

A Snapshot into Predicting Airline Delays in the U.S.

Project 4 - Northwestern Data Science Bootcamp 2022
By Connor Grant, Alfredo Garcia, Yousuf Amin AlFatwa, & Neel Patel

Description

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flights is published in DOT's monthly Air Travel Consumer Report and in this dataset of 2015 flight delays and cancellations..

We used this dataset to find out whether a delay prediction can be made for future flights based on Airline, Airport locations, Departure and Arrival Time, Delays by various causes (Security, Weather, Airline, etc...)

Factors explored in our data for predictions included differing airlines, departure and arrival airports, the day of week, and the month.

The data that we examined is composed of more categorical data and as a result of that, we predicted Random Forest Classifier would be a stronger model compared to Logistic Regression model.

Data Source

FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	...	ARRIVAL_TIME	AI
98	N407AS	ANC	SEA	5	...	408.0	-2
2336	N3KUAA	LAX	PBI	10	...	741.0	-9
840	N171US	SFO	CLT	20	...	811.0	5.1
258	N3HYAA	LAX	MIA	20	...	756.0	-9
135	N527AS	SEA	ANC	25	...	259.0	-2

- <https://www.kaggle.com/datasets/usdot/flight-delays>
- Data is collected and published by the Department of Transportation's Bureau of Transportation Statistics.
- Data consists of the whole year but it was condensed to 6 months to run the models.
- 5.8 million rows of data was examined with this dataset.

airlines.csv
IATA_CODE
AIRLINE

airports.csv
IATA_CODE
AIRPORT
CITY
STATE
COUNTRY
LATITUDE
LONGITUDE

flights.csv	
YEAR	DEPARTURE TIME & DELAY
MONTH	TAXI IN & OUT
DAY	CANCELLATION REASON
DAY_OF_WEEK	AIR_SYSTEM_DELAY
AIRLINE	SECURITY_DELAY
FLIGHT_NUMBER	AIRLINE_DELAY
TAIL_NUMBER	WEATHER_DELAY
ORIGIN_AIRPORT	AIR_TIME
DESTINATION_AIRPORT	DISTANCE

Technologies Used



Python - Pandas (Jupyter Notebook)

