A

PROJECT REPORT ON

"DESIGNING A MODEL THAT PREDICTS FIGHT DELAYS"

Submitted by

Group No: 3
ABHIJEET CHAVAN(1309)
VIVEK KUMAR SINGH(1357)

Centre Coordinator
Mr. PRASHANT KARHALE

Under the Guidance of
Mr. AKSHAY TILEKAR

In the fulfillment of e – Diploma in Big Data Analytics course from

Institute for Advance Computing and Software Development Akurdi, Pune

in the academic year

May 2021 – Sept.2021

e-DBDA
May 2021

Institute for Advance Computing and Software Development Akurdi,
Pune. 411044

ACKNOWLEDGEMENT

It gives us immense pleasure to present our report for project on “PREDICTING FLIGHT DELAYS.” The able guidance of all teaching staff of this department made the study possible. They have been a constant source of encouragement throughout this project. We would like to express our grateful thanks to Mr. Prashant Karhale Sir, Mr. Akshay Tilekar Sir who guided us properly for this project. We would also like to express our sincere thanks to Institute for Advance Computing and Software Development Akurdi, Pune for giving us an opportunity to explore the subject and use our knowledge by conducting this project.

Group No: 29

VIVEK KUMAR SINGH(1357)

ABHIJEET CHAVAN(1309)e-

DBDA, May 2021

Institute for Advance Computing and Software Development, Akurdi

CONTENTS

1. INTRODUCTION_ 1
2. GLOSSARY_ 2
3. OBJECTIVE_ 3
4. BLOCK DIAGRAM_ 4
5. WORKING METHODOLOGY

    5.1 Data Gathering_ 5

    5.2 Data Cleaning_ 8

    5.3 EDA _ 11

    5.4 Modelling ML Algorithims Analysis_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 15
6. SIGNIFICANCE_ 17
7. FUTURE SCOPE_ _18
8. APPLICATIONS_ 18
9. CONCLUSION_ 19

BIBLIOGRAPHY_ 20

Chapter - 1

INTRODUCTION

Flight delay has become a very important subject for air transportation all over the world because of the associated financial loses that the aviation industry is continuously going through.

According to data from the Bureau of Transportation Statistics (BTS) of the United Stated, over 20% of US flights were delayed during 2018, which resulted in a severe economic impact equivalent to circa 41 billion US$.

These delays not only cause inconvenience to the airlines, but also to passengers. With the increased travel time comes and increase in expenses associated to food and lodging and this results in added stress among passengers, but this doesn't account for the growing distrust towards the airlines, who also suffer from extra costs such as those associated to their crews, aircraft repositioning, increased fuel consumption while trying to reduce their elapse time, and many others that tarnish the airlines reputation and often result in the loss of demand by passengers.

The reasons for these delays vary a lot going from air congestion to weather conditions, mechanical problems, difficulties while boarding passengers, and simply the airlines inability to handle the demand given its capacity.

So what can be done as a passenger to avoid delayed flights? is it possible to know if your flight will be delayed before it comes up on the departure boards? or before you being inside the plane? The answer to these questions is maybe. By using Machine Learning (ML) Algorithms you can try to predict if your flight will be delayed in many ways. Of course, all of these different algorithms will have pitfalls and a certain degree of accuracy, and they will all depend on the data that they are fed.

GLOSSARY

Training data: The data that are used to train regression models.

Validation data: The data that are used to test the performance of the regression model during the training process. Validation data are used to fine tune parameters in the classification model and the data should not be part of the training data.

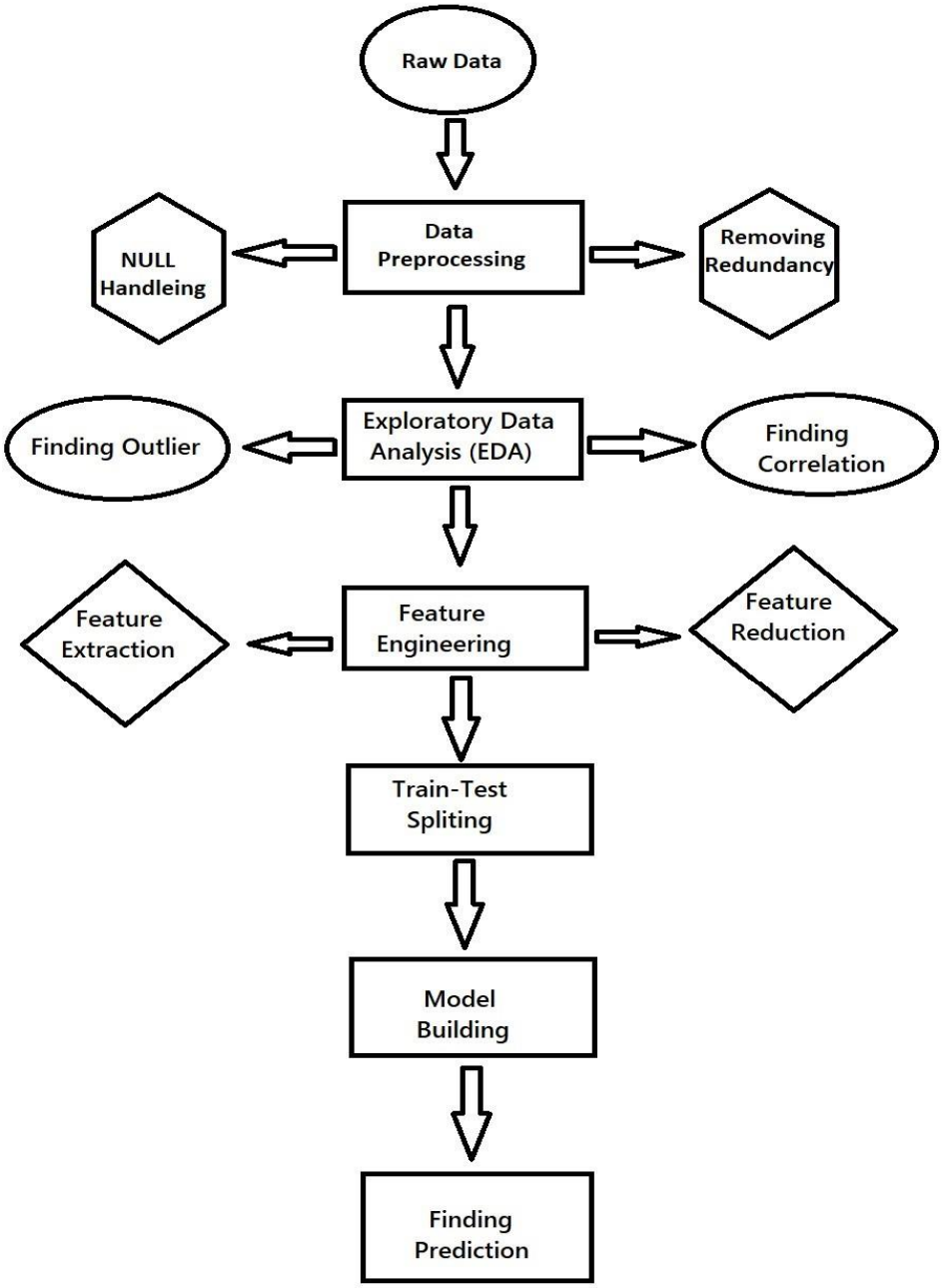
Testing data: The data that are used to test the performance of the trained regression model (after training process). Testing data are used to evaluate the final performance of the regression model. Testing data should not be part of training data or validation data. No fine tune should be made based on the result of testing data.

