Flight Delays in Tampa Airport and Simulation Study

Hasmik Grigoryan, Hanyu Xiao, Rose Porta, Sandani Kumanayake

December 11, 2023

1 Abstract

This project presents a comprehensive analysis of flight delays at Tampa Airport, augmented by an in-depth simulation study, with the ultimate goal of creating a Shiny application. This application is designed to predict the probability of flight delays at Tampa Airport, with the aim to deliver an intuitive user experience.

Initially, the study involved a detailed examination of the current state of flight operations at Tampa Airport. The analysis of the patterns and frequency of delays using various data visualization techniques, including bar plots, histograms, mosaic plots, box plots, and correlation plots. Through detailed analysis of historical data, the study identified common causes of delays, periods of peak delays, and carriers that are most likely to experience delays at Tampa airport.

Subsequently, the study employed advanced statistical models, focusing on logistic regression, to study the underlying complexities of flight delays. The analysis also incorporated variable selection methods, specifically stepwise regression, and shrinkage methods, i.e., Ridge and Lasso regression, to establish the optimal model for explaining flight delays. Moreover, tools such as the confusion matrix were utilized after the introduction of the logistic regression model. Additionally, the ROC(Receiving Operator Characteristics) [12] curve was employed to visualize the performance of the classifier models, complemented by the use of the AUC(Area under the ROC Curve) [12] metric for a numerical comparison of the models' relative performances.

Following the analytical phase, the project moves to a simulation study. This phase involved the application of logistic regression models to simulate various operational scenarios, categorized into high AUC and low AUC models. The study then integrated Stepwise regression and Lasso regression models to identify the highest and lowest AUC model. Finally, the highest and lowest AUC simulated models were compared with actual models. This com-

parison was conducted using confusion matrices and predictive accuracy assessments, thus providing a comprehensive understanding of the models' efficacy.

2 Background

The data set comprises 33,536 observations from Tampa International Airport for the period between April and September 2022. It encompasses 11 variables, detailing flight data on specific days of the week (e.g., number of flights on Mondays, Tuesdays, etc.). Additionally, information on the carriers operating at Tampa Airport is included, with identifiers such as "AA" for American Airlines and "AS" for Alaska Airlines. In total, 11 airline companies operate from this airport. The data set also captures the destination airports and states for these flights. For instance, destination airports are represented by codes like "ACY" for Atlantic City International Airport, "ATL" for Hartsfield-Jackson Atlanta International Airport, and "CLT" for Charlotte Douglas International Airport, among others. There are 64 such destination airports and 35 destination states, including California, Alaska, Texas, and even Florida itself. The dataset also indicates flight delays, with "0" denoting no delay and "1" signifying a delay. Departure times are provided for flights leaving Tampa International Airport, which, being an international hub, sees departures every hour. Flight distances in the data set range from 197 to 2,520 miles [3] [7] [8] [10] [9] [1] [16].

The study of flight delays, particularly at a bustling hub like Tampa Airport, requires a nuanced understanding of various statistical models and methods, as explored in the series of eight comprehensive assignments. These assignments lay the groundwork for the analytical techniques employed in this project, offering a rich foundation in advanced statistical modeling [15] [4].

Logistic Regression: The cornerstone of our analysis, logistic regression, has been extensively explored in the assignments to model binary outcomes, such as the occurrence of flight delays. This approach allows for a detailed examination of how different variables, such as departure time, duration of the flight, and carriers, etc., influence the likelihood of delays.

Model Evaluation Using AUC and ROC Curves [17]: The assignments have also emphasized the importance of rigorous model evaluation. AUC (Area Under the Curve) and ROC (Receiving Operating Characteristic) curves have been instrumental in assessing the performance of our predictive models, ensuring their reliability and accuracy in real-world scenarios.

Stepwise Regression [14]: Employed in refining our models, stepwise regression [17] techniques have helped in selecting the most relevant variables, ensuring our analysis remains focused and robust. This method is particularly valuable in handling the vast datasets typical of airport operations, where numerous factors can potentially influence delays.

Ridge Regression: Ridge regression is one common shrinkage method, which takes both goodness of fit to the data and variance into account by introducing a “shrinkage penalty” term into the estimation process, which shrinks coefficients toward zero.

Lasso Regression: Lasso regression, another vital shrinkage technique, improves model accuracy while preventing overfitting. It identifies and prioritizes the most significant predictors of flight delays by penalizing the absolute size of the regression coefficients, thereby refining the model for optimal predictive accuracy.

Simulation Techniques: Beyond analytical modeling, our project incorporates simulation studies, as outlined in the assignments. These techniques allow for the creation and analysis of virtual models that closely emulate real-world systems, particularly beneficial where direct experimentation is unfeasible or prohibitively expensive. By simulating various operational scenarios, we can conduct experiments and make predictions about system behavior under different hypothetical conditions [11].

3 Methods

3.1 Data visualization

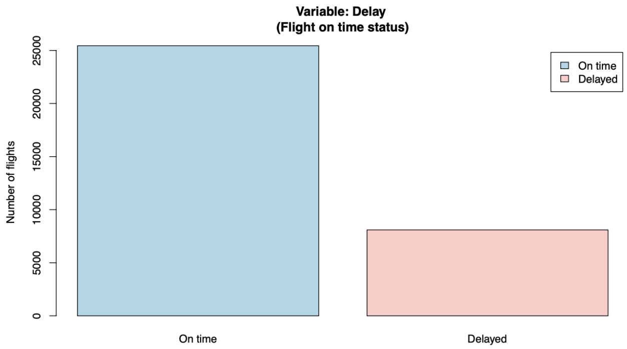


Figure 1: Barplot of Flight on time status

(1) Figure1

Figure 1 unequivocally illustrates a predominant trend of timely operations, showcasing Tampa Airport’s commitment to operational excellence and reliability.

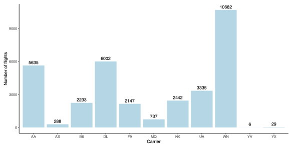


Figure 2: Barplot of Flights by Carrier

(2) Figure2

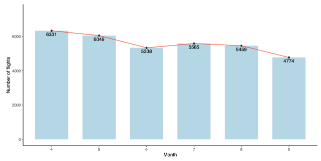


Figure 3: Barplot of Flights by Month

(3) Figure3

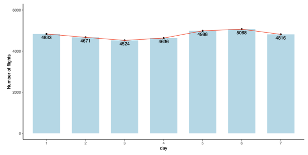


Figure 4: Barplot of Flights by Day

(4) Figure4

The study also aimed to discern the prominent airlines at Tampa Airport, Figure 2 is revealing Southwest Airlines Corporation, Delta Air Lines, and American Airlines as the leading carriers over a six-month span in 2022. Such insights are crucial for understanding market dynamics and consumer preferences within the region's airline industry; A meticulous examination of the data set offered valuable insights into monthly flight volumes from April to September(Figure 3). We observed a consistent decremented trend, with April experiencing the peak and September marking the lowest in-flight volumes, providing essential data for informed strategic planning and operational adjustments by stakeholders; Further, our analysis of flight frequencies uncovered a higher concentration of flights on Fridays and Saturdays, possibly due to a preference for weekend travel, highlighting areas for potential scheduling optimization and service enhancement [5].

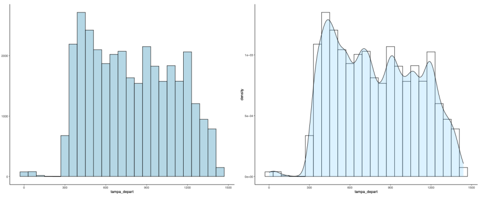


Figure 5: Histogram and Density Plot of Depature Time

(5) Figure5

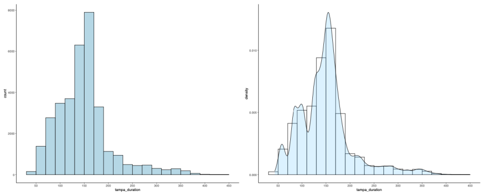


Figure 6: Histogram and Density Plot of Duration

(6) Figure6

We also explored the distribution of departure times using histogram and density plots, identifying peak flight activity around the early morning hours, with a secondary peak at 10 a.m. and a subsequent decline post 8 p.m. This pattern, potentially indicative of passenger preferences and operational strategies, emphasizes the significance of optimizing departure schedules; According to Figure 6, an exploration of flight duration revealed a great number of flights within the 150 to 180 minute range, indicating a strategic focus on short-haul flights due to passenger preferences and geographical considerations, aligning with the operational and economic objectives of the airlines.

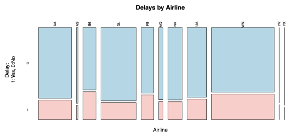


Figure 7: Mosaic Plot of Flight Delays by Airline

(7) Figure7

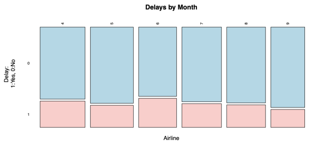


Figure 8: Mosaic Plot of Flight Delays by Month

(8) Figure8

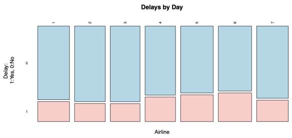


Figure 9: Mosaic Plot of Flight Delays by Day

(9) Figure9

Our detailed examination of mosaic plots offered nuanced insights into the interrelation between flight delays, airlines, and scheduling, emphasizing the prevalent trend of on-time operations and operational efficiency. However, notable variations were observed among airlines, with AS and YX airlines showcasing exemplary punctuality, contrasted by a higher delay propensity in B6 airlines; Further, a deep dive into the monthly delay patterns highlighted June as a critical month with heightened delay occurrences, possibly due to increased travel demand during the summer vacation period, leading to operational strains; Our evaluation of weekly delay patterns indicated heightened delays during Thursdays, Fridays, and Saturdays, reflecting increased end-of-week travel preferences and potential operational bottlenecks, emphasizing the need for strategic interventions to enhance operational efficiency and punctuality.

