Flight Prediction Project: Final Report

Authors: Dae Hyun Chung, Xinyao Liu, Brittany-Lauren Todd

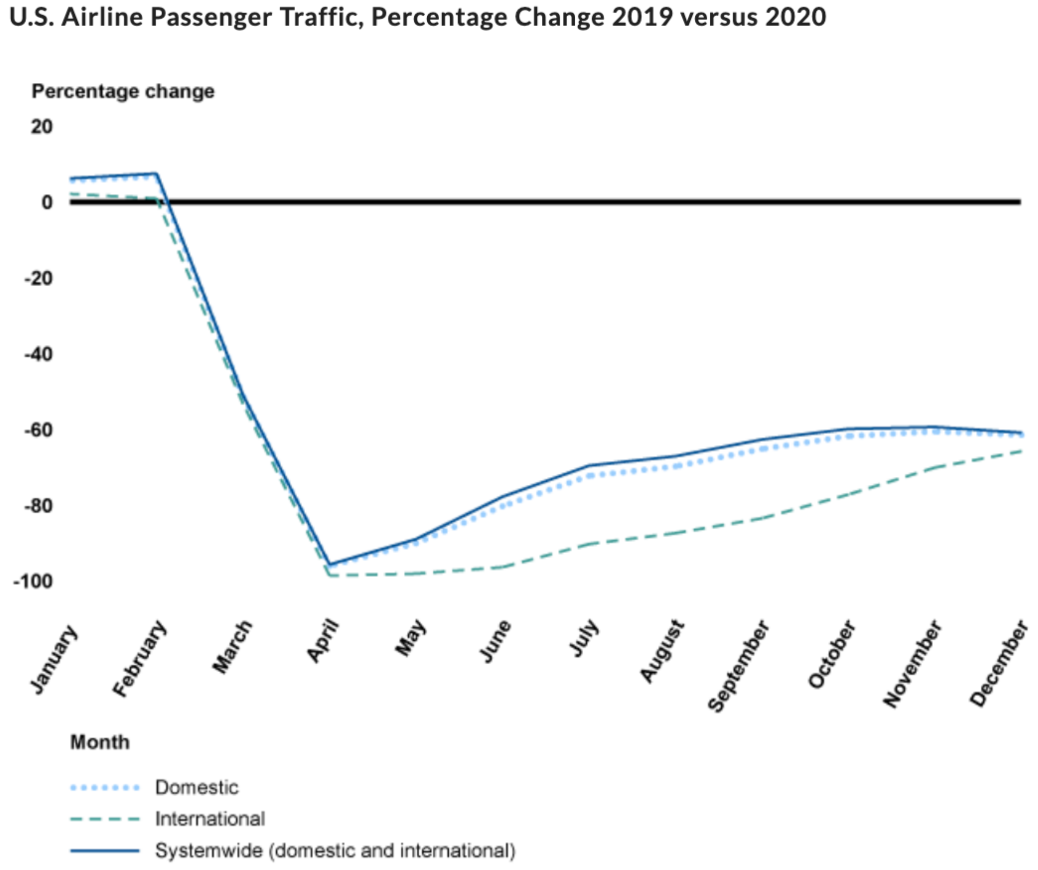
CS 6220 Data Mining Techniques, Fall 2022

Credit: Stock Photo

Introduction

1.1 COVID-19's Impact on the Airline Industry

The COVID-19 pandemic exacerbated flight delays and cancellations in record numbers, pushing both travelers and airline workers to the brink. These disruptions fractured customer relations and made the typically straightforward purchase of airline tickets an uncertain and daunting task. According to the U.S. Government Accountability Office (GAO), airline travel was cut by 60% in the year 2020 compared to 2019 (GAO). As a result, the airline industry collectively had an "after-tax net loss of over $43 billion" in 2020 (BTS).


U.S. Airline Passenger Traffic, Percentage Change 2019 versus 2020

Source: GAO analysis of Department of Transportation Bureau of Transportation Statistics data. | GAO-22-104429

Figure 1: Displays the drastic drop in domestic and international flights from January 2020 to April 2020. After April 2020, airline traffic begins to increase but does not make a full recovery by the end of the year. Credit: (GAO)

1.2 Project Goals

Being flyers ourselves, we sought to create an API that would benefit both travelers and airline companies. Our interface allows individuals to enter the details of a single U.S. domestic flight, and in return receive an estimate of the likelihood of the flight being delayed. We hope our site will help individuals feel confident about their flight ticket selections. Consequentially, the airline industry could regain the trust of its customers, which in turn could greatly increase its profits.

1.3 Applications in Data Science

Our flight prediction API is a useful application of data science because it allows us to take copious amounts of airline data from the years 2018-2022 and make future predictions on flight delays. The use of data science within the airline industry can lead to increased customer retention rates, as well as changes in airline procedures and policies. For example, if users learn which airports typically have the most delays, they might choose to change their destination or origin airport. Additionally, if they are aware of which flights are likely to be delayed, they might be able to plan ahead by opting for flight protection, booking a backup flight, or switching their travel plans to off-peak season. Similarly, airlines might use data science insights to increase staff (pilots, flight attendants, and customer service staff) on certain days or at certain airport locations.

Data Collection

2.1 The Data Set

After scouring multiple data sets we settled on the Kaggle data set titled: “Flight Status Prediction” by Rob Mulla. We chose this dataset for its relevant features and its high usability score of 10. Additionally, this dataset provided CSV files for the years 2018-2022. We knew that to reliably predict the status of future flights, we would need to train our machine-learning model with various years of robust flight data.

For our machine learning model to efficiently make a flight status prediction, we were very intentional in focusing on a set number of features. The original dataset contained 61 features, ranging from “FlightDate” to “DivertedAirportLandings”. For each year's dataset, we narrowed the number of features to 16. These features included “Year”, “FlightDate”, “Airline”, “Origin”, “Dest”, “Canceled”, “DepTime”, “DepDelayMinutes”, “DepDel15”, “DepartureDelayGroups”, “DepTimeBlk”, “Diverted”, “DayOfWeek”, “DayofMonth”, “Month” and “AirTime”. Moreover, this sample size of features was small enough to prevent our Google Colab notebook from crashing when loading our datasets but large enough to create useful visualizations.

2.1 Data Preprocessing

To train and test our model, we needed to do some pre-processing on our sample dataset. The first step we took was to drop all missing values from each year's CSV file. Next, we decided to combine all of our subset features from each year into a single data frame called “df_multiple”. Although a single data frame was a lot easier to work with, it was still too large for our Colab notebook to handle. As a result, we decided to get a random sample of 1,000,000 rows called “df_multiple_sample”. At this stage, we were able to create useful visualizations and train our models.

VISUALIZATIONS

3.1 Creating Flyer-Friendly Visuals

Since the goal of our API is to benefit both customers and airline companies, it was necessary to create straightforward and useful visualizations. These visualizations will act in conjunction with the flight prediction service, and provide extra insight into flight trends between the years 2018- 2022.

3.2 Dataset size

As mentioned above, the dataset we used for analysis was incredibly large. For each year from 2018-2022, there are around 5 million lines of flight records and 61 columns of features. During the preprocessing stage, we selected 13 columns of features for all datasets and combined all five years of data into one data frame titled “df_mulitple”.

Table 1: A summary of each year’s dataset

no. of flights	no. of features	no. of airlines	no. of origin/dest
~5 million	61	29	>300

Figure 3.1 shows a bar chart summary of the number of flights in each year. We can see that 2019 has the most number of flights, around 8 million. Figure 3.2 shows a bar chart summary of the number of delayed flights in each year and indubitably, 2019 has the most delays. Figure 3.3 shows a bar chart summary of the number of canceled flights for the years 2018-2022, with 2020 having the highest cancellation rate. This finding is unsurprising since 2020 was the start of the global pandemic. Figure 3.4 shows the bar chart summary of the number and percentage of

delayed flights for the years 2018-2022. Surprisingly, 2022 has the most percentage of delays, even though the data was only collected up to July 31st, 2022.

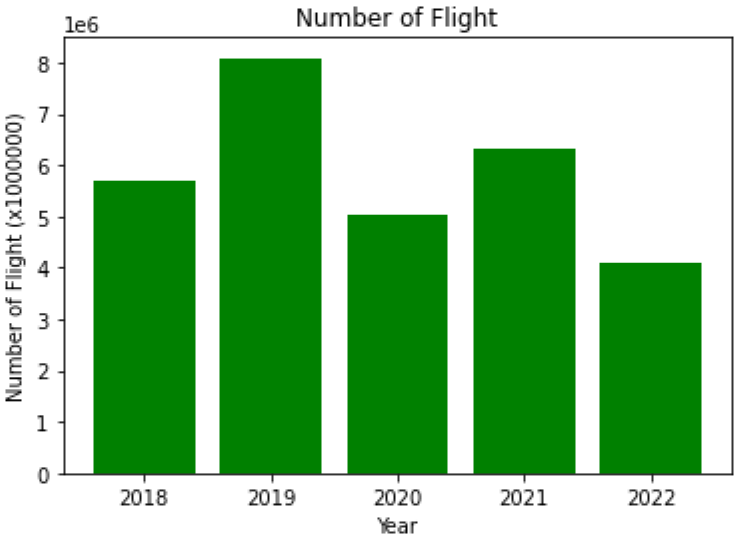


Figure 3.1: Number of flights in each dataset from 2018-2022. Note 2022's dataset only contains flight records up to July 31st, 2022

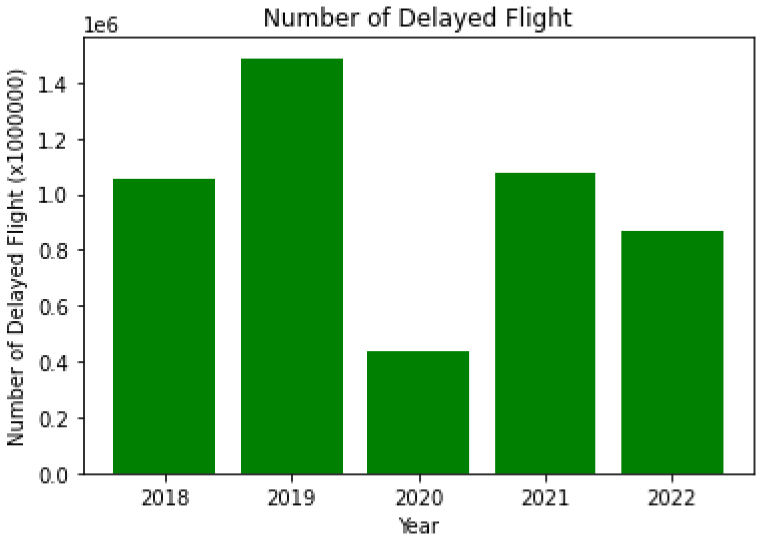


Figure 3.2: Number of Delayed flights in each dataset from 2018-2022.

