ORD Flight Delays

Aden Ramirez

March 24, 2023

Abstract

Air travel is a dominant form of travel in the United states with millions of hours a year being wasted to delays. In this paper I investigate some causes for delay of a period of November 2021 to November 2022 at O'Hare International Airport in Chicago Illinois. This airport is a major hub for several American airlines, and proves to be a challenge to minimize this issue. After some exploration of the issue, a couple of machine learning methods are explored to predict air travel delays in both departure and arrival. These methods are compared for predicting delays, and more tight delay groups. The methods are put against each other having different usage that could prove useful in developing further research and models to apply to minimizing waste in the complex system that is domestic United States air travel.

Contents

1 Project Plan

1.1 Data:

Source: Bureau of Transportation Statistics
About: The data being analyzed is sourced from the U.S Bureau of Transportation Statistics, part of the Department of Transportation. This data is collected by the Bureau is sourced from each U.S based airline self-reports each month. This includes each flight details that would be seen at any airport, its operation time on the ground, air, and deviation from scheduled times. This also includes some data beyond the flight itself such as the cause of delay and how long each is contributing to the overall total. In the event of a diversion or cancellation that is also observed and coded with reasons, and its destination. There is a lot of terms used that will need to be well defined in the project to assist in the readers understanding but is well documented by the organization.
Organization Details:

Primary Company Details:
Address:
Bureau of Transportation Statistics (BTS)
1200 New Jersey Avenue, SE
Washington, DC 20590
United States
Company Communication:
Website: https://www.bts.gov/
Phone: 800-853-1351
Alt Phone: 202-366-3282
Business Hours:
8:30am-5:00pm ET, M-F
Key Leaders:
Ms. Patricia S. Hu: Director of the Bureau of Transportation Statistics
Dr. Rolf R. Schmitt: Deputy Director of the Bureau of Transportation Statistics

1.2 Research Motivation:

The U.S air travel system is a convoluted complex system, this past year we have seen the effects of natural disasters effecting holiday travel. Even after we witnessed a meltdown of Southwest airlines that was completely operational. Airlines have huge amounts of data available to them, and the U.S. Department of Transportation collects huge amounts from them. Data

on delays, cancellations and diversions we can look more closely at the fragile system and areas that have the most effect of these stutters. With data like this, airlines can focus changes on specific areas, or be able to plan around common weather conditions in areas of the country.

1.3 Research Questions:

Research Question 1:

Can you predict flight delays?

Research Question 2:

Can you predict flight departure delays, and their effect on an overall delay?

1.4 Hypotheses

Hypothesis 1:

Looking at different categories of flight delays and difference in actual air time vs. actual flight time we can observe the effect of factors that are less directly related to flying the plane, but the business surrounding it. Seeing airline meltdowns outside of a winter storm as this past year, there are more problems than unpredictable weather. I intend to identify, quantify, and predict where these problems can lead to delays in flights.

Hypothesis 2:

When traveling often delays given have a wide range of reasons, delays often start from waiting at your gate for the airplane to arrive or be ready. Obviously leaving later would assume a flight arrival delay, though this is not always the case as many factors go into a planes flight such as wind, route and the speed they fly at. At a certain point a departure delay would become a guaranteed delay, and predicting this could allow for better planning.

2 Literature Review

As the air travel has grown substantially over the past 60 years, the investment in new facilities, aircraft and improved efficiency has grown along with it. Airlines in the United States pour huge amounts of time and effort into reducing delays each year. In the past 10 years we have observed delays hover around 20% in the country, with the exception of 2020. To further understand delays and what exists in the data science realm, I explore some of the research and their discoveries.

To begin identifying some areas of interest is important to further understand the topic and begin the design of a model. "As seen in Figure 1, there are six major themes regarding the flight delay: departure, arrival, propagation, airline, airport, and air system"[1].As they point out all these areas have significant impact on a flight, from facilities to actual flying, focusing on one of these and using the rest as aspects in a prediction will allow for a good model to be developed, but be loosely applicable to the rest. Knowing areas of interest allow me to break down further and apply data science to a smaller area.

The data to be used is a good foundation for this exploration. In multiple articles On the relevance of data science for flight delay research: a systematic review. Transport Reviews, Estimating Flight Departure Delay Distributions-A Statistical Approach With Long-Term Trend and Short-Term Pattern, and Predictive Modelling: Flight Delays and Associated Factors, Hartsfield–Jackson Atlanta International Airport applicable data techniques were applied to sets sourced mainly from two organizations. These organizations being the Federal Aviation Administration (FAA) and Bureau of Transportation Statics (BTS). Both reputable government organizations, the FAA regulating all encompassing air travel, and BTS reporting data around transportation. The set we have chosen has been referenced in multiple studies, pulling from the BTS database of on-time flight statistics.

Applying methods on this data would build towards some next steps of research that provides further progress in the field. "One step in that direction would be to try to extract, from individual airline–airport models, the effects that contribute to NAS-wide delay. Such an approach would provide more insight into the general structure of delays, and also would be easier to maintain and update on a NAS-wide basis"[3]. This set will be used to predict delays in Chicago's O'Hare International Airport (ORD), using methods such as regression, bringing another airport model that could support or refute findings at other airports contributing to a national level of predictions.

Looking at airport specific studies some consideration must be given to

the data with methods used. For example in "Since we faced a problem of unbalanced classes with 86% of flights without delay and 14% of delayed flights and since this problem can lead to a false classification accuracy [20,21] we applied a sampling technique to minimize it"[2]. Classification being used in their studies had to build on top of cleaning to maximize accuracy. Methods being used include regression, classification, and neural networks. I plan to build on top of their work and with working on a specific airport and tailoring data preparation. Data is tailored to each airport or airline, or even airport and airline in these articles and examining their methods they can be tweaked to be useful for ORD.

Research in flight delays is extensive, though much of airline delay prediction is a more closely guarded secret as they use this to build their flight schedules. Looking at different reasons for delays reveals some areas of focus. My research will focus on the airport and using other identifiers for the model, but this allows the narrowing of the dataset, and can contribute to a national prediction adding an airport to extrapolate national identifiers. The data being used is reliable and referenced in multiple research articles, allowing for some questions to be expanded from others. Methods and cleaning must be tailored to the airport, but may be similar to previous research, which can help finding the most prominent variables that effect a flight delay. This research will expand the knowledge base with ORD delay research, introducing more up to date data.

3 Exploratory Data Analysis

Abstract

ORD is one of the biggest and busiest airports in the United States supporting hundreds of thousands of flights a year. The data that will be analyzed November 2021 - November 2022 flights at O'Hare international Airport obtained from the Bureau of Transportation Statistics. This exploratory data analysis will explain te process of cleaning the data and its methods. Then will explore some basic stats on delays, and how the airport itself is used and delayed. Last a couple intuitions of some possible variable relationships will be visualized.

3.1 Important Terms

This contains some terms and information that will be used throughout the EDA that may not be common knowledge to help you understand.

3.1.1 General

Delay - A flight delay is defined by the Federal Aviation Administration (FAA) as a flight arriving 15 minutes or longer after scheduled arrivals time

3.1.2 Governing Bodies

FAA - Federal Aviation Administration is (FAA) is the governing body of all aviation in the U.S from aircraft certification to air traffic control
DOT - Department of Transportation (DOT) is a governing body that regulates all transportation, in this case they offer many unique identifiers and are the parent body to the Bureau of Transportation Statistics (BTS) that the data is sourced from.
IATA - International Air Transport Association (IATA) is an organization that makes a standard for transport type aircraft and in our case create the airport codes referenced throughout this research.
ICAO - International Civil Aviation Organization (ICAO) is a United Nations body that has standards for worldwide air travel. They are less important in this research as we look at domestic travel, they do play in role in how the U.S forms their regulations.

3.1.3 Airline Theory

Hub and Spoke - This is a major theory of how to operate and airline, in short it means the airline has many big airport hubs that fly to other hubs. Each hub will then fly the passenger to their final destination.
Point to Point - This is another rivaling theory to hub and spoke, this means the airline will just fly the airport to the destination with not stop.

3.2 The Data

This data was collected from the Bureau of Transportation Statistics (BLS) as part of the online Airline On-Time Arrival Performance Data. I selected one year of data to begin, choosing the most recent as of January 2023. The data range spans all U.S. Carriers flying domestically over the course of November 2021 - November 2022. The data has many fields totaling to 120

columns. BLS provides an excellent readme that explains each data field, though I will highlight some very important ones to my research here.

Note: _ is not shown in these data types as they appear in the .CSV

3.2.1 Basics

There are a lot of columns that are basic information that would be expected of most sets, there is alot of options in this set with plenty of formating opertunity.

Year - Year (4 digit)
Quarter - Quarter of the year (1-4)
Month - Month
DayofMonth - Day date of the month
DayofWeek - Day in words, such as 'Monday'
FlightDate - Flight Date Aggregation (yyyymmdd)

3.2.2 Indentifiers

There are many identifiers included that are not entirely useful for my use as a DOT Marketing ID, but worth having should there be NA data columns that can be cross referenced. There is also multiple ways to identify *anything* in aviation, so there are international Air Transport Association(IATA) and US identifiers in this set. The most important for my analysis are:

Marketing Airline Network - These are common codes for airlines someone would see on their ticket, such as UA for United Airlines

Tail Number - Aircraft Tail Number, can uniquely identify a register aircraft

Flight Number Operating Airline - Flight number as you would see on a ticket for example: UA1234

3.2.3 Flight Information

3.2.4 Route Information

All the information pertaining to the route of the flight:

Origin - Origin Airport in IATA standard (3 Letter) Example: ORD or PHX

OriginCityName - Origin Airport City (There are special cases such as CLT)

OriginState - Origin State Code

OriginStateFips - Origin State FIPS code, this will allow for adding state geometry in maps

OriginStateName - Origin State Name string

Dest - Destination Airport in IATA standard

DestCityName - Destination Airport City

DestState - Destination State Code

DestStateFips - Destination State FIPS code, this will allow for adding state geometry in maps

DestStateName - Destination State Name string

3.2.5 Delay Information

All the information pertaining to the timing of the flight, and delay if applicable. There is much more, but this is a good focus for the analysis and EDA. Extensive diversion information is not included, as it is noted, but not the focus of the research.

CRSDepTime - CRS Departure Time (local time: hhmm)

DepTime - Actual Departure Time (local time: hhmm)

DepDelay - Difference in minutes between scheduled and actual departure time

DepDelayMinutes - Difference in minutes between scheduled and actual departure time

DepDel15 - Departure Delay Indicator if the flight is at least 15 minutes delayed

DepartureDelayGroups - Departure Delay intervals, every (15 minutes from ≤ 15 to > 180)

DepTimeBlk - CRS Departure Time Block, Hourly Intervals

TaxiOut - Taxi Out Time, in Minutes

WheelsOff - Wheels Off Time (local time: hhmm)

WheelsOn - Wheels On Time (local time: hhmm)

TaxiIn - Taxi In Time, in Minutes

CRSArrTime - CRS Arrival Time (local time: hhmm)

ArrTime - Actual Arrival Time (local time: hhmm)

ArrDelay - Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers.

ArrDelayMinutes - Difference in minutes between scheduled and actual arrival time. Early arrivals set to 0.

ArrDel15 - Arrival Delay Indicator, 15 Minutes or More (1=Yes)

ArrivalDelayGroups - Arrival Delay intervals, every (15 minutes from ≤ 15 to > 180 minutes)

ArrTimeBlk - CRS Arrival Time Block, Hourly Intervals

Cancelled - Cancelled Flight Indicator (1=Yes)

CancellationCode - Specifies The Reason For Cancellation

Diverted - Diverted Flight Indicator (1=Yes)
CRSElapsedTime - CRS Elapsed Time of Flight, in Minutes
ActualElapsedTime - Elapsed Time of Flight, in Minutes
AirTime - Flight Time, in Minutes
Flights - Number of Flights
Distance - Distance between airports (miles)
DistanceGroup - Distance Intervals, every 250 Miles, for Flight Segment
CarrierDelay - Carrier Delay, in Minutes
WeatherDelay - Weather Delay, in Minutes
NASDelay - National Air System Delay, in Minutes
SecurityDelay - Security Delay, in Minutes

3.3 Data Cleaning

3.3.1 Aggregation

Collecting the initial one year (11/2021-11/2022) data from BTS yields twelve separate .zip files containing a .csv file and .html documentation readme. The first step to using this data is to concatenate all this information together into one usable format. I accomplish this with a Python script (10.2.1) using the *Pandas* library. This script pulls in all .csv files into a list of Pandas Dataframes. Once these are Pandas Dataframes it is quite simple with all variables matching, they get concatanated. Now with a single monolithic Dataframe, it is written to a new .csv file. (10.2.1)

3.3.2 Subset Files

This research is focusing on a single airport, Chicago O'Hare International Airport. The data sourced is for all U.S Airline Carriers that fly domestically. The first step in cleaning this data is to properly subset the information into a smaller, less bloated file to reference (10.1.1). My approach will produce to files as output.

First, one main file will use the *Pandas* library in Python to pull in our single monolithic csv file into a Dataframe. From here it is quite simple to mask only the data we want. I filter by either the *Origin* or *Dest* to be the airport of interest ORD. Once this mask is applied, the values are verified to be an expected value, and once again exported using the next masked dataframe to a separate .csv file.