Overfitting: The regression model performs consistently better on training data than on validation data and testing data.

OBJECTIVE

- The objective of this project is very clear as described in the introduction: **"Design a Model that predicts flight delays before they are announced on the departure boards.**
- Analyse the computation model to better understand the important features for predicting delays. Based on this understanding, validate, and further improve the performance of the model flight delays.
- Based on the result of the computation model, develop a model design for flight delays prediction

BLOCK DIAGRAM



Chapter – 2

WORKING METHODOLOGY

1- DATA GATHERING

The dataset comes from Kaggle, and it consists of a multi-year data ranging from 2009 to 2018 separated in 10 different files.

Each one of these datasets has 28 categories/features in average with a few million rows. Because of the size of each file I chose to work with only one, corresponding to the 2018. This one consists of 28 categories with just over 7.2 million rows.

Below is the glossary of all the features/categories available

Glossary

FL_DATE = Date of the Flight
OP_CARRIER = Airline Identifier
OP_CARRIER_FL_NUM = Flight Number
ORIGIN = Starting Airport Code
DEST = Destination Airport Code
CRS_DEP_TIME = Planned Departure Time
DEP_TIME = Actual Departure Time
DEP_DELAY = Total Delay on Departure in minutes
TAXI_OUT = The time duration elapsed between departure from the origin airport gate and wheels off
WHEELS_OFF = The time point that the aircraft's wheels leave the ground
WHEELS_ON = The time point that the aircraft'ss wheels touch on the ground
TAXI_IN = The time duration elapsed between wheels-on and gate arrival at the destination airport
CRS_ARR_TIME = Planned arrival time
ARR_TIME = Actual Arrival Time = ARRIVAL_TIME - SCHEDULED_ARRIVAL

ARR_DELAY = Total Delay on Arrival in minutes
CANCELLED = Flight Cancelled (1 = cancelled)
CANCELLATION_CODE = Reason for Cancellation of flight: A - Airline/Carrier; B - Weather; C - National Air System; D - Security
DIVERTED = Aircraft landed on different airport that the one scheduled
CRS_ELAPSED_TIME = Planned amount of time needed for the flight trip
ACTUAL_ELAPSED_TIME = AIR_TIME+TAXI_IN+TAXI_OUT
AIR_TIME = The time duration between wheels_off and wheels_on time
DISTANCE = Distance between two airports
CARRIER_DELAY = Delay caused by the airline in minutes
WEATHER_DELAY = Delay caused by weather
NAS_DELAY = Delay caused by air system
SECURITY_DELAY = caused by security reasons
LATE_AIRCRAFT_DELAY = Delay caused by security

Source: Kaggle

As I mentioned in the Introduction, I will be only considering features that you are aware of before the plane takes off. This way what I am predicting is before you board the plane and not while you are in the plane in mid air, which wouldn't be of much use as you would want to know if you will be late before you board the plane. Adding any of the features listed below would increase your accuracy to at least 85%, which sounds great, but then again, what's the point if you are already in the air or about to take off?

- TAXI_OUT
- WHEELS_OFF
- WHEELS_ON
- TAXI_IN
- ARR_DELAY
- ACTUAL_ELAPSED_TIME

Now, there is an additional feature that will biased the models, and that is the DEP_DELAY (Departure Delay), which yes, if your plane is leaving late then your chances of arriving late to your destination will increase. The plot on Figure_1, which is part of the EDA done, shows this. There I compared the DEP_DELAY with the ARR_DELAY by airline, and as you can see, normally when your flight leaves late, the

airlines pushes for the flights to have shorter elapse times to compensate for the delay, and in some cases, this is accounted for and the flight ends up arriving either on time, or earlier, such as with Delta Airlines and Alaska airlines, which have both negative arrival averages, meaning an early arrival.