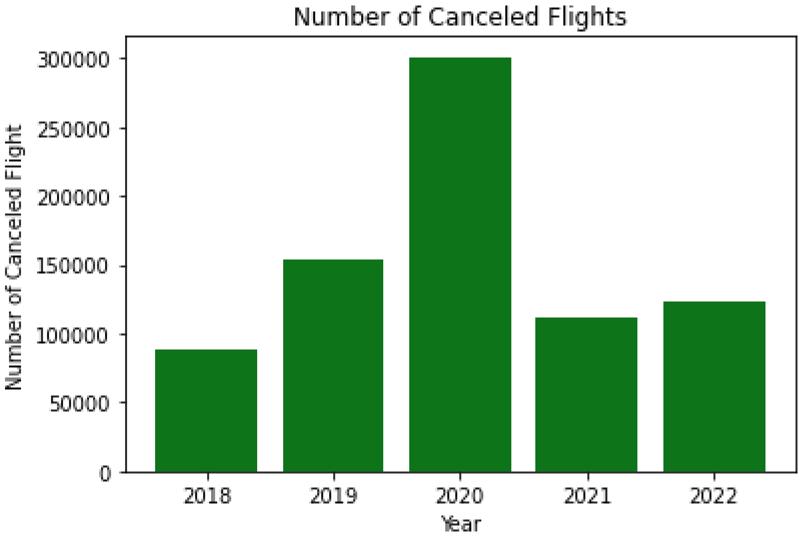


Figure 3.3: Number of Canceled flights in each dataset from 2018-2022.

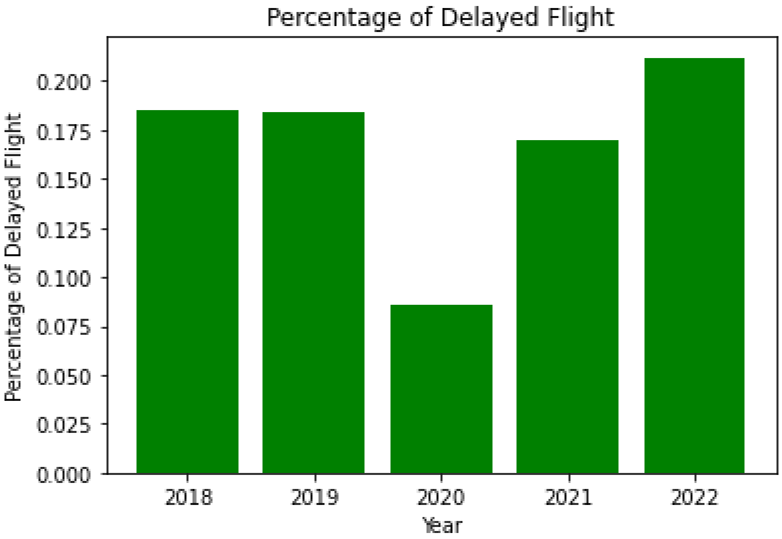


Figure 3.4: Percentage of delayed flights in each dataset from 2018-2022.

We counted the number of departures at each airport using 2019's data and made Figure 3.5 to visualize the distribution of busy airports in the US. We can see that airports on the west and east coast, as well as around the Great Lakes, Texas and the Hawaiian Islands have a high number of flight departures compared to inland airports.

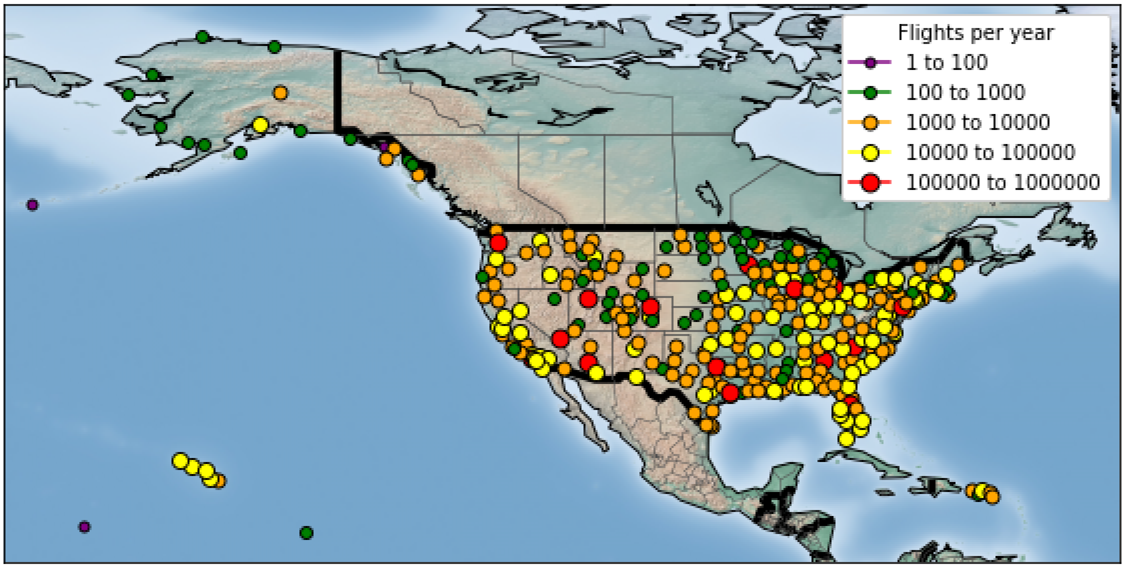
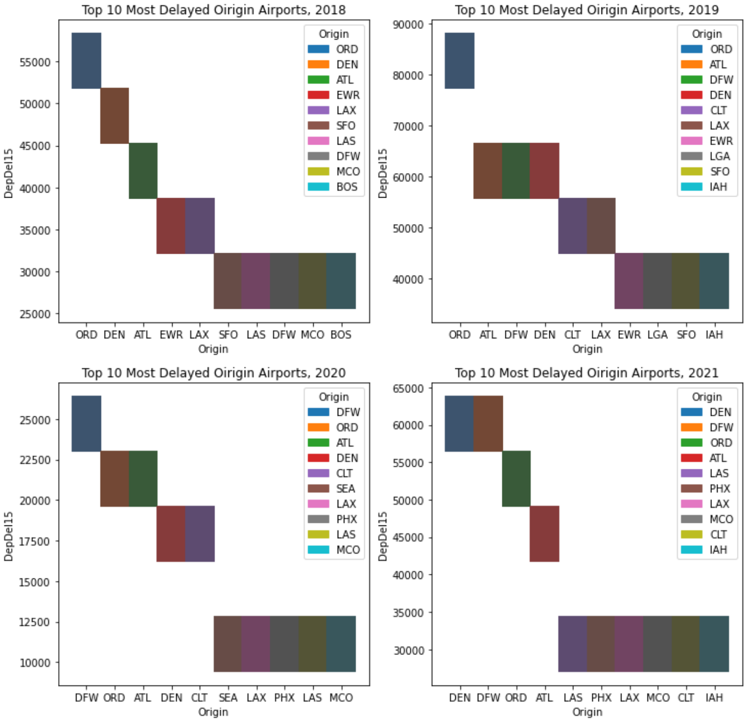


Figure 3.5: Distribution of busy US domestic airports plotted based on 2019's dataset

Next, we counted the top 10 most delayed origin airports in the year 2018-2021 as shown in Figure 3.6. The location of these airports matches the distribution of those busy airports.


Top 10 Most Delayed Oirigin Airports, 2018

Top 10 Most Delayed Oirigin Airports, 2019

Top 10 Most Delayed Oirigin Airports, 2020

Top 10 Most Delayed Oirigin Airports, 2021

Figure 3.6: Top 10 most delayed origin airports from the years 2018-2021

Finally, we collected the information for each airline. Figure 3.7 shows the market share of each airline. Figure 3.8 shows the number of on-time (< 60 minutes of delay), small delay (60-120 minutes of delay), and large delay (>120 minutes) flights for each airline. Figure 3.9 shows the accumulated delay minutes for each airline. We can see that since Southwest Airlines has the most market shares, it also has the most on-time flights and the highest accumulated delay minutes. Each airline has approximately the same ratio of on-time, small delay, and large delay flights. This also implied that our dataset is highly imbalanced, where the majority of flights are on-time.