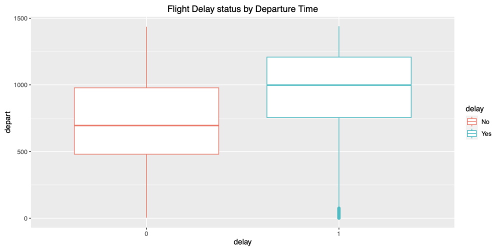


Figure 10: Box Plots of Flight Delay status by Departure Time

(10) Figure10

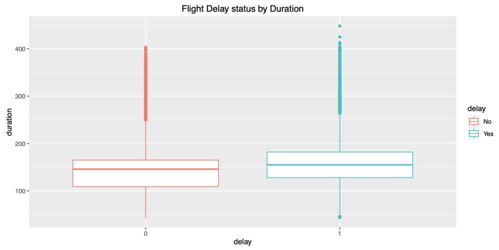


Figure 11: Box Plots of Flight Delay status by Duration

(11) Figure11

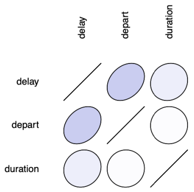
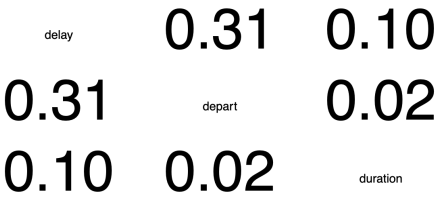


Figure 12: Correlation Plot of Flight Delays, Departs and Duration

(12) Figure12



delay	0.31	0.10
0.31	depart	0.02
0.10	0.02	duration

Figure 13: Correlation Plot of Flight Delays, Departs and Duration

(13) Figure13

Lastly, our examination of box plots and correlation plots corroborated the significant association between departure times and delay occurrences, underscoring the critical role of departure scheduling in maintaining operational efficiency and mitigating delays.

In conclusion, our comprehensive analysis has illuminated the intricate dynamics of flight operations at Tampa Airport, providing insights into operational strategies, passenger preferences, and areas requiring optimization to enhance the passenger travel experience and operational coherence.

3.2 Logistic Regression

3.2.1 Logistic Regression Methodology

In mathematical terms, logistic regression models $P(Y = yes|X)$ where Y represents our binary response and X represents the explanatory variable(s). In order to model this, we use the logistic function defined as

$$p(x) = \frac{e^{(\beta_0+\beta_1 X)}}{1 + e^{(\beta_0+\beta_1 X)}}$$

The use of this function ensures that we will get values between [0, 1]. We estimate the coefficients using maximum likelihood estimation (MLE) [14].

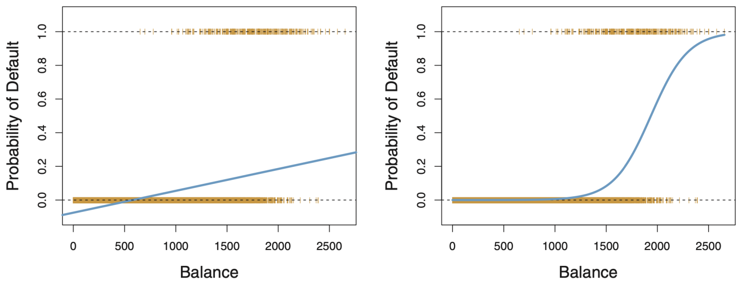


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default(No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

(14) Figure14

Since this model is not as straight-forward as linear regression, it is more difficult to interpret the meaning of the coefficients when we use this form of the function. For this reason, we consider a different manipulation of the same function:

$$\frac{p(X)}{1-p(X)} = e^{(\beta_0+\beta_1 X)}$$

The expression on the left hand side of the equation is called the odds. Odds are related to probabilities, but can take on any value between [0, infinity). We can think of odds as a comparison of the relative chances of two possible outcomes. For example, an odds value of four (also could be written as four to one) means that for every four "success", we observe one "failure" on average. A helpful reference point is that an odds value of 1 (or 1 to 1) is equivalent to a 0.5 (random chance) probability. This means that odds values between 0 and 1 are associated with probabilities lower than 0.5, and odds values larger than 1 are associated with probabilities higher than 0.5 [14]. Even in the above form of the equation, the coefficients are still difficult to interpret because they are still exponentiated. Thus, we consider the natural log of both sides and arrive at,

$$log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$$

The left hand side expression is called the logit or log-odds, and it is linear in X. Therefore, we can say that for each unit increase in X, the log-odds of Y increase by a factor of β_1 on average. Although this is still not an extremely intuitive interpretation, it does give us a clear idea of the relationship between X and Y. If we exponentiate the coefficients, we can interpret the exponentiated versions as the multiplicative increase in the odds of Y for each unit increase in X, and this is generally a more intuitive interpretation [14]. Specifically,

e^{β_0} represents the mean odds of success for the reference category (for the explanatory variable). e^{β_1} represents the multiplicative increase in the odds of success for each unit increase in X [16].

With regard to prediction, the original form of the logistic function is actually most intuitive because it directly outputs the probability of success for any input value of X. For any new X value of interest, we can find

$$P(Y \hat{=} yes|X) = P(\hat{X}) = \frac{e^{(\hat{\beta_0}+\hat{\beta_1}X)}}{1+e^{(\hat{\beta_0}+\hat{\beta_1}X)}}$$

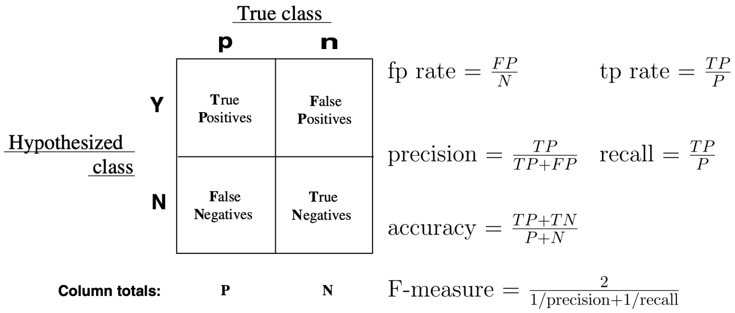
by plugging the new X-value into the right hand side of the equation. This produces a point-estimate for a new x-value. We can also find prediction intervals and confidence intervals for logistic regression, but they will not be discussed here.

3.2.2 AUC and ROC curve

Receiving Operator Characteristics (ROC) curves are a tool for visualizing the performance of classifier models. Before going into ROC curves directly, we will establish some definitions which we will need to understand ROC curves.

A classifier is a type of model which aims to classify data points into categories. There are two main types of classifiers: discrete classifiers and probabilistic classifiers. Discrete classifiers output only a class label, while probabilistic classifiers output a numerical value expressing the degree to which an observation is likely to be in each category (usually a probability). Classifiers can involve classification into more than two categories, but for now we will focus on binary classifiers, which aim to classify observations into one of two categories. Logistic regression is a binary probabilitic classifier which outputs a probability of success. We generally refer to the two categories for a binary classifier as positives (successes) and negatives (not successes). Discrete binary classifiers will directly label observations as positive or negative. For probabilistic classifiers, we use a threshold to determine the final classification based on the probability of success produced by the model. For example, we may classify all observations with success probability greater than or equal to 0.5 as positive and those with success probability less than 0.5 as negative. We will return to this idea of thresholds later [12].

When we evaluate binary classifiers, we are interested in four main metrics: true positive rate, false positive rate, true negative rate, and false negative rate, which can be summarized in a confusion matrix (shown in Figure 2). We are also interested in other metrics derived from these four main ones such as accuracy, precision, and recall, but for ROC curves, we will focus on true positive rate (tp) and false positive rate (fp). The definitions of these basic metrics are summarized in Figure 2 (Figure 1 from [12]).



$\text{fp rate} = \frac{FP}{N}$ $\quad$ $\text{tp rate} = \frac{TP}{P}$

$\text{precision} = \frac{TP}{TP+FP}$ $\quad$ $\text{recall} = \frac{TP}{P}$

$\text{accuracy} = \frac{TP+TN}{P+N}$

$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$

Fig. 1. Confusion matrix and common performance metrics calculated from it.

(15) Figure15

The true positive rate is defined as the number of observations that are true positives (or successes) and also classified as positive (success) by the model divided by all true positive values. The false positive rate is defined as the number of observations identified as positive (success) which were in fact negative (not success) divided by the total number of true negative values. ROC graphs plot the true positive rate on the y-axis versus the false positive rate on the x-axis. The plot of these two metrics represents the trade off between the benefits of our model (true positives) and the costs of the model (false positives). Since both metrics are rates (expressed as probabilities), both the x and y axes are defined on the interval [0,1]. Thus, the ROC space is the unit square [12].

If we have a discrete classifier, it will only produce one point on the ROC graph. When we consider probabilistic classifiers such as logistic regression, we can generate an ROC curve by considering a continuous range of thresholds to use when determining whether to classify a value as a positive or negative. At each threshold, we re-compute the true positive and false positive rates and plot a new point on the curve. There are more efficient ways than this of actually computing the values on the curve, but this is the intuitive idea behind it. Figure 3 demonstrates this idea with an example of what some of the points would look like at different threshold values [12].

When interpreting ROC graphs, it is helpful to have some landmark points for reference. If we consider a discrete classifier, the point (0,0) on the graph would mean that the classifier did not classify any points as positive, so we have 0 false positives but also 0 true positives. The point (1,1) would mean that the classifier categorized all points as positive. The point (0, 1) represents perfect classification of all values. Any point along the line y = x represents random guessing, and any point below this line (lower right triangle of the graph) means that the model is performing worse than random guessing. When we have discrete classifiers represented as single points, it is fairly straight-forward to compare them based on the relative positions of the points. However, when we have probabilistic classifiers represented as curves, it is more complex to compare which one has a better performance. For this reason, we define a metric called AUC explained in the next section [12].

Given multiple ROC curves, we want a concise metric for comparing them. The most common metric used for this purpose is the area under the ROC curve, abbreviated as AUC. AUC values can be anywhere in the interval [0, 1], but generally would not be less than 0.5 because a value less than 0.5 would mean that the model is performing worse than random guessing. A higher AUC value (closer to 1) indicates better performance of the classifier. If we have a randomly chosen positive value and a randomly chosen negative value from the true data, we can interpret the AUC value to be the probability that the model assigns a higher probability of success to the true positive value than to the true negative value. Figure 4 shows a visual representation of how we might use AUC to compare the relative performance of two classifier models [12].

There are many different approaches to comparing classifiers, but the ROC/AUC method has some useful properties which make it a popular method. One of the most important properties is that the values of ROC curves are unaffected by changes in the distribution of positives versus negatives in the data. This aspect is useful in certain domains of research [12].