Supervised Machine Learning - SKLearn

Seaborn

Matplotlib

HTML / CSS / Bootstrap

Javascript

Flask

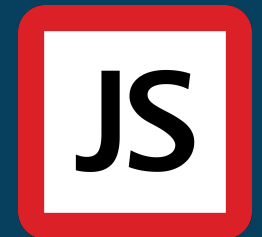
Tableau



seaborn



matplotlib



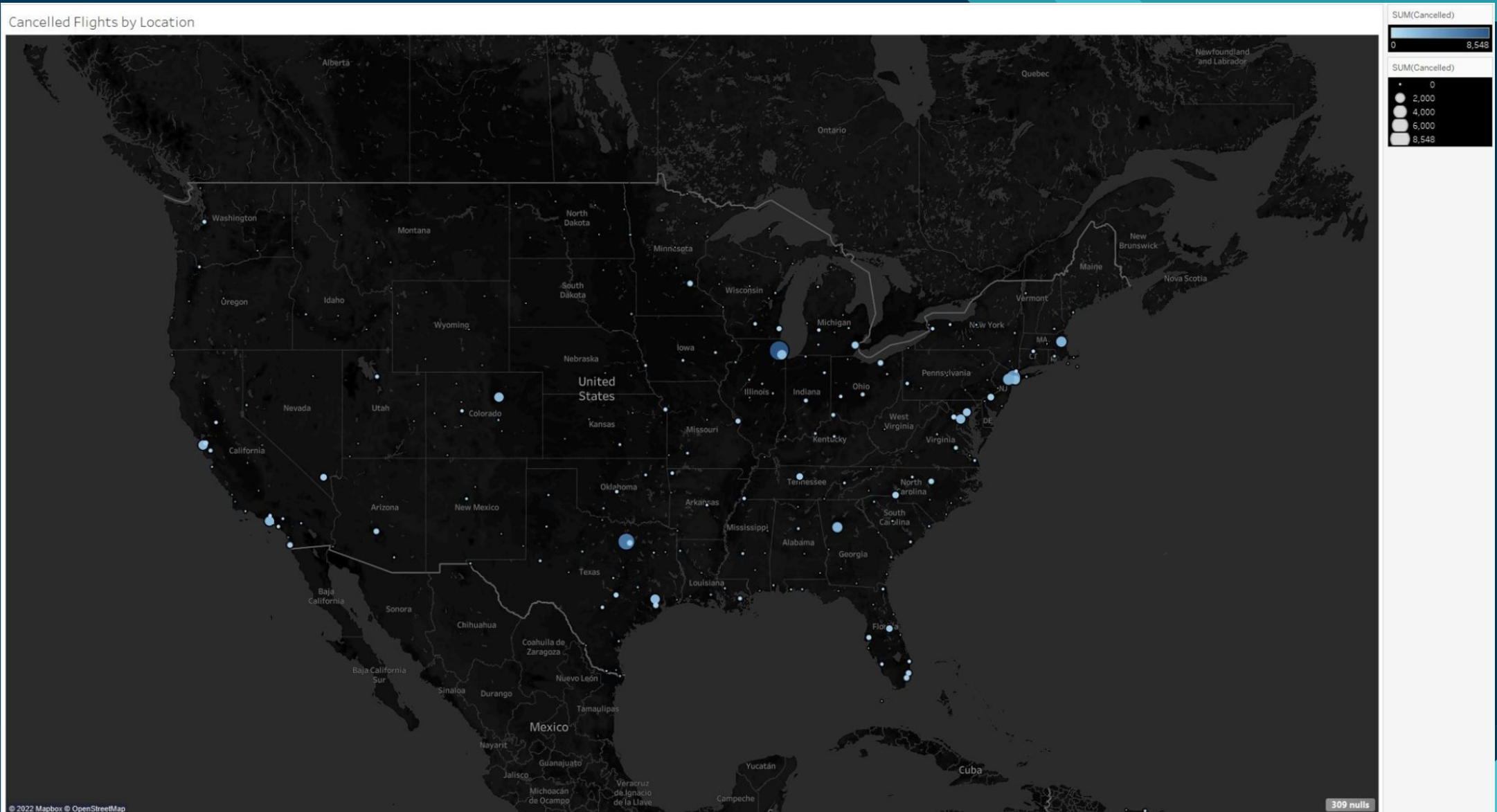
Airlines

IATA_CODE	AIRLINE	IATA_CODE	AIRLINES
UA	United Airlines (515,723)	NK	Spirit Airlines (117,379)
AA	American Airlines (725,984)	WN	Southwest Airlines (1,261,855)
US	US Airways (198,715)	DL	Delta Airlines (875,881)
F9	Frontier Airlines (90,836)	EV	Atlantic Southeast Airlines (571,977)
B6	JetBlue Airways (267,048)	HA	Hawaiian Airlines (76,272)
OO	SkyWest Airlines (588,353)	MQ	American Eagles Airlines (294,632)
AS	Alaska Airlines (172,521)	VX	Virgin America (61,903)