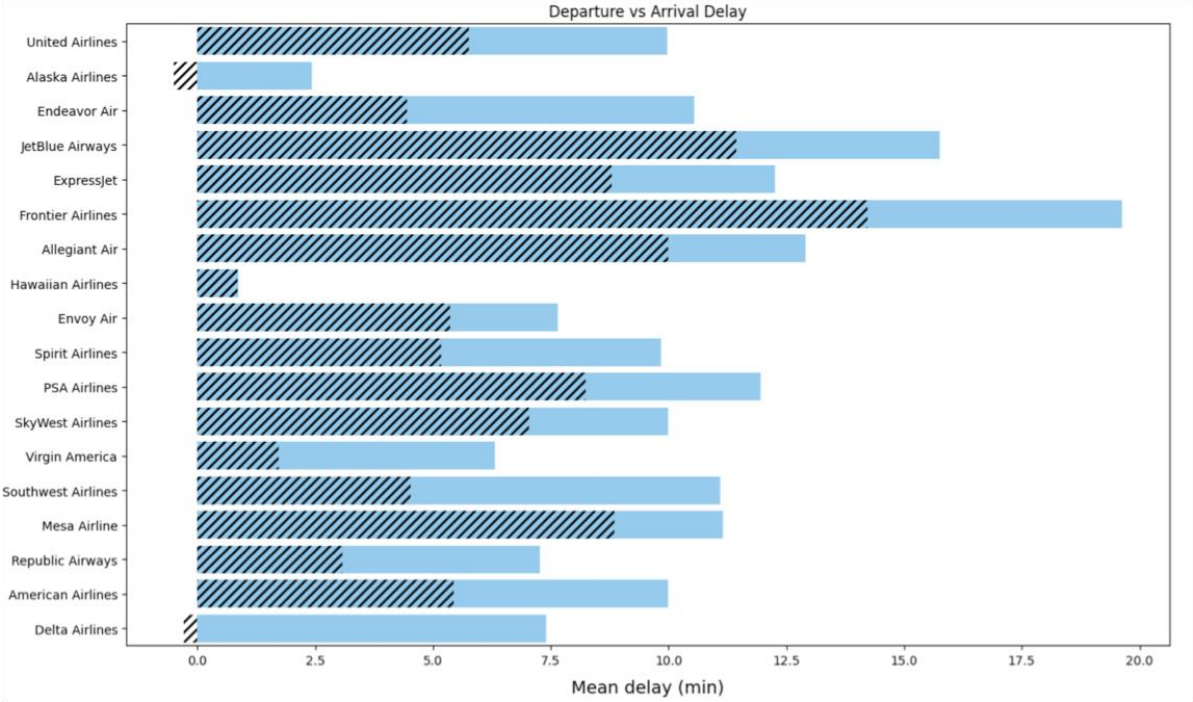


Figure_1. "Departure Delays" compared to "Arrival Delays" by airline

Some people might argue, that if your flight's departure is delayed, you will see it on the screens before you board the plane, so that means that I should leave it on my predictive model, right? well yes and no. Yes I should leave it because you are right about seeing the flight's departure being delayed before you board the plane, but then no, because a late departure will most probably mean a late arrival (Figure 1) even when the airline tries to compensate by reducing the elapsed time as the above plot suggests. So this will definitely affect the accuracy of my predictions in a positive but unrealistic predictive way. Still, I have ran two models for each ML and Neural Network algorithm that I have tested, one with the DEP_DELAY and a second without the DEP_DELAY. You will notice that there is a large difference in the accuracy of the models and respective metric, but that is because of the nature of the predictions being made.

2- DATA CLEANING

The data preprocessing and cleaning was done in two separate parts, documented in two notebooks to make it easier to follow up due to their length.

The first section is a standard cleaning involving minimal feature engineering, and the second is driven after the 20 most common arrival destinations were defined based on the number of flights and is the one that contains the most feature engineering done.

The first step before going into the data cleaning was to define what I will be considering a **delayed flight**. This is important because it will determine if I can drop or not any other columns and how I will be choosing the predictive features to work with. So, for a flight to be considered delayed, it has to meet the following criteria:

*** Arrive late at its destination**

Quite simple, and this means, that even if a flight has a delay from its departure, but still arrives to its destination on time, it will not be considered a delayed flight

Based on the above, also a canceled flight will not be a delayed one either. Therefore, you can assume that I dropped that column, but not only for this reason, but also due to the high number/percentage of missing values (~81%). This could have been very useful for EDA, but unfortunately most of it was not available.

Each one of the columns within the main dataframe was analyzed individually with the exception of the following 5:

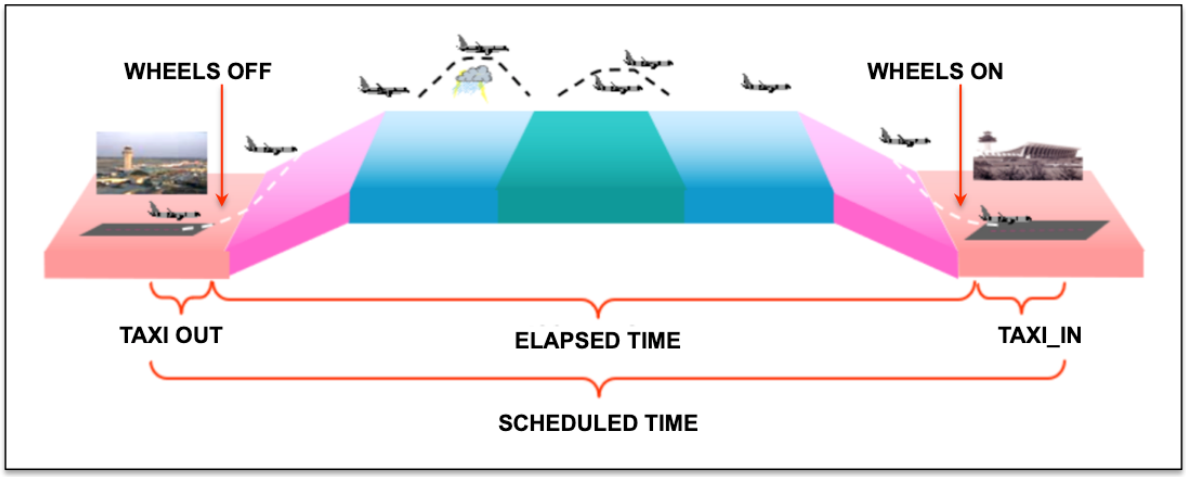
1. CARRIER_DELAY
2. WEATHER_DELAY
3. NAS_DELAY
4. SECURITY_DELAY

5. LATE_AIRCRAFT_DELAY

These 5 represent the different reasons why a flight is delayed. Unfortunately, for the 2018 dataset, 5,744,152 rows are missing, adding an 81% of the entire dataset. Therefore, a decision to drop those columns was also made.