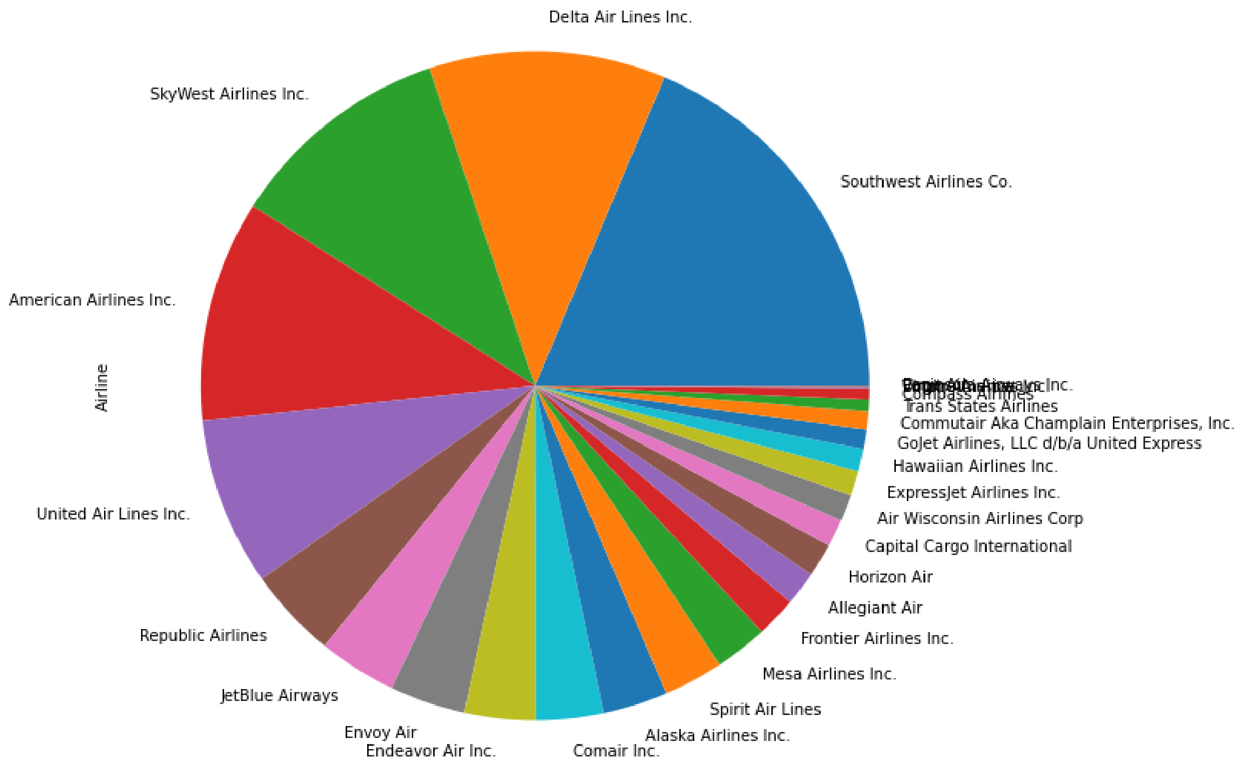


Figure 3.7: Percentage of flights belonging to each airline in our combined dataset

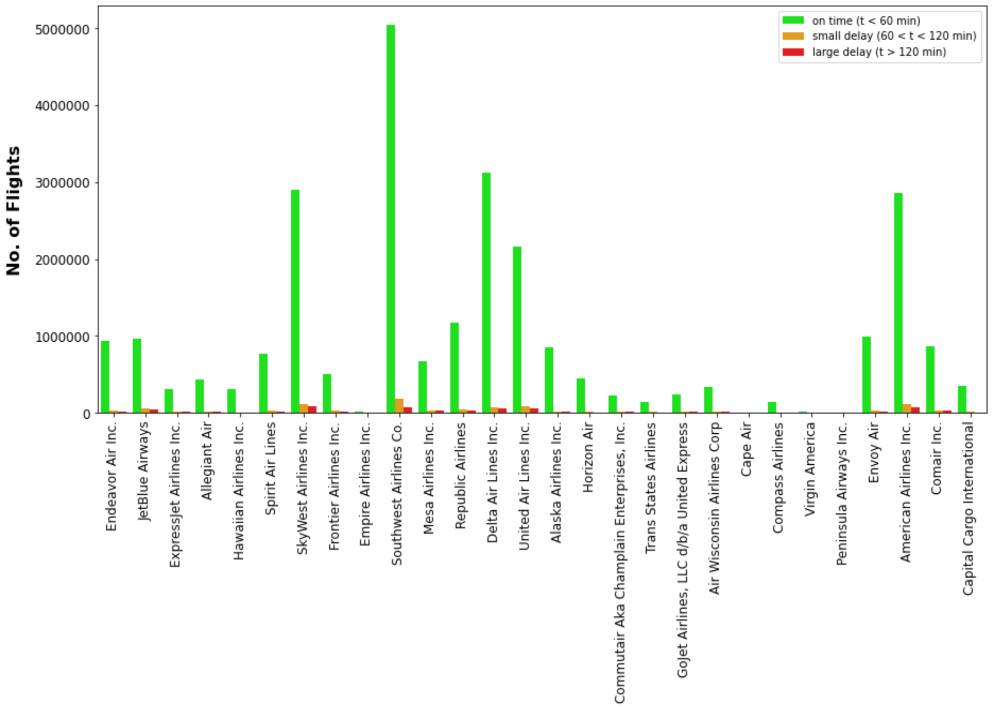


Figure 3.8: Number of on-time, small delay, and large delay flights for each airline

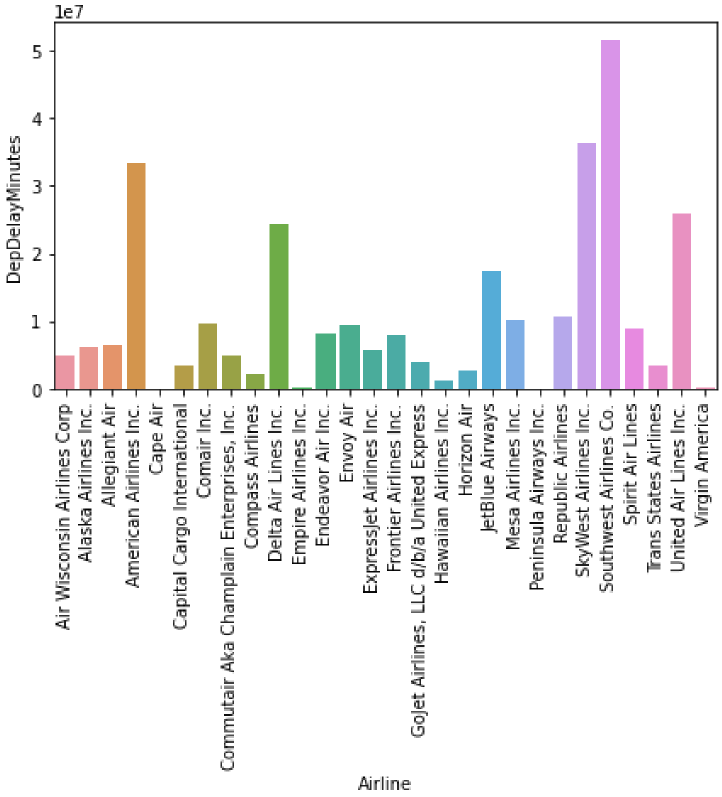
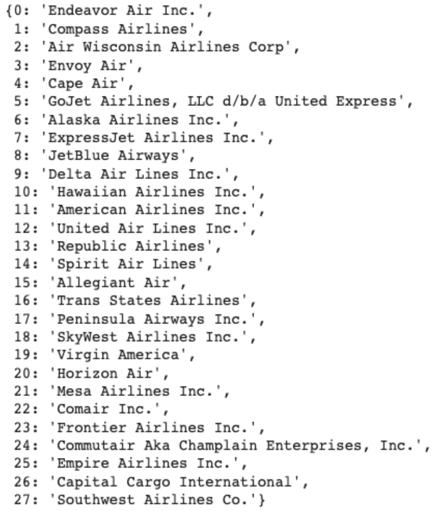


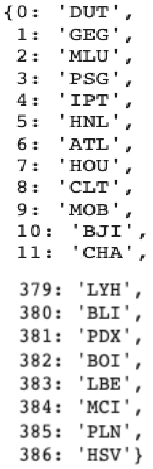
Figure 3.9: Accumulated delay minutes for each airline

4.1 The Process of Choosing, Training, and Testing Our Model

Before we could consider the appropriate features for the training variable, we had to do more preprocessing. First, we dropped any features deemed unnecessary, then we had to change our string features of Airline, Origin, and Dest to integers so that they could be accepted by our model. To change these features from categorical to numeric, we tried dummy encoding but there were too many cases in each feature(27 in Airline and 386 in each Origin and Dest) resulting in errors. Next, we tried one-hot encoding which was also unsuccessful. Thus, we decided to do label encoding ourselves and create new features titled, “Airlines”, “Origin”s and “Destinations”.