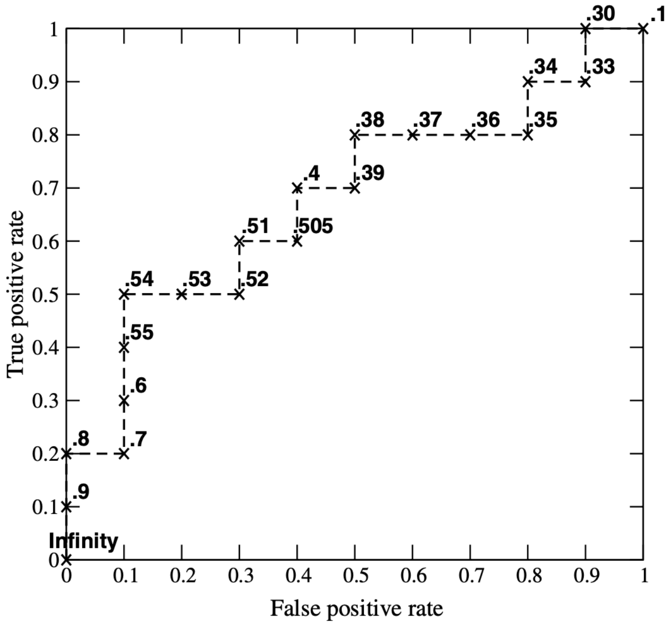


Fig. 3. The ROC "curve" created by thresholding a test set. The table shows 20 data and the score assigned to each by a scoring classifier. The graph shows the corresponding ROC curve with each point labeled by the threshold that produces it.

(16) Figure16

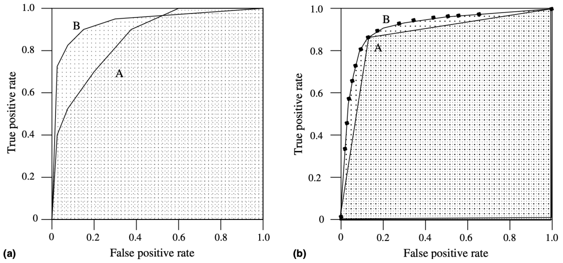
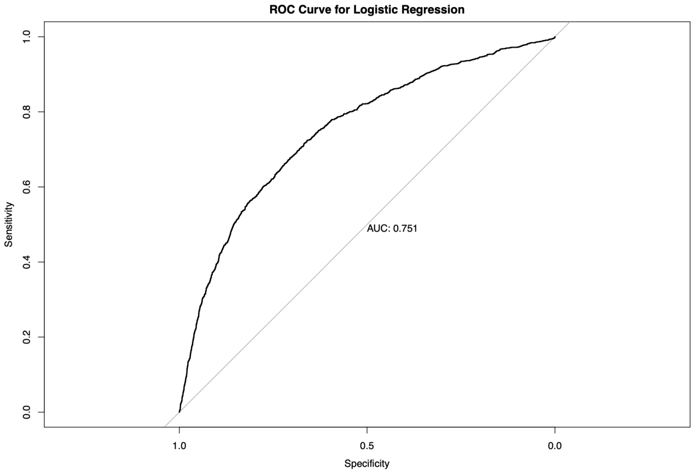


Fig. 8. Two ROC graphs. The graph on the left shows the area under two ROC curves. The graph on the right shows the area under the curves of a discrete classifier (A) and a probabilistic classifier (B).

(17) Figure17

3.2.3 Logistic Regression Results



(18) Figure18

In our analysis of Tampa airport flights using a logistic regression model, we have identified several important findings:

Each day has its own coefficient estimate, representing the change in the log odds of flight delay compared to the reference level (day1). For example, ”day4” has a positive coefficient estimate of approximately 0.3028, indicating that flights scheduled on Thursday are associated with higher log odds of being delayed compared to flights scheduled on Monday.

Each carrier level has its own coefficient estimate, indicating how it influences the log odds of flight delays compared to a reference carrier (carrierAA). For example, ”carrierAS” has a negative coefficient estimate of approximately -1.987, suggesting that flights operated by carrier AS are associated with lower log odds of being delayed compared to carrier AA.

Similar to the other categorical variables, each month level has its own coefficient estimate, indicating how it affects the log odds of flight delays compared to a reference month (month 4). For example, ”month5” has a negative coefficient estimate of approximately -0.2321, suggesting that flights in May are associated with lower log odds of being delayed compared to April.

The coefficient estimate for ”depart” is approximately 0.002. This means that for each unit increase in the depart variable, the log odds of flight delay increase by approximately 0.002.

The coefficient estimate for "duration" is approximately 0.006. This means that for each unit increase in the duration variable, the log odds of flight delay increase by approximately 0.006.

Highly Significant Variables: Among the days of the week, we found that flights scheduled for Thursday (day4), Friday (day5), and Saturday (day6) are highly significant predictors of flight outcomes. These days are associated with distinct patterns that impact flight delays. Certain carriers, including AS, B6, DL, and NK, significantly influence flight outcomes, with some carriers more likely to experience delays than others. The variables depart (departure time) and duration of the flight have a substantial impact on predicting flight delays. The months of May (month5), July (month7), August (month8), and September (month9) are highly significant in understanding flight delays.

Significant Variables:Wednesday (day3) and carrierWN are also significant factors affecting flight delays. June (month6), while not highly significant, still plays a notable role in predicting flight outcomes.

Moderately Significant Variables: Tuesday (day2) and Carrier UA are moderately significant predictors of flight delays. Variables with p-values above 0.05, including carrierF9, carrierMQ, carrierYV, carrierYX, and Sunday (day7), were found not to be statistically significant at the 0.05 significance level. Regarding the performance of our logistic regression model, we assessed it using the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve. An AUC value of 0.75 indicates that our model possesses a meaningful level of discriminatory power.

3.2.4 Findings

Our in-depth examination of flight delays at Tampa airport has unveiled significant factors shaping flight punctuality.

The estimate for depart and duration suggests that a later departure time is associated with a higher likelihood of a flight being delayed and longer flight duration are associated with a higher likelihood of a flight being delayed.

Days of the Week (Thursday, Friday, Saturday): These specific days of the week (Thursday, Friday, and Saturday) exhibit noteworthy patterns in our analysis. They display positive coefficient estimates in our logistic regression model, implying that flights scheduled on these days have a higher likelihood of being delayed compared to flights scheduled on Mondays, which serves as our reference day. This finding underscores the importance of considering the day of the week when planning flight schedules and managing delays.

JetBlue Airways (carrierB6): JetBlue Airways, represented as carrierB6 in our analysis, stands out as a significant contributor to flight delays. It is associated with a positive coefficient estimate, suggesting that flights operated by JetBlue Airways are more likely to experience delays when compared to American Airlines (carrierAA). This insight is valuable for both the airline and passengers as it highlights areas where improvements in scheduling and operations can be made.

Airlines with Negative Coefficient Estimates (carrierAS, carrierDL, carrierNK): On the other hand, Alaska Airlines Inc (carrierAS), Delta Air Lines Inc. (carrierDL), and Spirit Airlines (carrierNK) emerge as airlines with negative coefficient estimates. These estimates indicate that flights operated by these carriers have a lower chance of being delayed compared to flights operated by American Airlines (carrierAA), which serves as our reference carrier. This finding reflects positively on the operational efficiency of these airlines and could serve as a benchmark for other carriers in terms of reducing delays.

Months (May, July, August, September): Our analysis reveals interesting patterns regarding the influence of different months on flight delays. Months such as May, July, August, and September display negative coefficient estimates in our model. These estimates indicate that flights operated during these months are less likely to experience delays when compared to flights operated during April, which is our reference month. This information offers insights into seasonal variations in flight delays and underscores the importance of optimizing operations during peak months to minimize disruptions.

When evaluating the overall performance of our logistic regression model, we observed an Area Under the Curve (AUC) score of 0.75. This AUC score signifies that our model possesses a moderate level of predictive accuracy. These findings collectively provide valuable insights for airlines and airport authorities, offering a foundation for strategic decision-making to effectively manage and mitigate flight delays at Tampa airport.

3.3 StepWise Regression

3.3.1 Stepwise Regression Methodology

One of the biggest challenges when building a logistic regression model is deciding which variables and interactions to include. This is especially difficult when we want to include polynomial terms and high level interactions as the potential number of covariates grows very fast. Hence, many methods have been developed for automatically selecting features.

One of the most common selection methods for feature selection is stepwise regression. Stepwise regression is an automated method for selecting predictors in regression models. It iteratively identifies the optimal subset of variables that most effectively predict the response variable, adding or removing predictors based on their statistical significance [14].

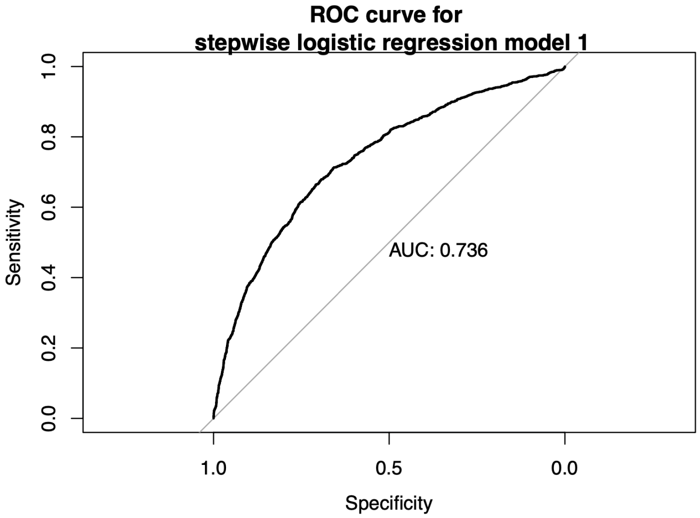
Forward Stepwise Regression: Begin with no variables in the model; Test the addition of each variable using a chosen model fit criterion (often the significance level of the F-statistic for the added variable); Add the variable that gives the most statistically significant improvement of the fit; Repeat this process until adding another variable does not improve the model fit significantly.

backwards Stepwise Regression: Start with all candidate variables in the model; Remove the least significant variable (the one with the highest p-value, for instance); Test the performance of the model without this variable; Repeat this process until no further variables can be removed without a significant loss of fit.

Stepwise Method: This combines forward selection and backward elimination; Variables are added or removed based on their significance, and the process continues iteratively until no variables can be added or removed within the specified criteria. [17]

3.3.2 Stepwise Regression Results

We continue to use logistic regression with a more rigorous model selection algorithm in order to model flight delays at Tampa International Florida. We explore three variations of starting models and use backwards stepwise regression to select a final model. The results are below.