Another set of features that were interesting but were not taken into account for the predictive modeling were the following four:

1. TAXI_OUT
2. WHEELS_OFF
3. WHEELS_ON
4. TAXI_IN



These four, as Figure_2 illustrates, add up the elapse time, which is the amount of time initially planned for the flight. Unfortunately, these don't add much value plus they can biased the model, so as a result they were dropped. An interesting fact about these columns though, is that a significant number of WHEELS_ON and TAXI_IN didn't have any values, whereas their respective TAXI_OUT and WHEELS_OFF did. How should this be interpreted? is it because the airlines responsible for them made mistakes and forgot about them? or is it because these aren't that relevant for them? more on this can be seen on the Cleaning and Preprocessing notebook where I tried to explain my findings and relate them to the responsible/owner airlines, but for the time being these will enter the category of what are known as ghost flights.

Because there are quite a few features on this dataset, I won't explain the work done on each one of them, instead I will just mentioned some that I found interesting, and if you would like to see more detail, the two cleaning and processing notebooks have every step explained in depth.

After a brief look at the data, the key features that needed some immediate work were the Airline (OP_CARRIER), and departing (ORIGIN) and arrival city/airport (DEST). These needed to have their abbreviations and their IATA codes changed to the airline and airports names respectively.

The dataset for this particular year (2018) didn't have available the airport.csv file with their name and IATA codes, and because this dataset contains 358 airports, adding them manually was not an option given the time for this project. The airlines was the easy part as they were only 18 of them, so that was done with the help of Wikipedia. For the IATA codes, the solution was to use the older file from 2015 by using its list of airports, then compared it to the one from the 2018 that I extracted from my main dataset (.csv file). That gave a difference of 41 airports that needed to be found online plus 4 airports that were on the 2015 list but not on the 2018 list, therefore those needed to be dropped. This still involved a bit of manual work but considerably less than the initial 358.

In terms of engineered features, the first one to be calculated was the target (FLIGHT_STATUS) which was the flight being delayed or not. This is a binary column, with a 0 for flights arriving on time, and a 1 for flights arriving late, calculated from the "Arrival Delay" (ARR_DELAY) column. With this column ready, the next step was a quick check for the data distribution, meaning, checking if the data is balanced or not. Results are plotted on Figure 3 and they suggest a severe imbalance dataset with an almost 2:1 ratio, this means right away that looking at accuracy on its own will not be enough to evaluate the models, but I will also need to look at other metrics such as Precision, Recall and F1.

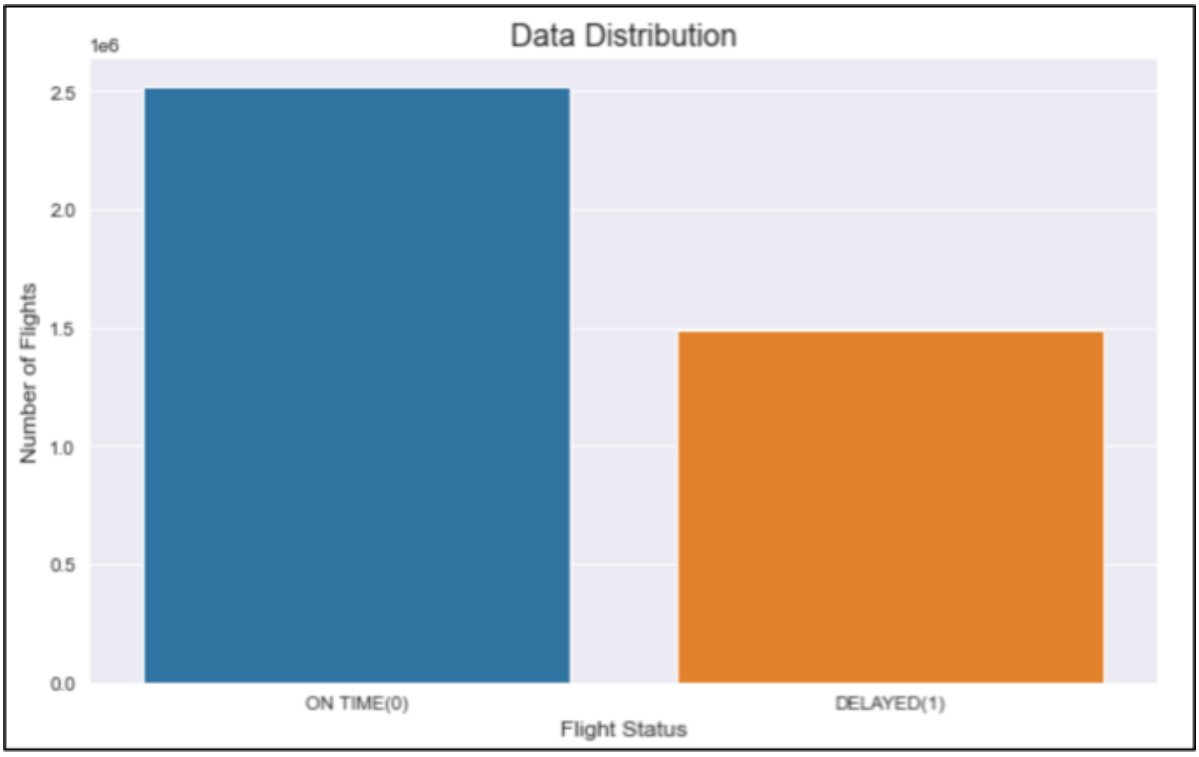


Figure 3. Data distribution showing a high imbalance dataset.

The imbalanced data means that I will need to weight these two classes while training my models.

Other features were engineered mainly to perform the EDA. Among those, some of the most relevant were:

- Calculating the total number of flights and total numbers of delayed flights (from departure and arrivals separately) by airline
- Extracting the "weekday" from the date using the "datetime" function from Pandas. Using the same function, the "month" and "day of the month" were also extracted
- Calculating percentages of delayed departures and arrivals by airlines and by cities
- Extracting the top destinations with average delays and arrivals
- Calculating best weekday to travel in terms of delays (departures and arrivals)
- Impact of late departure on arrival time (with difference between both)

As with the cleaning and preprocessing, if you wish to see more detail about the feature engineering, refer to the respective notebooks.