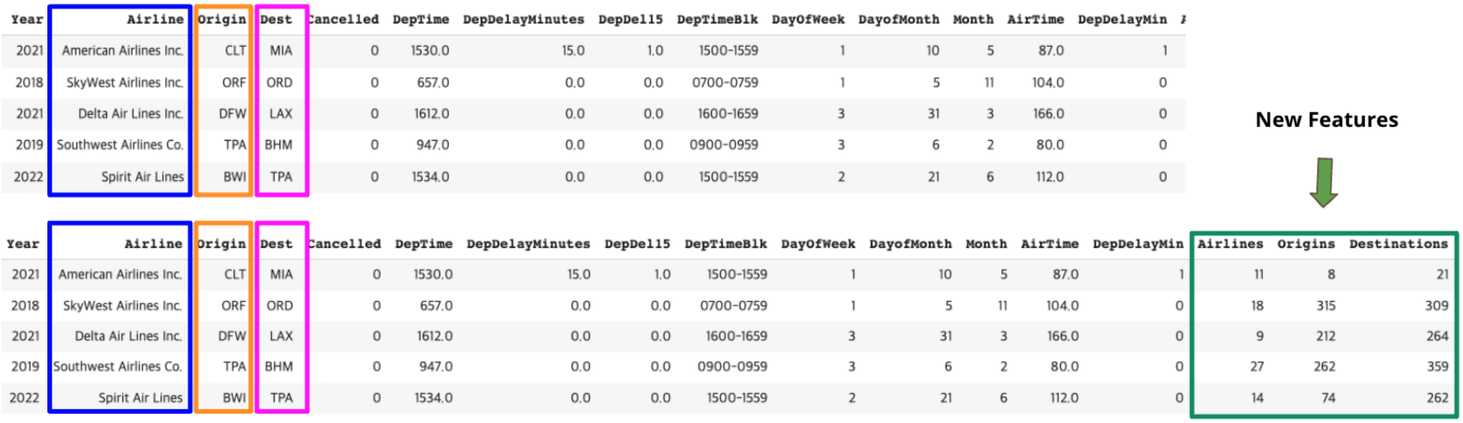
```
{0: 'Endeavor Air Inc.',
 1: 'Compass Airlines',
 2: 'Air Wisconsin Airlines Corp',
 3: 'Envoy Air',
 4: 'Cape Air',
 5: 'GoJet Airlines, LLC d/b/a United Express',
 6: 'Alaska Airlines Inc.',
 7: 'ExpressJet Airlines Inc.',
 8: 'JetBlue Airways',
 9: 'Delta Air Lines Inc.',
 10: 'Hawaiian Airlines Inc.',
 11: 'American Airlines Inc.',
 12: 'United Air Lines Inc.',
 13: 'Republic Airlines',
 14: 'Spirit Air Lines',
 15: 'Allegiant Air',
 16: 'Trans States Airlines',
 17: 'Peninsula Airways Inc.',
 18: 'SkyWest Airlines Inc.',
 19: 'Virgin America',
 20: 'Horizon Air',
 21: 'Mesa Airlines Inc.',
 22: 'Comair Inc.',
 23: 'Frontier Airlines Inc.',
 24: 'Commutair Aka Champlain Enterprises, Inc.',
 25: 'Empire Airlines Inc.',
 26: 'Capital Cargo International',
 27: 'Southwest Airlines Co.'}
```



```
{0: 'DUT',
 1: 'GEG',
 2: 'MLU',
 3: 'PSG',
 4: 'IPT',
 5: 'HNL',
 6: 'ATL',
 7: 'HOU',
 8: 'CLT',
 9: 'MOB',
 10: 'BJI',
 11: 'CHA',

 379: 'LYH',
 380: 'BLI',
 381: 'PDX',
 382: 'BOI',
 383: 'LBE',
 384: 'MCI',
 385: 'PLN',
 386: 'HSV'}
```

Figures 4.1 Labeling Airline

Figures 4.1 Labeling Origins and Dest



Year	Airline	Origin	Dest	Cancelled	DepTime	DepDelayMinutes	DepDel15	DepTimeBlk	DayOfWeek	DayofMonth	Month	AirTime	DepDelayMin
2021	American Airlines Inc.	CLT	MIA	0	1530.0	15.0	1.0	1500-1559	1	10	5	87.0	1
2018	SkyWest Airlines Inc.	ORF	ORD	0	657.0	0.0	0.0	0700-0759	1	5	11	104.0	0
2021	Delta Air Lines Inc.	DFW	LAX	0	1612.0	0.0	0.0	1600-1659	3	31	3	166.0	0
2019	Southwest Airlines Co.	TPA	BHM	0	947.0	0.0	0.0	0900-0959	3	6	2	80.0	0
2022	Spirit Air Lines	BWI	TPA	0	1534.0	0.0	0.0	1500-1559	2	21	6	112.0	0

Year	Airline	Origin	Dest	Cancelled	DepTime	DepDelayMinutes	DepDel15	DepTimeBlk	DayOfWeek	DayofMonth	Month	AirTime	DepDelayMin	Airlines	Origins	Destinations
2021	American Airlines Inc.	CLT	MIA	0	1530.0	15.0	1.0	1500-1559	1	10	5	87.0	1	11	8	21
2018	SkyWest Airlines Inc.	ORF	ORD	0	657.0	0.0	0.0	0700-0759	1	5	11	104.0	0	18	315	309
2021	Delta Air Lines Inc.	DFW	LAX	0	1612.0	0.0	0.0	1600-1659	3	31	3	166.0	0	9	212	264
2019	Southwest Airlines Co.	TPA	BHM	0	947.0	0.0	0.0	0900-0959	3	6	2	80.0	0	27	262	359
2022	Spirit Air Lines	BWI	TPA	0	1534.0	0.0	0.0	1500-1559	2	21	6	112.0	0	14	74	262

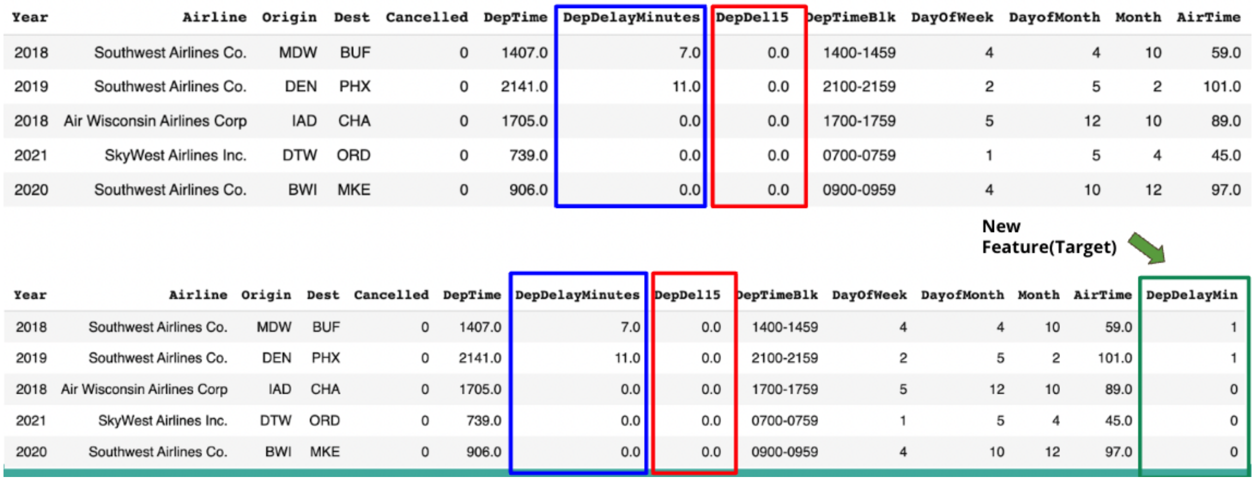
Figures 4.3 Label Encoding Airline, Origin, and Dest

After completing the previous step, we chose DepTime, DepDelayMinutes, DayOfWeek, DayOfMonth, Month, Airlines, Origins, and Destinations as our independent "X" training variables. We selected these features because they had a lot of influence on our predictions and would be the most useful to customers. During the process of choosing the target value, the categorical feature for the "Y" testing variable was the single feature DepDel15. DepDel15 is binary and is represented as 0 or 1 if the flight is on-time or delayed for 15 minutes or more.

Since we are trying to predict a delayed period instead of a binary outcome, DepDel15 is not important for our prediction. Consequently, we needed to make new features for the target variable. Our data file provides DepDelayMinutes which is departure delay minutes and with that feature, we made DepDelayMin which is a categorical feature that labels the numbers according to delay minutes.

Condition:

1. If DepDelayMinutes is 0, then the label is 0
2. If DepDelayMinutes is less than 15, then the label is 1
3. If DepDelayMinutes is more than 15 but less than 30, then the label is 2
4. If DepDelayMinutes is more than 30 but less than 60, then the label is 3
5. Lastly, if DepDelayMinutes is more than 60 minutes, the label is 4.



