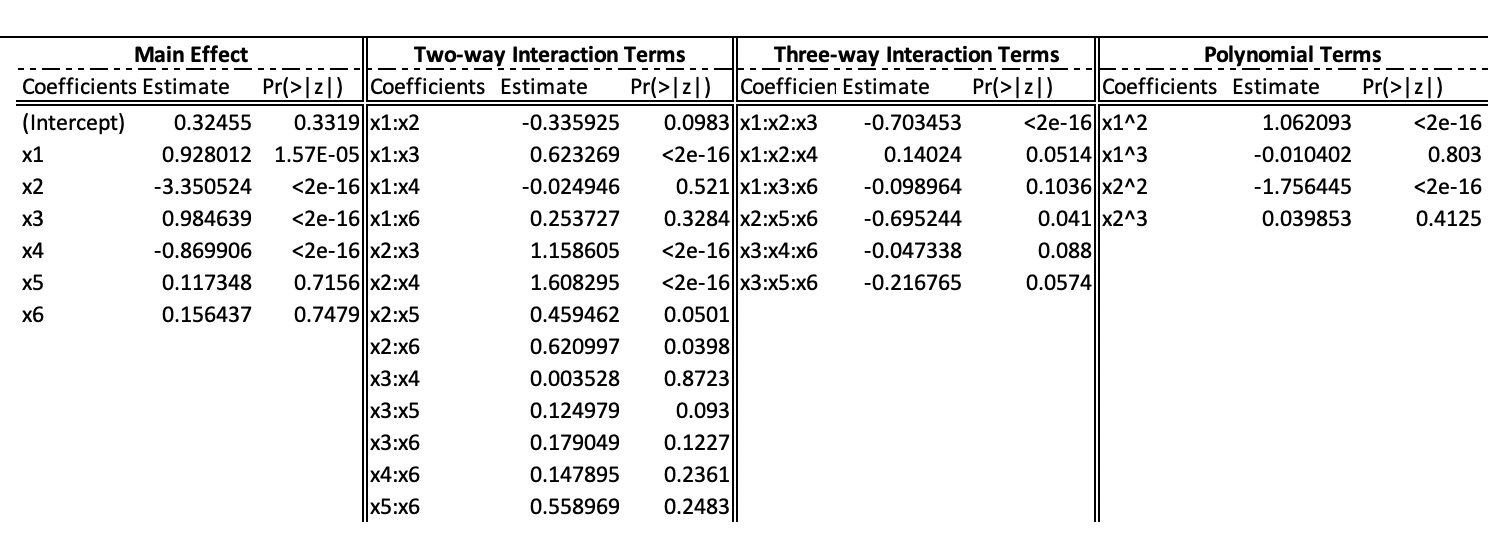


Figure 3.12 shows us the misclassification rate of the stepwise regression model is slightly lower than that of the Lasso model. Therefore, we decided to apply the stepwise regression model with cutoff equals 0.5 to predict Y value for the test set. Table 3.6 shows the predictors and the estimated betas for our final estimated regression model for training data set 1.

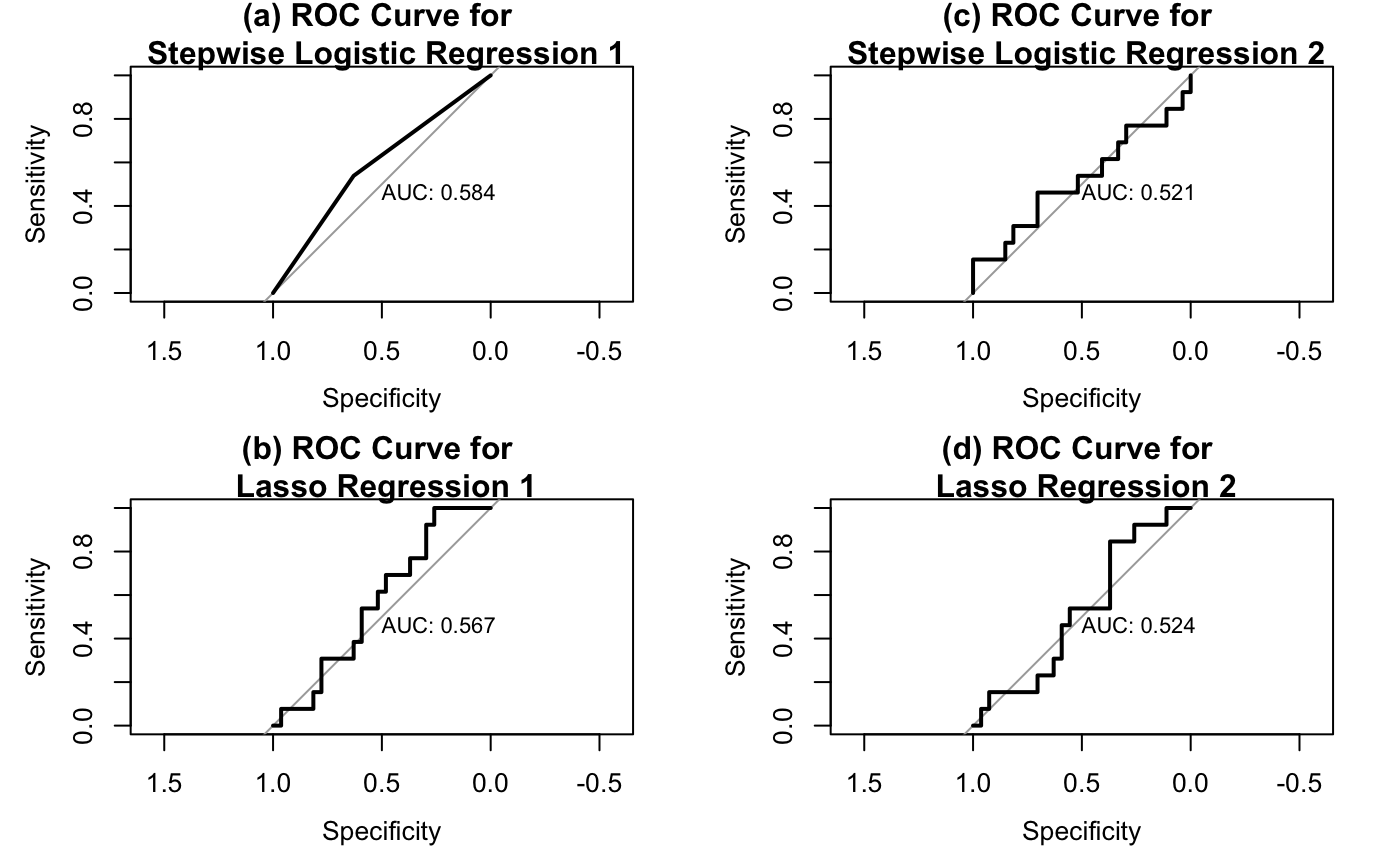
*Table 3.6: Final Estimated Regression Model for low AUC data set*

**

In fitting the simulated data expected to have a low AUC, we fitted a total of four models. For the first Stepwise Logistic Regression model and Lasso regression model, we include all the 6 main features, all pairwise interactions terms among the main effects, all three-way interaction terms and the 2nd and 3rd polynomial terms for the continuous main effect predictors . The performance is then evaluated using the validation data set and the result is captured in Figure 3.13a. and 3.13b.