Second, a second file that contains only variables being heavily used, to help with operation speed. All the variables listed in section Figure 3.2 will be maintained. The process will follow the same process as section Figure

3.3.2. The output will be a much more compact file to operate on, to aid speed for the EDA and future model development.

3.3.3 NA Values

Now that the data has been subset into what will be used to explore, the next concern is NA values. There are some columns where this is acceptable, but other such as out delay times that are not. The first step is finding all the columns that have an NA value and then determining what the value should be replaced with. Many of the delay related columns have many NA values, the delay types (Weather, NAS, Security, and Late Aircraft) are null if they are not applicable. These columns get their NA values set to 0, as in no delay minutes. This takes care of the NAs that could corrupt data. The last big concerning item is flights that have no delay value, either arrival delay or arrival delay minutes, which could not be created. (10.2.1) Any NAs of these are dropped. This operation dropped *17004* data points. This is significant, but still far less than 5% of all the data. This now leaves the data with *582721* observations.

3.3.4 Miscellaneous

Outliers in this dataset can get large. I decided to explore with a data subset where all delays are *6 hours (720 min.)* or less (still including non-delays). This decision is not final as a better consideration for what an outlier is should be conducted in this set. This does simplify understanding the data, and making the point through visualization easier for the purpose of this EDA. With this data mask the observations left is now *582045*.

3.4 Exploratory Data Analysis

To begin exploring the data there is some basic information to gather to learn about this data and begin to show some areas that may be worth investigating further with machine learning models.

ORD Amount of Flights: 582045
ORD Amount of Delays: 106187
ORD Percent of Flights Delayed: 18%
ORD All Flights Delay Mean: 3 minutes
ORD All Flights Delay Median: -8 minutes (8 minutes early!)
ORD Delayed Flights Mean: 67 minutes
ORD Delayed Flights Median: 42 minutes

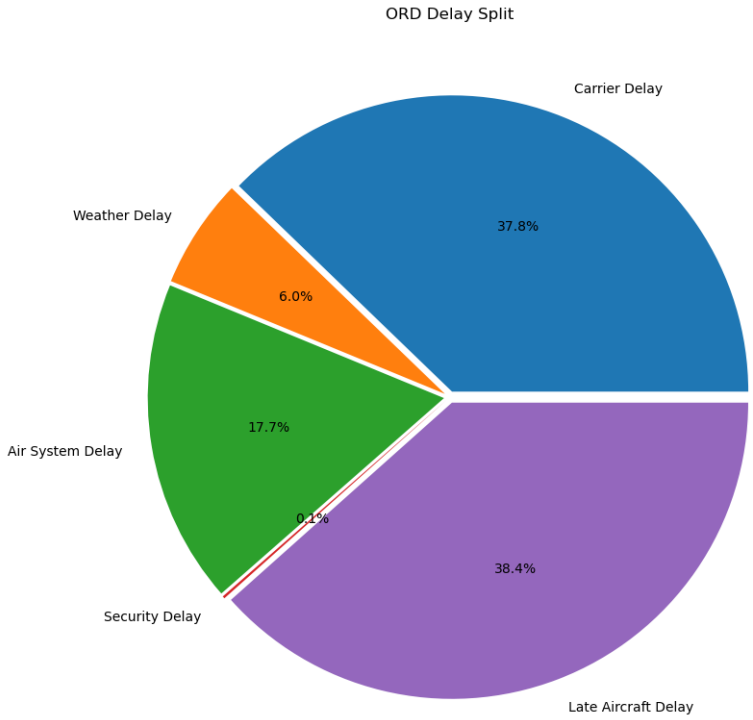

Figure 1: ORD Delays by Type

As seen in Figure 1 the two biggest delays are by carrier and late aircraft. The research questions are going to dive much deeper in how these delays are created and interacted with by airline and airport staff. Prediciting and eliminating these delays could say millions of hours a year.

To begin breaking down the delays that occur, there are five categories officially made by the FAA in this data set tracked. Carrier Delay, Late Aircraft Delay, National Air System Delay, Security Delay, and Weather Delay.

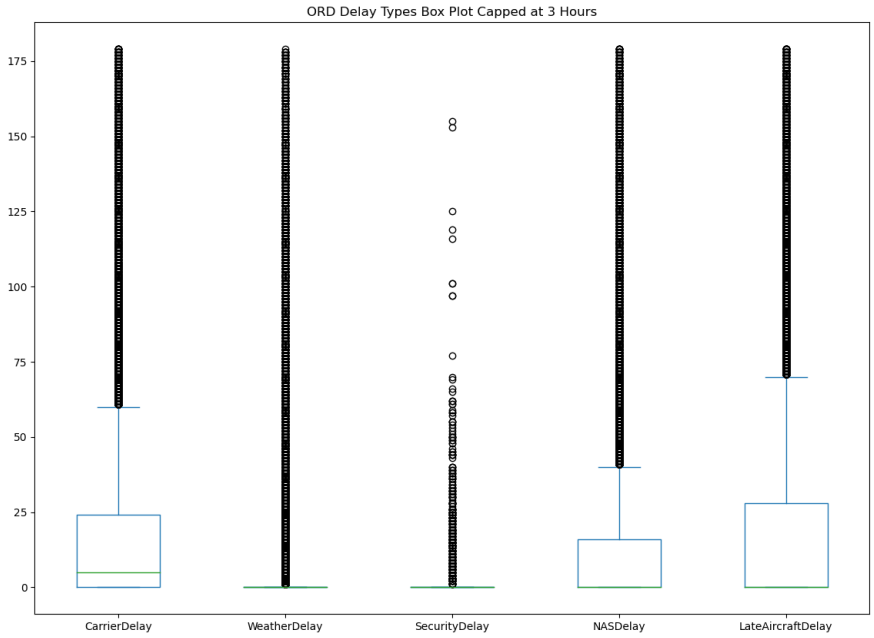

Figure 2:
In Figure 2 This box plot is capped at the highest delay group in the data, 180 minutes or 3 hours. The large amount of flights are delayed for short amounts of time, but you can see the two largest areas cover much higher ranges of delays.

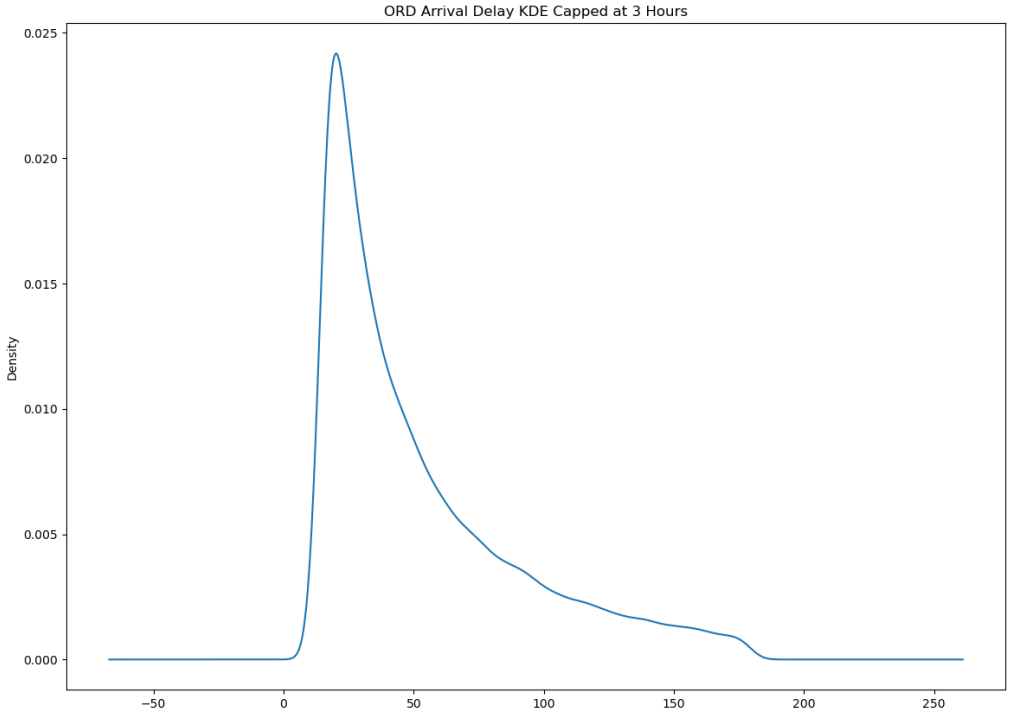

Figure 3:
In Figure 3 looking at arrival delays, we can see the highest density peaks well before the hour mark sharply decreasing to a non zero before the max included amount (180 minutes).

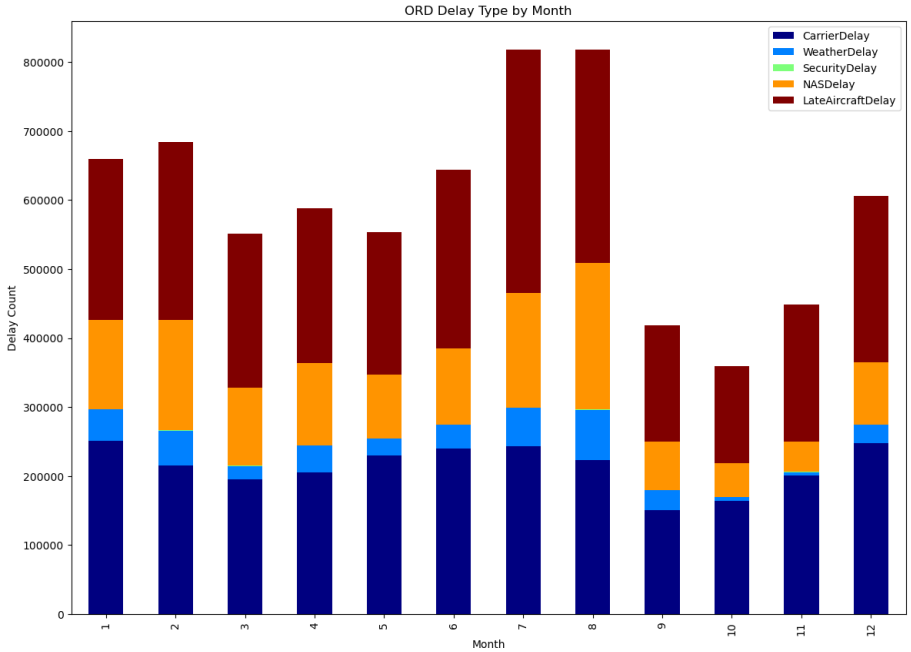

Figure 4:
In Figure 4 this reflects similar to the pie chart, of note is the summer months have a much higher delay count. The busy travel season could be contributing to this, and worth further investigation.

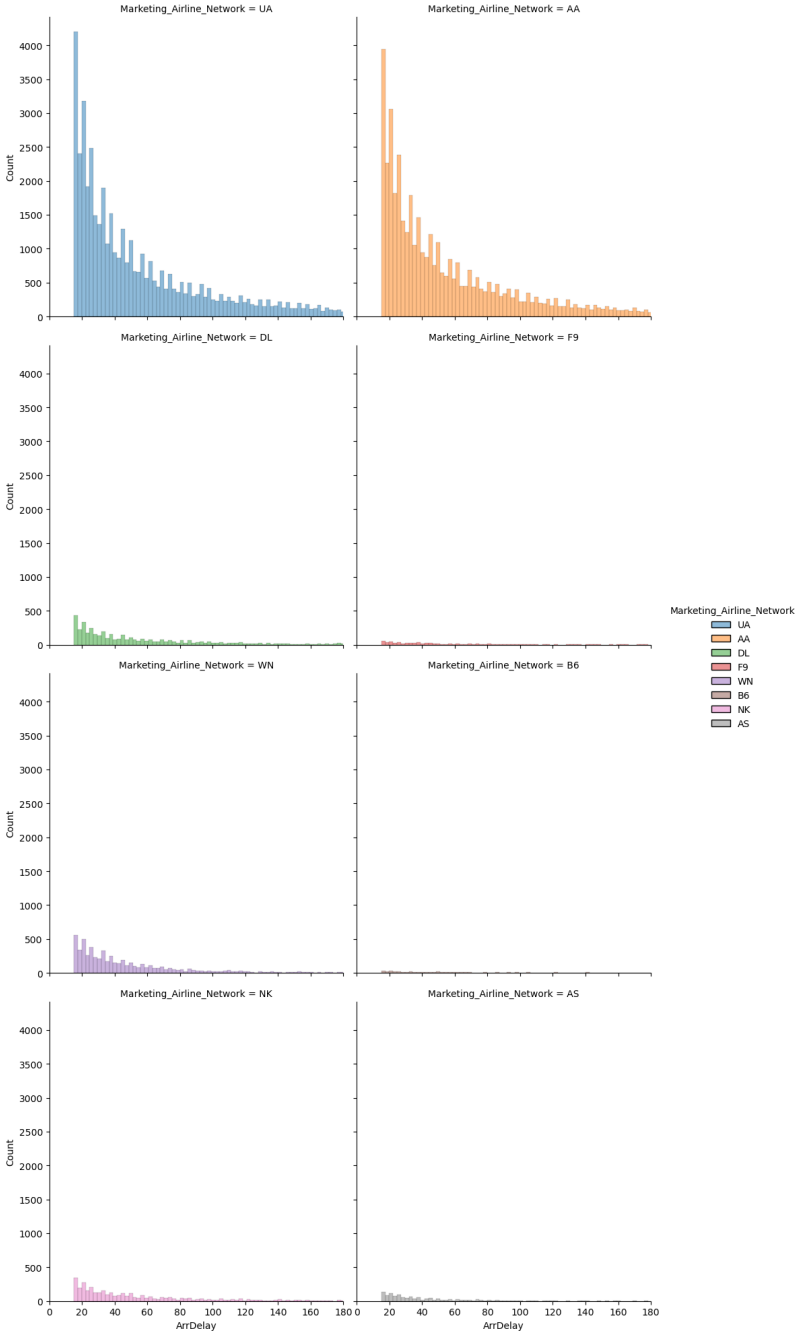

Figure 5:
In Figure 5 this shows the amount of delays distribution by each operating airline. United and American have the largest, which can be expected as ORD is a hub airport for American Airlines, and United Airlines who both fly a hub and spoke model.