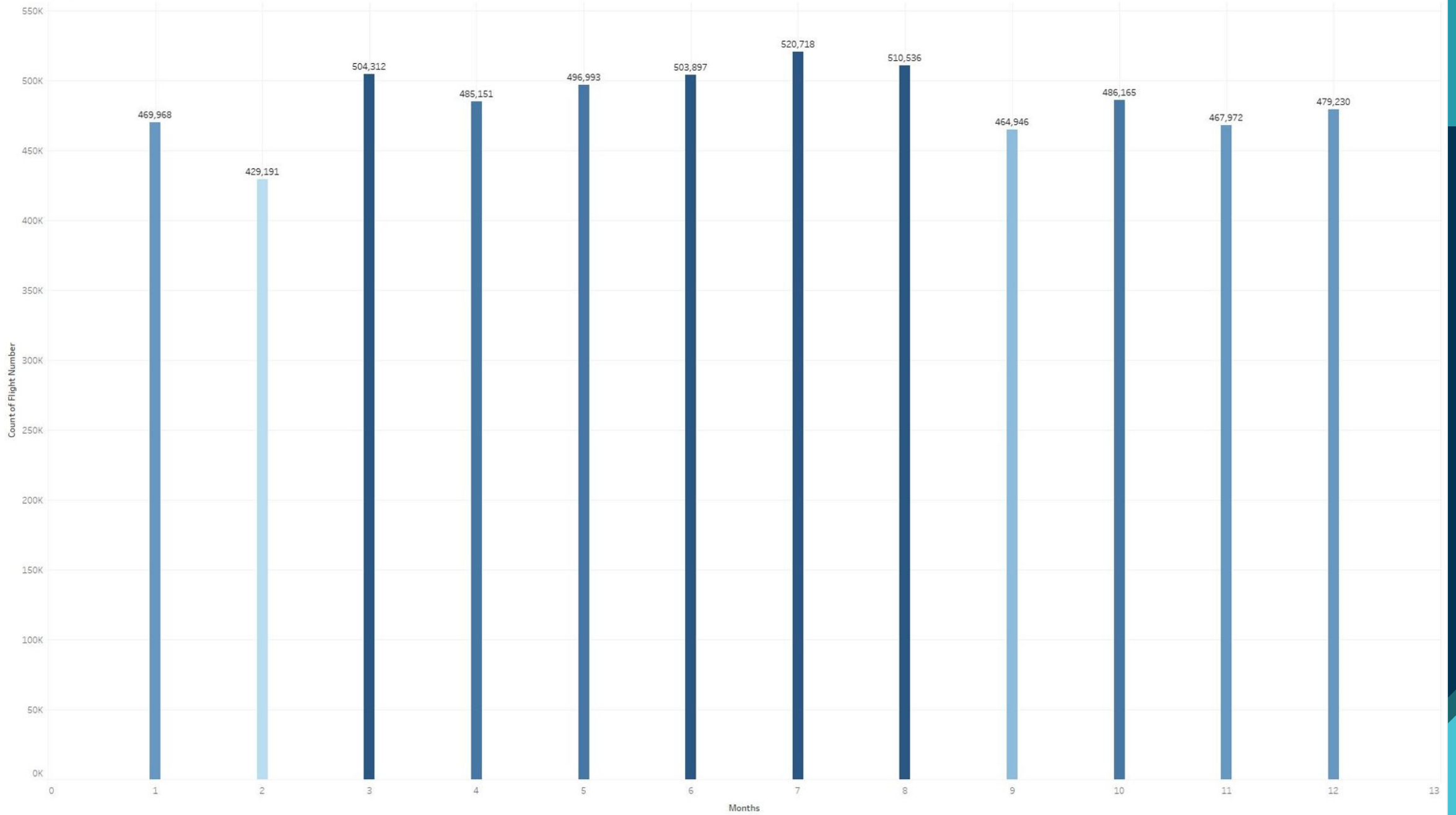


Tableau

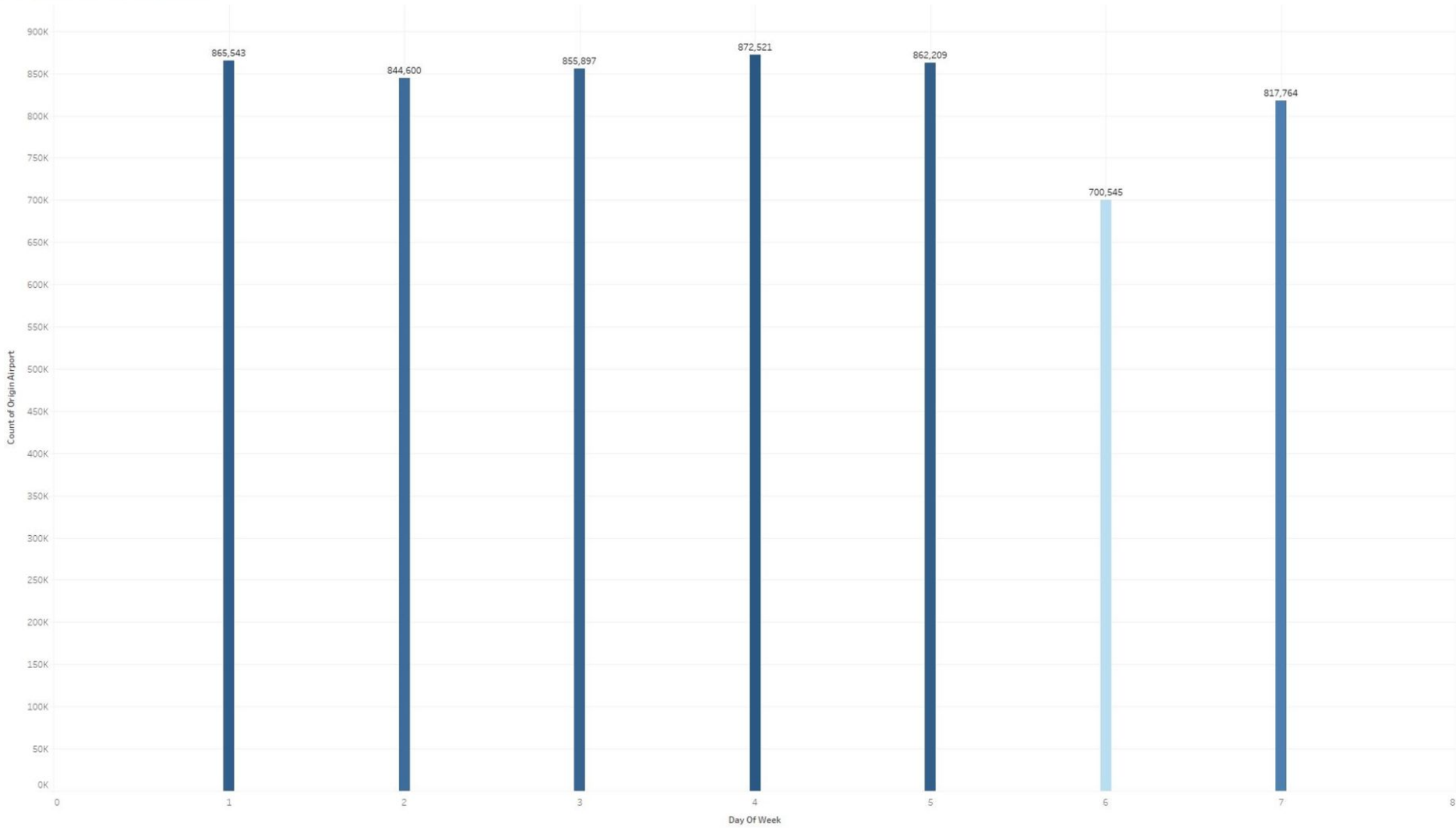
- 50 States, 308 Cities totaling to 322 Airports examined
- 14 Airline Companies
- 5,819,079 rows and 31 columns examined