The second Stepwise Logistic Regression model and Lasso regression model includes all the 6 main features, all pairwise interaction terms among the main effects, and the 2nd and 3rd polynomial terms for the continuous main effect predictors . The performance is then evaluated using the validation data set and the result is presented in Figure 3.13c. and 3.13d.

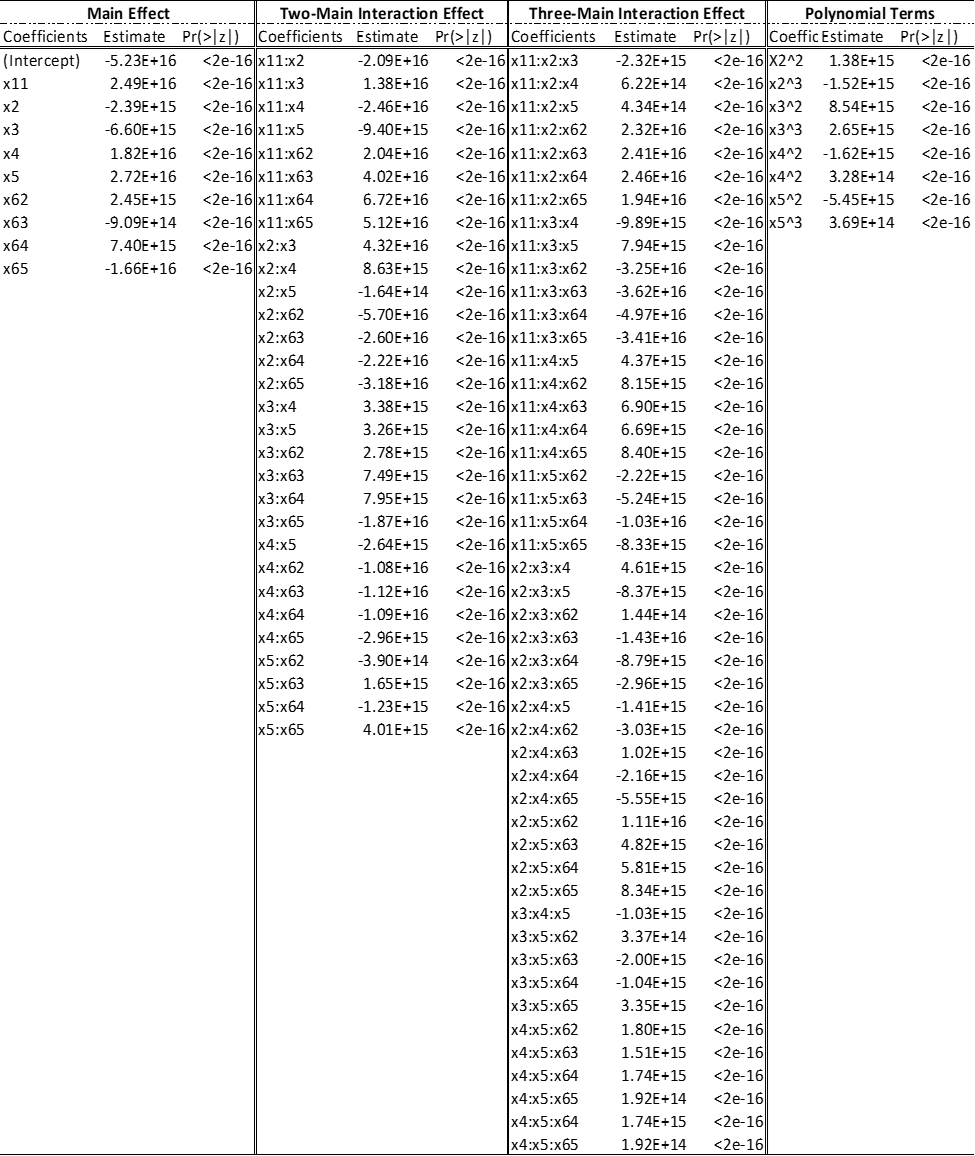
*Figure 3.13: ROC Curves for Low AUC dataset*



From Figure 3.13a, it is clear that the first model using the Stepwise Logistic model produces the highest AUC among all the other models considered with an AUC of 0.584 followed by the first model using Lasso Regression with an AUC of 0.567. It is worth noting that these models are more complicated as they consider three-way interaction between the main effect predictors. Also, the AUC is notably higher than the AUC of 0.504 achieved by the team that simulated the dataset. Next, the summary results of the best performing model is provided in Table 3.7.

It is quite intriguing to see that the summary results for the best performing model from Table 3.7 has really large coefficient estimates for the covariates, however all these covariates have p-values which are almost zero indicating that they are significant. Moreover, the AUC of this model was quite low (0.584) which suggests that having covariates that are significant might not be a good indication of the performance of a model.