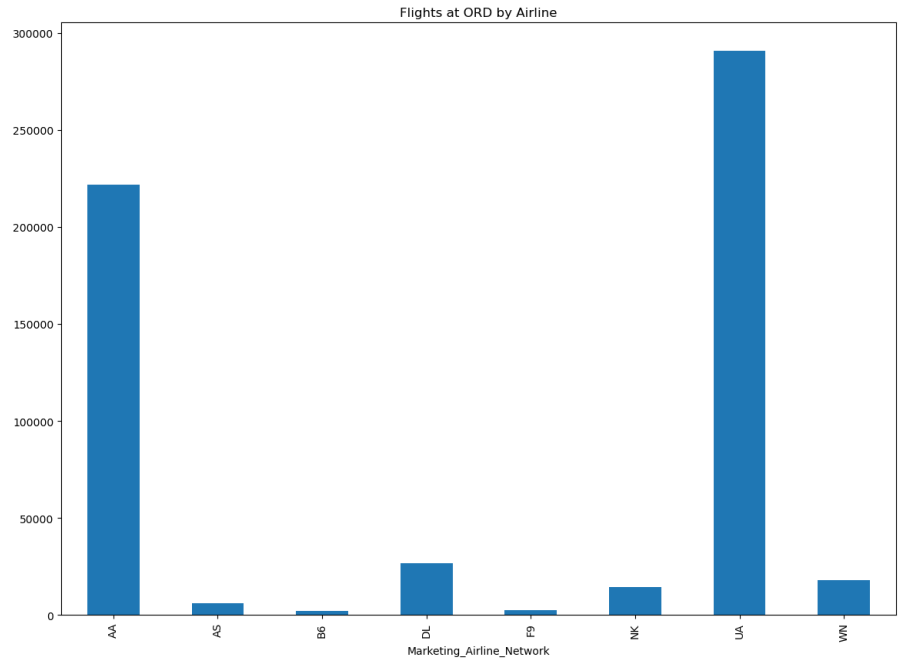

Figure 6:
In Figure 6 To show the previous figure, American and United fly the most to ORD, though there is a significantly larger amount operated by United. In Figure 5 there is not as large a difference in their delay distributions.

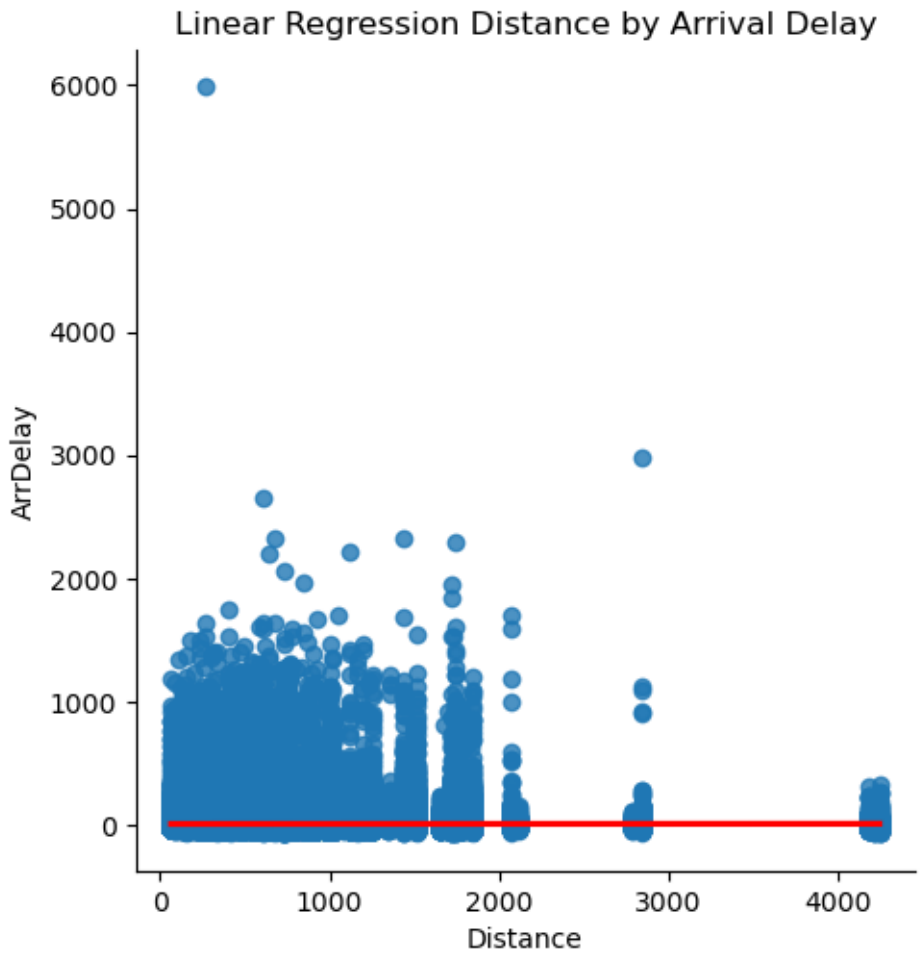

Figure 7:
In Figure 7 One of the relationships I wanted to briefly look at to further think about in the coming day was distance of the flight and the arrival delay. This is a very stange visual I want to look into more, there is almost a recognizable cluster for certain distance airports. There's has no relationship between just these two variables.

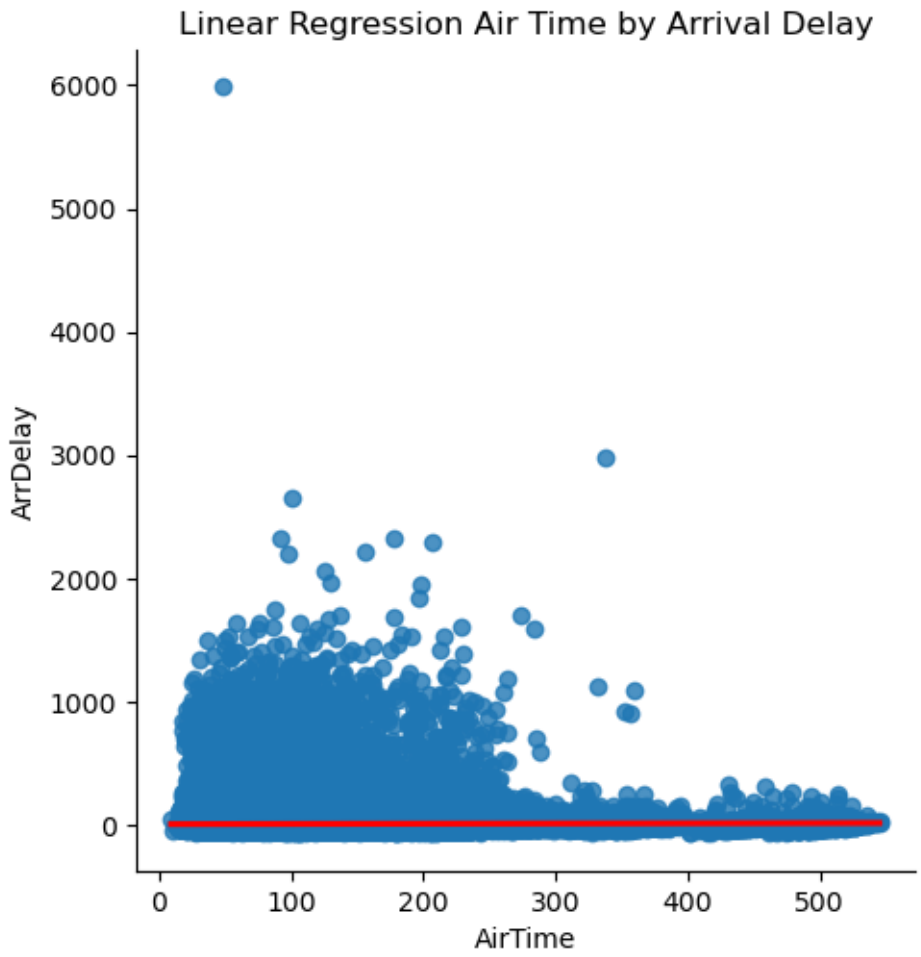

Figure 8:
In Figure 8 The other relationship I wanted to preview was airtime by arrival delay, the thinking being more time in the air laves possibility to make up time flying faster or catching good wind. This very simple linear regression shows not simple relationship, though at a glance looks like more airline has lower delays, but there is obviously a more complex relationship.

3.5 Conclusion

This exploratory data analysis cleaned the data to be more usable, identified some areas the data could be improved, and explored some basic relationships. Moving forward to modelling some this EDA will prove valuable to understanding the data, and helping work towards the data being easily workable for machine learning models.

3.5.1 Data Improvement

During the EDA there is some spots of the data that can be improved and expanded. For future visualizations it would be helpful to have a nicer way of showing information such as airline names or even a map showing where flights go. There are supporting tables provided by BTS that would assist in expanding tha data. Another consideration is how to deal with outliers when building models there is a large range, that shouldn't eliminate all high delays. This should be solved mathematically and researched what the FAA considers long delays, or a point of where delay is considered a new flight. Working with the data could also be made easier with some table work, on the names, and adding new information that is used more than once.

3.5.2 Relationships and Methods

In the EDA only a couple areas were explored, and both showed the problem is much more complex than a linear regression. Thinking ahead I think the possibility of a logistic regression is there that could provide a good predictor or a delay. Another area I did not consider earlier is trees that could prove promising. There may be the need for some dimension reduction as these complex relationships may start adding uo in variable size. Lastly clustering may still be a good option where there are splits on more focused data.

4 Methodology

4.1 Research Question 1

Can flight delays be predicted?

For this question we want a good overall look at stages of a flight, taxi, departure, landing, taxi, and arrival. The goal is to categorize if a flight will be delayed, this is useful for airlines and people for expectations and planning around it. However there should also be an attempt at predicting the length of the delay to make the information more meaningful. This question will be solved with three different methods.

4.1.1 Method 1

The first method for this question that will be applied is logistic regression. This method will have two possible categorical outputs, being the flight is delayed or the flight is not delayed (Using >15m FAA definition). Logistic regression is great for two class problems, and will fit well here.

4.1.2 Method 2

The second method for this research question is also a logistic regression, but using more categorical outputs. These being our delay groups that are used in the dataset. They are not delayed, 15 minutes - 30 minutes, and builds all 15 minute increments up to 180 minutes for a total of 15 possible groups. The attempt in this method is to see with the data if logistic regression can be expanded as a good predictor for more specific flight delay information.

4.1.3 Method 3

The third method for this research question is a random forest decision tree, but using the group of 15. These being our delay groups that are used in the dataset. They are not delayed -15 minutes , 15 minutes - 30 minutes, and builds all 15 minute increments up to 180 minutes for a total of 15 possible groups. The power of the decision tress will allow to go beyond saying a flight is delayed or not, and offer a range following the ranges the DOT provides.

4.2 Research Question 2

Can flight departure delays be predicted?

4.2.1 Method 1

The first method for this question that will be applied is logistic regression. This method will have two possible categorical outputs, being the airport is the main delay or other.

4.2.2 Method 2

The second method being applied for this research question is a random forest decision tree. This method will output multiple categories corresponding to the type fo delay possible.

4.3 Research Question 3

Can flight delays be prevented through operational changes?

From the EDA flight delays obviously have a bias towards late aircraft and airlines operations such as preparing the plane after the previous flight. Minimizing delays means more people flying and more money for the airline, looking at how these operations play into a predicition can allow understanding for the effect working towards minimization could have on the industry.

This research question is supported by the machine learning methods of the previous questions, and will be explored through model training results and data validation.

4.4 Validation

With the methods being used, there will need to be many validation methods used. For logistic regression there will need to be simple classification measures such as accuracy and precision that can be determined testing the model on a subset of the data. With this info sensitivity and specificity can also be calculated, and all reported in a table before the next method. An ROC curve will also be used as a nice visual for these values calculated. For decision trees there will be variable importance measures and predictions verified on a subset of the data. Both will start with 75% of the data being used for training and the last 25% to be used in validation.

4.5 Data

For the models in this research question we will provide:

Airline
Origin
Departure Block

Departure Delay
Destination
Arrival Block
Arrival Delay (Time in minutes)
Arrival Delay (True or False)
Distance
Carrier Delay
Late Aircraft Delay
Air System Delay
Security Delay
Weather Delay
Taxi Time

5 Ethical Recommendations

There are some ethical considerations for the tooling and analysis developed in this project looking closer at O'Hare International Airport flight delays. Predicting flight delays is hugely beneficial to the industry, allowing for carriers to plan ahead and work towards eliminating preventable delays. With this information though there are so problems that could arise such as delaying a flight that needs to be cancelled, prolonging the issues for the customer, or over adjusting for predicted delays by moving the flight time dramatically to attempt to avoid a delay.