3-EDA(Exploratory data analysis)

We did the EDA part in jupyter notebook , however the difference here is that the visualizations done on each of the EDAs were done with different libraries. The libraries used are matplotlib and Seaborn.

On the EDA notebook, the following questions were addressed:

1. Total Number of Flights by Airline
2. Number of Delayed Flights by Airline
3. Percentage of Delayed Flights by Airline
4. Total Minutes Delayed by Airline
5. Average Delay Time by Airline
6. 30 Most Common Destination (Cities)
7. Worse and Best months to travel
8. Is there a Better day of the month to travel?
9. Best weekday to avoid delays
10. Impact of Delays (Departure vs Arrival Delay)
11. Most Popular Destinations with Average Arrival Delays
12. Number of Destination by Airline
13. Recommended airlines based on lowest delay times

You will notice that each one of these questions were addressed and discussed individually and afterward, put together to answer question 13.

Again, I won't go through all of them here, but just share a few interesting findings:

Total Number of Flights by Airline: The plot from Figure 3 talks by itself, therefore, it is quite easy to interpret. Basically stating that the top 5 airlines in terms of number of flights are:

- SouthWest Airlines
- Delta Airlines
- American Airlines
- SkyWest Airlines
- United Airlines

With no additional comments about this, I will come back to this list after looking at other plots.

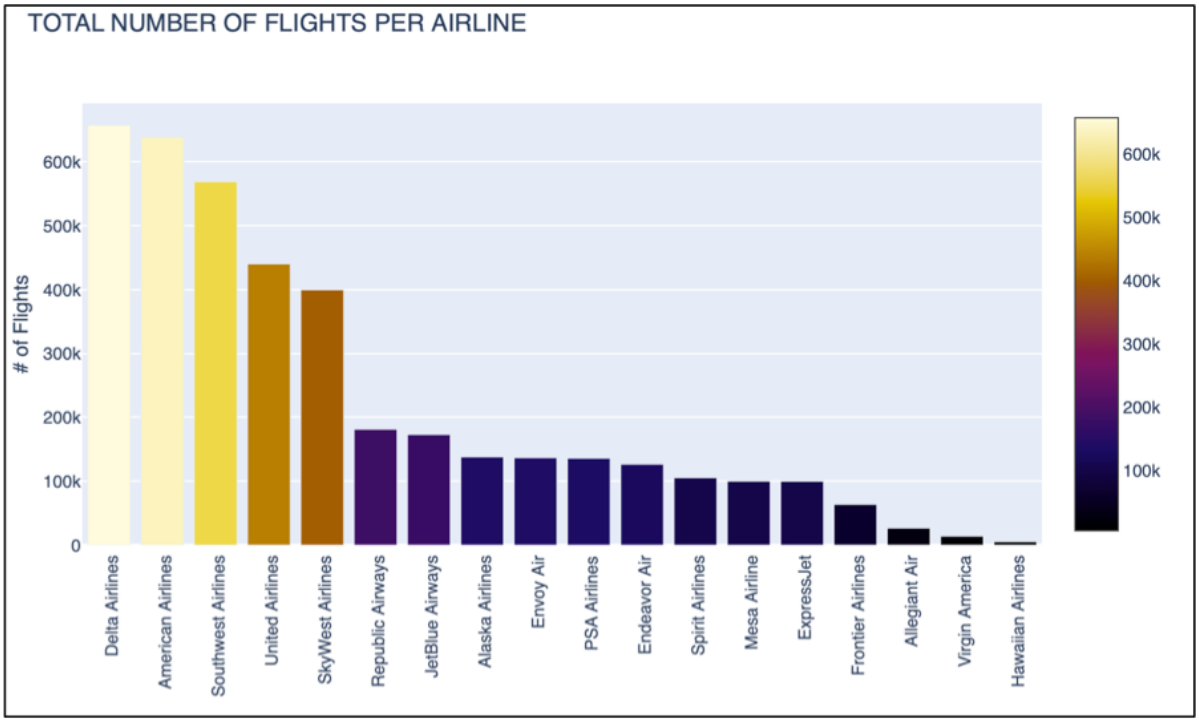


Figure 3. Total number of flights by airline sorted is descending order.

Percentage of Delayed Flights by Airline: It seems normal to think that the most flights you have the more likely it is that you will end up having more delayed flights. It's simple math right? For example, lets assume a fix percentage of delayed

flights such as 30%, well 30% of 100 is 30, whereas 30% of 1000 is 300. We translate that into flights, and there is a huge difference with a ratio of 10:1 in terms of numbers, but the percentage remains the same.

Now according to this dataset, the average of delayed flights in the US for 2018 was 37.52%, which is the red horizontal line on plot from Figure 4. I know that in the introduction I mentioned a 20% of flights within the US being delayed, but that number if overall for the 58 airlines that operate domestic US flights, whereas my dataset only looks at 18 airlines which I am assuming are the major carriers.

You as the airline don't want to be above that red line/threshold, you want to be as far as possible below it. If you pay attention to Delta Airline, they are top 5 in terms of number of flights, but they are dead last in terms of delay percentage. It is quite interesting the relationship that they have managed to achieve.

Another interesting observation is that SouthWest Airlines and American Airlines are two of the other top 5 in terms of number of flights and they are both above that threshold that we want to avoid.