*Table 3.7: Summary Results of Best Performing Model*



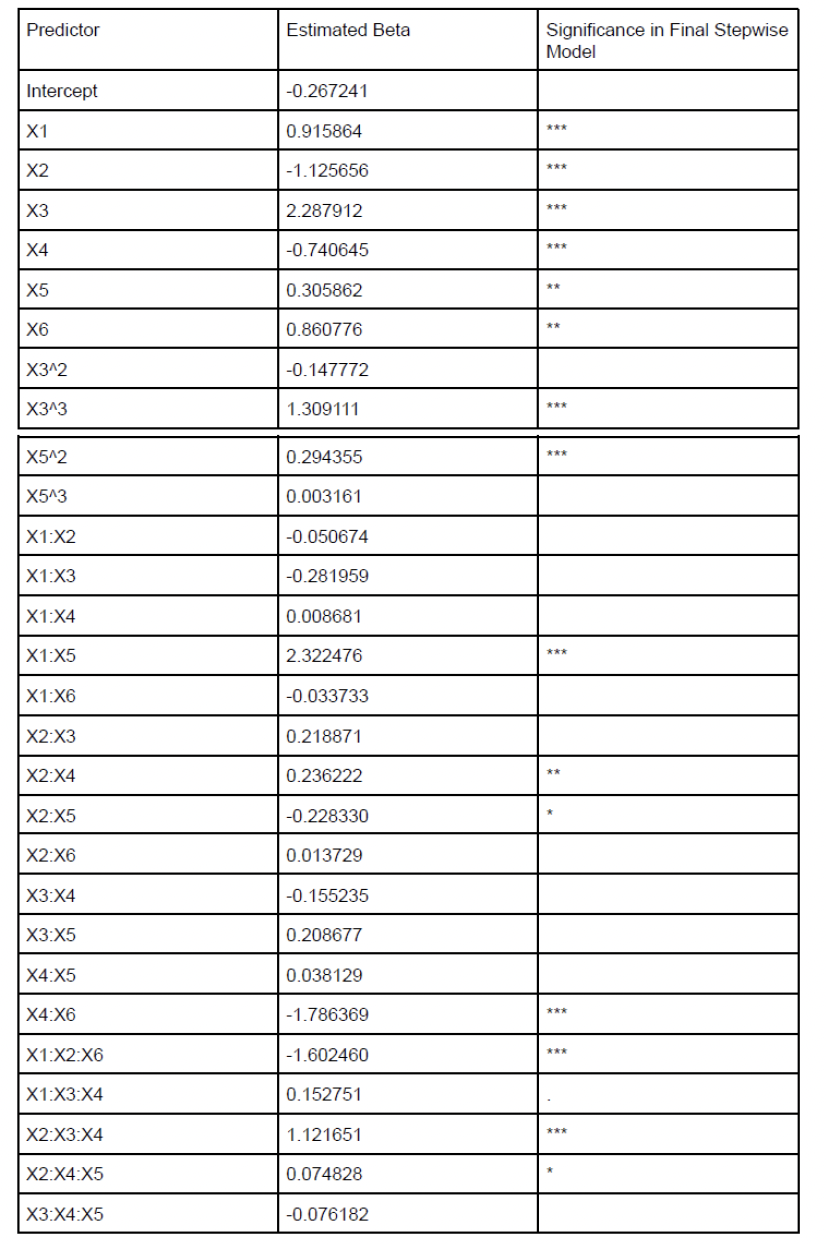
### 

### 3.6.3 Assessment of Fitted Model to Simulated Dataset

Here, we assess the model fitted by the other group to our simulated datasets. This assessment is based on the estimated beta values and predicted the “*y*” values for the testing data sets for both the low and high AUC simulated datasets received from the assigned group.

We first compare our “true” beta values with the estimated beta values, as well as the original predictors with the predictors chosen by the other group for the high AUC dataset. The estimated beta values for the high AUC dataset is shown in Table 3.8 and the comparison to the true beta values is presented in Table 3.9.

*Table 3.8: Estimated Easy Model*



*Table 3.9: Predictors and Beta Values of Original Easy Model and Estimated Easy Model*

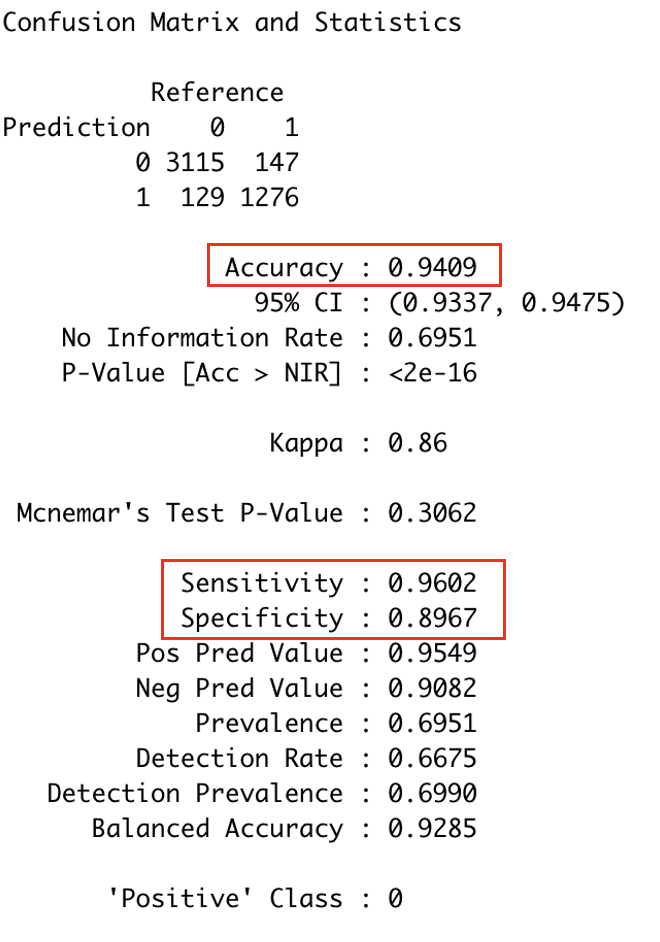
Table

Description automatically generated

From Table 3.9, the beta values in the estimated model for *x2*, *x4*, *x4\*x6*, and *x1\*x2\*x6* are negative, which is consistent with our original model. Secondly, we find that the beta value of *x2* in the estimated model is relatively smaller than that in the original model. The beta value of *x3* and the beta value of *x6* are relatively larger than that of the original model. Otherwise, the rest of the beta values are very similar to each other. All the information shows the estimated model is very similar to our original model.

In addition, we investigate how the estimated model for the high AUC performs on the test dataset. From Table 3.10, we can see the accuracy is 0.9409, which is very good. It indicates 94% of the cases in the test set are classified correctly. Meanwhile, we examine the other two important measures of performance, sensitivity and specificity. The results show us that the sensitivity is very high (*at 0.9602*) and the specificity is not so low (*at 0.8967)*. It denotes that the estimated model performs well in detecting both positive cases and negative cases. In conclusion, the estimated model is pretty good.