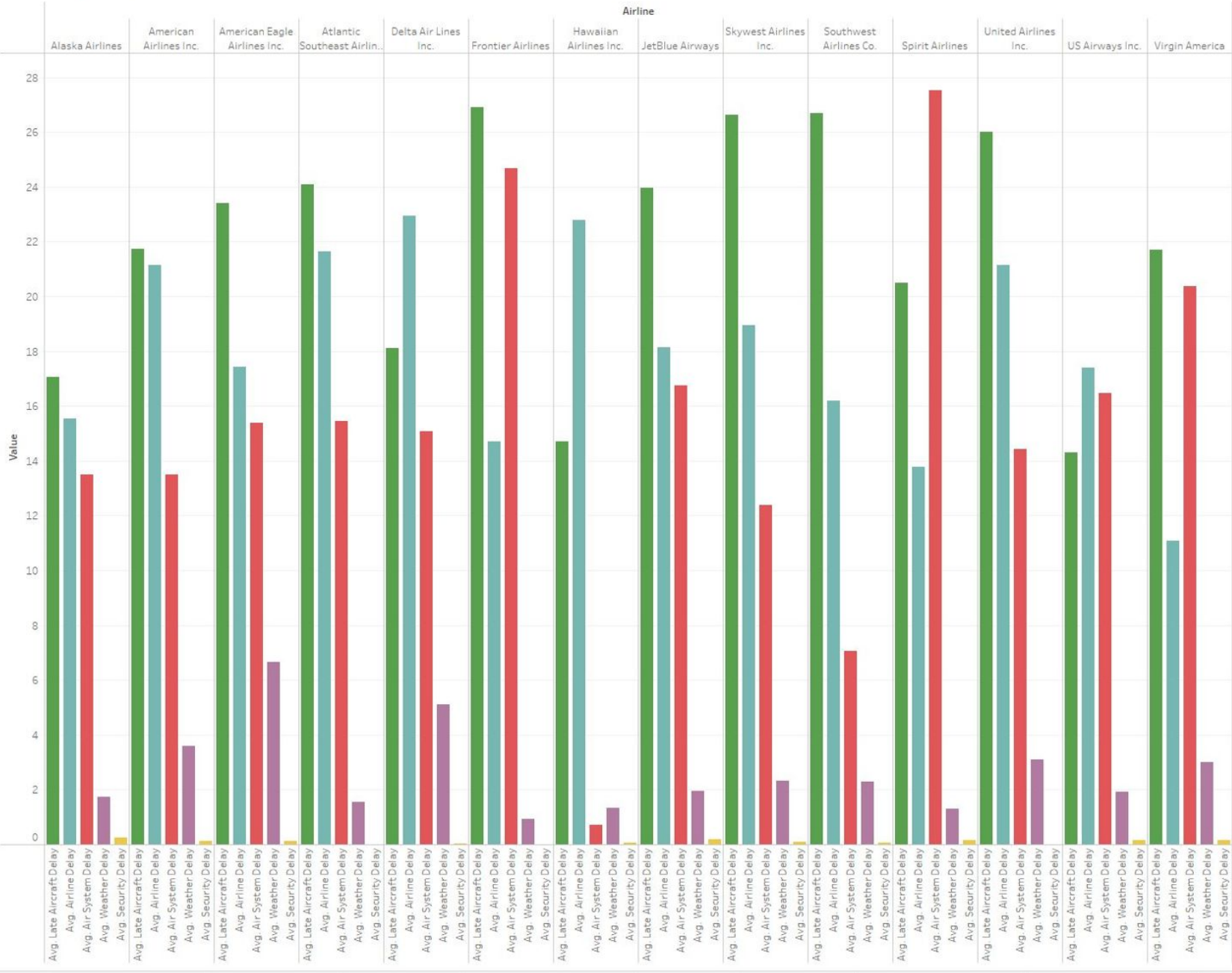
Of Flights by Months



Of Flights by Day of the Week



Delay Reasons by Airlines



Measure Names

- Avg. Air System Delay
- Avg. Airline Delay
- Avg. Late Aircraft De...
- Avg. Security Delay
- Avg. Weather Delay

Models Used

- Logistic Regression
- Random Forest Classifier
- Confusion Matrix

Logistic Regression

```
In [20]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

In [21]: from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier

Out[21]: LogisticRegression()

In [22]: classifier.fit(X_train, y_train)

C:\Users\Connor\anaconda3\envs\PythonData38\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs
failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
  https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
  https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(

Out[22]: LogisticRegression()

In [23]: print(f"Training Data Score: {classifier.score(X_train, y_train)}")
print(f"Testing Data Score: {classifier.score(X_test, y_test)}")

Training Data Score: 0.9977666666666667
Testing Data Score: 0.9952
```

Random Forest Classifier

```
In [24]: # Train a Random Forest Classifier model and print the model score
from sklearn.ensemble import RandomForestClassifier

In [25]: # Scale the data
scaler = StandardScaler()
scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

In [26]: randomForestClass = RandomForestClassifier(random_state=42)
randomForestClass.fit(X_train, y_train)

Out[26]: RandomForestClassifier(random_state=42)

In [27]: print("RandomForestClassifier score: ", randomForestClass.score(X_test,y_test))

RandomForestClassifier score: 1.0

In [28]: # Create a RandomForestClassifier model, fit it to the data, and print the model's score.
randomForestClass_scaled = RandomForestClassifier(random_state=42)
randomForestClass_scaled.fit(X_train_scaled, y_train)
print("RandomForestClassifier scaled score: ", randomForestClass_scaled.score(X_test_scaled, y_test))

RandomForestClassifier scaled score: 1.0
```