(19) Figure19

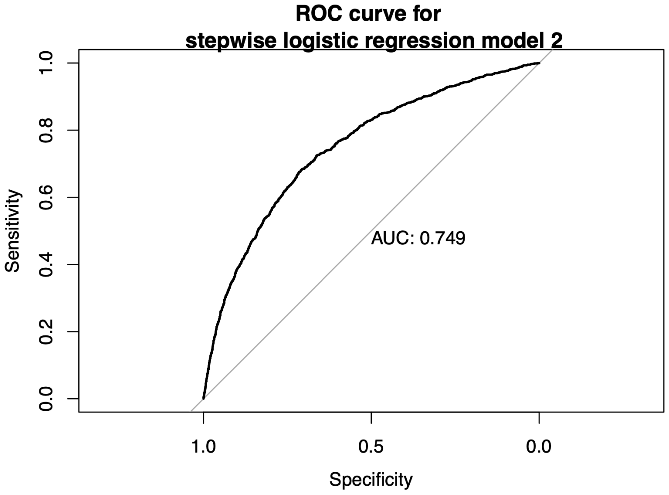
Model1 For the first model we look at day, carrier, depart, duration, and month as main effect as well as pairwise interaction terms between day, depart, duration, and month. Using backwards elimination model selection we get a model with ROC AUC score of 0.736. The resulting model has 77 different main effect and interaction covariates, many of which are categorical factors. Out of those, slightly more than 30 reach statistical significance using

95% CI. In addition to statistical significance we take note of the magnitude of coefficients and the following covariates prove to be the most strongly associated and relevant:

We found strong negative association between delay and specific combinations of month and day of the week (e.g. September Sundays, August Fridays, etc.). Examples of the model terms that were significant include Day7:month9, Day5:month8, Day4:month8, Day2:month8, Day7:month7, Day4:month7, Day3:month7, and others representing day-month interactions.

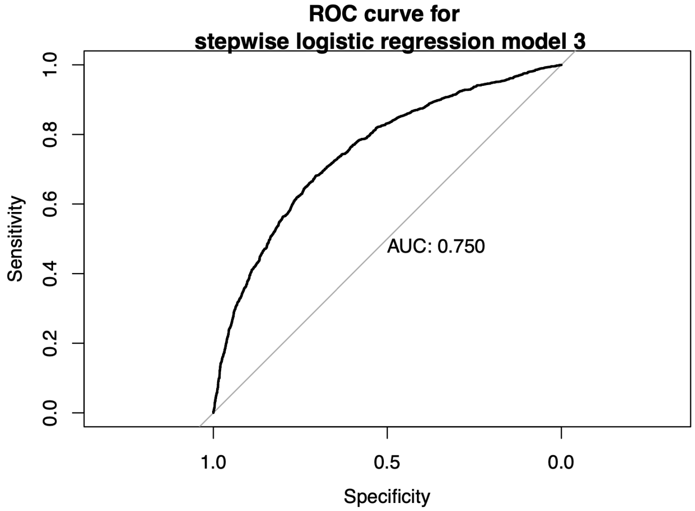
CarrierAS, CarrierDL, CarrierNK have negative associations with delay while CarrierWN and CarrierB6 have positive associations.

Besides, We found a strong positive association between July (Month7) and delays and a strong negative association between Wednesday (Day3) and delays.



(20) Figure20

Model2 Next we consider adding second and third polynomial degrees to test for non linearity and other potential interactions. Our new model does perform better with a new ROC AUC score of 0.749. With this model we get a few additional covariates after model selection with 81 final covariates. In terms of significant covariates we still see many of the same associations as with the first model. We can now see very small but statistically significant 3rd degree polynomial associations with departure and duration so the addition of 3rd degree polynomials was not wasted.



(21) Figure21

Model3 Finally, we consider a much larger model that considers up to 3 way interactions in addition to 3rd degree polynomials. The resulting selected model has 111 covariates with many significant terms. However the resulted ROC AUC remains 0.750. As such we do not recommend using this third model since it adds complexity without improving predictive power. That said, it is noteworthy that some 3-way interactions between specific days, months and departure times were statistically significant. These can later be added to existing models when manually selecting features to see if they improve the model.

3.3.3 Conclusions

Model 2 provides the best performance for the least complexity, hence the 2nd model would be our choice for prediction. That said, we can see that many of the covariates selected by stepwise selection do not reach statistical significance. As such, potentially manually modifying selected features in order to achieve more simplicity may be useful depending on the application of the model - whether it is for prediction, exploration, or testing.

3.4 Shrinkage Methods: Lasso and Ridge

3.4.1 Shrinkage Methodology

Shrinkage methods are a technique used in regression analysis for estimating coefficients in a way that optimally balances goodness of fit to the data (bias) with variability.

Standard methods for coefficient estimation in regression models such as Ordinary Least Squares (OLS) for linear regression or Maximum Likelihood Estimation (MLE) for logistic regression, prioritize fitting the data well, i.e., minimizing bias in the coefficient estimates, but often allow for large variability of the estimates. When we have a large number of predictors (p) relative to the number of data points (n), variance of regression model tends to be large, which is a problem because the model fit could change drastically based on adding just a small set of new training data.

Shrinkage methods are a techique which allows us to control variability of estimates while also ensuring that the estimates are a good fit for the data. In order to balance these two priorities, shrinkage methods do necessarily introduce some bias to coefficient estimates (by underestimating). However, the amount of bias introduced is minimal enough to justify using this method for the purpose of reducing variability in cases where the variability for standard regression estimates is very large.

The shrinkage method technique involves constraining coefficient estimates such that they shrink toward zero. This method is different from variable selection methods because it does not necessarily eliminate variables, and instead just makes coefficient estimates smaller.

3.4.2 Ridge Regression

Ridge regression is one common shrinkage method, which takes both goodness of fit to the data and variance into account by introducing a "shrinkage penalty" term into the estimation process, which shrinks coefficients toward zero.

Ridge regression can be applied to coefficient estimation for any type of generalized linear regression model, but for now we will discuss the process in the context of linear regression in order to clearly convey the basis of how the method works. Ridge Regression coefficients for linear regression β^R are estimated by minimizing the following expression:

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

where RSS refers to residual sum of squares, and is the quantity that we minimize when estimating Ordinary Least Squares (OLS) coefficients for linear regression. It is defined as follows:

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2$$

Looking at the expression used for estimating $\hat{\beta^R}$, we can see that it has two terms added together. The first term, RSS, is exactly the expression which is minimized when using OLS estimation, and this term seeks to fit the data well. The second term, $\lambda \sum_{j=1}^{p} \beta_j^2$ represents the shrinkage penalty, which controls variance by shrinking estimates toward zero. Another way to conceptualize this second term is as a constraint on how large the coefficients can be. In other words, we are aiming to find the coefficients which best fit the data, except the coefficients must be small enough to fit the constraint set by the second term.

The parameter λ is called the tuning parameter (also called regularizaton parameter). This parameter is important because it determines the balance between how influential each of the two terms is. In other words, it dictates how strict or lenient the constraint on the size of the coefficients is. λ can be any real number greater than or equal to zero.

While OLS and MLE estimation will each only produce one unique set of coefficient estimates which is the "best fit", ridge regression will produce different coefficient estimates based on what λ is chosen to be. An intuitive way to understand the effect of λ is to consider what happens to the coefficient estimates when lambda is an extreme value on either end. A $\lambda$of 0 means shrinkage penalty has no effect on estimates, i.e. the estimates are equivalent to OLS. We can see that this is the case based on the expression above because when $\lambda = 0$, the second term goes away entirely, and we are left with the first term, which is simply the RSS. As λ approaches infinity, the estimates shrink closer toward 0. A very large lambda corresponds to roughly the null model, with no regressors. However, it is important to note that in ridge regression, no coefficient estimates will ever shrink all the way to zero. The estimates can come very close to zero, but will not be zero.

The tuning parameter λ can be chosen to be any value greater than or equal to zero by the statistician. When considering how to choose the value for λ, we want to pick a value such that the shrinkage effect is high enough to reduce variability by a meaningful amount while also minimizing the amount of bias introduced. A common method for determining what that optimal value is is cross-validation. The process works as follows. First, we choose a grid of possible λ values. Then, we compute the cross-validation error for each value. Finally, we choose the value with smallest cross-validation error.

When performing ridge regression, there are a few important technicalities to keep in mind. Firstly, we note that the shrinkage penalty is applied to all coefficient estimates except the intercept β_0. We care about shrinking effects of less significant regressors, but we have no reason to want to shrink the intercept.

Secondly, it is important to standardize the predictors before using ridge regression so that they are all on the same scale. The following formula can be used for scaling the predictors:

$$\tilde{x_{ij}} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x_j})^2}}$$

In essence, this formula divides each predictor by its estimated standard deviation such that all standardized predictors have a standard deviation of 1. As compared to OLS, ridge regression performs best when standard regression estimates are highly variable, such as when the number of predictors (p) is large compared to the number of data points (n). One specific case where ridge regression is especially useful is when the number of predictors exceeds the number of data points ($p > n$). In this case, the OLS method is not only highly variable, but in fact is unable to fit a unique model. Despite this limitation of OLS, ridge regression can successfully fit a model which may be reasonably useful even when $p > n$. Furthermore, ridge regression has an extreme computational advantage over subset selection methods such as stepwise. The reason for this is that ridge regression only has to fit 1 model, while subset selection methods work by searching through 2^p models to decide which is best. Thus, subset selection can be extremely computationally resource intensive, while ridge regression has roughly equivalent computational complexity to fitting one OLS model.

3.4.3 Lasso Regression

Lasso is another common shrinkage method which can be used instead of ridge regression depending on the context. The main difference between Lasso and Ridge Regression is that Lasso allows coefficient estimates to shrink all the way to zero, which may drop some regressors from the model. Thus, we can use lasso for variable selection, which is not the case for ridge regression. Like ridge regression, lasso serves to reduce model variance, but it also serves to simplify complex models with many regressors by dropping less significant regressors. This can be a major advantage with respect to interpretation of the model.

The lasso technique is very similar to ridge regression in that it works by introducing a shrinkage penalty (or constraint) when performing coefficient estimation. However, the term representing the shrinkage penalty is defined slightly differently, as outlined in the next section.