*Table 3.10: Confusion Matrix for Easy Model*

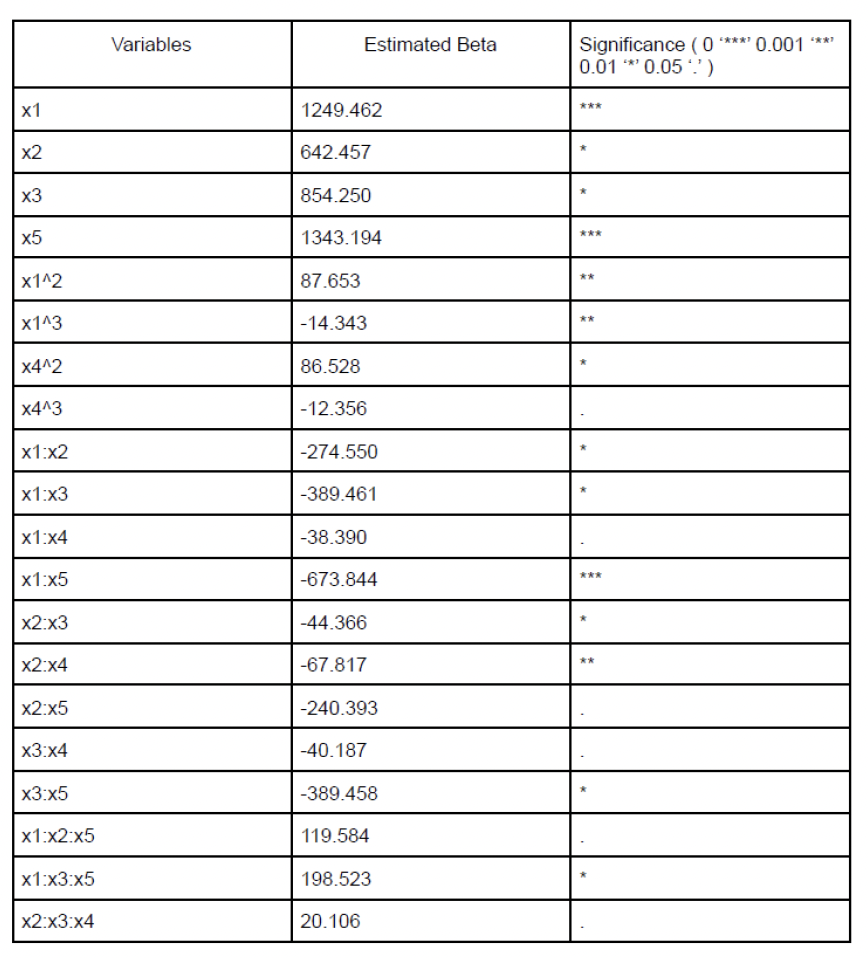


Next, we compare our “true” beta values with the estimated beta values, as well as the original predictors with the predictors chosen by the other group for the low AUC dataset. The estimated beta values for the low AUC dataset is shown in Table 3.11 and the comparison to the true beta values is presented in Table 3.12.

## 

## 

*Table 3.11: Estimated Difficult Model*



*Table 3.12:* *Predictors and Beta Values of Original Difficult Modal and Estimated Difficult Model*

Table

Description automatically generated

We have a total of 12 predictors in our original model, which are shown in Table 3.12. However, there are 20 predictors in the estimated model, which are shown in Table 3.11. From Table 3.12, the beta values are significantly different for each predictor. Not only for the beta value signs but also for the magnitude of beta values. In terms of predictors and beta values, the estimated model is not so close to our original model.

In addition, we investigate how the estimated model for the low AUC performs on the test dataset. From Table 3.13,the estimated difficult model performs very badly on the testing data set in terms of accuracy. However, when we focus on the sensitivity and specificity of the model, we find the specificity is not very low. We note that a Stepwise regression method was used to determine the estimated betas whereas K-nearest neighbor was deployed to predict the outcome for the test set.

*Table 3.13: Confusion Matrix for Difficult Model*

Table

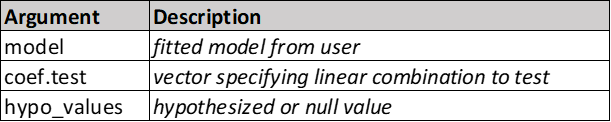
Description automatically generated

## 3.7 Illustration of R Functions and Package

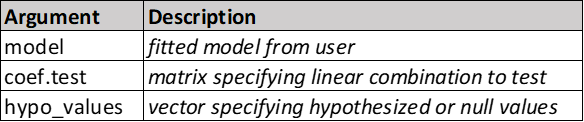
For the purpose of doing the tests described in section 2.7, we present three functions which will all be hosted under one package in R. The functions are **t.test\_coef** and **f.test\_coef**, corresponding to the *t*-test described for testing one linear combination and the *F*-test for several linear combinations, respectively. In addition, we use the **coef\_conf.int** function to obtain a confidence interval for the case of one linear combination.

### 3.7.1 Description of Arguments of Function

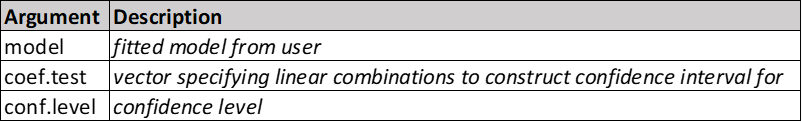
* **t.test\_coef**



* **f.test\_coef**



* **ceof\_conf.int**

****

### 3.7.2 Testing with Linear Regression

We first conduct the test for a multiple linear regression using the popular **iris** data in R. We fit a multiple regression model with *Sepal.Length* as the response variable and *Sepal.width* and *Petal.width* as the independent variables. A summary of the fitted regression model is shown below:

Call:

lm(formula = Sepal.Length ~ Sepal.Width + Petal.Width)

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.4573 0.3092 11.18 < 2e-16 \*\*\*

Sepal.Width 0.3991 0.0911 4.38 0.000022 \*\*\*

Petal.Width 0.9721 0.0521 18.66 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.451 on 147 degrees of freedom

Multiple R-squared: 0.707, Adjusted R-squared: 0.703

F-statistic: 178 on 2 and 147 DF, p-value: <2e-16

**Example I: Testing One Linear Combination**

Here, we test the hypothesis that the coefficient of the *Sepal.width* and *Petal.width* are equal using the **t.test\_coef** function. The null hypothesis can be expressed as:



t.test\_coef(model=lm.fit,coef\_test=c(0,1,-1),hypo\_values=0)

Test for Linear Combination

lm

Sepal.Length ~ Sepal.Width + Petal.Width

Results:

t-test with 147 degrees of freedom

Alternative: the linear combination equals the specified value

Estimate Std. error t value P(T>|t|)

[1,] -0.5730589 0.08683009 -6.599773 3.496626e-10

**Example II: The Confidence Interval**

Next illustrate the **coef\_conf.int** function by constructing a 95% confidence interval for the linear combination of . This is imputed in R as shown below:

coef\_conf.int(model=lm.fit,coef\_test=c(0,1,-1),conf.level=0.95)