Confusion Matrix

```
In [29]: from sklearn.metrics import confusion_matrix

y_true = y_test
y_pred = classifier.predict(X_test)
confusion_matrix(y_true, y_pred)

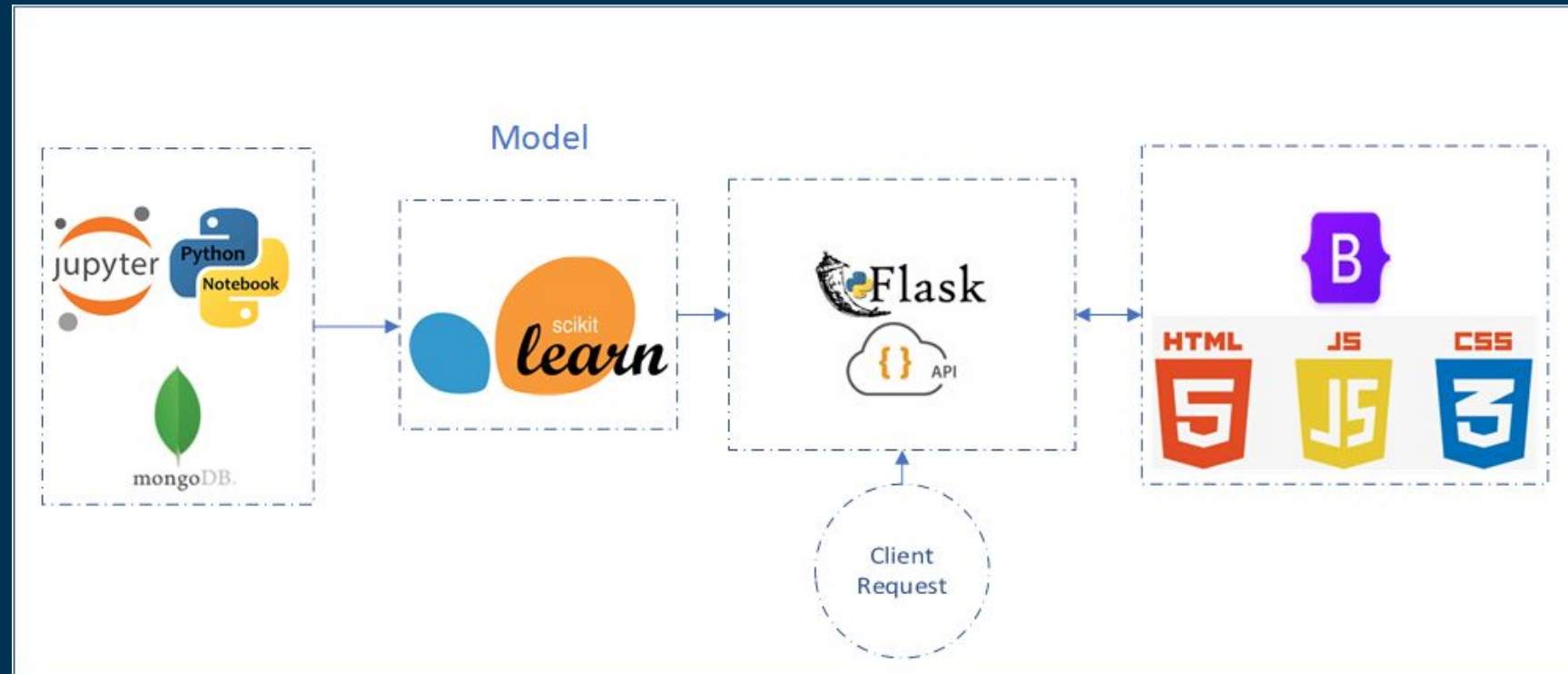
Out[29]: array([[6110,  0],
               [ 48, 3842]], dtype=int64)