Like ridge regression, lasso can be applied for any type of generalized linear regression model. We will discuss the process in the context of linear regression for simplicity. Lasso coefficients for linear regression $\hat{\beta^L}$ are estimated by minimizing the following expression:

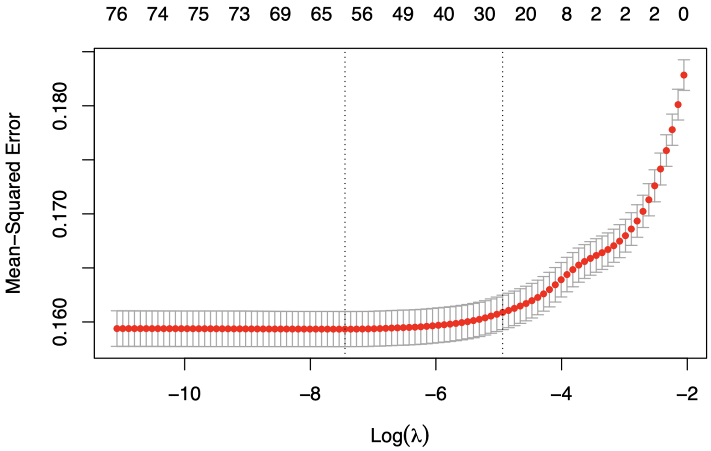
$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

We notice that this is very similar to ridge regression, except we put absolute value signs around the β_j instead of squaring them inside the summation in the second term. This difference has the effect of forcing some coefficient estimates to exactly zero when λ is large enough.

Ridge regression and lasso are similar techniques used for similar purposes, but in some cases one technique is more useful than the other. The choice of which one to use will depend on the context of the analysis. One important consideration is the extent to which the analysis prioritizes interpretability versus prediction accuracy. Lasso can perform subset selection, which helps us to create simpler and more interpretable models. However, if we only care about prediction, it may not be important to eliminate regressors. For prediction, lasso generally performs better when only a few of the many predictors have a significant relationship with the response. In this case, it is helpful to force the coefficients which represent non-significant relationships to zero. On the other hand, ridge regression generally performs better when most of the predictors are associated with the response, and most of the coefficients are similar in size. In this case, it is not as useful to shrink coefficient estimates to zero when there are not many coefficients which are zero in the true model. For real data sets, we would not know in advance how many predictors are significantly related to the response, so we would use cross-validation to determine which method fits the analysis better. Note: all information in this background section including the specific formulas is sourced from [14]. The information is re-paraphrased in my own words based on my own understanding and interpretation of the material.

3.4.4 Shrinkage Method Results

Our first lasso model, Model 01, set out to predict flight delay status based on a combination of predictors: day of the week, carrier, scheduled departure time, flight duration, and month of the year. Additionally, to capture complex relationships, we included pairwise interaction terms between day, departure time, duration, and month.

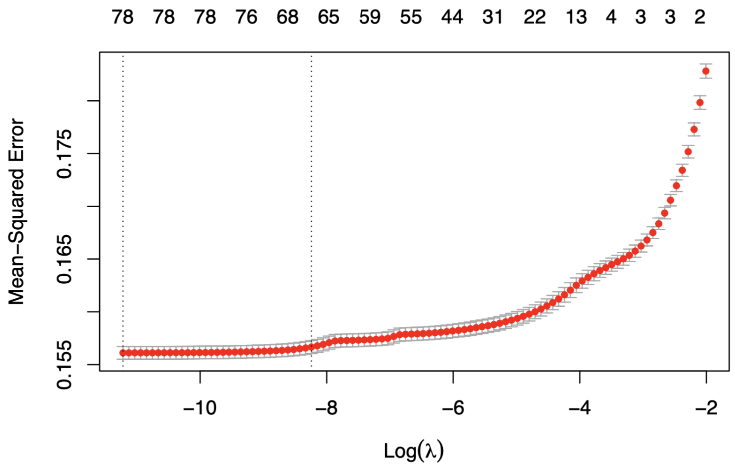


(22) Figure22

Model 01 Results The optimal regularization parameter, λ identified through cross-validation, is approximately 0.0006. The resulting coefficients represent the change in the log odds of flight delays for a one unit change in the predictor, while keeping all other predictors constant.

Examining the coefficients of Model 01, we observe significant effects associated with certain days, months, carriers, and two-way interactions. For instance, flights on day 02 and day 03 show a reduced likelihood of delay compared to day 01, assuming all other factors remain constant. Conversely, flights on day 06 are more likely to be delayed compared to day 01. The interaction term day4:month7 carries a negative coefficient, suggesting that the combined effect of day4 and month7 on flight delays differs from the sum of their individual impacts. Specifically, flights scheduled on Thursdays in July have a decreased probability of delay in comparison to what one might anticipate based solely on the separate effects of day4 and month7.

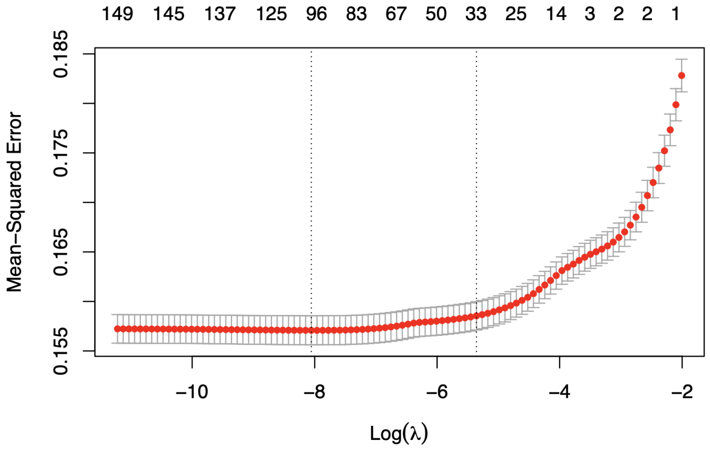
Following the initial Lasso regression model, Model 02 was developed to further investigate the relationship between flight delays and a combination of categorical and continuous predictors. In addition to the basic predictors included in Model 01, this model incorporated second degree and third-degree polynomial terms for 'depart' and 'duration' to capture potential non-linear relationships. The polynomial terms are expected to provide a more nuanced understanding of how changes in departure times and flight durations might influence the likelihood of delays.



(23) Figure23

Model 02 Results The optimal regularization parameter, λ was determined to be approximately 1.342e-05. This value minimizes the cross-validated mean squared error and provides a balance between model fit and complexity. The model coefficients offer insights into the relationship between each predictor and the response variable, flight delay. For instance, the positive coefficient for 'day6' suggests that flights on this particular day of the week are more likely to be delayed compared to day 01. Conversely, negative coefficients, such as for 'day3', indicate a decreased likelihood of delay compared to day 01. The third-degree polynomial terms for 'depart' and 'duration' also yielded coefficients, though their interpretation is more complex. The interactions between variables, such as 'day' and 'month', provide insights into how the effect of one predictor might change depending on the level of another. For example, the negative coefficient for the interaction 'day4:month7' suggests that the combined effect of flying on day 4 and in month 7 on flight delays differs from the sum of their individual impacts. Specifically, flights scheduled on Thursday in July have a reduced likelihood of delay compared to what we might anticipate based solely on the separate effects of Thursday and July. In summary, Model 02 offers a more detailed view of the factors influencing flight delays, capturing both linear and non-linear relationships, as well as interactions between predictors.

Following the development of our second Lasso regression model, Model 03 was constructed to explore deeper into the relationships affecting flight delays. This model embraces the complexity by including three-term interactions among the predictors: day, depart, duration, and month. In addition, it incorporates second-degree polynomial terms for 'depart' and 'duration' to ascertain potential non-linear effects these predictors might have on the likelihood of flight delays. The rationale behind adding these interactions and polynomial terms is to thoroughly explore the intricate relationships that might exist between flight characteristics and the propensity for delays.



(24) Figure24

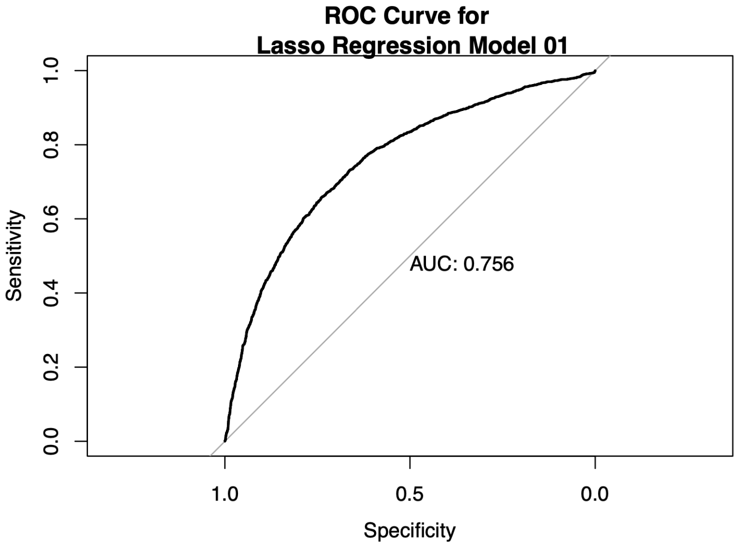
Model 03 Results The optimal regularization parameter, λ was was found to be approximately 0.00032. Interpreting the coefficients for Model 03, we discern that various predictors and interactions play significant roles in determining flight delays. The polynomial terms for 'depart' and 'duration' manifest coefficients which imply non-linear relationships, indicating that the effect of these predictors on delays is multifaceted.

The three-way interactions, such as 'day:depart:duration', provide a more granulated understanding of the relationships. For instance, the coefficient for 'day1:duration:month4' suggests that the effect of flight duration on delays on mondays of the month is different in April compared to other days and months. Model 03 provides a comprehensive overview of the factors influencing flight delays, capturing both linear and non-linear relationships, as well as intricate interactions between predictors.