95% Confidence Interval for linear combination

lm

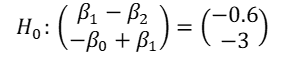
Sepal.Length ~ Sepal.Width + Petal.Width

Estimate lower limit upper limit

-0.5731 -0.7447 -0.4015

**Example III: Testing Several Linear Combinations**

Lastly, we demonstrate how the **f.test\_coef** function can be used to test an hypothesis involving several linear combinations of the coefficients in the fitted regression model. We test the following hypothesis:



my\_matrix <- matrix(c(0,1,-1,

-1,1,0), byrow = TRUE, nrow = 2)

my\_hyp <- c(-0.6,-3)

f.test\_ceof(model=lm.fit,coef.test=my\_matrix,hypo\_values=my\_hyp)

Test for Linear Combinations

lm

Sepal.Length ~ Sepal.Width + Petal.Width

Results:

F-test with 2 and 147 degrees of freedom

Alternative: the linear combination equals the specified vector

F value P(F>f)

[1,] 0.9875927 0.3749297

### 3.7.3 R Package

Now, we present a new package, **LinearHypoTest**, in R which houses these functions. This package is available on *Github* and could be downloaded and installed (free of charge). How does one obtain this package for use?

* First, one needs to install and load R development tools. The user could use install.packages("devtools"), followed by library(devtools).
* Download and install the package from Github repository using install\_github("[eamponsah/LinearHypoTest](https://github.com/eamponsah/LinearHypoTest)"), followed by library(LinearHypoTest).

The package has now been installed and ready for use by the user. A brief description of the functions and their usage has been provided in the form of R documentation and could be accessed by running ?t.test\_coef, ?f.test\_coef, or ?coef\_conf.int.

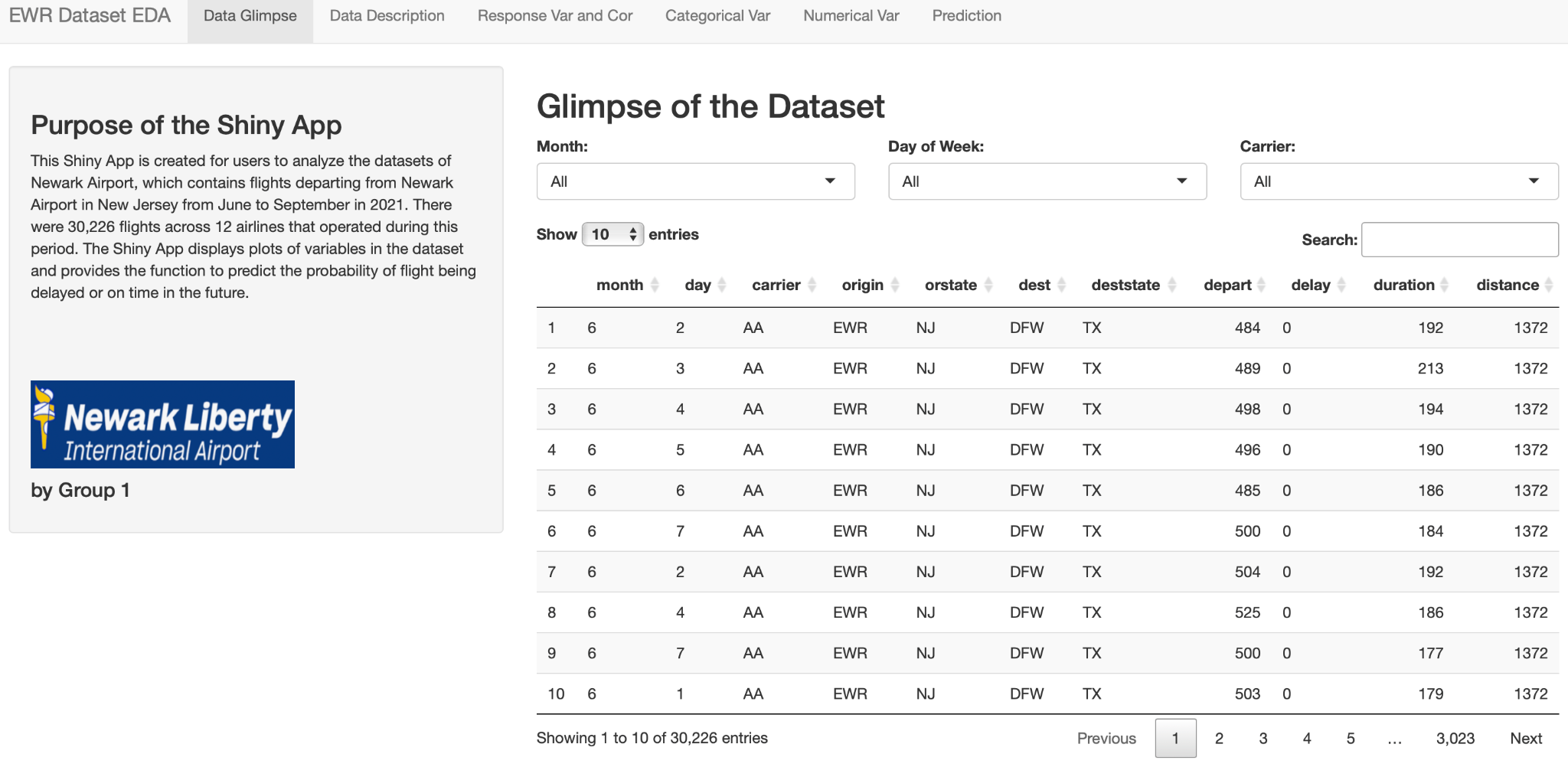
## 3.8 Illustration of R Shiny App

For our Shiny app, we built an interactive web app that performs the exploratory data analysis and visualization as we did in assignment 2. In addition to data visualization, our app also allows a user to predict the probability of a flight departing from the Newark airport being delayed or on-time.