Year	Airline	Origin	Dest	Cancelled	DepTime	DepDelayMinutes	DepDel15	DepTimeBlk	DayOfWeek	DayofMonth	Month	AirTime
2018	Southwest Airlines Co.	MDW	BUF	0	1407.0	7.0	0.0	1400-1459	4	4	10	59.0
2019	Southwest Airlines Co.	DEN	PHX	0	2141.0	11.0	0.0	2100-2159	2	5	2	101.0
2018	Air Wisconsin Airlines Corp	IAD	CHA	0	1705.0	0.0	0.0	1700-1759	5	12	10	89.0
2021	SkyWest Airlines Inc.	DTW	ORD	0	739.0	0.0	0.0	0700-0759	1	5	4	45.0
2020	Southwest Airlines Co.	BWI	MKE	0	906.0	0.0	0.0	0900-0959	4	10	12	97.0

New Feature(Target)

Year	Airline	Origin	Dest	Cancelled	DepTime	DepDelayMinutes	DepDel15	DepTimeBlk	DayOfWeek	DayofMonth	Month	AirTime	DepDelayMin
2018	Southwest Airlines Co.	MDW	BUF	0	1407.0	7.0	0.0	1400-1459	4	4	10	59.0	1
2019	Southwest Airlines Co.	DEN	PHX	0	2141.0	11.0	0.0	2100-2159	2	5	2	101.0	1
2018	Air Wisconsin Airlines Corp	IAD	CHA	0	1705.0	0.0	0.0	1700-1759	5	12	10	89.0	0
2021	SkyWest Airlines Inc.	DTW	ORD	0	739.0	0.0	0.0	0700-0759	1	5	4	45.0	0
2020	Southwest Airlines Co.	BWI	MKE	0	906.0	0.0	0.0	0900-0959	4	10	12	97.0	0

Figures 4.4 Label Encoding DepDelayMinutes to DepDelayMin

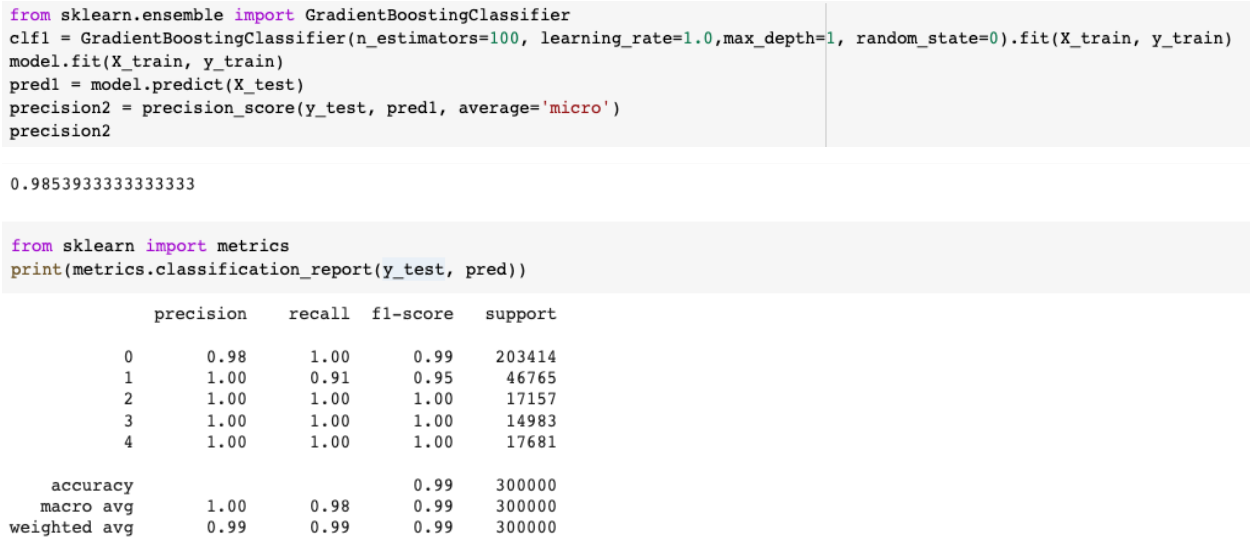
If a flight is delayed more than 60 minutes, we assumed that more than likely the customer will not want to make the selected flight purchase, so we set the maximum delay minutes to 60. With this condition, DepDelayMin has a label value of 0 to 4. The higher the number of DepDelayMin the longer the flight delay. 8 features of "X" training variables and 1 feature of "Y" testing variables were used for our prediction model.

4.2 Initial Model Selections and Precision Metric

To find the correct machine-learning model for the back-end of our Flight Predictor API, we trained our datasets on 3 different classification models using the scikit-learn package. We chose to focus on classification models because we wanted our flight prediction output to be a range of delayed minutes, not just the binary output, of either “delayed” or “not delayed”. Initially, we decided to use accuracy as a means for measuring the performance of our machine-learning models but switched to measuring with precision on our chosen model. Our models included Gradient Boost Classifier, Decision Tree Classifier, and Logistic Regression.

Additionally, our flight delay prediction wasn’t focusing on how many times the model was correct overall. Avoiding the accuracy metrics, we decided to use Precision since we are focusing on how good the model is at predicting a specific condition.

Gradient Boost Classifier functions by combining “many weak learning models together to create a strong predictive model” (Nelson). The results of using the Gradient Boost Classifier gave us a precision score of 98.54%.



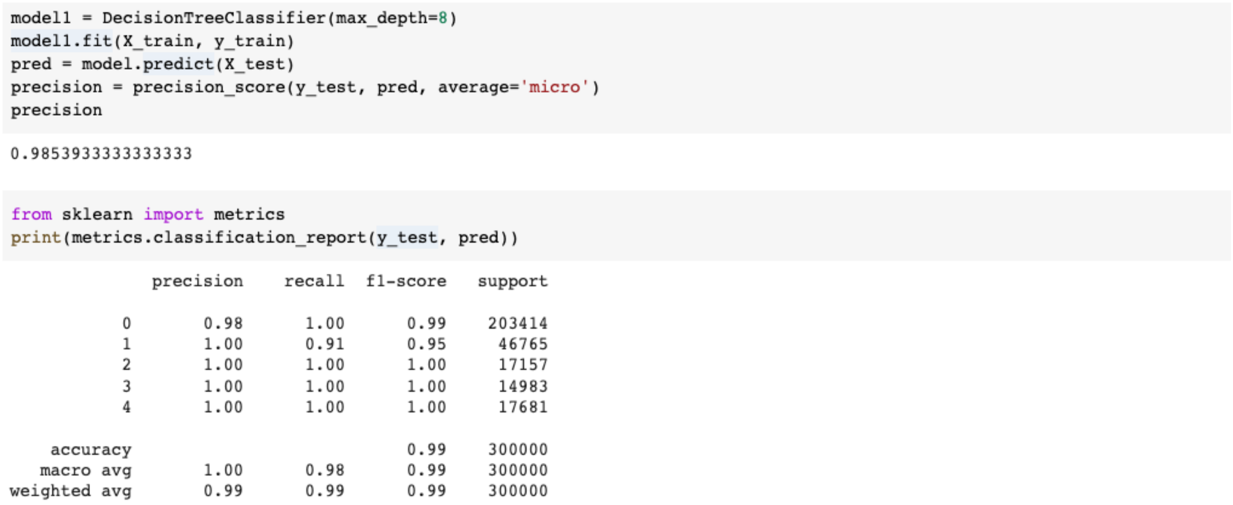
```
from sklearn.ensemble import GradientBoostingClassifier
clf1 = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,max_depth=1, random_state=0).fit(X_train, y_train)
model.fit(X_train, y_train)
pred1 = model.predict(X_test)
precision2 = precision_score(y_test, pred1, average='micro')
precision2
```

```
0.9853933333333333
```

```
from sklearn import metrics
print(metrics.classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99    203414
           1       1.00      0.91      0.95     46765
           2       1.00      1.00      1.00     17157
           3       1.00      1.00      1.00     14983
           4       1.00      1.00      1.00     17681

    accuracy                           0.99    300000
   macro avg       1.00      0.98      0.99    300000
weighted avg       0.99      0.99      0.99    300000
```

Next, we tested the Decision Tree Classifier. This algorithm is sometimes preferred for its “ability to handle numerical and categorical data”, and its reduced need for data normalization (scikit-learn.org). This model gave us a precision score of 98.54%.



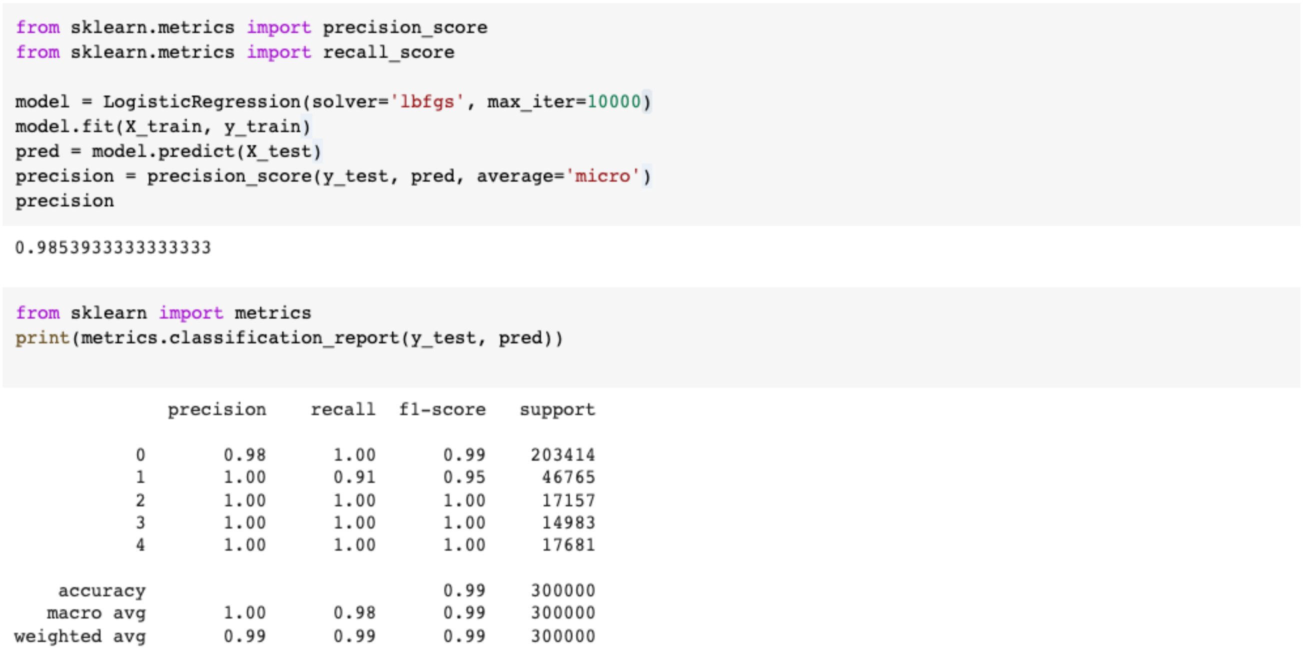
```
model1 = DecisionTreeClassifier(max_depth=8)
model1.fit(X_train, y_train)
pred = model.predict(X_test)
precision = precision_score(y_test, pred, average='micro')
precision
```

```
0.9853933333333333
```

```
from sklearn import metrics
print(metrics.classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99    203414
           1       1.00      0.91      0.95     46765
           2       1.00      1.00      1.00     17157
           3       1.00      1.00      1.00     14983
           4       1.00      1.00      1.00     17681

    accuracy                           0.99    300000
   macro avg       1.00      0.98      0.99    300000
weighted avg       0.99      0.99      0.99    300000
```