Airlines are widely known to have a low margin for profits, cutting costs, and maximizing money from flights. Carriers want to avoid any extra costs possible especially having to accommodate a passenger who's flight has been cancelled. Predicting flight delays, and getting the precision down to a block, even if the block is large could allow for an airline to hold out cancelling a flight in the name of cost savings, since they believe they can see it would be a long delay, or even just extend the time, being more inconvenient for the passengers. This would not necessarily be industry breaking, but would leave an opportunity for airlines to skirt costs and create situations that would not be in the favor of the customer.

Another consideration that is closely related, predicting flight delays, especially when looking at block departure or arrival times historically airlines would likely adjust their schedules each year to account for this. This is common, and isn't bad in itself, flights are large airports often have more padding with a higher average taxi time. This does leave the ability to "over-adjust" when seeing a flight it likely to be delayed where they could add a later arrival time not only to pad from a long taxi, but an anticipated delay. This over adjustment could lead to longer flight (total not in air) times that could have an effect throughout the system, but not actually having a delayed flight since with new pad, it is technically on time or even early if the adjustment becomes too extreme.

Carriers reputation is important to their business with a low margin, highly competitive industry, they will take every chance to increase their reputation or cut cost. With this analysis and implementation this information could be used to assist and airline with that, and of course could effect the future of airline schedules where total flight time is increased more than it should be in order to padding time to avoid delays. The information must be use responsibly and not adding unnecessary time to flights, that would cause these problems.

6 Analysis

6.1 Research Question 1

Research question one set out to predict flight delays at O'Hare International Airport. There were two different model types used and four total models, each with a set of challenges and varying success. The first model used was a two class logistic regression (10.3.1) that aimed to determining wether a flight was either delayed or not. This model did not require much extra preprocessing beyond the basic model level cleaned data The data was subset (10.3.1) to include some columns including the time blocks of flight and the departure delay. Building the first logistic regression for this question brought up an issue that will continue to be encountered for many other models. Since delays are a minority, the dataset has a heavily skewed imbalance toward on-time or early flights, following closely to the national numbers. Addressing the imbalance found here adds class weights inverse of the proportion found in the dataset. In the EDA section (10.2.2) these proportions were used as weights, with class 0 (no delay) having a weight of 15 and class 1 (delay) having a weight of 85. With only two classes these weights work well as simple hyperparameters and greatly increased AUC and accuracy values. Other adjustments included random state, which is set to the class 0 weight, and a high max iteration with the amount of data to allow the lbfgs solver converge. Overall the model worked very well for predicting a flight delay once the plane has begun its trip, with an accuracy of 89%. This is demonstrated with a confusion matrix as the first evaluation (9). In the ROC curve a great area under curve (AUC) value of 0.93 (10) was observed. The recall vs. precision curve (11) was slightly less promising, but still having a high enough value to be somewhat reliable. Of note with this model, with more data before flight perhaps with airline operations departure delays could have a lighter effect on the model's accuracy. This model would work well in determining a flight delay on a flight tracker or something of the same.

The second model was also a logistic regression model, but with 15 separate classes, one for each departure block (-15 min to 180 minutes in 15 minute increments). This model was much less successful, showing an accuracy of 37% and a more diverse confusion matrix (12). Logistic regression as it stands is not the best approach for this many classes in the attempts made here. There could be improvement with further preprocessing and accounting for the imbalanced data.

The third and fourth model for research question one was based on a random forest decision tree. These models included a new type of preprocessing from the previous models, synthetic minority oversampling technique

(SMOTE) which provides a similar desire to the first logistic regression, that oversamples the classes to bring the imbalance to a more equal level. With this another random state was added to the classifier similar to the logistic regression. Before SMOTE was applied , the random forest classifier was seeing numbers similar to the multi class logistic regression of around 30%, now currently having an accuracy of 60% with multi-class and 36% with 2 class, a large improvement. The confusion matrix (13) shows there is still spread, but predicting within one or two classes meaning the error for an everyday user is not much longer than the minimum for a flight to be considered delay.

6.2 Research Question 2

Research question two, predicting flight departure delays became of interest during research question one when analyzing random forest. In predicting flight delays (6.1) when predicting with departure delay information it provided the most important variable. Wanting to be able to use this information as early as possible, predicting a departure delay could assist with using in further modeling. Similar to the arrival flight delays there is four models two logistic regression and two random forest decision trees. The data used is similar to 6.1 using a 2-class classification for predicting a delay or not, and multi-class following the DOT groups from 15 minutes early up to 180 minutes late. Without a strong predictor like departure delays are in predicting arrival delays, there is much more work to getting a good accuracy without more data that airlines may have access to. A predictor that was dropped that had a massive effect on the models was taxi out time. This was in the objective of observing if departure delays could be predicted further out in advance., with this value accuracy could theoretically be higher, but would be used mostly in the event of predicting a delay while the flight is already left.

Both logistic regression models are using SMOTE to resample the imbalanced data of delays, which with departure delays having a distribution of 75% of on-time departed flights. Predicting the two class departure delays the regression model uses balanced class weights, fit intercept with an l2 penalty. With the high need of preprocessing with weaker predictors I created a grid search for some values in the regression models (10.3.2). The outcome of this showed balanced class weights with fit intercept and l2 penalty was showing the highest accuracy score. This model proved to be okay the accuracy was 57% (21). The AUC values was 60% (18). This model was far less successful than logistic regression for arrival delays. Multi-class would proved to fall even further with an accuracy of 21%. This was improved by adding another parameter in the multi-class regression model, C. C allows for adjusting the

penalty, setting this to 0.15 showed better results which is a very strong penalty. Logistic regression here, shows there may be a better model method for these predictions.

As above preparation for the random forest models is very similar to the previous trained models for arrival delays. (6.1) Two class prediction was decently successful with an accuracy of 77%. This is much higher than the logistic regression, but also had a much longer runtime. Multi-class got a lower accuracy 54% but still passable over the logistic regression. In both of these models importance features were dominated by surprisingly day of the week and scheduled airtime. With this information this model could feed into an overall prediction or expanded to give a numeric value to allow more accurate overall delays from further time out and not day of or almost in flight.

7 Challenges

This topic is notoriously difficult, and airlines spend millions of dollars each year trying to maximize their on time performance and reputation. There are several challenges I encountered in this research including understanding the industry data, preparing the data, and deciding models.

Understanding the data on face value was simple, during EDA there were few issues as the data allowed for multiple ways to express similar subjects such as a delay and delay group that sectioned off into 15 minute segments that were used extensively. The challenge became harder when working on models, as the initial list od data subsets was not sufficient and wildly inaccurate in all models. Adding in factors such as the time did not come until understanding that time could show an uptake in congestion with could be a large effect. The hardest hurdle was an ethical one, the data is laid out in a way where unless the flight is delayed 15 minutes or more, there is not specific delay information since there isn't technically a delay. This realization made it harder to justify using late aircraft information or any delay information at all in predicting departure delays, since it is so connected to a delay the model or a human could figure out the link very easily.

Data preparation was an extremely time consuming subject, along with tuning models. This is likely not the best way to do it already. I had to investigate multiple ways to prepare data that is imbalanced since inherently knowing the national values, about 85% or more of flights are on-time. Even with this implementation, SMOTE in higher amount delays effected the end models as it was only able to access a small sample size, which proved difficult in accuracy of delay group models. A significant amount of time was used attempting grid search on many models, which automated some hyperparameter tuning, but still could be improved.

Lastly, deciding the models to use was difficult, though early on decided with this amount of data to stick to categorical outputs that would already be hard to get right. Numeric predictions would need much more data, which with computing limitations and other variables such as the early 2020 travel dips due to COVID-19 would prove to be too much to bear in the scope of this project.

8 Recommendations

This project had a limited scope in what is possible furthering the research. As mentioned in 2 many other forms of research have been conducted on this topic including specific airports such as Hartsfield Jackson International Airport in Atlanta Georgia. Further steps would be to make reliable models for more hub airports that could accurately predict per airport and contribute to a national or international delay prediction profile. This could be utilized by all regulatory bodies such as the FAA or DOT, and airlines alike to improve existing models or fill a gap some of these entities may have in this subject area. Cutting waste with machine learning could prove to save millions if not more each year in both hours and dollars.