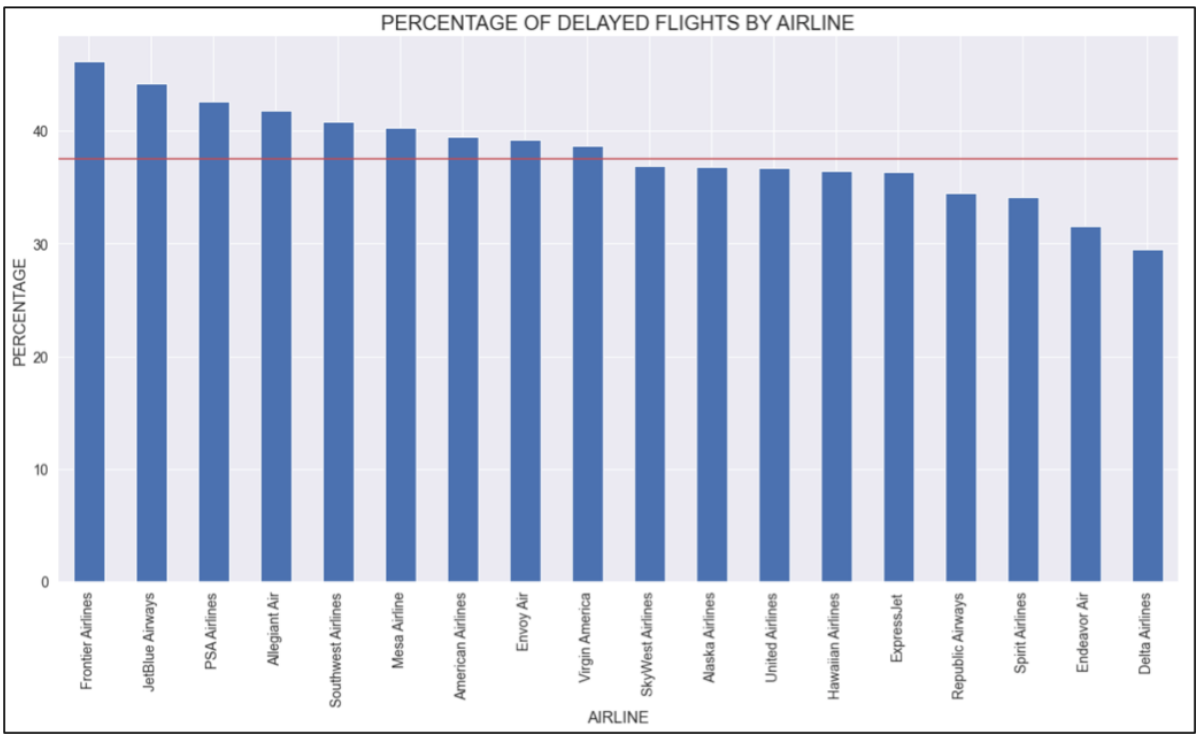


Figure 4. Percentage of delayed flights by airline

Most Popular Destinations with the largest arrival delay: Because there are a total of 358 destination airports within 341 cities, I decided to focus only on the top 30.

Chicago, Atlanta, New York, Dallas-Fort Worth and Denver are the top 5 destination, with Chicago being number 1, but interesting enough it has a pretty high average of annual delays, so if you are traveling to Chicago, there is a high chance that your flight will be delayed. Atlanta in the contrary, is the second most popular destination and with a very low delay at arrivals. New York and Dallas-Fort Worth aren't great, and Denver is just within the average.

Out of the top 15 destinations, the city with the most delays is by far Newark, where you are almost guaranteed to arrive late. Others cities that have very negative records are San Francisco, Orlando, Boston, Philadelphia, Ft. Lauderdale, Tampa and Chantilly.

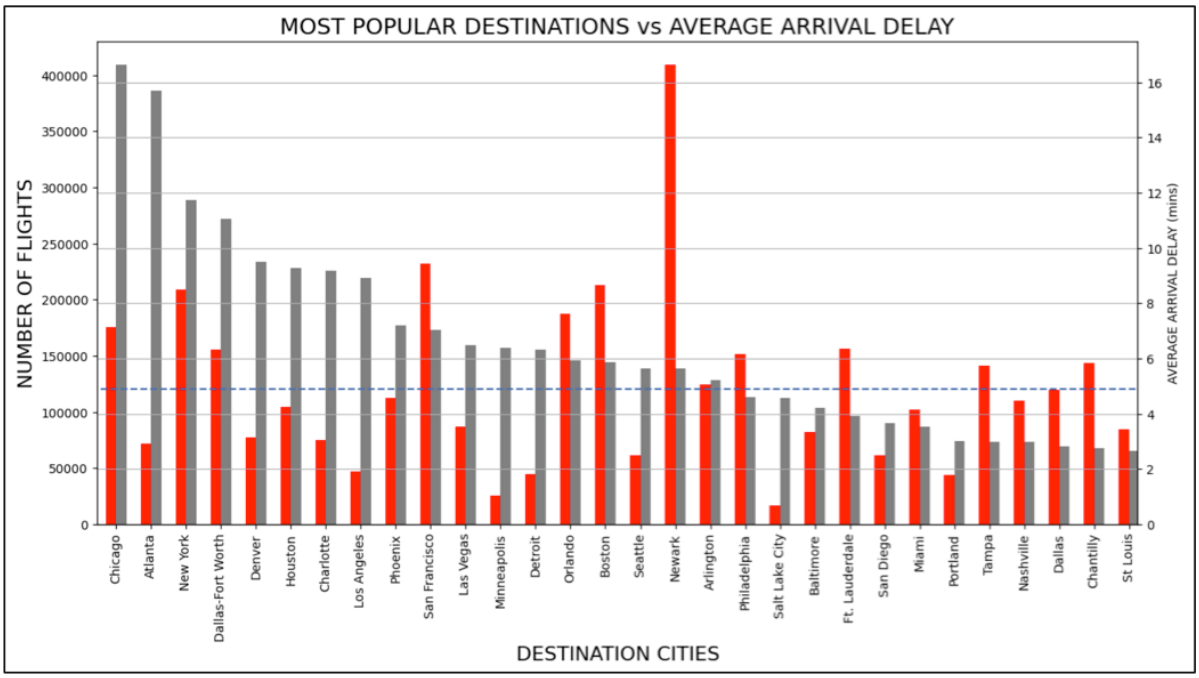


Figure 5. Most popular destinations (cities) with their average arrival delay (min)

Now the plot on Figure 5 compares the most popular destinations again with the average departure delays, with the dashed line being the average. So again, you would want to be below that threshold, but in this case we are talking about cities and multiple airlines at the same time.