Our last model, Logistic Regression, was our chosen model, for its simplicity, and high precision score, which we ultimately used as our measurement of reliability. Like other models, this model also gave us the same precision score of 98.54%. We also obtained the recall score of three models, all of which were equally high at 98.54%



```
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

model = LogisticRegression(solver='lbfgs', max_iter=10000)
model.fit(X_train, y_train)
pred = model.predict(X_test)
precision = precision_score(y_test, pred, average='micro')
precision
```

```
0.9853933333333333
```

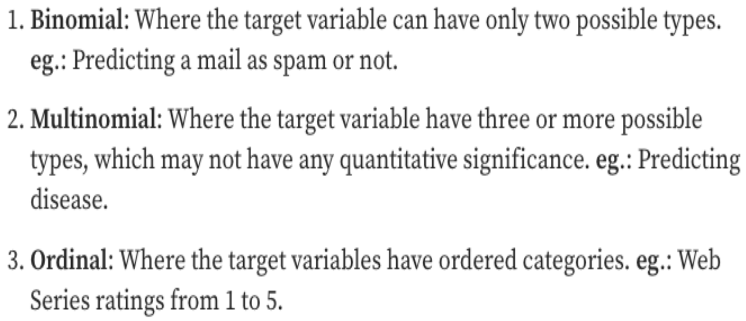
```
from sklearn import metrics
print(metrics.classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99    203414
           1       1.00      0.91      0.95     46765
           2       1.00      1.00      1.00     17157
           3       1.00      1.00      1.00     14983
           4       1.00      1.00      1.00     17681

    accuracy                           0.99    300000
   macro avg       1.00      0.98      0.99    300000
weighted avg       0.99      0.99      0.99    300000
```

Chosen Model: Logistic Regression

5.1 What is Logistic Regression and Why We Chose It

Although titled Logistic Regression, this algorithm is a classification model. Classification is a technique used to analyze a dataset in which there are one or more independent variables that determine an outcome. It uses a logistic function to model the dependent variable and is often used when dealing with binary data. However, it can be extended and further classified into three different types. Since our target variable DepDelayMin has 4 possible types, we used the multinomial Logistic Regression. Not only was its precision the highest score, but it was also easier to implement, and interpret and was efficient to train, compared to other models.



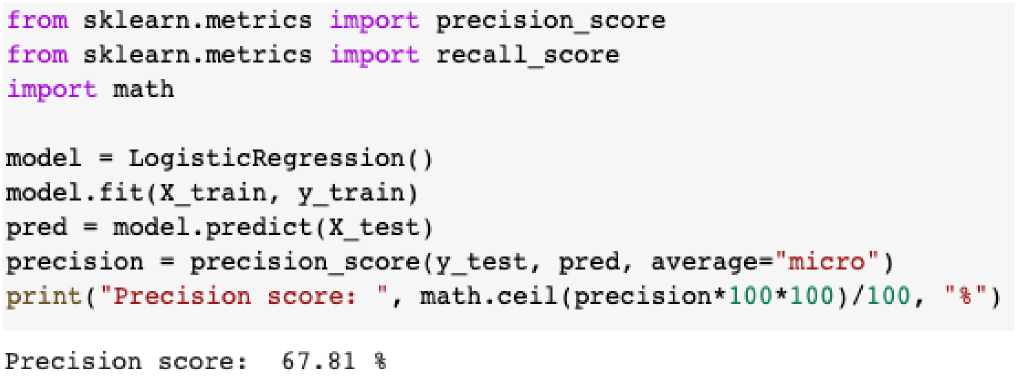
1. **Binomial:** Where the target variable can have only two possible types. **eg.**: Predicting a mail as spam or not.
2. **Multinomial:** Where the target variable have three or more possible types, which may not have any quantitative significance. **eg.**: Predicting disease.
3. **Ordinal:** Where the target variables have ordered categories. **eg.**: Web Series ratings from 1 to 5.

Figure 5.1 Types of Logistic Regression

5.1 Our Results and Modification

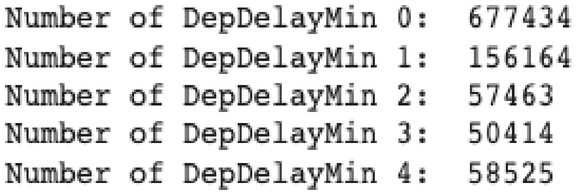
Using Logistic Regression and 98% precision, we created our API where customers could select their desired inputs. Customers could choose DepTime, DepDelayMinutes, DayOfWeek, DayOfMonth, Month, Airlines, Origins, and Destinations to predict their flight delay. However, we realized that users will not know their flight DepDelayMinutes. We, therefore, had no choice but to remove DepDelayMinutes from the "X" training variable, resulting in a total of 7 features instead of 8.

We expected no significant changes from reducing our feature by only one, but our expectation was wrong. Our precision score of 98.54% drastically dropped to 67.81%. DepDelayMinutes affected 21% of the decrease.



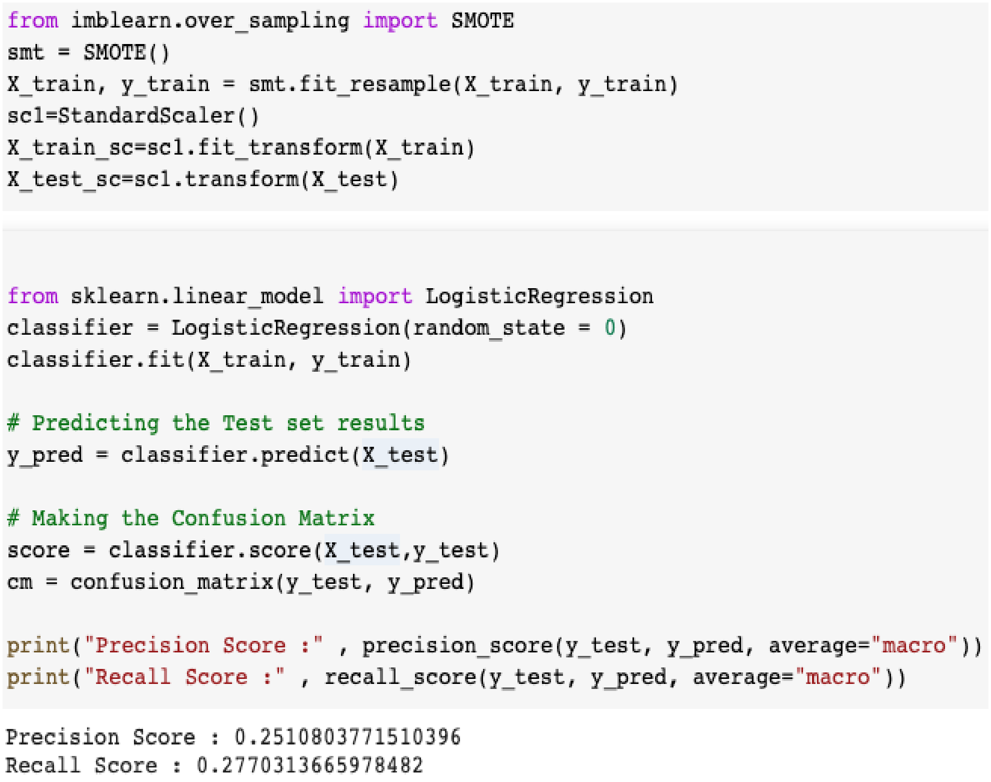
```python
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
import math

model = LogisticRegression()
model.fit(X_train, y_train)
pred = model.predict(X_test)
precision = precision_score(y_test, pred, average="micro")
print("Precision score: ", math.ceil(precision*100*100)/100, "%")
```

```
Precision score:  67.81 %
```



```
Number of DepDelayMin 0:  677434
Number of DepDelayMin 1:  156164
Number of DepDelayMin 2:  57463
Number of DepDelayMin 3:  50414
Number of DepDelayMin 4:  58525
```

Without using DepDelayMinutes, we had to increase the precision score. Unfortunately, there weren't any features that we can include in the training variable since the other feature options are unknown to customers for user input. Comparing the number of each label of DepDelayMin, DepDelayMin with 0 had almost 12 times of others. Because of this, we thought that our dataset was imbalanced, so we tried using SMOTE to improve the model performance, even though it generally increases recall at the cost of lower precision. Unfortunately, both precision and recall decreased to 0.25 and 0.27, so we selected to not use SMOTE.