9 Appendix

9.1 Figures

9.1.1 Research Question 1

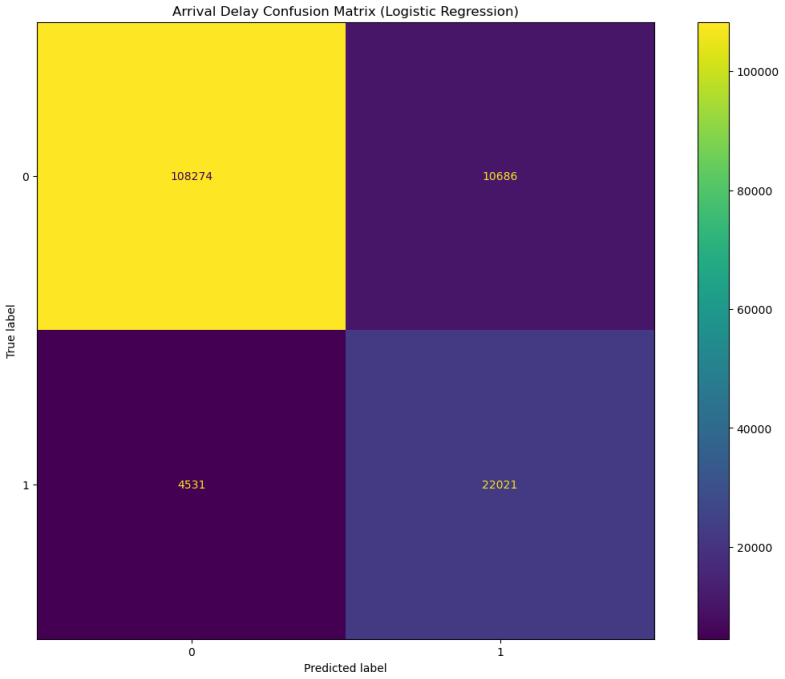

Figure 9: Two Class Logistic Regression Confusion Matrix

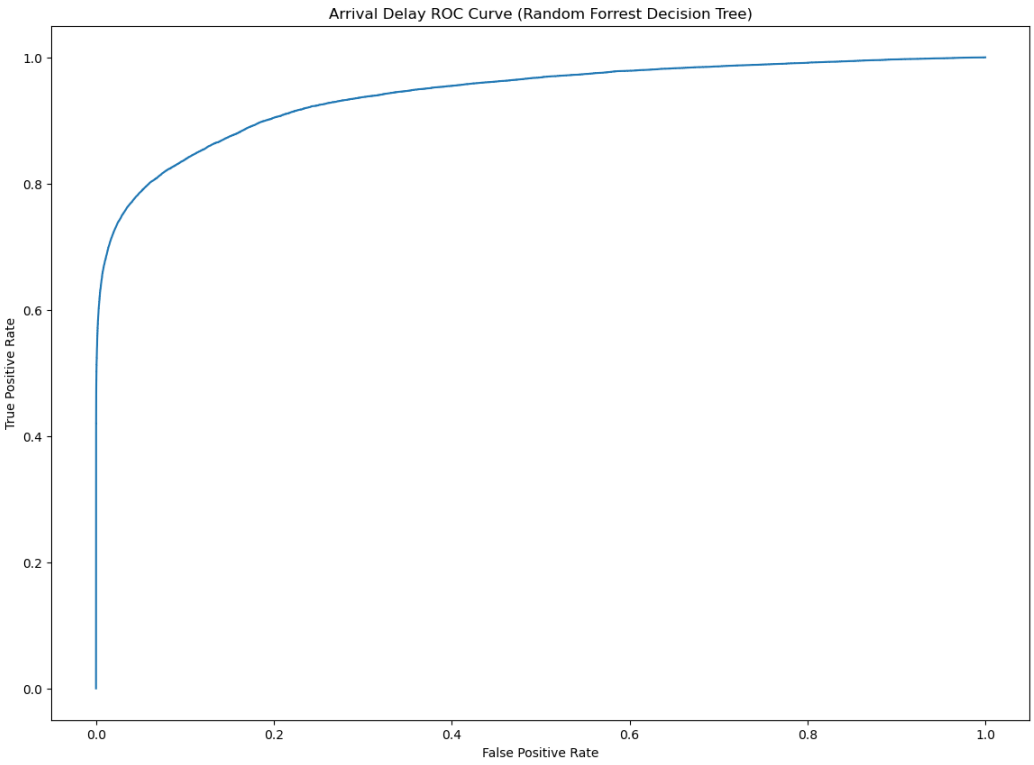

Figure 10: Two Class Logistic Regression ROC Curve

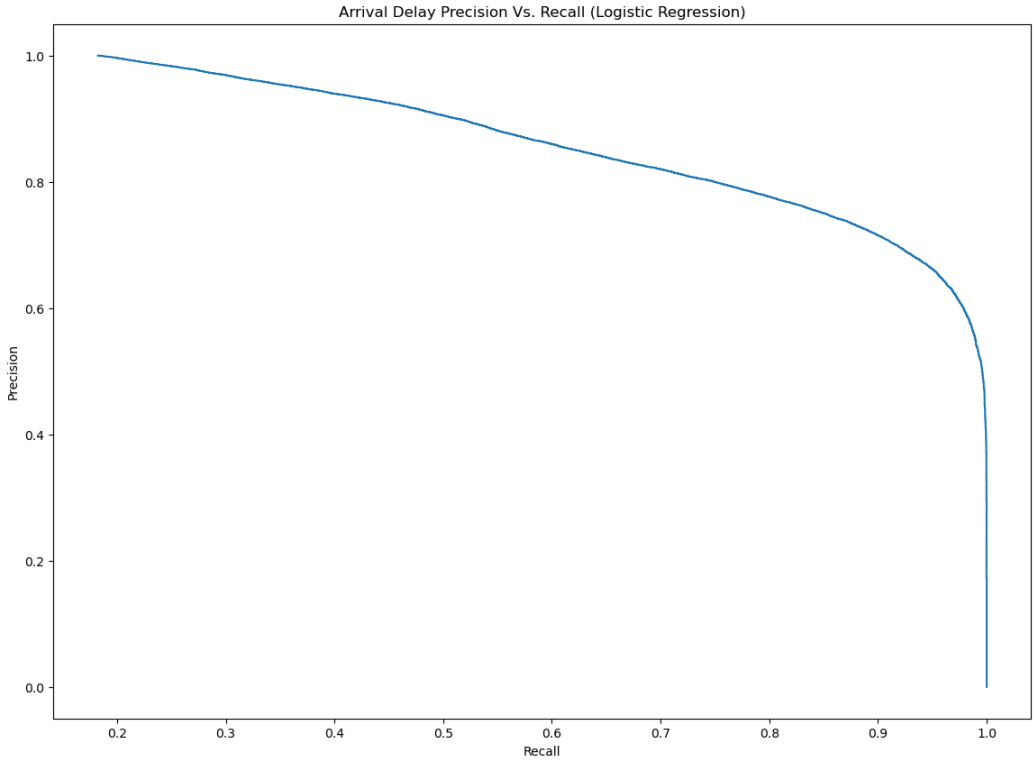

Figure 11: Two Class Logistic Regression Precision Vs. Recall

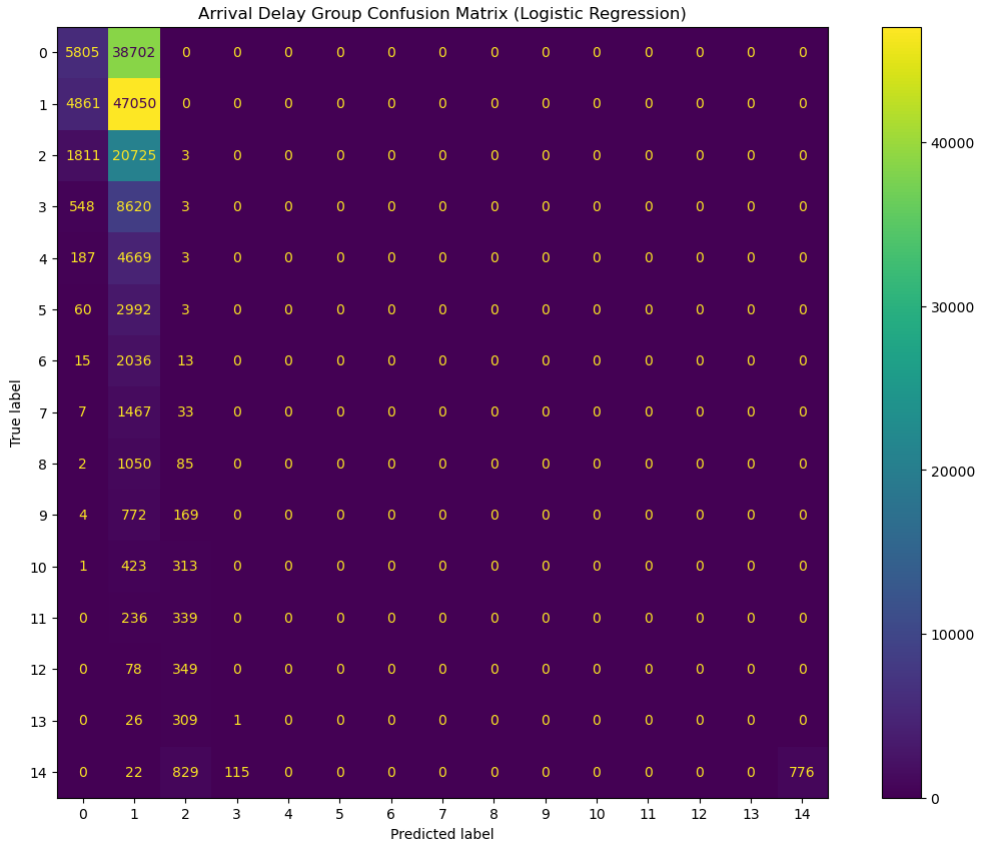

Figure 12: Multi-Class Logistic Regression Confusion Matrix

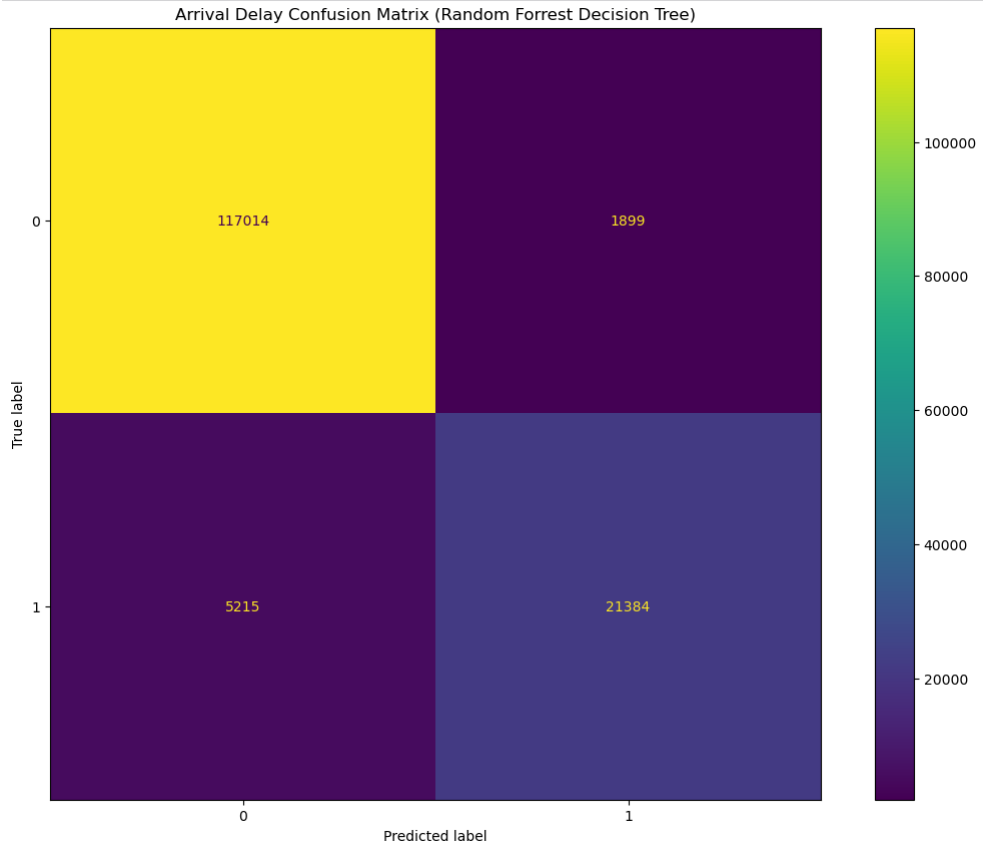

Figure 13: Multi-Class Random Forest Decision Tree Confusion Matrix

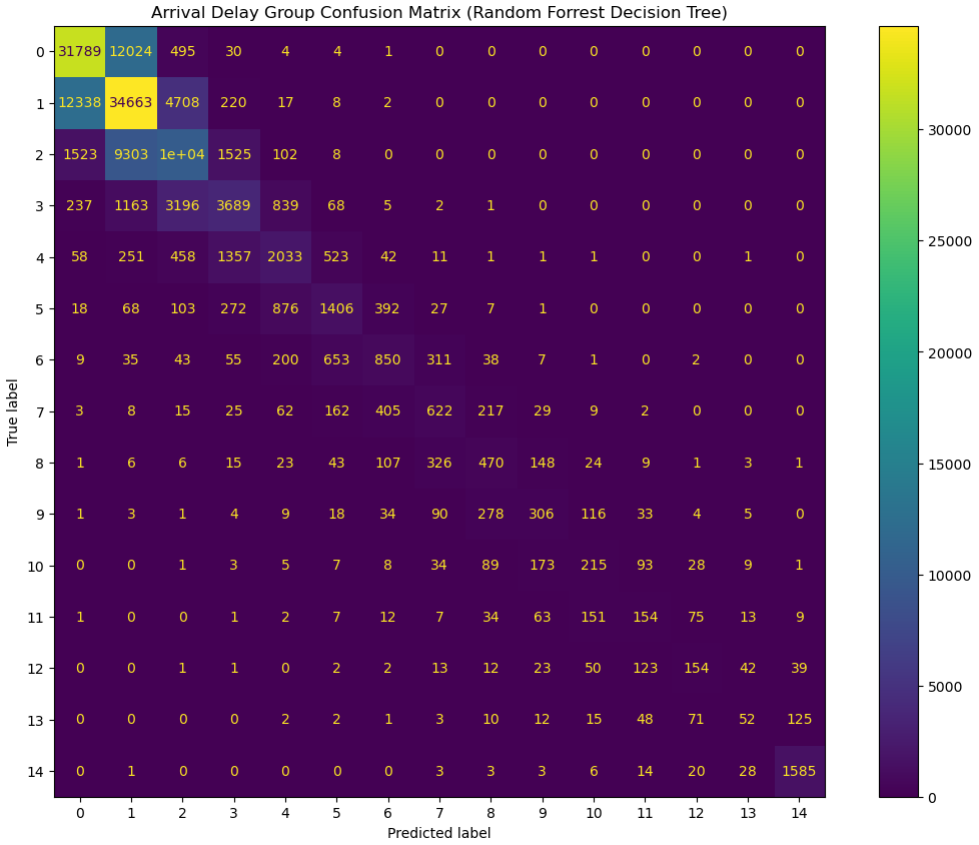

Figure 14: Multi-Class Random Forest Decision Tree Confusion Matrix

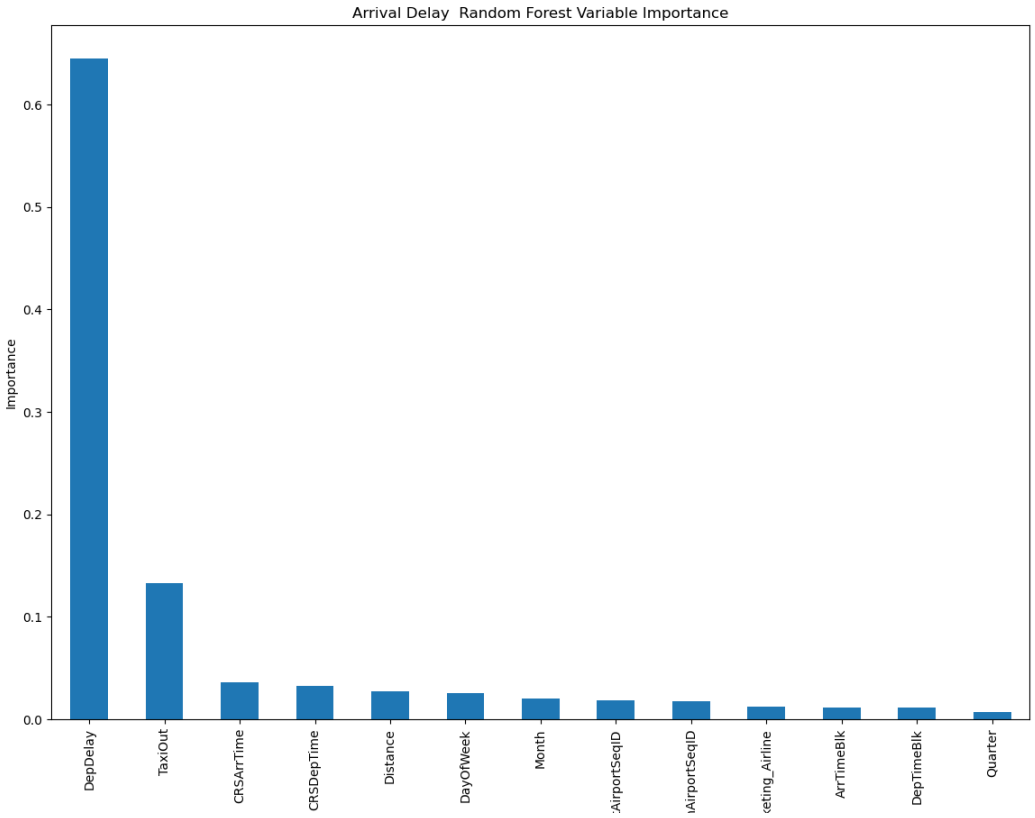

Figure 15: Multi-Class Random Forest Decision Tree Variable Importance

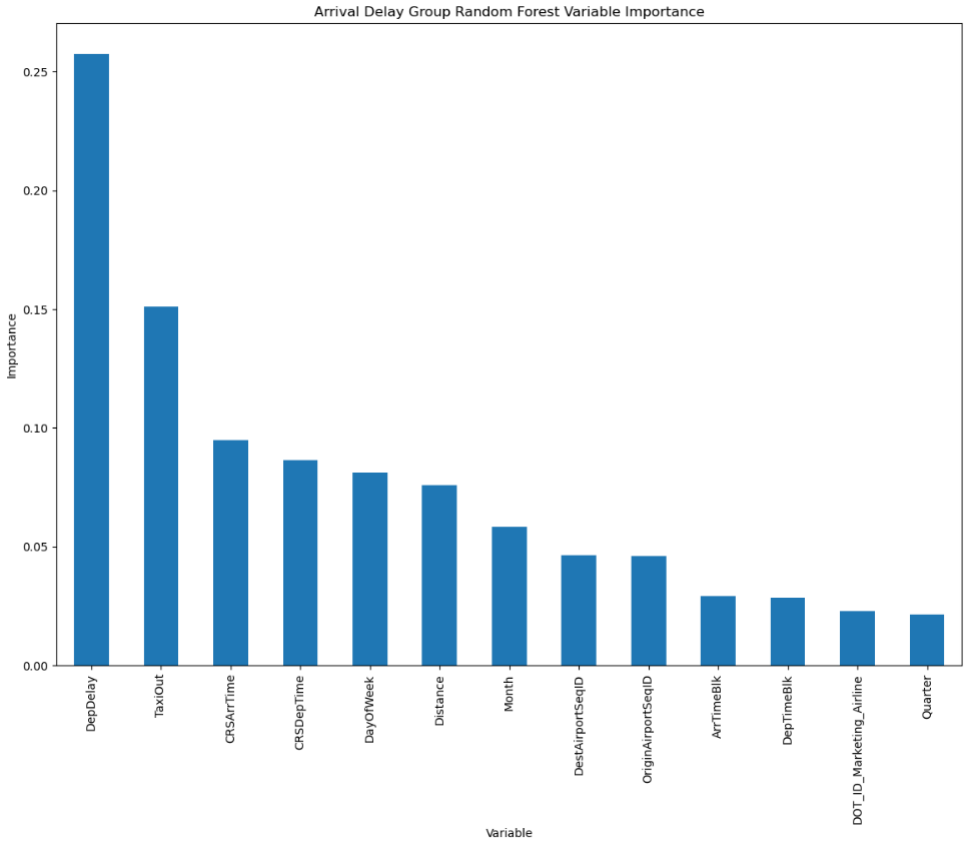

Figure 16: Multi-Class Random Forest Decision Tree Variable Importance

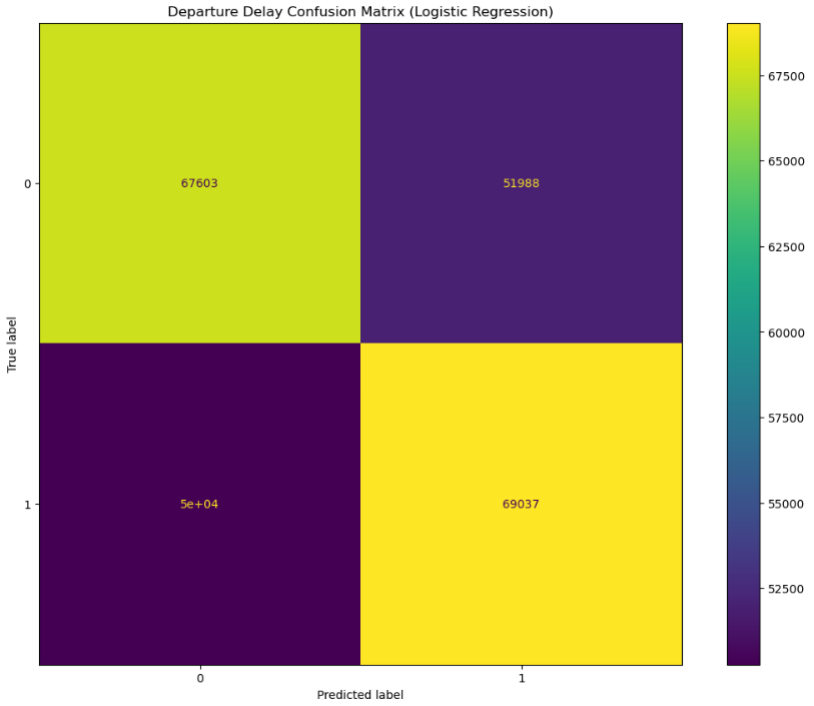

Figure 17: Two Class Logistic Regression Confusion Matrix

9.1.2 Research Question 2

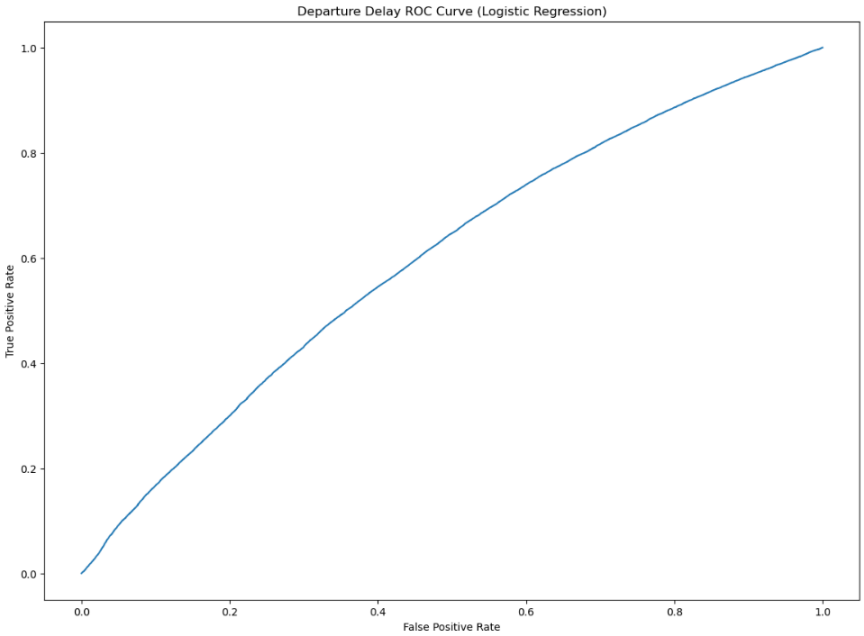

Figure 18: Two Class Logistic Regression ROC Curve

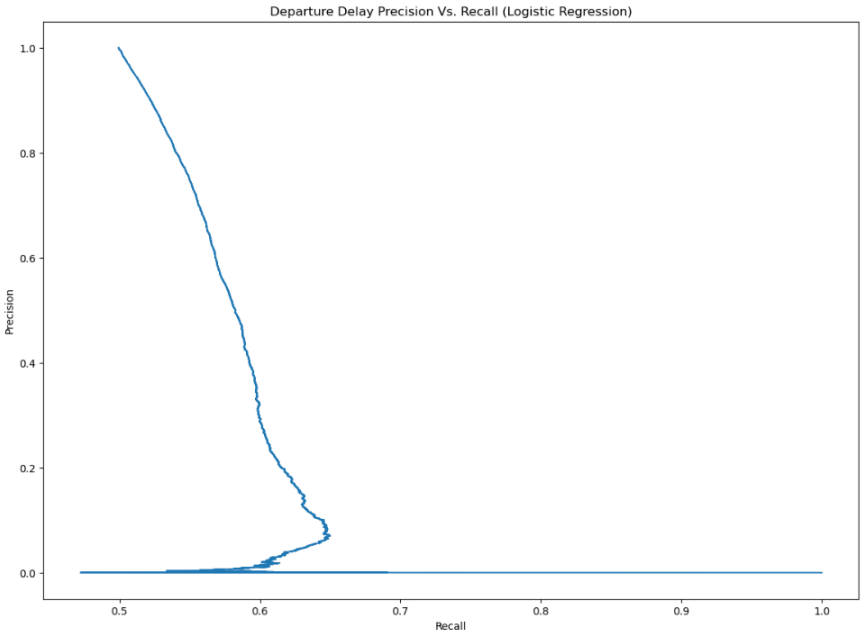

Figure 19: Two Class Logistic Regression Precision Vs. Recall

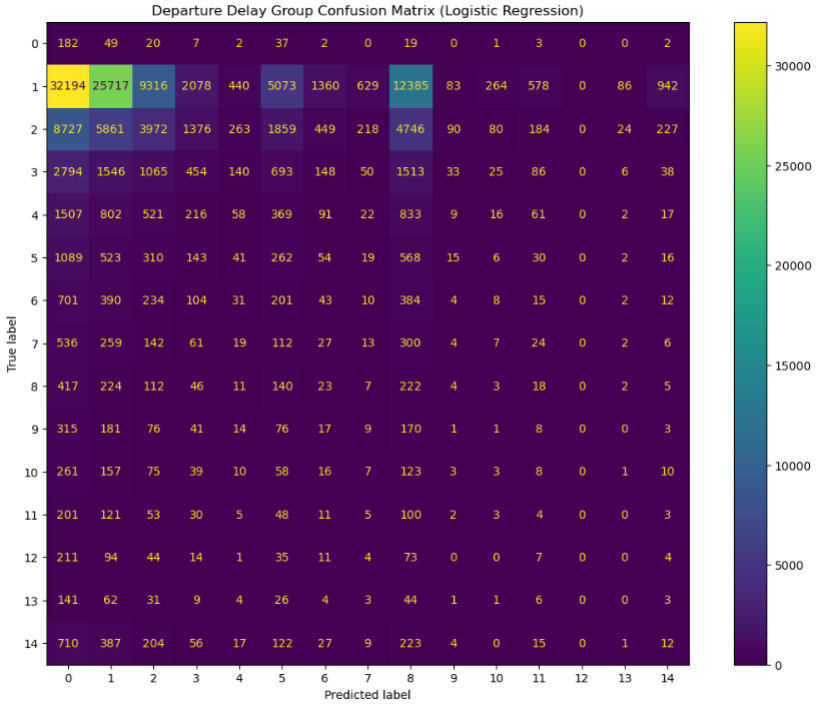

Figure 20: Multi-Class Logistic Regression Confusion Matrix

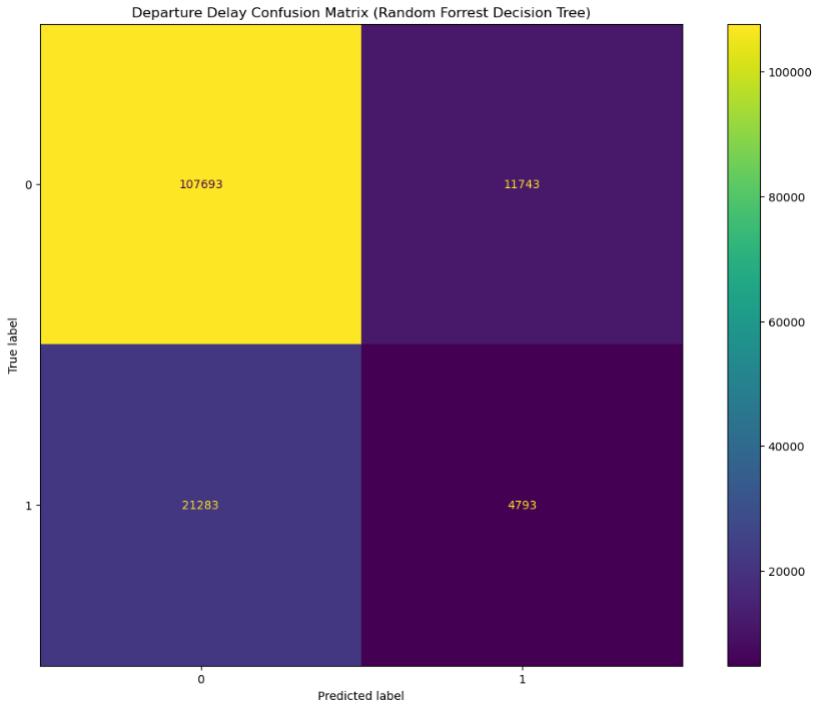

Figure 21: Two Class Random Forest Decision Tree Confusion Matrix

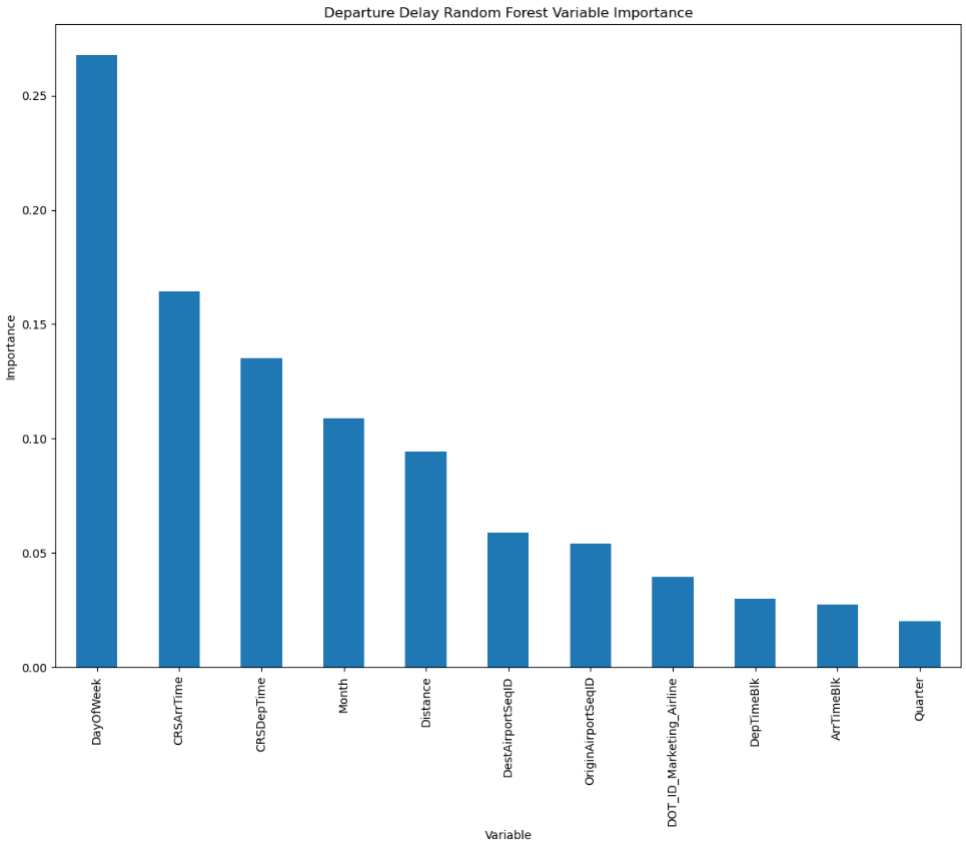

Figure 22: Two Class Random Forest Decision Tree Variable Importance

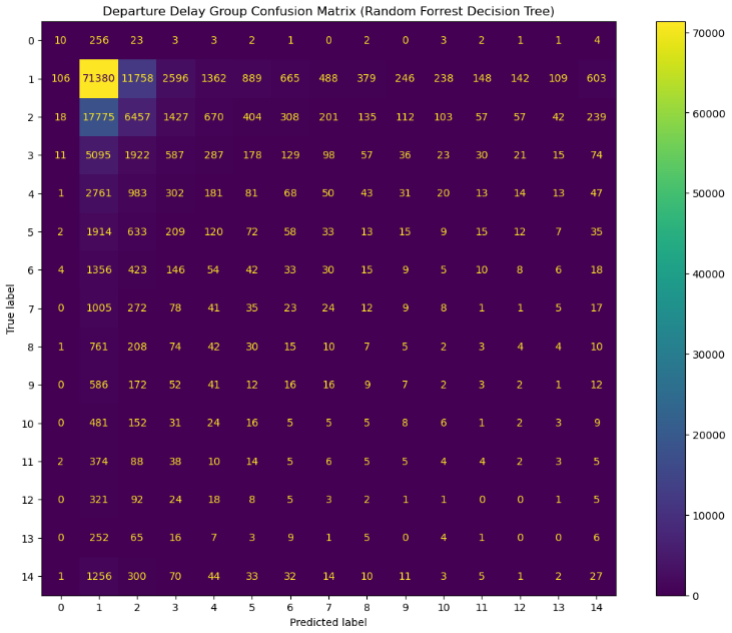

Figure 23: Multi-Class Random Forest Decision Tree Confusion Matrix

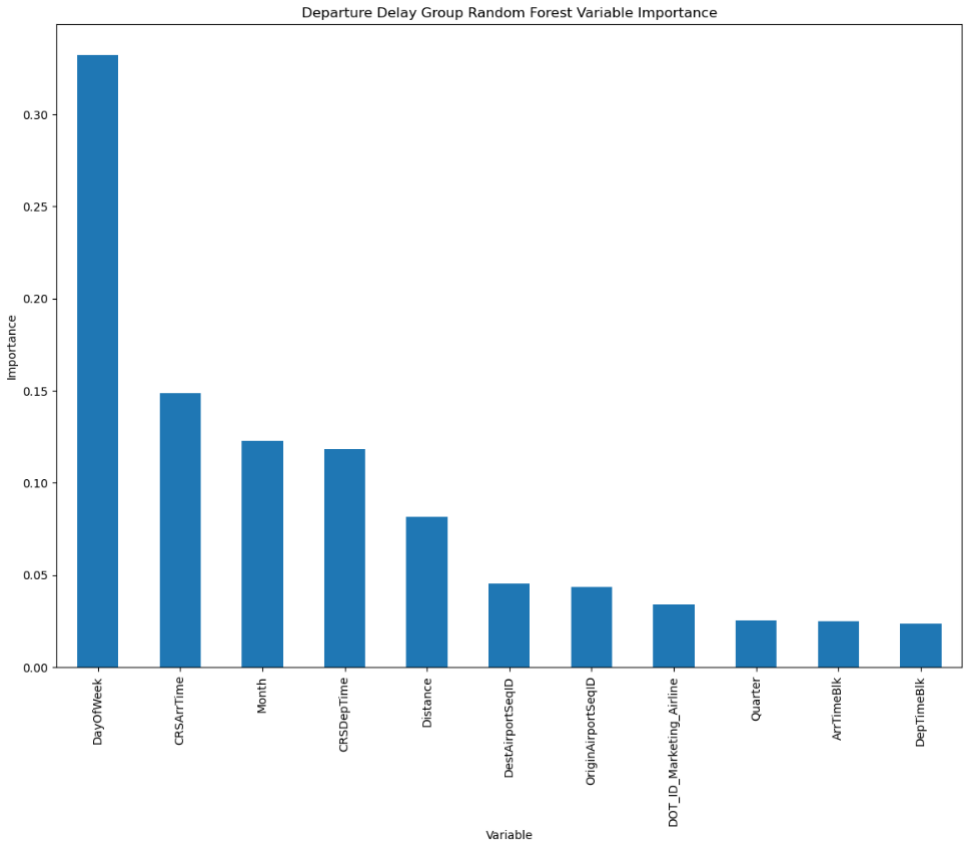

Figure 24: Multi-Class Random Forest Decision Tree Variable Importance

10 Code

10.1 Cleaning

10.1.1 O'Hare Subset

This script gathered all the .CSV files, compiled them to one dataframe, subset to O'Hare International Airport's flights and exported to a single .CSV file that is referenced throughout the project.

```
csv_path = 'C:\Code\school\DAT-490-Data\csvs'
data_lst = []

csv_files = [f for f in listdir(csv_path) if isfile(join(
                                csv_path, f))]

for data_file in csv_files:
    this_csv_path = join(csv_path, data_file)
    data_lst.append(pd.read_csv(this_csv_path))

assert len(data_lst) == 12

flight_delays_df = pd.concat(data_lst)
print("Length of Flight Delay Data:", len(flight_delays_df))

ord_flight_delays_df = flight_delays_df.loc[ (
                                flight_delays_df['Origin'] ==
                                'ORD' ) | ( flight_delays_df['
                                Dest'] == 'ORD' ) ]

print("Length of ORD Data:", len(ord_flight_delays_df))
ord_flight_delays_df.to_csv('../data/ORD_11_21-11-22.csv')
```

10.1.2 Model Level Data

This script cleans the data to model level and is only used in machine learning models. This drops NA values, fills 0 for delays types and subsets to only columns that may be needed when creating a model for this project.

```
import pandas as pd
df = pd.read_csv("../data/ORD_11_21-11_22.csv")

#fill NA values of flight delay times by types
df['CarrierDelay'] = df['CarrierDelay'].fillna(0)
df['WeatherDelay'] = df['WeatherDelay'].fillna(0)