### 3.8.1 Description of R Shiny App

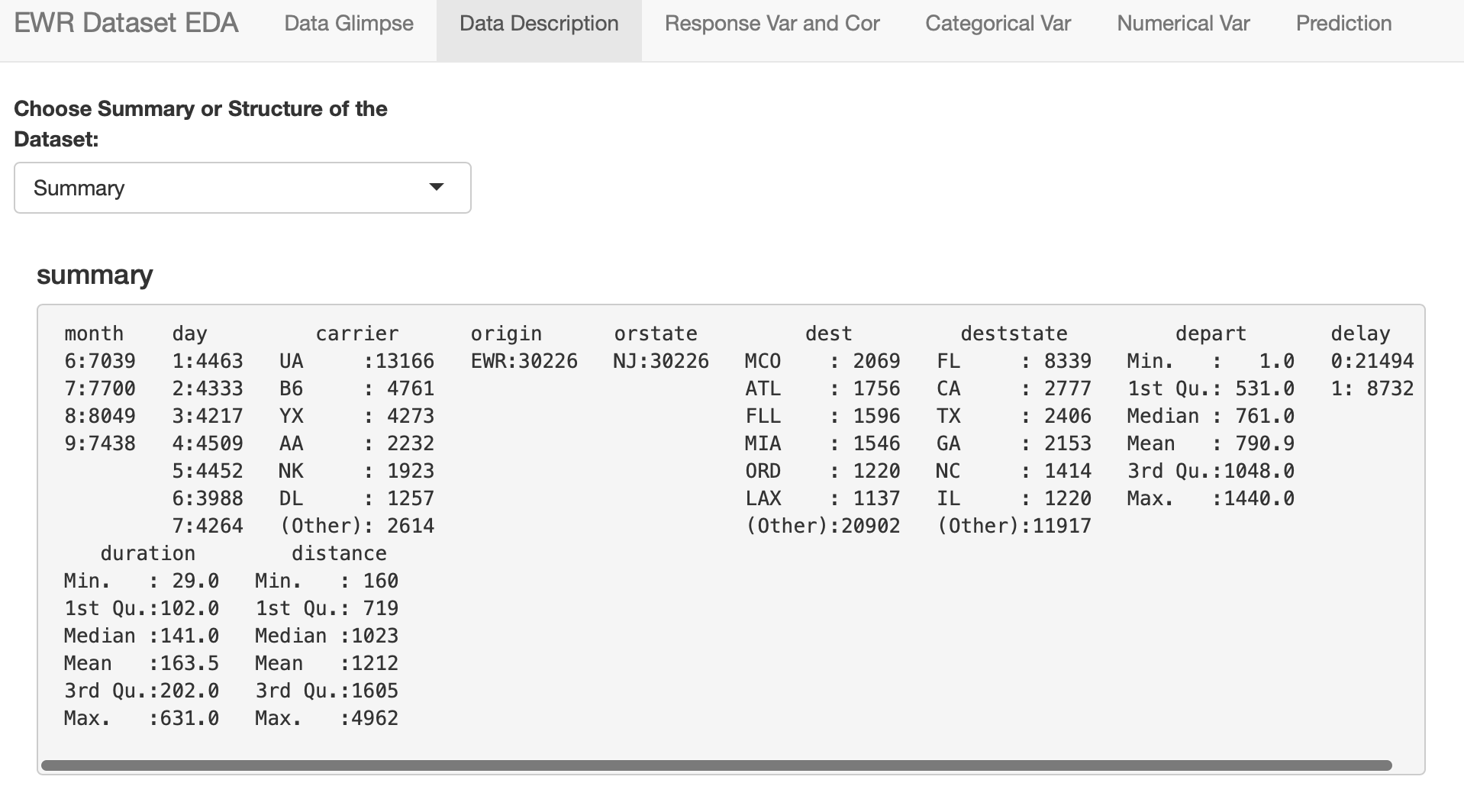
The Shiny App consists of six navigable tabs created by using Navbar functionality. The main page, which the user first lands on, includes a description of the Shiny App along with its purpose on the left hand side of the screen. Additionally, this main page is titled Glimpse of the Dataset and allows the user to actually view all 30,226 entries with increments on 10, 25, 50, or 100 visible at once as shown in Figure 3.14. The data can be further subset by Month, Day of the Week, or Carrier. The user can also use a search bar for particular values.

*Figure 3.14: Glimpse of Dataset from Shiny App*

**

The second page in our Shiny App, navigable by a tab bar at the top of the page, is labeled Data Description. This page has a dropdown menu with two options: summary, and structure. The summary option displays a summary of all 11 variables such as counts for the categorical variables, and quartiles, median and mean for the continuous variables as shown in Figure 3.15. The structure option gives the structure of each variable such as data type and number of levels. The page also features a description of each variable which is visible for both summary and structure.

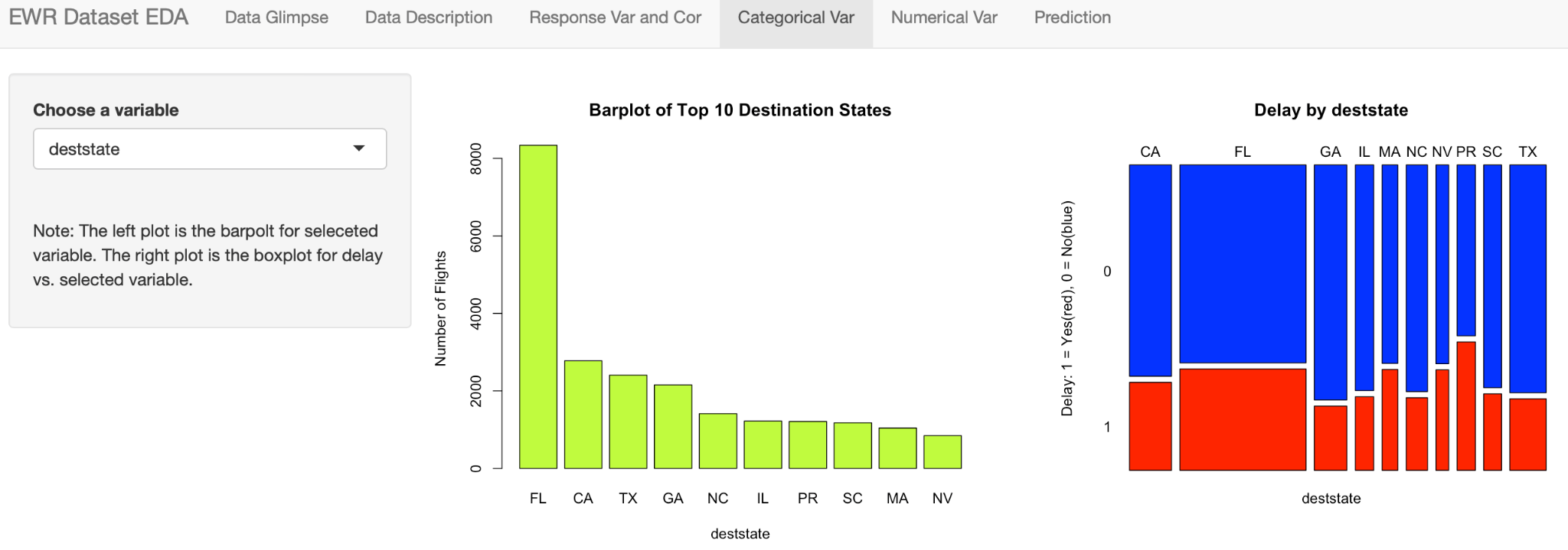
*Figure 3.15: Data Description Summary from Shiny App*

**

Following the Data Description Page is Response Var and Cor which provides a bar plot of our response variable Delay and a correlation matrix for the three continuous variables (depart, duration, and distance) and delay. The dropdown menu on the page allows the user to choose whether to display the correlation matrix with ovals or numerically.