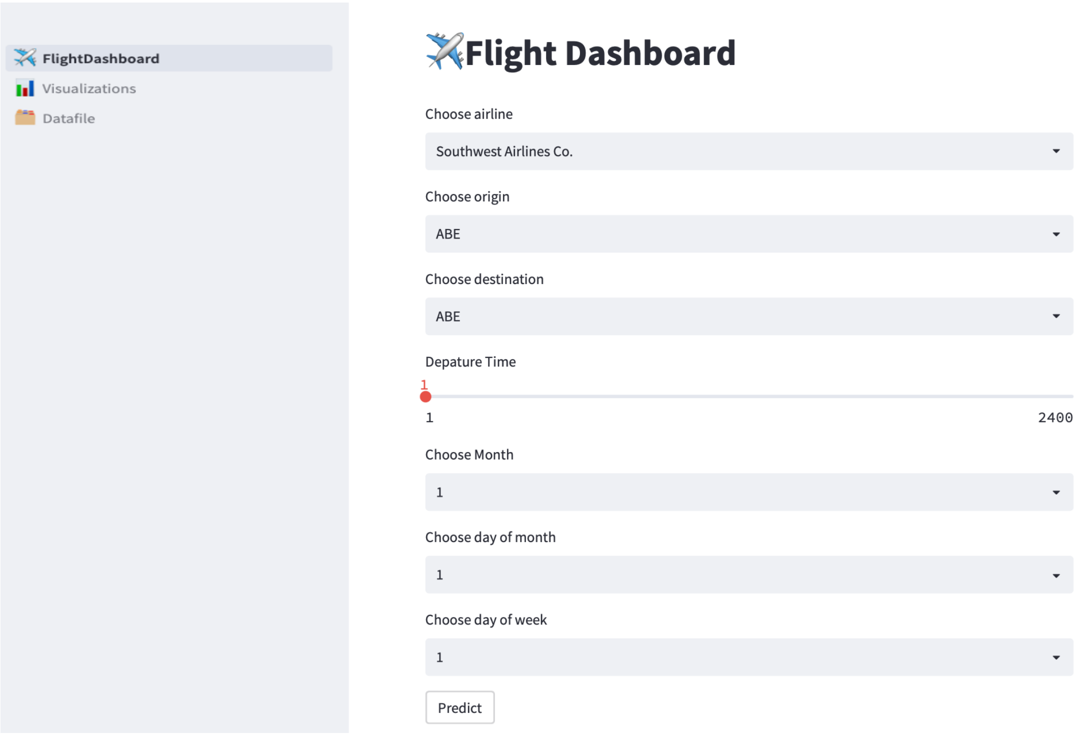
```
from imblearn.over_sampling import SMOTE
smt = SMOTE()
X_train, y_train = smt.fit_resample(X_train, y_train)
sc1=StandardScaler()
X_train_sc=sc1.fit_transform(X_train)
X_test_sc=sc1.transform(X_test)
```

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
score = classifier.score(X_test,y_test)
cm = confusion_matrix(y_test, y_pred)

print("Precision Score :" , precision_score(y_test, y_pred, average="macro"))
print("Recall Score :" , recall_score(y_test, y_pred, average="macro"))
```