If we look at Chicago, we can see that it has quite a high average departure delay, but combining this information with the one from Figure 4, we can infer that flights going to Chicago try to compensate for late departures by reducing the elapse time, and in average it seems as they succeed. With regards to Atlanta, it still is in a good position by being the second most popular destination, with low arrival delay and still with an average delay below the average. I am not sure if this is related to the arrival or departure airports, the weather in this area, or why exactly this happens, and in order to explain it, I would need some additional data which I don't have and that goes beyond the scope of this project anyways, but perhaps is something that can be added later on.

Once again Newark is in bad shape by having the highest average of departures delayed. Orlando and Boston and two others that combined with Figure 4, puts them in bad position. And then you can see the cities which are in pretty bad shape going way above the threshold, such as Philadelphia, Baltimore, Ft. Lauderdale, Miami, Tampa, Nashville and Dallas. Reasons for this? again not enough data nor time to find out.

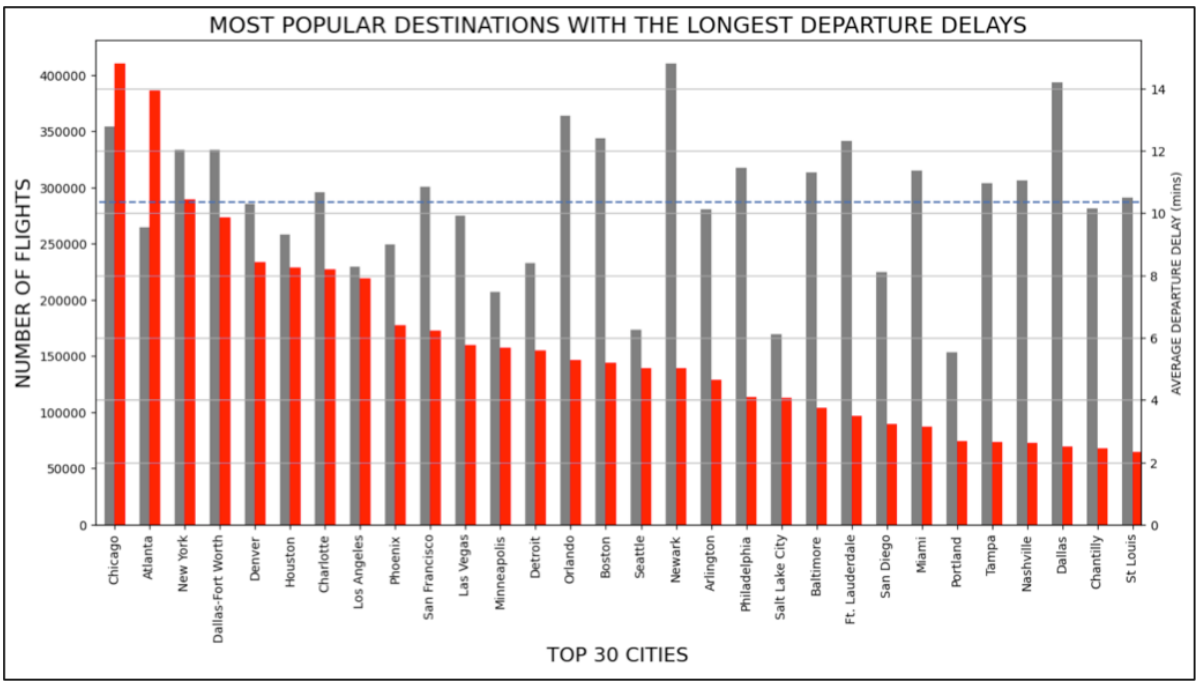


Figure 6. Most popular destinations (cities) with longest average departure delays (min)

Number of destinations by Airlines The plot from Figure 7 is the last one that I will comment on this introductory README. Here you see the number of destinations per airline and once again it's interesting because it shows as highlighted on that plot, that Delta Airlines is the third with most destination. Remember, that it is also top 5 in terms of number of flights, it has the lowest percentage of delayed flights, and it is in negative with regards to the total delayed minutes. It seems as they perform quite well from this pack of 18 airlines so it is the one that I would recommend based on this information for the year 2018. Now this might have changed, I really could say. What I could do and add it later on to this project, is extend the study to all the files cover the 10 years available and that way see if this is a one year trend, or if it is really a historical one, which in that case, it will become more solid to make such a recommendation, but for now I will have to live with what I have.