In [30]: tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
accuracy = (tp + tn) / (tp + fp + tn + fn) # (111 + 128) / (111 + 5 + 128 + 6)
print(f"Accuracy: {accuracy}")

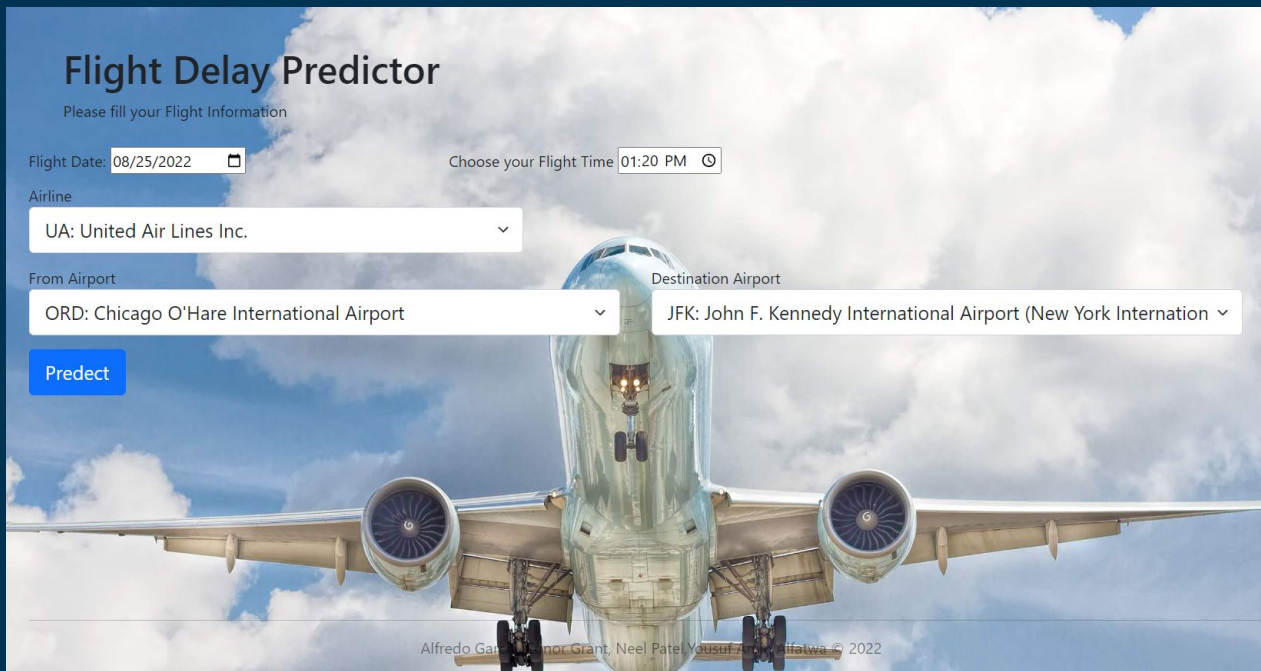
Accuracy: 0.9952
```