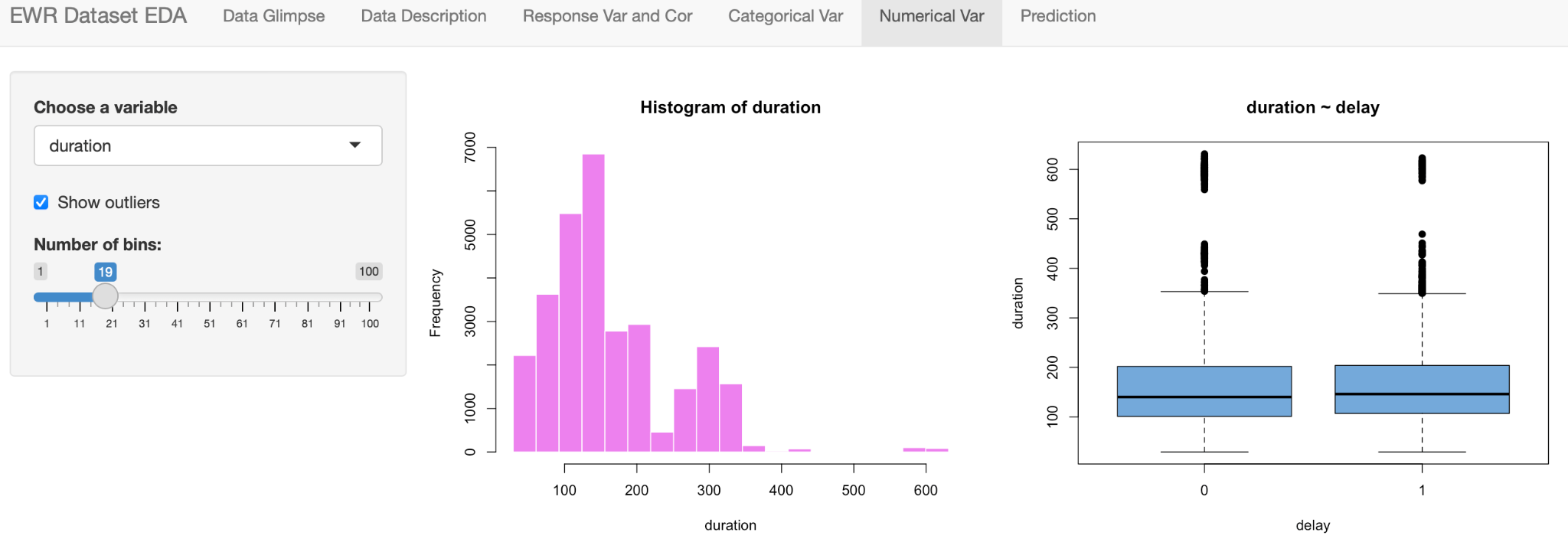
The fourth page available to the user of our Shiny App is labeled Categorical Var and displays a bar plot and mosaic plot. The bar plot displays the count of five categorical variables (month, day, carrier, destination airport, and destination state) along with a mosaic plot of the chosen variable and flight delays. The plots for destination and destination state only display the top 10 as there are too many options to fit comfortably on the plot. Figure 3.16 displays a bar plot for the top 10 destination states along with the corresponding mosaic plot. There is a drop down menu which allows the user to select which categorical variable is displayed in the plots.

*Figure 3.16: Bar plot and Mosaic Plot from Shiny App*

**

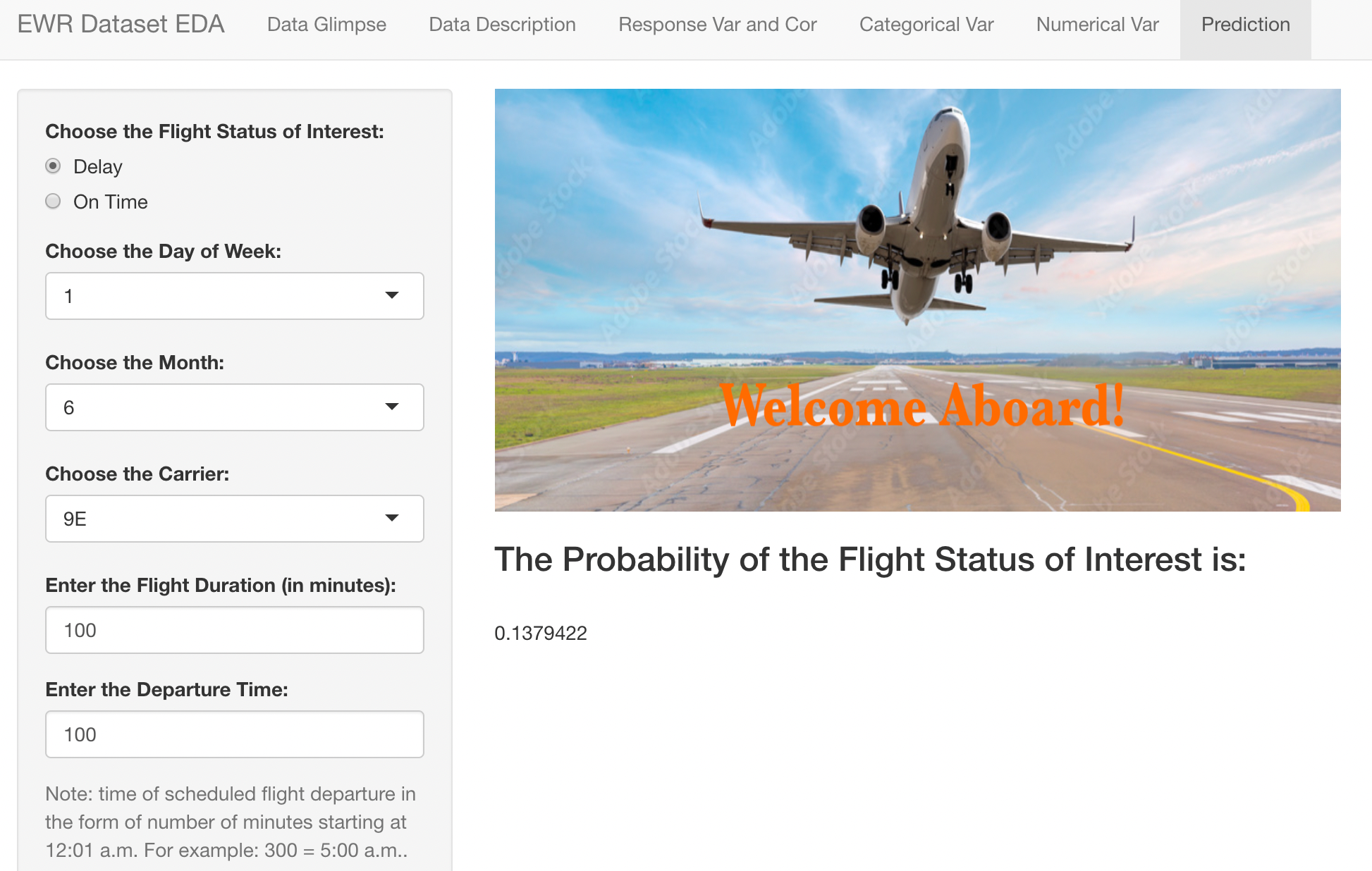
The fifth page shows a histogram of three numerical variables (depart time, duration, and distance) and the corresponding box plots for the variables and delay status. The drop down box allows the user to select which numerical variable is displayed, and there is a slider allowing the number of bins on the histogram to be altered. The number of bins can range from 1 to 100. There is also a checkbox allowing the user to determine whether outliers are displayed on the box plot. Figure 3.17 displays an example histogram and box plot for flight duration.