df['NASDelay'] = df['NASDelay'].fillna(0)
df['SecurityDelay'] = df['SecurityDelay'].fillna(0)
```

```
df['LateAircraftDelay'] = df['LateAircraftDelay'].fillna(0)



init_test_lst = ['Marketing_Airline_Network', 'Origin', '
                                DepTimeBlk', 'DepDelay', '
                                ArrTimeBlk',
                  'ArrDelay', 'Distance', 'CarrierDelay', '
                                                    WeatherDelay'
                                                    , 'NASDelay',
                                                     '
                                                    SecurityDelay
                                                    ',
                    'LateAircraftDelay', 'TaxiIn', 'TaxiOut']
model_col_lst = ['DOT_ID_Marketing_Airline', '
                                OriginAirportSeqID', '
                                DepTimeBlk', 'DepDelay',
'DestAirportSeqID', 'ArrTimeBlk', 'ArrivalDelayGroups', '
                                ArrDelay','Distance',
'CarrierDelay', 'WeatherDelay', 'NASDelay', 'SecurityDelay',
                                'LateAircraftDelay',
'TaxiIn', 'TaxiOut', 'ArrDel15', 'CRSElapsedTime','CRSDepTime
                                ', 'CRSArrTime']

# blk_mapping.csv as a dict
blk_map_dict ={
    '0001-0559' : 0,
    '0600-0659' : 6,
    '0700-0759' : 7,
    '0800-0859' : 8,
    '0900-0959' : 9,
    '1000-1059' : 10,
    '1100-1159' : 11,
    '1200-1259' : 12,
    '1300-1359' : 13,
    '1400-1459' : 14,
    '1500-1559' : 15,
    '1600-1659' : 16,
    '1700-1759' : 17,
    '1800-1859' : 18,
    '1900-1959' : 19,
    '2000-2059' : 20,
    '2100-2159' : 21,
    '2200-2259' : 22,
    '2300-2359' : 23
}

# copy and apply block time mapping
model_df = df.copy()
```

```
model_df['DepTimeBlk'] = df['DepTimeBlk'].apply(lambda x:
                                blk_map_dict.get(x))
model_df['ArrTimeBlk'] = df['ArrTimeBlk'].apply(lambda x:
                                blk_map_dict.get(x))

model_df = model_df[model_col_lst]
print(model_df.columns)

# final drop of all left over NA values
model_df = model_df.dropna()
model_df.isna().values.any()

# write to csv for later use
print(len(model_df))
model_df.to_csv('../data/ORD_11_21-11-22_model.csv')
```

10.2 Exploratory Data Analysis

10.2.1 EDA Specific Cleaning

The EDA uses some different data cleaning, as it was prepared before models. This is less strict to mappings, and uses more human readable strings.

```
# libraries used
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

ord_df = pd.read_csv("../data/ORD_11_21-11_22.csv")

# Columns with mixed type warning, lets investigate
ord_df.iloc[:, [12, 14, 87, 94]]

# None of these columns are needed, we can ignore and create
                                our second subfile
include_lst = ['Year', 'Quarter', 'Month', 'DayofMonth', '
                                DayOfWeek',
'FlightDate','Marketing_Airline_Network', 'Tail_Number',
'Flight_Number_Operating_Airline','Tail_Number', 'Origin', '
                                Dest',
'OriginCityName', 'OriginState','OriginStateFips',
'OriginStateName','DestCityName', 'DestState', 'DestStateFips
                                ',
'DestStateName','CRSDepTime', 'DepTime', 'DepDelay',
'DepDelayMinutes', 'DepDel15', 'DepartureDelayGroups', '