In predictive modeling, evaluating the performance of a model on out-of-sample data is crucial. It helps in understanding how well the model will generalize to new, unseen data. We have implemented three Lasso regression models to predict flight delays, each with varying complexities and interactions among predictors. To assess the effectiveness of each model, we examined their performance on the test data set. Key metrics for evaluation include the confusion matrix, which provides a breakdown of true positives, false positives, true negatives, and false negatives, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

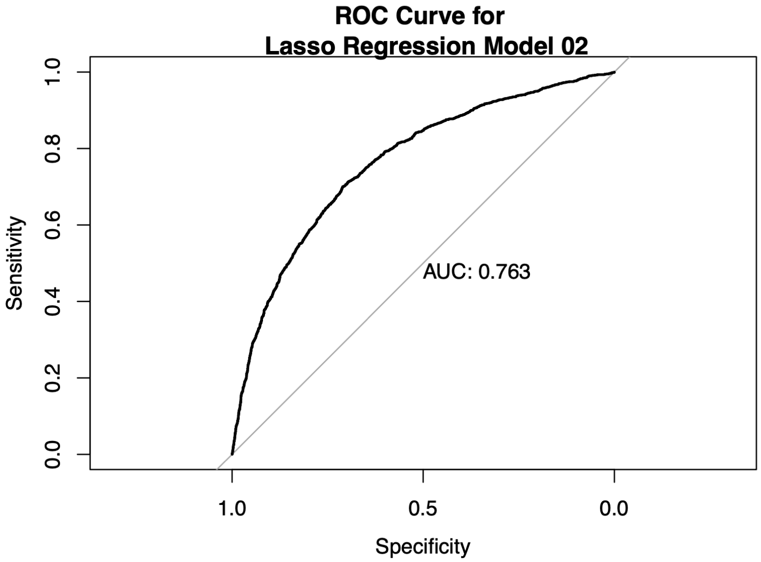
In this scenario, False Positive (Type I Error) happens when the model predicts a flight will be delayed (1), but it actually is not (0). The implications of this error could include passengers arriving too early for their flights or operational adjustments being made unnecessarily, potentially leading to wasted time and resources. False Negative (Type II Error) happens when the model predicts a flight will not be delayed (0), but it actually is (1). The consequences of this error could be more severe. Passengers might miss their flights, leading to dissatisfaction, compensation claims, and logistical challenges. Airlines might also miss opportunities to preemptively manage the delay, leading to cascading operational disruptions.

Confusion matrix and Statistics for Three Models The first model provides an accuracy of approximately 77.75%, with an Area Under the Curve (AUC) of 0.7568. However, there are 1284 instances where the model incorrectly predicts no delay, which is a significant concern.



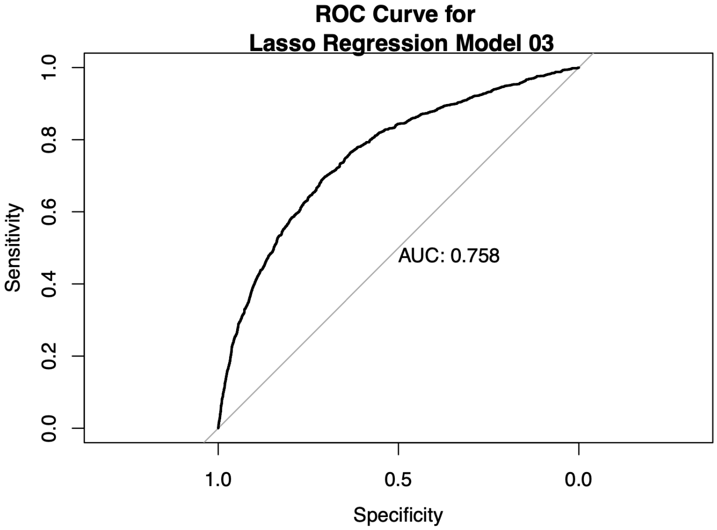
(25) Figure25

The second model shows a slight improvement in accuracy at 78.43%, and an AUC of 0.7634. Notably, the False Negatives are reduced to 1205, indicating a better performance than Model 1 in the aspect we are most concerned about.



(26) Figure26

The third model yields an accuracy of 78.16%, with an AUC approximately similar to the first model at 0.7575. The number of False Negatives is 1225, which places its performance between Model 1 and Model 2 in this critical aspect.



(27) Figure27

While all three models have approximately similar accuracy rates and AUC values, the difference in False Negatives makes Model 2 the most preferable choice for our specific context. Reducing False Negatives is of utmost importance, as these errors can lead to significant disruptions for passengers and operational challenges for the airline. As we move forward, further refinement of Model 2 or exploration of other predictive techniques might yield even better results, especially in reducing False Negatives.

3.5 Simulation

In this study we will have simulated data with exactly 12 covariates, out of which 6 are main effects. Our goal is to simulate two datasets: one which can be fitted very well using a logistic regression, and one which cannot. More specifically, our aim is to get a model with very close to 1 AUC score and one with an AUC of near 0.5. Here, we discuss our approach to achieving these models.

Afterwards, we get data similarly simulated for high and low AUCs and do our best to recover the original covariates. Our goal was to find a set of regressors which does the best job (defined as highest AUC) of predicting the y-values for the given simulation data. We are aiming for an AUC close to 0.98. Once the set of regressors has been identified, we use the best model to predict unknown y-values for a separate set of "testing" simulation data. We approach this goal by using several different variable selection methods, namely backward stepwise regression, lasso, forward stepwise regression, and both ways stepwise regression to reduce the set of possible regressors down to only the most useful ones. Then we compare the resulting AUC between all of the methods in order to decide on a final model based on which method produces an AUC closest to 0.98. We split our data set with known y-values into a training set including 80 percent of the data and a validation set containing 20 percent of the data. We then fit each model using the training set and assess performance using the validation set.

3.5.1 High AUC Model

High AUC: Overall approach: Our initial hypotheses were that simpler models (e.g., lower degree polynomials, few interactions, many significant main effects) are more likely to be well fit. Additionally, lower noise is more likely to be well fit. In this case, we can 'simulate' lower noise by having the coefficients be significantly larger than the variances of the covariates. Finally, we hypothesize that lower correlation between variables will make for a better fit, and more significant variables.

Through experimentation we found that indeed our hypothesized approach worked. The following model was built:

Main effects:

x_1, x_2, x_3 - Three of our main effects are correlated as per the original code. We decided to reduce the correlation significantly to almost independent. The correlation we ended up with after some experimentation was 0.05.

x_4, x_5 - Two of the remaining main effects are normally distributed. We set their means to 0 and reduced the standard deviation significantly to 0.1.

x_6 - The last of the main effects follows a binomial distribution as per the assignment requirements. We decided to keep it as a binary variable (Bernoulli) but set the probability of success to 0.5. This way about half of the trials will be successful. This may be important later when we create interactions between this variable and others.

Interactions and polynomial terms:

We decided to keep the other terms as simple as possible. After some experimentation the following covariates were decided upon. $x_7 : {x_1}^2, x_8 : {x_2}^2, x_9 : {x_5}^2, x_{10} : {x_4}^2, x_{11} : {x_5}^3, x_{12} : x_6 * x_4$

Note that the only interaction we have is x_{12} which is interacted with a binary variable. This is equivalent to setting some of the values to 0 and keeping the rest as is. Additionally, we have a few 2nd degree and one 3rd degree covariates.

Coefficients:

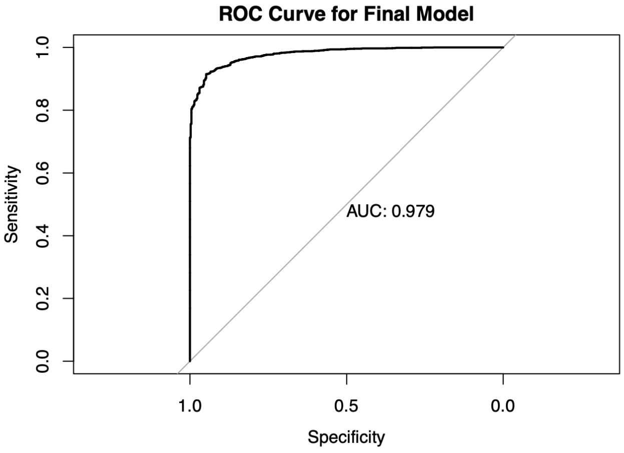
Experimentally, we found that setting larger coefficients makes for a higher AUC score. This is as expected based on the theoretical behavior of the logistic regression as we can expect to get a better fit with lower residuals when in relation to the variance of the predictors the coefficients are high. Some coefficients were then experimentally modified to achieve significance for enough covariates. The coefficients we ended up with are as follows: $b_1 : 80.2, b_2 : -20.1, b_3 : 10.05, b_4 : -60.5, b_5 : 30, b_6 : 50, b_7 : 60, b_8 : -70, b_9 : 90, b_{10} : 40.1, b_{11} : -110.2, b_{12} : 120$

Final Model: Include all regressors which were included in at least 95 percent of Lasso Models, 17 Regressors

In order to determine the final model for prediction, we decided to include all regressors which were included in at least 95 percent of the lasso models. This includes 16 regressors which were included in all 100 models along with one additional regressor which was included in 99 of the 100 models (plus an intercept). We made this choice because choosing only regressors included in most of the models with different seeds minimizes the chance that we are including a regressor which only turned up as useful because of random variation in the train-validate split of the data.

The regressors included in the final model are as follows:

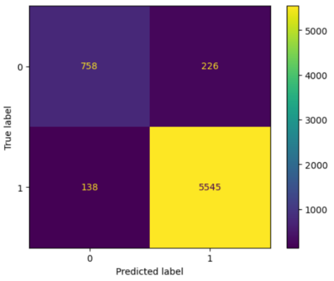
- *all main effects*: $x_1, x_2, x_3, x_4, x_5, x_6$
- *Second Degree Polynomials*: x_2^2, x_3^2
- *Third Degree Polynomials*: x_1^3
- *Two-Way interactions*: $x_1 : x_5, x_2 : x_4, x_3 : x_4, x_4 : x_5$
- *Three-Way interactions*: $x_1 : x_2 : x_3, x_2 : x_3 : x_4, x_2 : x_4 : x_5, x_3 : x_4 : x_5$



(28) Figure28

Based on the AUC analysis, we find that our final model has an AUC of 0.9786603. This is very slightly better than what we had with only one lasso model, so it is difficult to tell if running the model many times and choosing only the most frequent regressors was useful for prediction or not. Nevertheless, we used this model as our final model for generating predictions for the testing data set with missing y values.