```
Precision Score : 0.2510803771510396
Recall Score : 0.2770313665978482
```

We tried several different techniques other than SMOTE, but nothing increased the precision. Therefore, we decided to use the previous logistic regression with 67.81% even though it has low precision. With this, we created our API for customers to predict their flights using Streamlit.

Streamlit: Making our API



✈️Flight Dashboard

Choose airline: Southwest Airlines Co.

Choose origin: ABE

Choose destination: ABE

Depature Time: 1 (1 – 2400)

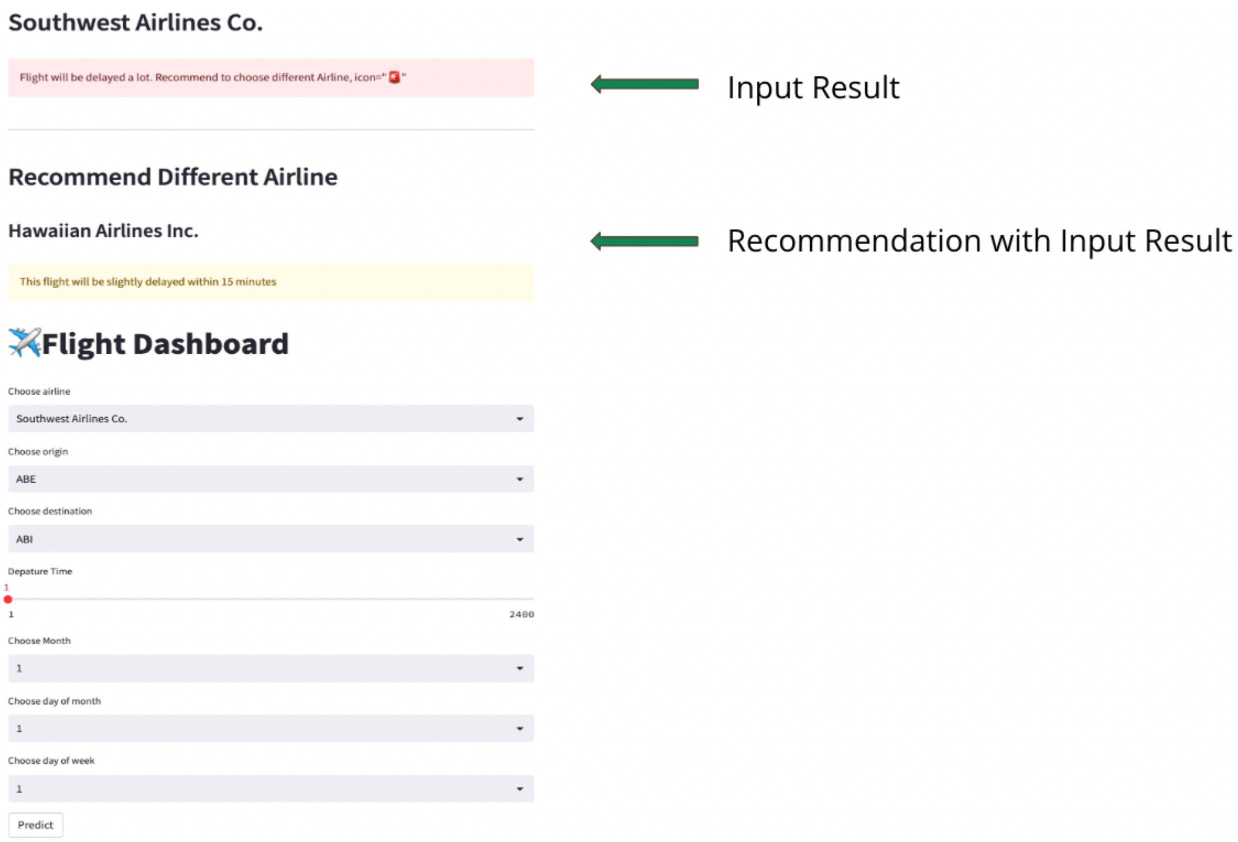
Choose Month: 1

Choose day of month: 1

Choose day of week: 1

Figure 6.1 Flight Dashboard Page

Figure 6.1 reflects our user interface using Streamlit. At the sidebar, there are a total of 3 different pages, FlightDashboard, Visualization, and DataFile. The first page is FlightDashboar, the main page of our interface where customers will go to get their flight delay prediction. Customers can input their known Airline, Origin, Destination, Departure Time, Month, Day of Month, and Day of Week and it will predict the delay for their specified flight details. After customer selections, they can click the predict button to see the output.



Southwest Airlines Co.

Flight will be delayed a lot. Recommend to choose different Airline, icon=" 🚨 "

Recommend Different Airline

Hawaiian Airlines Inc.

This flight will be slightly delayed within 15 minutes

✈️Flight Dashboard

Choose airline

Southwest Airlines Co.

Choose origin

ABE

Choose destination

ABI

Depature Time

1

1 2400

Choose Month

1

Choose day of month

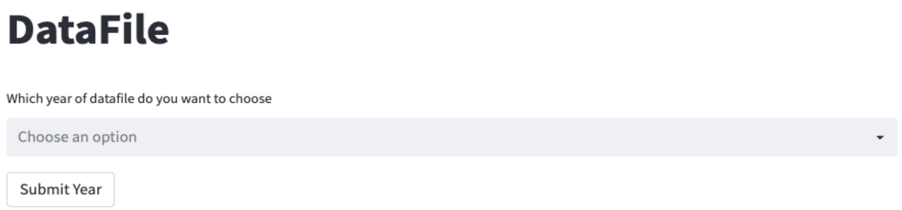
1

Choose day of week

1

Figure 6.2 Output of Flight Dashboard Page

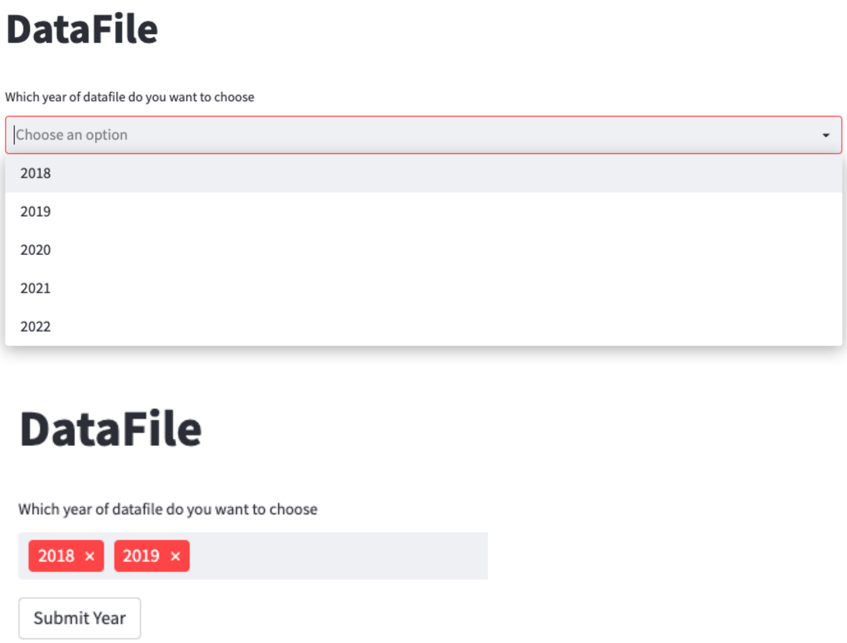
At the top, it shows the prediction result of the flight delay with the chosen airline. Under the Airline, it will always show a comment about your flight delay prediction and each condition has a different visual effect. The red box and flash icon mean the flight will be delayed more than 60 minutes indicating that customers should avoid this flight. If there's less time delay than 60 minutes, there will be a yellow box with different icons. If there is no predicted delay, the box will be green. Not only does the API show the flight delay status, we thought it would be useful to recommend users alternative airlines in the event they want to change their flight. This recommendation will only appear if there is statistically a better flight option with a smaller probability of delay than the customer's original choice.



DataFile

Which year of datafile do you want to choose

Figure 6.3 DataFile Page



DataFile

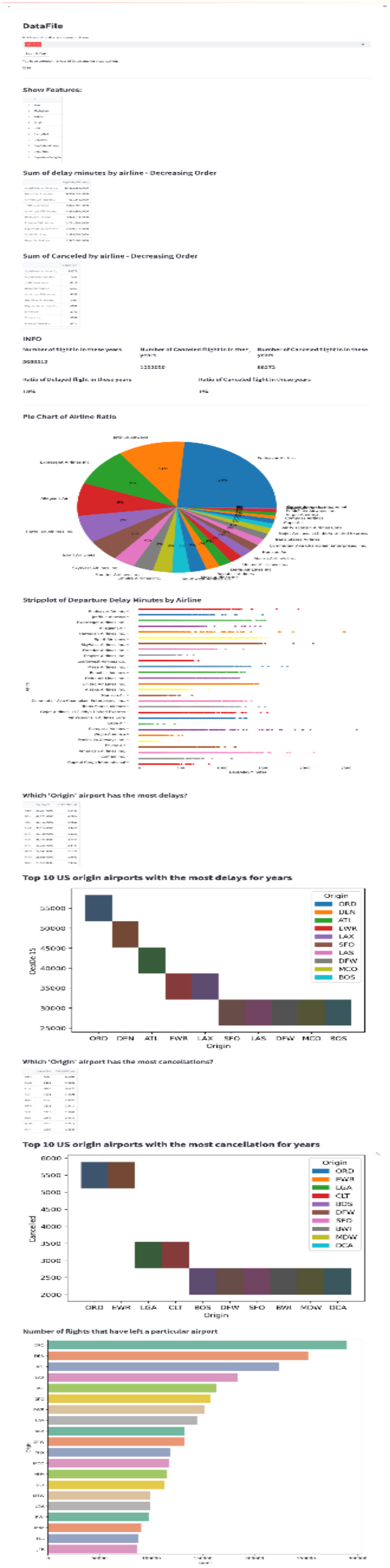
Which year of datafile do you want to choose

DataFile

Which year of datafile do you want to choose

Figure 6.4 Multiple Selection of year in DataFile Page

DataFile is the page where customers can select a specific year and view useful visualizations. These visualizations can help them make educated decisions when planning their trip, or they can just provide them with interesting information. As shown in Figure 6.3, customers can choose a single year or multiple years from 2018 to 2022 and would click the Submit Year button. Alternatively, the Visualization page shows all the visualization for the 2018 to 2022 datasets. Each graph is explained in the "Visualizations" section of our report.



DataFile

Show Features:

Sum of delay minutes by airline - Decreasing Order

Sum of Canceled by airline - Decreasing Order

INFO

Number of flight in in these years	Number of Canceled Flight in in these years	Number of Canceled flight in in these years
5683512	1353050	66373

Ratio of Delayed flight in these years	Ratio of Canceled flight in these years
18%	1%

Pie Chart of Airline Ratio

Stripplot of Departure Delay Minutes by Airline

Which 'Origin' airport has the most delays?

Top 10 US origin airports with the most delays for years

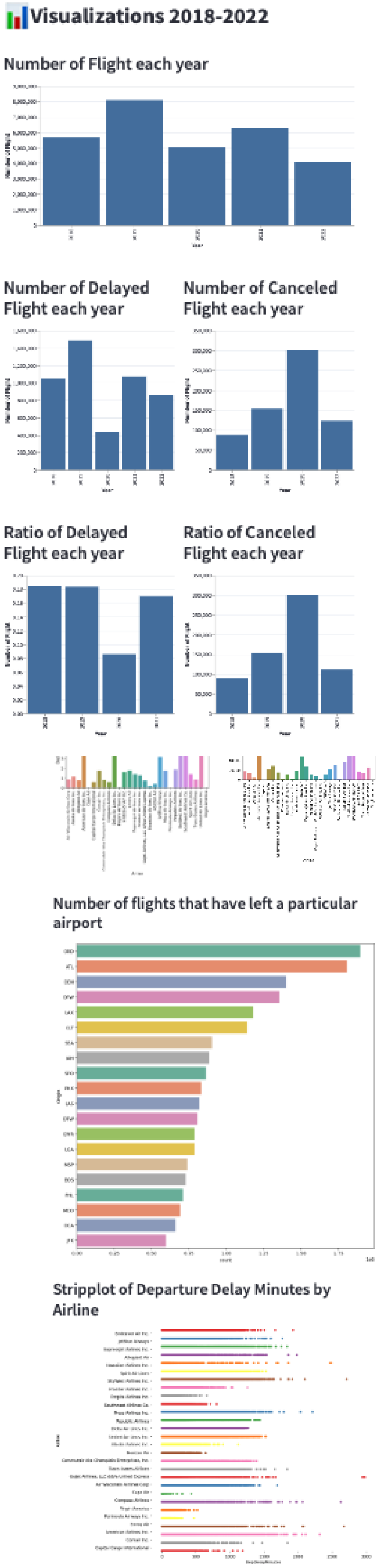
Which 'Origin' airport has the most cancellations?

Top 10 US origin airports with the most cancellation for years

Number of flights that have left a particular airport

Figure 6.5 Brief Visualization of Output of Datafile Page

Figure 6.6 Visualization Page of 2018-2022



Visualizations 2018-2022

Number of Flight each year

Number of Delayed Flight each year

Number of Canceled Flight each year

Ratio of Delayed Flight each year

Ratio of Canceled Flight each year

Number of flights that have left a particular airport

Stripplot of Departure Delay Minutes by Airline

Conclusion & Future Implementations

7.1 Conclusion

In closing, our use of the logistic regression machine-learning model, allowed us to create a relatively reliable flight-predicting API. We are confident that airline customers will be able to use our service to help guide their travel plans. In addition to being able to enter a flight and get its likely hood of it being delayed, users can also take advantage of our platform Visualizations, before committing to their travel arrangements. These graphs can provide insights into flight delays, cancellations, and airline trends for the current and previous 4 years. By deploying our site we ultimately hope to reduce travel stress while increasing customer confidence in their flight selections.

7.2 Future API Implementations

If given more time we would like to make our API a lot more detailed and interactive for our users. We would first want to allow customers to be able to enter multiple flights at once, instead of a single flight. Next, we would want our site to be able to predict the likelihood of flight cancelations alongside delay rates. We also think it would be beneficial to our users to get airline and flight route suggestions when given a warning of a flight cancellation. In addition to airline and route suggestions, we would like to focus on implementing an interactive U.S. map that allows users to click on a particular state and its airports to get detailed information about that particular location. Lastly, in order to make our site a true digital companion for travelers, our customers would benefit from real-time updates on their selected flights.

References

https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022?select=Combined_Flights_2021.csv&group=owned

https://www.gao.gov/products/gao-22-104429

https://www.bts.gov/newsroom/us-airlines-2020-net-profit-down-35-billion-2019#:~:text=U.S.%20scheduled%20passenger%20airlines%20reported,consecutive%20annual%20pre%2Dtax%20profits.

https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/

https://scikit-learn.org/stable/modules/tree.html

https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592#:~:text=Logistic%20regression%20is%20a%20classification,variables%20that%20determine%20an%20outcome.

https://github.khoury.northeastern.edu/toddbr/cs6220_FinalProject

https://www.kaggle.com/code/brunovinicius154/predicting-the-delay-of-flights-auc-0-88