                                DepTimeBlk',
'TaxiOut', 'WheelsOff', 'WheelsOn',
```

```
'TaxiIn', 'CRSArrTime', 'ArrTime', 'ArrDelay', '
                                ArrDelayMinutes',
'ArrDel15','ArrivalDelayGroups', 'ArrTimeBlk', 'Cancelled',
'CancellationCode', 'Diverted', 'CRSElapsedTime','
                                ActualElapsedTime',
'AirTime', 'Flights', 'Distance', 'DistanceGroup',
'CarrierDelay', 'WeatherDelay', 'NASDelay', 'SecurityDelay',
'LateAircraftDelay']
new_ord_df = ord_df[include_lst]

ord_df = new_ord_df

# gets all NA columns to further clean
def lst_na_cols():
    na_col_lst = []
    no_na_lst = []
    for i in ord_df.columns:
        if ord_df[i].isna().values.any():
            na_col_lst.append(i)
        else:
            no_na_lst.append(i)
    assert len(ord_df.columns) == (len(na_col_lst) + len(
                                    no_na_lst))

    print("NA exists in:", na_col_lst)
    print("There are no NAs in :", no_na_lst)

lst_na_cols()

# Before dealing with NA we should fill the NAs that should
                                be zero
# These are delay times for sure
ord_df['CarrierDelay'] = ord_df['CarrierDelay'].fillna(0)
ord_df['WeatherDelay'] = ord_df['WeatherDelay'].fillna(0)
ord_df['NASDelay'] = ord_df['NASDelay'].fillna(0)
ord_df['SecurityDelay'] = ord_df['SecurityDelay'].fillna(0)

#info for development
pre_ord_df_len = len(ord_df)
ord_df = ord_df.dropna(subset=['ArrDelayMinutes', 'ArrDelay']
                                )
print(f"Dropped {pre_ord_df_len - len(ord_df)} data points")

# lastly cap to 12 hours
ord_df = ord_df[ord_df['ArrDelay'] < 720]
```

10.2.2 EDA Statistics Code

```
ord_df.describe()

ord_delay_mean = ord_df['ArrDelay'].mean()
ord_delay_median = ord_df['ArrDelay'].median()

print("ORD All flights Delay Mean:", ord_delay_mean)
print("ORD All flights Delay Median:", ord_delay_median)

# use FAA definition of delay
ord_delays_df = ord_df['ArrDelay'] >= 15
ord_delays_df = ord_df[ord_delays_df]
ord_delay_mean = ord_delays_df['ArrDelay'].mean()
ord_delay_median = ord_delays_df['ArrDelay'].median()

print("ORD Delay Mean:", ord_delay_mean)
print("ORD Delay Median:", ord_delay_median)

flights_num = ord_df['ArrDelay'].count()
delays_num = ord_delays_df['ArrDelay'].count()
print("Amount of Flights:", flights_num)
print("Amount of Delays:", delays_num)
print("Precentage of Delayed Flights:", (delays_num/
                                flights_num))
```

10.2.3 EDA Plots Code

Code for Pie Chart of delay type split at ORD.

```
pie_chart_lst = [ord_df['CarrierDelay'].sum(), ord_df['
                                WeatherDelay'].sum(),
 ord_df['NASDelay'].sum(), ord_df['SecurityDelay'].sum(),
                                 ord_df['LateAircraftDelay'].
                                 sum()]

pie_chart_labels_lst = ["Carrier Delay", 'Weather Delay', '
                                Air System Delay',
 'Security Delay', 'Late Aircraft Delay']

fig, ax1 = plt.subplots(1,1, figsize=(14,10))
ax1.pie(pie_chart_lst, labels=pie_chart_labels_lst, autopct='
                                %1.1f%%', explode=(.02, .02, .
                                02, .02, .02));
ax1.set_title("ORD Delay Split");
```

Bar plot of delays by month at ORD.