*Figure 3.17: Flight Duration Histogram and Box Plot in Shiny App*

**

The final page of our Shiny App allows the user to input various variable values to determine the probability of a flight departing from Newark either being delayed or on-time. There are six variable choices for the user to determine on this page. The first is whether the user wishes to know the probability of a flight being delayed or on-time. This choice is made using a radio button. Following this, there are three drop down selection menus where the user chooses the day of the week, month, and carrier for the analysis. Finally, there are two numerical input boxes for flight duration (in minutes) and departure time. Departure time is measured in minutes after 12:00 am; for example if a user inputs 300 then the corresponding departure time is 5:00 am. Using these inputs for our variables, the Shiny App then uses two models developed for Assignment 5, found with lasso regression, to determine the associated probability. Figure 3.18 shows an example of this page with the user interested in the probability of delay.

*Figure 3.18: Probability of Delay in Shiny App*

**

# 

# 

# 4. Conclusion

In this project, we have examined flight data from the Newark Liberty International Airport. Among other things, we found that flying on Wednesday, Thursday, and Friday were associated with higher odds of getting delayed as compared to flying on Monday. Flying with the carriers Frontier, Jetblue, Allegiant Air, and American Airline, one was more likely to get delayed than flying with Endeavor. Also, relative to the month of June, there was increased odds of getting delayed for those who flew in July and August. On the other hand, the odds were decreased for flights that occurred in September. Overall, we noticed that increasing the complexity of the model led to a marginal improvement in the AUC, although this improvement did not necessarily translate to gains in other performance metrics. It is also important to note that the stepwise selection technique tends to overstate statistical significance, and should be treated with caution.

Ultimately, the choice of which model to use would largely depend on the class that is of great interest to the end user — i.e whether predicting on-time flights is more important than predicting flight delays or vice versa. For example, in the results obtained when fitting the Lasso logistic regression, if our interest is to do better in predicting on-time flights, then *Model 3* would be preferred, since it does better on the positive class (ie., on time). However, *Model 2* would be appropriate to use if we are more interested in predicting delayed flights.

In the simulation exercise, we observed that results obtained were specific to “seeds” used to generate and/or partition the data. Essentially, how one obtains or partitions a given data set significantly impacts results from that data set. Also, the simulation exercise showed that having several of the covariates in a model being significant may not necessarily be a good indication of the performance of a model.

In the analysis of the results from the other group part, we observed that the estimated easy model performs very well on the testing data set. All three important metrics values are very high. On the other hand, the estimated difficult model performs very badly on the testing data set in terms of accuracy. However, when we focus on the sensitivity and specificity of the model, we find the specificity is not very low. It means this model may be used for an imbalanced data set that contains more negative cases. Even though accuracy is one of the most commonly used metrics to assess the performance of a classification model, it sometimes could mislead us. ​​This is because accuracy counts all of the true predicted values, not for a specific label. Lower accuracy does not mean that we have a bad performance in predicting a specific label.

Also, we have proposed R functions and a package for making inference about linear combinations of parameters in a linear regression model. Although it would not be explored in this current study, it would be interesting to also look at how these functions translate into other types of regression. We further developed an R Shiny app for handling data visualization on the Newark Airport data.

Finally, we built an R Shiny App for users to analyze datasets of Newark Airport and predict the probability of a flight departing from the Newark airport being delayed or on-time. It helps users better understand the dataset with an interactive web application.

# References

1. Bureau of Transportation Statistics (2022). <https://www.transtats.bts.gov/homedrillchart.asp> (accessed 9/15/2022).
2. Fawcett, T. (2006). An Introduction to ROC Analysis. Pattern Recognition Letters. 27 (8): 861–874.
3. Harrell, F. E. (2001). Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis (Vol. 608). New York: springer.
4. Hosmer, D.W., Lemeshow, S. (2000). Applied Logistic Regression, 2nd Ed. Chapter 5, John Wiley and Sons, New York, NY, pp. 160-164.
5. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R (sixth printing). New York: Springer.
6. Port Authority of New York and New Jersey (2015). <http://www.panynj.gov/about/pdf/property-owned-leased-port-authority.pdf> (accessed 9/15/2022).
7. Sperandei, S. (2014) Understanding logistic regression analysis. Biochemia Medica 24: 12-18.
8. Theil, H. (1971). Principles of econometrics. New York: Wiley.
9. Weisberg, S. (2005). Applied linear regression (Vol. 528). John Wiley & Sons.
10. Weiss GM. 2013. Foundations of imbalanced learning. Imbalanced Learning: Foundations, Algorithms, and Applications, 13-41.
11. Xu, N., Sherry, L. and Laskey, K.B., 2008. Multifactor model for predicting delays at US airports. *Transportation Research Record*, *2052*(1), pp.62-71.

# Contributions