Compare with true model:



(29) Figure29

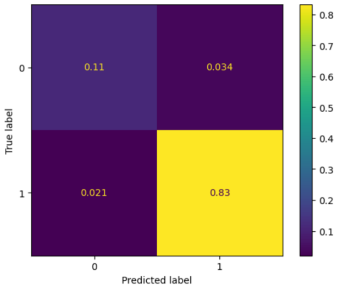


Figure 5: High AUC Normalized Confusion Matrix

(30) Figure30

We have a very good accuracy of 95%; However, we see a relatively low sensitivity (can also be seen by the high number (226) of false negatives; continuing from the previous bullet point, we note that we have 63 more false positives than false negatives; the true model is imbalanced in terms of having significantly more positive responses than negative ones; in these cases we often see this imbalance in sensitivity vs specificity seen above; the balanced accuracy of 87% (much lower than the accuracy) hints at the fact that our model, while close to the truth, is not perfectly correct.

3.5.2 Low AUC Model

Low AUC: The goal was to obtain a lower AUC, ideally close to 0.5. We hypothesized that complex models—those with higher-degree polynomials, numerous interactions, and fewer dominant main effects—might result in deficient outcomes. Concurrently, an increase in noise levels could weaken the model's predictive power. To simulate this increased noise, we shifted the coefficients closer to zero, thereby reducing their relative impact compared to the variances of the covariates. Moreover, we suggest that a heightened correlation among variables could adversely affect the model's fit, leading to less influential predictors.

Variables x_1, x_2, x_3, x_4 and x_5 represent the main effects in our model and are intercorrelated. Specifically, x_5 is normally distributed with a mean of 2 and a standard deviation of 0.1. This configuration adds complexity because the normally distributed term may interact with the response variable differently than the other terms do. The inherent correlation among these predictors can also introduce noise, potentially making the model's estimates less stable and hard to interpret.

For x_6, in line with our previous setup, we assigned it a binomial distribution with a success probability of 0.5. We also explored the effects of imbalanced outcomes, such as p = 0.2, and found that they introduced some noise to the model. Conversely, when the success probability was set to 0.8 (i.e., p = 0.8), it enhanced the model's predictive capacity just a little bit. This suggests that varying success probabilities in a binomial distribution could influence the model's noise level, depending on how well the distribution reflects the true relationship between the predictor and response.

Interaction and Polynomial terms:

To introduce complexity into our model, we incorporated the following interaction and polynomial terms: $x_7 = x_1x_3, x_8 = x_1x_2x_6, x_9 = {x_2}^4, x_{10} = {x_3}^5, x_{11} = {x_4}^2, x_{12} = x_1x_2x_3x_4x_5x_6$

It's noteworthy that x_{12} encompasses a six-way interaction term. Additionally, our model now possesses higher-degree coefficients than Model 1, evident from x_{11} being of the 4th degree. Our intent behind these design choices is to increase the model's complexity since excessively complicating the model can result in overfitting, thereby diminishing its generalizability to the dataset.

Coefficients: $b_1 = 0.081, b_2 = -0.021, b_3 = 0.0105, b_4 = -0.065, b_5 = 0.0075, b_6 = 0.06, b_7 = 0.007, b_8 = 0.009, b_9 = 0.011, b_{10} = -0.014, b_{11} = 0.0087, b_{12} = -0.00136$

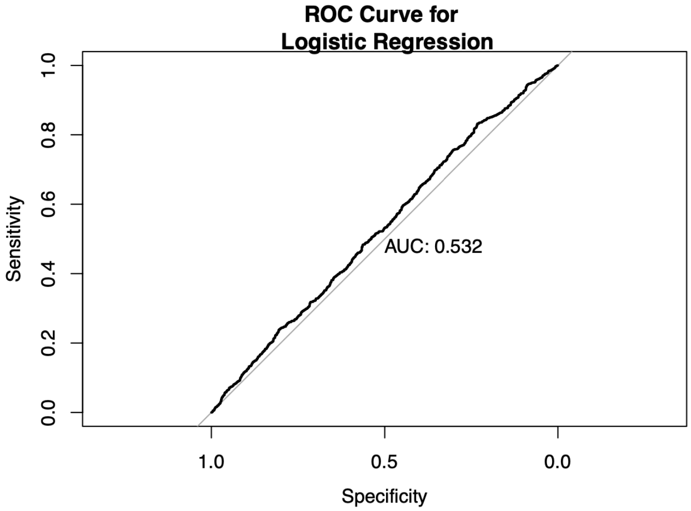
While larger coefficients typically lead to higher AUC scores, we think that by adjusting the coefficients (beta values) of a logistic regression model closer to zero, it will diminish the impact of predictors on the outcome. This approach is likely to yield predictions that remain consistent regardless of the input features.

Recovering the original model:

In this section, we aim to develop a predictive model for the given data set with a low Area Under the Curve (AUC) close to the given low AUC score of 0.53. We will utilize a logistic regression approach, experimenting with various model structures to optimize our predictive performance. This analysis will provide a comprehensive summary of each model we have explored, including their respective AUC scores. We will conclude by presenting our final, optimized model.

For our fifth and final model, we chose to revert to the basics by using a logistic regression model. This model focuses on the main effects as well as the second-degree polynomial terms of x1 to x5. The predictors used were x1 through x6.

- *all main effects*: $x_1, x_2, x_3, x_4, x_5, x_6$
- *Second Degree Polynomials*: $x_1^2, x_2^2, x_3^2, x_4^2, x_5^2$



(31) Figure31

The logistic regression model, with an AUC of 0.532, is the one that most closely matches the desired AUC score of 0.53. Notably, there's a minor distinction between the AUC of the forward stepwise model and that of the binary logistic model. Additionally, the binary logistic model is simpler in terms of the predictors it includes.

Compare with true model:

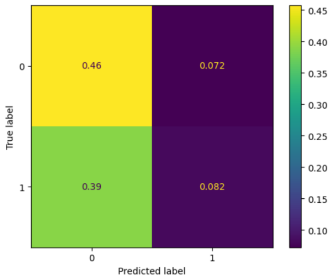


Figure 10: Low AUC Normalized Confusion Matrix

(32) Figure32

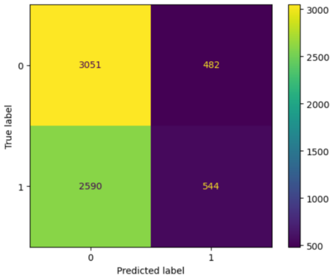


Figure 9: Low AUC Confusion Matrix

(33) Figure33

Our accuracy is very low. At 54% it reflects the low AUC; We have a relatively high sensitivity and a very low specificity; In this case the dataset is relatively balanced, with 3134 positive cases, and 3533 negative cases. However, our model 'prefers' to predict cases as negative (0). This causes the high sensitivity. The balanced accuracy is slightly lower than the accuracy.

3.6 R Package and Shiny App

3.6.1 R Function

We have made an R function that takes as input data about a specific flight leaving from TPA, Florida and returns as an output the probability of the flight being delayed. We believe this function (and package) will serve as a prototype for a package of high value to anyone buying flights or planning itineraries involving flights. Below we describe our approach to making this function and show some implementations of it.

The function needs to have access to the trained model. Since this function is later going to be turned into a package we needed to have the model saved as an Rdata file. The code shows how we trained and saved the data. The code outputs two files: *lasso_model.Rdata*: the trained lasso model; *model_columns.Rdata*: a list of comprehensive interactions and polynomials used in the lasso model. We could read the list of coefficients from the lasso model directly. However, a decision was made to include the interactions for the full model, because this would make it easier to in the future change the model slightly if needed. This way, only the model file needs to be modified but no other code.

The main function - called predict.delay() takes the following inputs:

- month - the month of the flight
- day - the day of the flight
- carrier - the carrier

- depart - departure time
- duration - duration of the flight

The function outputs one number - the probability of the flight getting delayed.

We tested the function by comparing some of its outputs against the outputs of the validation dataset used as part of evaluating the model.

3.6.2 R Package: TPAdelays

Based on our R function presented, we created an R package called *TPAdelays* which serves as a user-friendly interface for predicting the probability of a flight delay based on month, day, carrier, departure time, and duration.

A few modifications have been made to the function in the process of creating the R package which are described below. The full code of the revised function will be included in the appendix at the end of this report, and the full package will be submitted separately with this assignment.

Firstly, the name has been changed to *predict_delay* instead of ‘*predict.delay*‘. The main reason for this is that the period (.) is used for other purposes like file extensions which can cause confusion when it is in the name of a function, while the underscore () is less frequently used for other purposes.

Secondly, the format of the user inputs has been changed slightly to facilitate the experience for the user. Previously, the argument for day of week was expected to be an integer 1-7. In order to make this more intuitive, the input is now expected to be a character string specifying the day of week directly, e.g. ”Thursday”. The other change to the argument format is in the departure time input. Previously, the expected input was a single integer representing departure time in minutes after 12am. This would be some work for the user to compute, so this argument has now been split into two separate arguments: departure hour (in 24 hour time) and departure minute. The first of these arguments expects an integer between 0-23 representing the hour of the departure time, and the second expects an integer between 0-59 representing the minute of the hour for the departure time. Some additional code has been added into the function to convert these more user-friendly inputs into the input formats needed for the prediction model.

Thirdly, several error and warning messages have been added to assist users who do not read the documentation and enter the arguments in an incorrect format or enter a value outside of the expected possibilities. Notably, a warning message has been added which warns users who ener a month outside of April-September that the model was only trained on data for flights in April-September, so the prediction may be less accurate for months outside of this range.

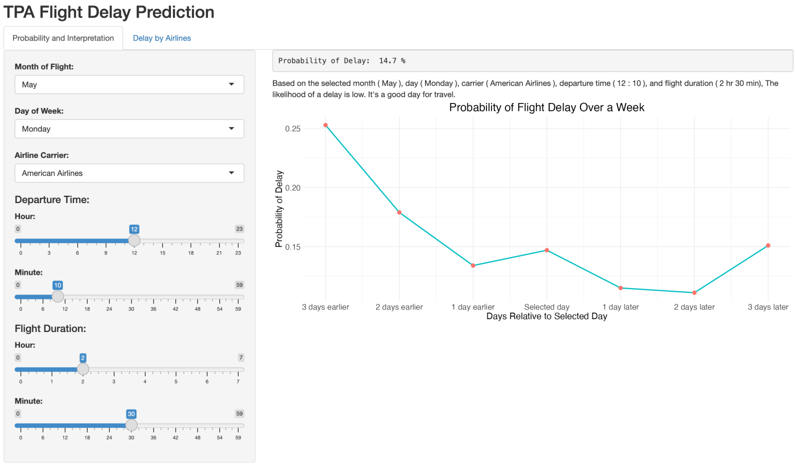
Beyond the function, we have added one other element to the package, which is the full dataset that we used to train the model (the one posted to Moodle at the beginning of the semester). The inclusion of this dataset provides transparency by allowing users to access the original data used to create the prediction model, and it also allows users to explore the data on their own in different ways in order to gain deeper insight into flight delay patterns beyond the predictions that the function provides. The user can access this data set using the command $TPAdelays :: tpa\_data\_2022$.

The aforementioned dataset is the only dataset available to the user from the package, but we also use two internal datasets, *glmmod* and *model_columns* to facilitate fitting the model.

3.6.3 Shiny App: TPA Flight Delay Prediction

Incorporating the functionality of the TPAdelays R package, we have developed a user friendly Shiny app [18] named ”TPA Flight Delay Prediction.” This Shiny app allows users to interactively predict the probability of flight delays departing from Tampa International Airport (TPA) based on various input parameters, including the month of the flight, day of the week, airline carrier, departure time, and flight duration. The primary goal of this app is to empower users to make informed decisions about their travel plans and to gain insights into the likelihood of flight delays [6] [2] [13].

The TPA Flight Delay Prediction Shiny app consists of two main tabs, which are Probability and Interpretation and Delay by Airlines. In Probability and Interpretation tab users can select the month of the flight, day of the week, airline carrier, departure time (hour and minute), and flight duration (hour and minute). The app leverages the *predict_delay* function from the TPAdelays package to calculate and display the probability of a flight delay based on the selected parameters. An interpretation of the prediction is given, offering practical advice to users based on the calculated probability. This interpretation assists users in understanding the significance of the predicted delay probability. Also in this tab, there is a visualization to see the probability of delay three days earlier and three days later. These days are relative to the day the user selected, and the probabilities for each day are again calculated using the *predict_delay* function. The following provides an overview of the ’Probabilities and Interpretation’ tab.



TPA Flight Delay Prediction

Probability and Interpretation | Delay by Airlines

Month of Flight: May

Day of Week: Monday

Airline Carrier: American Airlines

Departure Time:

Hour: 12

Minute: 10

Flight Duration:

Hour: 2

Minute: 30

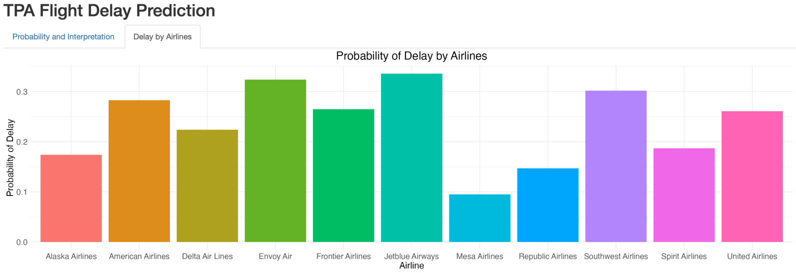
```
Probability of Delay:  14.7 %
```

Based on the selected month (May), day (Monday), carrier (American Airlines), departure time (12 : 10), and flight duration (2 hr 30 min), The likelihood of a delay is low. It's a good day for travel.

(34) Figure34

The interpretations are categorized into three distinct thresholds. When the calculated probability falls below 0.2, users receive a message indicating that the likelihood of a delay is low, making it a favorable day for travel. For probabilities between 0.2 and 0.5, the app advises users that there is a moderate chance of delay and suggests checking for updates closer to the departure time. Finally, when the probability exceeds 0.5, users are alerted to a high likelihood of delay and are advised to plan their travel accordingly.

In Delay by Airlines tab, it provides a visual representation of the probability of delay for different airline carriers, allowing users to compare the performance of airlines in terms of on time departures. A bar chart displays the probabilities, making it easy for users to identify which airlines have higher or lower chances of experiencing delays. The following provides an overview of the 'Delay by Airlines' tab.



TPA Flight Delay Prediction

Probability and Interpretation | Delay by Airlines

(35) Figure35

In summary, the app offers a highly intuitive and interactive interface. Users can effortlessly tailor their flight details through the use of drop down menus and sliders. This shiny app is a valuable tool for both travelers and industry experts. It offers transparency by providing insights into flight delay probabilities, empowers users to make informed decisions, and facilitates comparisons among airline carriers.

Furthermore, it's important to note a limitation of our prediction model. The model was developed using a dataset comprising flights exclusively from April to September (months 4 to 9). When users select months outside of this range, which correspond to months other than 4 to 9 in the calendar, the prediction model may encounter limitations in its performance. In such cases, the accuracy of the predictions may be compromised due to the model's reliance on a more limited dataset, and this warning is given to the user with the interpretation. We advise users to exercise caution and take this limitation into account when interpreting predictions for travel plans scheduled in months beyond the April to September time frame.

4 Conclusions

Throughout the entire project this semester, we gained valuable insights into flight delay trends out of TPA airport through a thorough logistic regression analysis. We also gained insights into the factors that contribute to high versus low predictive accuracy in logistic regression models in general through multiple simulation studies.

Through our initial logistic regression model for flight delays which included only main effects for month, day, departure time, duration, and carrier, we found that later departure time; longer duration; day on Thursday, Friday, or Saturday; carrier JetBlue; and month of April are all associated with the highest chance of delay. Most of these findings make intuitive sense and thus were not extremely surprising. One piece that is surprising is that JetBlue Airways turned up as the carrier with highest chance of delay. JetBlue is generally considered a reliable airline compared to some of the others, so this finding is a bit surprising.

After the basic main-effects analysis, we compared several models by adding interaction and higher order terms and then using variable selection methods including stepwise and lasso to reduce the large number of predictors to a useful subset. The model that we found to ultimately be the most useful was a lasso model including main effects for day of the week, carrier, scheduled departure time, flight duration, and month of the year; pairwise interaction terms between between day, departure time, duration, and month; as well as second and third degree polynomial terms for departure time and duration (model 2 from assignment 5). One surprising finding on this model is that out of 81 predictors included in the full model, none of the coefficients were shrunk all the way to zero by the lasso regularization procedure. Although this model is more difficult to directly interpret since it has so many complex regressors, it is interesting that all of the regressors were useful enough to the model to have non-zero coefficients.

In the first part of the simulation study, we aimed to use simulation to find a model with an AUC as close to 1 as possible and a separate model with AUC as close to 0.5 as possible. We found that a simple model with large true coefficients, low correlation between predictors, low variance of the distributions from which the predictors were drawn yielded very high AUC. On the other side of it, a complex model with many interactions and higher order terms, smaller true coefficients, higher variances for predictors, and large correlations between predictors yielded very low AUC. At first glance, this seems mostly intuitive and unsurprising, but a surprising aspect of this assignment was the vast variation in the approaches used by each group while all achieving very similar target AUC values. There was one group in particular which took almost an opposite approach to us (specifically one thing they did was using very large variances rather than small ones for high AUC) and still got very similar AUC values. This indicates that there is no one straightforward way that is the best to yield a model with a particular AUC, and many different approaches are valid.

In the second part of the simulation study, we aimed to use variable selection techniques to predict the terms included in the underlying true model of simulated data provided by the instructor. Our predictive accuracy was relatively close to what we were aiming for, yet the terms included in the model were quite different from those in the true model, especially for the low AUC model. One surprising finding from this piece was the large variation in which predictors were selected into the high AUC model when running lasso with different seeds. Given that the underlying true model was able to capture the relationship between the predictors and response almost perfectly, it seems that a variable selection method on the resulting data would be able to identify the terms in the model at least somewhat reliably. We found that when running lasso 100 times with different seeds, there were several core regressors which were selected every time, but there were several others which were included in some but not all models in a seemingly random pattern with respect to how many times each was included. This was very interesting to notice how the results of the lasso can be so different just based on a different random seed being used.

To sum up, our study offers valuable insights into the dynamics of flight operations at Tampa Airport. By using advanced statistical models and simulations, we can now better predict flight delays. This not only enhances the passenger experience by providing more accurate information but also helps airlines and airport authorities in managing and reducing delays. While the world of flight delays is complex, our study takes a significant step in making it more understandable and manageable.

References

[1] 2022 north america airport satisfaction study. https://www.jdpower.com/business/press-releases/2022-north-america-airport-satisfaction-study, 2022.

[2] On-time performance marketing carrier flight delays 2022. https://www.transtats.bts.gov, 2022.

[3] Airport administration | tampa international airport. https://www.tampaairport.com/airport-administration, 2023.

[4] Facts statistics financials tampa international airport. https://www.tampaairport.com/facts-statistics-financials, 2023.

[5] J.d. power names TPA #1 large airport in 2022 north american airport satisfaction study. https://news.tampaairport.com, 2023.

[6] On-time performance reporting operating carrier flight delays at a glance. https://www.transtats.bts.gov/homedrillchart.asp, 2023.

[7] Public art collection | tampa international airport. https://www.tampaairport.com/public-art-collection, 2023.

[8] RITA|BTS|transtats. https://www.transtats.bts.gov, 2023.

[9] Tampa international airport expects record 76k passengers a day | wtsp.com. https://www.wtsp.com/article/travel, 2023.

[10] Tampa international airport's phoebe wins CODAWorx award. https://news.tampaairport.com, 2023.

[11] Statista Research Department. Total operating revenue streams of u.s. airlines from 2004 to 2022. https://www.statista.com/statistics, 2023.

[12] Tom Fawcett. Introduction to ROC analysis. 27:861–874, 2006.

[13] Sally French and Meghan Coyle. How to book a flight that (likely) won't get cancelled. 2023.

[14] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer US, 2021.

[15] Staff Reports. Study lists tampa international as no. 3 airport in florida, puts st. pete-clearwater among the worst. https://floridapolitics.com, 2023.

[16] Sandro Sperandei. Understanding logistic regression analysis. pages 12–18, 2014.

[17] S. Weisberg. *Applied Linear Regression.* Wiley Series in Probability and Statistics. Wiley, 2014.

[18] Joe Cheng Winston Chang. *shiny: Web Application Framework for R 1.8.0*, 2023.

5 Contribution

Hasmik Grigoryan: Hasmik wrote the R function and code for the shiny app.

Hanyu Xiao: Hanyu knit the preliminary final report.

Rose Porta: Rose built the framework for the shiny app.

Sandani Kumanayake: Sandani made the slides and designed the Users interface.

All of team members actively participate in everyweek's project, wrote R code and summary about each part.