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))

ord_delay_types_df = ord_delays_df[['Month','CarrierDelay', '
                                    WeatherDelay', 'SecurityDelay'
                                    , 'NASDelay', '
                                    LateAircraftDelay']]
delay_cnt_month = ord_delay_types_df.groupby(by='Month').sum
                                    ()
delay_cnt_month.plot(ax=ax1, kind='bar', stacked=True, cmap='
                                    jet')
ax1.set_title("ORD Delay Type by Month");
ax1.set_ylabel("Delay Count")
```

Box plot of Delay types at ORD.

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))
ord_delay_types_df.plot.box(ax=ax1)
ax1.set_title("ORD Delay Types Box Plot");

ord_delay_types_df_2 = ord_delays_df[ord_delays_df['ArrDelay'
                                    ] < 360]
ord_delay_types_df_2 = ord_delay_types_df_2[['Month','
                                    CarrierDelay', 'WeatherDelay',
                                     'SecurityDelay', 'NASDelay',
                                    'LateAircraftDelay']]

fig, ax1 = plt.subplots(1,1, figsize=(14,10))
ord_delay_types_df_2.plot.box(ax=ax1)
ax1.set_title("ORD Delay Types Box Plot Capped at 6 Hours");

ord_delay_types_df_2 = ord_delays_df[ord_delays_df['ArrDelay'
                                    ] < 180]
ord_delay_types_df_2 = ord_delay_types_df_2[['CarrierDelay',
                                    'WeatherDelay', 'SecurityDelay
                                    ', 'NASDelay', '
                                    LateAircraftDelay']]

fig, ax1 = plt.subplots(1,1, figsize=(14,10))
ord_delay_types_df_2.plot.box(ax=ax1)
ax1.set_title("ORD Delay Types Box Plot Capped at 3 Hours");

fig, ax1 = plt.subplots(1,1, figsize=(14,10))
ord_delays_df['ArrDelay'].plot(ax=ax1, kind='kde')
ax1.set_title("ORD Delay Types Box Plot Capped at 3 Hours");
```

Distributions of airline flight delays at ORD.

```
airlines_distribution = sns.displot(data=ord_delays_df, x='
                                    ArrDelay', col='
```

```
                                Marketing_Airline_Network',
                                col_wrap=2, hue='
                                Marketing_Airline_Network')
airlines_distribution.set(xlim=(0,180))

fig, ax1 = plt.subplots(1,1, figsize=(14,10))

airline_nums_flights = airline_nums['Year']
airline_nums_flights.plot.bar(ax=ax1)
ax1.set_title("Flights at ORD by Airline")
```

KDE plot of flight delay times at ORD.

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))
ord_delays_180_df = ord_delays_df[ord_delays_df['ArrDelay'] <
                                  180]
ord_delays_180_df['ArrDelay'].plot(ax=ax1, kind='kde')
ax1.set_title("ORD Arrival Delay KDE Capped at 3 Hours");
```

Simple linear regression for exploration.

```
dep_time_lm = sns.lmplot(data=ord_df, x='LateAircraftDelay',
                                y='CarrierDelay', line_kws={'
                                color':'red'})
dep_time_lm.set(title='Linear Regression Air Time by Arrival
                                Delay')
```

10.3 Models

Imports Machine Learning Techniques.

```
# General Imports
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE

# R1.1 Imports
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, recall_score,
                                precision_score
from sklearn.metrics import roc_curve, precision_recall_curve
                                , roc_auc_score

#R1.2 Imports
from sklearn.metrics import confusion_matrix,
                                ConfusionMatrixDisplay
```

```
from sklearn.metrics import classification_report

#R1.3 Imports
from sklearn.ensemble import RandomForestClassifier

df = pd.read_csv("../data/ORD_11_21-11-22_model.csv")
```

10.3.1 Research Question 1

Logistic Regression (2 Classes)

```
# Two class subset
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', '
                                DestAirportSeqID', 'DepDelay',
                                 'ArrTimeBlk', 'Distance', '
                                TaxiOut', 'CRSDepTime',
 'CRSArrTime', 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['ArrDel15']

fit = LogisticRegression(solver='lbfgs', max_iter=10000,
                                class_weight={0:15, 1:85},
                                random_state=15).fit(x_train,
                                y_train)

y_pred = fit.predict(x_test)
y_prob = fit.predict_proba(x_test)
y_prob = y_prob[:,1] # want the positive values
```

Logistic Regression (Multi-Class)

```
# Multi class subset **overwriting variables**
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', 'DepDelay', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'TaxiOut', '
                                CRSDepTime',
 'CRSArrTime', 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['ArrivalDelayGroups']

x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
fit = LogisticRegression(solver='sag', max_iter=10000,
                                multi_class='multinomial').fit
                                (x_train, y_train)
y_pred = fit.predict(x_test)
y_prob = fit.predict_proba(x_test)
y_prob = y_prob[:,1] # want the positive values
```

Random Forest Decision Tree (2 Class)

```
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', 'DepDelay', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'TaxiOut', '
                                CRSDepTime', 'CRSArrTime',
 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['ArrDel15']

x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
oversample = SMOTE(k_neighbors=2)
df_x, df_y = oversample.fit_resample(df_x, df_y)
rfc = RandomForestClassifier(random_state=15).fit(x_train,
                                y_train)
rfc_pred = rfc.predict(x_test)

print(f"Accuracy of Model: {accuracy_score(y_test, rfc_pred)}
                                ")
print(classification_report(y_test, y_pred))
```

Random Forest Decision Tree (Multi-Class)

```
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', 'DepDelay', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'TaxiOut', '
                                CRSDepTime', 'CRSArrTime',
 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['ArrivalDelayGroups']

x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
oversample = SMOTE(k_neighbors=2)
df_x, df_y = oversample.fit_resample(df_x, df_y)
rfc = RandomForestClassifier(random_state=15).fit(x_train,
                                y_train)
rfc_pred = rfc.predict(x_test)

print(f"Accuracy of Model: {accuracy_score(y_test, rfc_pred)}
                                ")
print(classification_report(y_test, y_pred))
```

10.3.2 Research Question 2

Grid search

```
lst_tst = []
for i in range(0,101):
    for j in range(0,101):
        lst_tst.append({0:i, 1:j})

grid_params = [{'class_weight': lst_tst,
"penalty": ['l1', 'l2'],
'fit_intercept':[True, False]
}]

grid_search = GridSearchCV(fit, grid_params, cv=10, scoring='
                                accuracy')
grid_search.fit(x_train, y_train)
print(grid_search.best_params_)
```

Logistic Regression (2 Classes)

```
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'CRSDepTime',
 'CRSArrTime', 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['DepDel15']

oversample = SMOTE(k_neighbors=2)
df_x, df_y = oversample.fit_resample(df_x, df_y)
x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
fit = LogisticRegression(solver='lbfgs', max_iter=10000,
                                random_state=15, class_weight=
                                'balanced', fit_intercept=True
                                , penalty='l2', C=0.15).fit(
                                x_train, y_train)
y_pred = fit.predict(x_test)
y_prob = fit.predict_proba(x_test)
y_prob = y_prob[:,1] # want the positive values

print(f"Accuracy of Model: {accuracy_score(y_test, y_pred)}")
print(f"Precision of Model: {precision_score(y_test, y_pred)}
                                ")
print(f"Recall of Model: {recall_score(y_test, y_pred)}")
```

Logistic Regression (Multi-Class)

```
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'CRSDepTime',
 'CRSArrTime', 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['DepartureDelayGroups']
```

```
x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
fit = LogisticRegression(solver='lbfgs', max_iter=10000,
                                random_state=15, class_weight=
                                'balanced', multi_class='
                                multinomial', fit_intercept=
                                True, penalty='l2', C=0.15).
                                fit(x_train, y_train)
y_pred = fit.predict(x_test)
y_prob = fit.predict_proba(x_test)
y_prob = y_prob[:,1] # want the positive values

print(f"Accuracy of Model: {accuracy_score(y_test, y_pred)}")
```

Random Forest Decision Tree (2 Class)

```
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'CRSDepTime',
 'CRSArrTime', 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['DepDel15']

x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
oversample = SMOTE(k_neighbors=2)
df_x, df_y = oversample.fit_resample(df_x, df_y)
rfc = RandomForestClassifier(random_state=15).fit(x_train,
                                y_train)
rfc_pred = rfc.predict(x_test)

print(f"Accuracy of Model: {accuracy_score(y_test, rfc_pred)}
                                ")
print(classification_report(y_test, y_pred))
```

Random Forest Decision Tree (Multi-Class)

```
df_x = df[['DOT_ID_Marketing_Airline', 'OriginAirportSeqID',
                                'DepTimeBlk', '
                                DestAirportSeqID', 'ArrTimeBlk
                                ', 'Distance', 'CRSDepTime',
 'CRSArrTime', 'Quarter', 'Month', 'DayOfWeek']]
df_y = df['DepartureDelayGroups']

x_train, x_test, y_train, y_test= train_test_split(df_x, df_y
                                , test_size=.25)
oversample = SMOTE(k_neighbors=2)
df_x, df_y = oversample.fit_resample(df_x, df_y)
```

```
rfc = RandomForestClassifier(random_state=15).fit(x_train,
                                    y_train)
rfc_pred = rfc.predict(x_test)

print(f"Accuracy of Model: {accuracy_score(y_test, rfc_pred)}
                                    ")
```

10.3.3 Model Evaluation

Logistic Regression Confusion Matrix Evaluation

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))

matrix = confusion_matrix(y_test, y_pred)
ConfusionMatrixDisplay(confusion_matrix=matrix).plot(ax=ax1)
ax1.set_title("Arrival Delay Confusion Matrix (Logistic
                                    Regression)")

print(f"Accuracy of Model: {accuracy_score(y_test, y_pred)}")
print(f"Precision of Model: {precision_score(y_test, y_pred)}
                                    ")
print(f"Recall of Model: {recall_score(y_test, y_pred)}")
```

Logistic Regression ROC Curve Evaluation

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))
fpr, tpr, _ = roc_curve(y_test, y_prob)
auc = roc_auc_score(y_test, y_prob)

print(f"AUC Score: {auc}")

ax1.plot(fpr, tpr)
ax1.set_ylabel('True Positive Rate')
ax1.set_xlabel('False Positive Rate')
ax1.set_title("Arrival Delay ROC Curve (Random Forrest
                                    Decision Tree)")
```

Logistic Regression Recall Vs. Precision Curve Evaluation

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))
prec, recall, _ = precision_recall_curve(y_test, y_prob)

ax1.plot(prec, recall)
ax1.set_ylabel('Precision')
ax1.set_xlabel('Recall')
ax1.set_title("Arrival Delay Precision Vs. Recall (Logistic
                                    Regression)")
```

Random Forest Decision Tree Variable Importance

```
fig, ax1 = plt.subplots(1,1, figsize=(14,10))
feature_importance = pd.Series(rfc.feature_importances_,
                               index=x_train.columns).
                               sort_values(ascending=False)

feature_importance.plot.bar(ax=ax1)
ax1.set_xlabel("Variable")
ax1.set_ylabel("Importance")
ax1.set_title("Arrival Delay  Random Forest Variable
                               Importance")
```

11 References

References

[1] Leonardo Carvalho, Alice Sternberg, Leandro Maia Gonçalves, Ana Beatriz Cruz, Jorge A. Soares, Diego Brandão, Diego Carvalho, and Eduardo Ogasawara. On the relevance of data science for flight delay research: a systematic review. *Transport Reviews*, 41(4):499–528, 2021.

[2] Roberto Henriques and Inês Feiteira. Predictive modelling: Flight delays and associated factors, hartsfield–jackson atlanta international airport. *Procedia Computer Science*, 138:638–645, 2018. CENTERIS 2018 - International Conference on ENTERprise Information Systems / ProjMAN 2018 - International Conference on Project MANagement / HCist 2018 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018.

[3] Yufeng Tu, Michael O Ball, and Wolfgang S Jank. Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern. *Journal of the American Statistical Association*, 103(481):112–125, 2008.