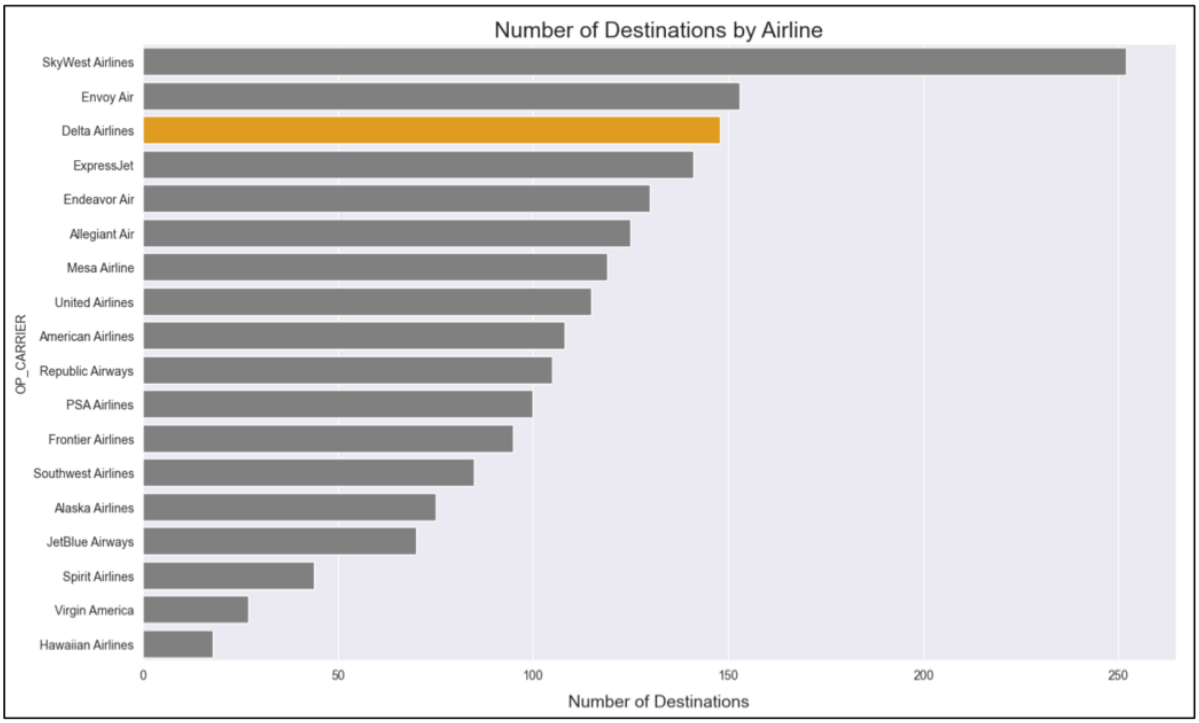


Figure 7. Number of destinations by airline

4-MODELLING

Now that the data has been cleaned and gone through a thorough EDA process done in two stages, its time to start with the modeling which will be a binary classification, where a "0" will correspond to a flight being on time, and a "1" to a flight being delayed.

This dataset consists of 28 features, out of which there are a series of them (listed above) that can affect the predictive model in a positive way in terms of predictions and therefore accuracy. However, when you use them, you are making the assumption that you are most probably already sitting in the plane, or in the best case scenario, your flight status on the departure boards has been changed to: "delayed". This is what the majority of the published models do, so I decided to do something slightly different by limiting the model to only features that won't directly indicate a delay.

For the ML the workflow was pretty straight forward by starting defining the target, which was the FLIGHT_STATUS, and then dropping it alongside the DEP_DELAY from the dataframe to define X (features). With this done, I split the data with a 25 and 75% for the test and training set respectively and used a typical random_State of 42.

This was followed by the next steps:

- Building a Regular Tree as Baseline
- Created Bagged Trees
- Ran a Random Forest with no Class Weighting (ran the feature importance as a QC tool)
- Random Forest with Bootstrat Class Weighting
- Ran a AdaBoost with and without the DEP_DELAY
- RAN Gradient Boosted Trees with and without the DEP_DLAY
- RAN XGBoost with and without the DEP_DELAY

And every model went through a performance evaluation, with the highest accuracy achieved wasn't great at 70% when the DEP_DELAY (departure delay) was dropped, without dropping the DEP_DELAY the highest accuracy obtained was 86, so a hard 16% better with only one feature that suggest a late arrival. Now you can get the

picture of how much the accuracy can improve if I add the rest of these predictive features, certainly it will increase above the 90%.

Figure 8 is a summary of the model evaluations done with and without balancing the data, and with and without the DEP_DELAY feature. Colored in green, is the XGBoost that outperform the other models, and without using any of the features that could biased the model towards a predicted delay.

		Inbalanced data			balanced data		
	Algorithm	Precision	Recall	Accuracy	Precision	Recall	Accuracy
1	Baseline Tree (without DEP_DELAY)	64.00	51.00	63.25	57.00	57.00	57.76
2	Bagged Tress (without DEP_DEPAY(	66.00	51.00	63.33	61.00	52.00	63.68
3	Random Forest (without DEP_DELAY)	74.00	50.00	62.89	57.00	57.00	56.76
4	Random with Bootstrat Weighting	57.00	57.00	56.67	57.00	57.00	57.25
5	AdaBoost_V1 (with DEP_DELAY)	84.00	82.00	81.72			
6	AdaBoost_V2 (without DEP_DELAY)	65.00	54.00	64.64			
7	Gradient Boosted Trees (with DEP_DELAY)	85.00	79.00	83.09			
8	Gradient Boosted Trees (without DEP_DELAY)	70.00	57.00	66.84			
9	XGBoost (with DEP_DELAY)	88.00	82.00	86.00	87.00	83.00	85.65
10	XGBoost (without DEP_DELAY)	71.00	61.00	69.37	69.00	63.00	69.68

Figuree 8. Model Evaluation Summary

The classification report for what I chose to be my best model until this point can be seen on Figure 9 and as you can see the metrics aren't the best but at lest this is predicting the flight delay before you are even on your way to the airport by a good 70%

	precision	recall	f1-score	support
0	0.70	0.90	0.79	1889408
1	0.67	0.36	0.47	1116784
accuracy			0.70	3006192
macro avg	0.69	0.63	0.63	3006192
weighted avg	0.69	0.70	0.67	3006192

Figure 9. XGBoost Classification report of the model with the best performance so

Chapter - 4

FUTURE SCOPE

- Add to the EDA a time of the day analysis, to understand if there is a time more prone to delays than others. Because there are 24 hours a day, maybe make this every 3 hours, ending up with 8 categories. It is known that early and late in the day flights tend to have less delays, so this would be interesting to try to validate
- Do the EDA with the 10 year dataset and not just the 2018. This will require additional cleaning and pre-processing but will definitely give more insight as to the airline performance and hence put me in a position to give a more accurate recommendation
- Re-run the ML and Neural Network model with the best metrics again but with more cities with the objective of adding them into a dash application.

APPLICATIONS

- Apps to predict delays in flight .

Chapter - 6

CONCLUSION

Flights dealys prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data.

Data cleaning is one of the processes that increases prediction performance, yet insufficient for the cases of complex data sets as the one in this research. Applying single machine algorithm on the data set accuracy was less than 50%. Therefore, the ensemble of multiple machine learning algorithms has been proposed and this combination of ML methods gains accuracy of 83.71%.

This is significant improvement compared to single machine learning method approach. However, the drawback of the proposed system is that it consumes much more computational resources than single machine learning algorithm. Although, this system has achieved astonishing performance in flight delays prediction problem our aim for the future research is to test this system to work successfully with 10 years of dataset and use pyspark for data cleaning.

BIBLIOGRAPHY

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In Proceedings of the International Workshop of Ambient Assited Living, 2012.

[2] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.