Flask ,Web page

- Logistic Regression model
- Save the model as pkl
- Use it in flask
- Got info from client
- Make prediction
- Prediction result on the screen




Flask ,Web page




The image shows a web application titled "Flight Delay Predictor" with a background image of a large commercial airplane flying through a cloudy sky. The interface includes a title, a subtitle "Please fill your Flight Information", and several input fields: "Flight Date" with a date picker set to "08/25/2022", "Choose your Flight Time" with a time picker set to "01:20 PM", "Airline" with a dropdown menu showing "UA: United Air Lines Inc.", "From Airport" with a dropdown menu showing "ORD: Chicago O'Hare International Airport", and "Destination Airport" with a dropdown menu showing "JFK: John F. Kennedy International Airport (New York Internation". A blue "Predect" button is located below the input fields. At the bottom, there is a small copyright notice: "Alfredo Garcia, Connor Grant, Neel Patel, Yousuf Ahmed, Afiatwa © 2022".


Flight Delay Predictor

Please fill your Flight Information


Flight Date: 08/25/2022 

Choose your Flight Time: 01:20 PM 


Airline

UA: United Air Lines Inc. 

From Airport

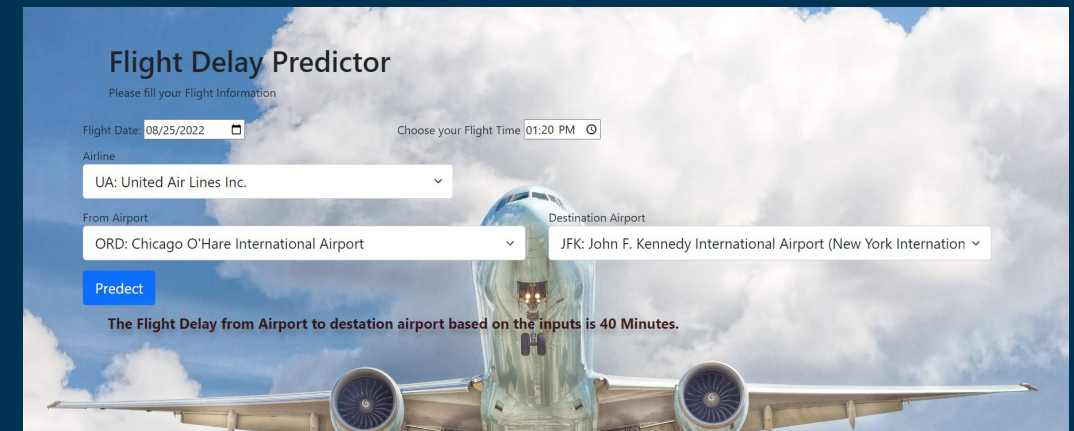
ORD: Chicago O'Hare International Airport 

Destination Airport

JFK: John F. Kennedy International Airport (New York Internation 

Predect


Alfredo Garcia, Connor Grant, Neel Patel, Yousuf Ahmed, Afiatwa © 2022




The image shows the same web application as the previous one, but with the "Predect" button highlighted in blue. Below the button, a message is displayed: "The Flight Delay from Airport to destation airport based on the inputs is 40 Minutes." The background image of the airplane and the input fields remain the same.


Flight Delay Predictor

Please fill your Flight Information


Flight Date: 08/25/2022 

Choose your Flight Time: 01:20 PM 


Airline

UA: United Air Lines Inc. 

From Airport

ORD: Chicago O'Hare International Airport 

Destination Airport

JFK: John F. Kennedy International Airport (New York Internation 

Predect

The Flight Delay from Airport to destation airport based on the inputs is 40 Minutes.

Summary

Our prediction was that a Random Forest would produce a better prediction with our data. Random Forest are generally stronger with using categorical data, and given our cleaning and preparation, we had to convert our categorical data to numeric (airports, flight airlines, etc) which influenced this.

The results were very close, however, our prediction held true. Our Random Forest model did provide a stronger prediction (1.0), compared to our Logistic Regression model (0.9997).

The results of our data did show that Southwest Airlines (IATA: WN) is the airline with the highest amount of delayed flights, followed by Delta (IATA: DL), Skywest Airlines (IATA: OO), and Atlantic Southeast Airlines (IATA: EV).

The airports with the highest amount delayed arrival flights (IE: flight arrives past the original estimated arrival time) are: Denver (DEN), Atlanta(ATL), O'Hare(ORD), Dallas Fort Worth (DFW), Los Angles (LAX), and Phoenix AZ).

The day of the month did not have any direct correlation to a pattern of predicting delays, however, our data did suggest the 1st -> 5th and the 25th + 26th of each month had the highest counts of delays.

Lesson Learned

Our team came across hurdles throughout the project, but we continued to address and fix them in a timely manner.

We needed to reassess our data size (we bounced around from working with a full year, to only 3 months, ultimately to ending with 6 months). Our models would crash when trying to run due to such a high size, but we were able to work around this by using a random sample size when running our code. (shoutout to TA Erin for this suggestion!).

Assisting with changing factors in our dataframe columns for our Flask App and HTML prediction page to properly generate. (Changing data to numeric values, etc).

The amount of data used (6 months) could have an influence on our prediction models and their outcome, and future prediction models used could try to incorporate multiple years' of flight data to provide further insight on these models.

Thank You!

Repo Link:

