

# Predicting Flight Delays

TAMIDS Competition - Spring 2020





# Team InFlightful



Undergraduate 1<sup>st</sup> Prize Team Inflightful: Cameron Brill, Samarth Dave, Allyson King, Nathan Reddy



# Challenge

Build models for **predicting expected flight delays** by airline and flight for the third quarter of 2019.

Considerations...

1. Who are the beneficiaries of such a model?
  - a. *Airlines and travelers*
2. How can we answer the research question?
  - a. *Model flight delays by airline and by flight*
3. What can we do to go beyond the ask?
  - a. *Create a visualization tool to help understanding of delays*



# Data Collection

Given the following data sets...

1. Flight Delays (flight logs for 2018 and first ½ of 2019)
2. Airports (airport locations and codes)
3. Routes (distance for routes)
4. Airfares (average airfares between cities)

Considerations...

1. Is the data given complete?
  - a. *To merge Airports and Flight Delays, we found more airport data*
2. Would additional data be influential?



# Data Cleaning

Steps taken...

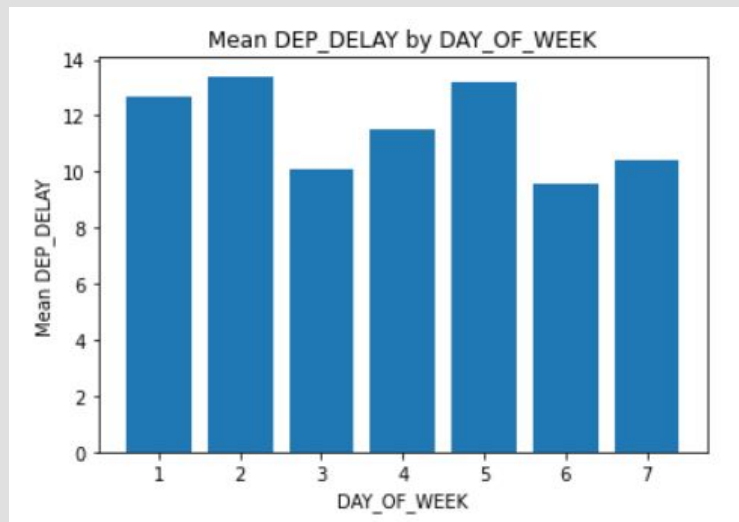
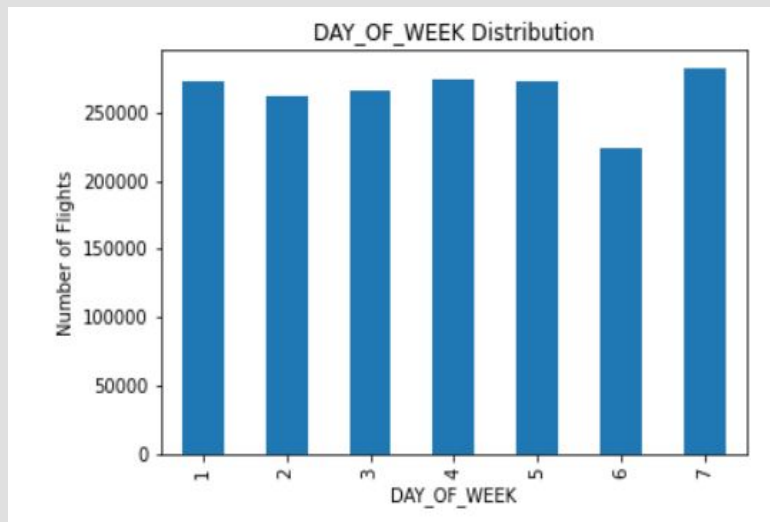
1. Merge data sets
  - a. *Merged Flight Delays, Airports, and external airport info*
2. Handle null values
  - a. *Column with  $< 1\%$  null  $\Rightarrow$  drop missing rows*
  - b. *Columns with  $> 70\%$  null  $\Rightarrow$  drop columns*
  - c. *In between  $\Rightarrow$  calculate values from other columns*
3. Change data types
4. Feature engineering
  - a. *Made DEP\_HOUR and ARR\_HOUR variables*



# Data Exploration

Steps...

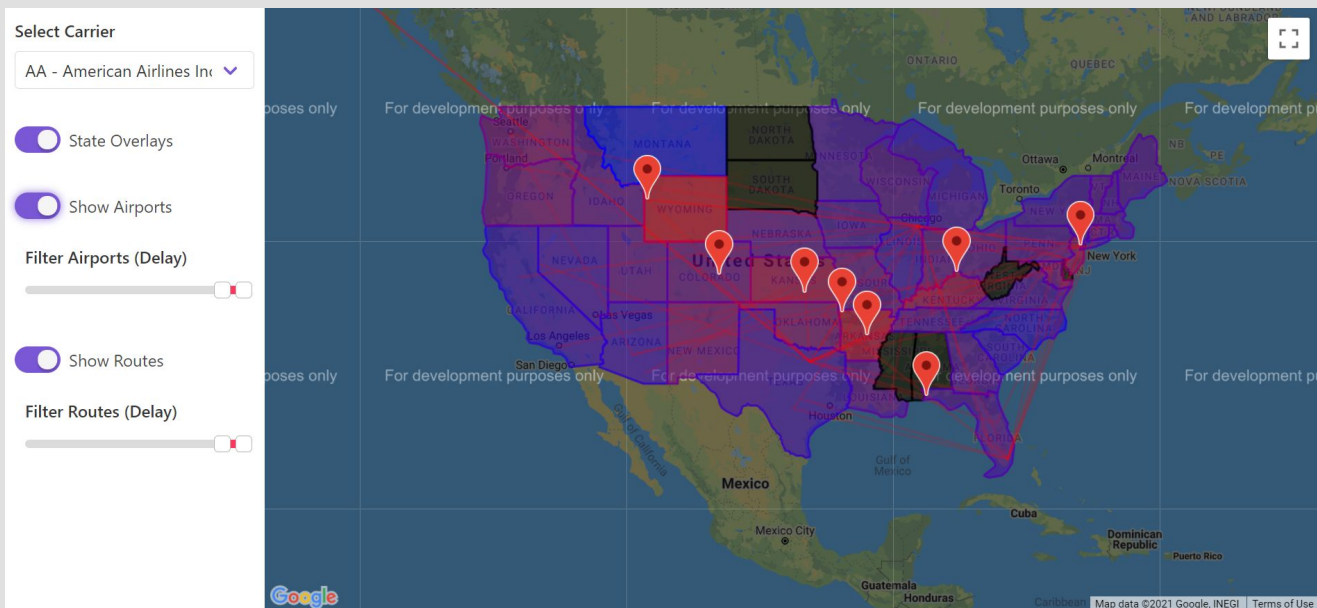
1. Visualize the distribution of each column
2. Visualize the relationship between each column and target variable





# Data Visualization

[tamids.herokuapp.com](https://tamids.herokuapp.com)

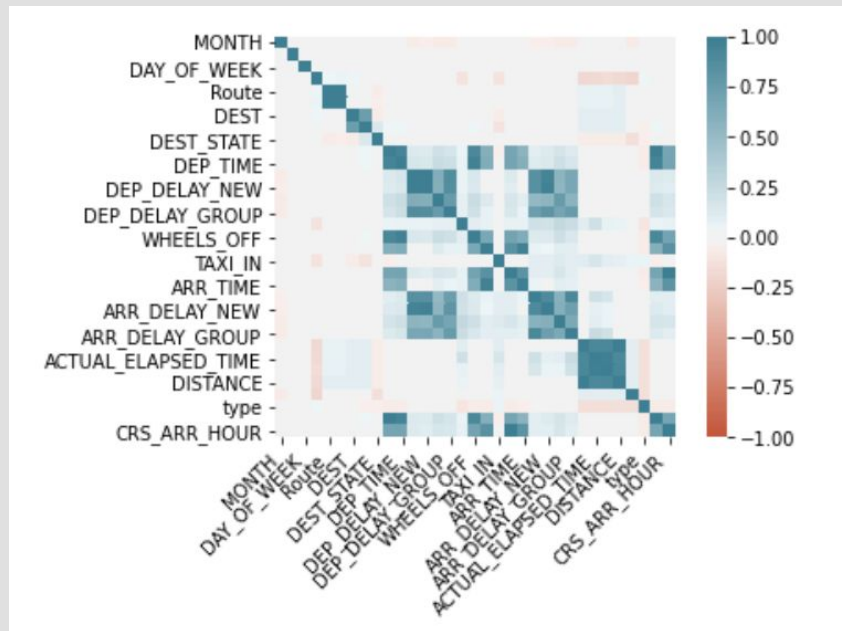




# Model Fitting

Steps...

1. Model selection
  - a. *Random Forest*
2. Feature Selection
  - a. *Variables with low correlation*
3. Cross validation
  - a. *Avoids overfitting*







# Model Fitting (ARR)

```
[ ] ## feature selection
feature_list = ['DEST_STATE', 'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY', 'TAXI_OUT', 'WHEELS_OFF', 'CRS_ARR_TIME', 'ARR_DELAY']
features = df[feature_list]
```

```
[ ] rf.score(test_features, test_labels)
```

```
↳ 0.6373086291946746
```

```
[ ] from sklearn.model_selection import cross_val_score
scores = cross_val_score(rf, features, labels, cv=3)
scores
```

```
↳ array([0.8230642 , 0.49102801, 0.85999882])
```

```
↳ [('DEP_DELAY', 0.83),
    ('TAXI_OUT', 0.05),
    ('CRS_DEP_TIME', 0.04),
    ('DEP_TIME', 0.02),
    ('WHEELS_OFF', 0.02),
    ('CRS_ARR_TIME', 0.02),
    ('DEST_STATE', 0.01)]
```



# Model Fitting (DEP)

```
[ ] ## feature selection
feature_list = ['CRS_DEP_TIME', 'CRS_ARR_TIME', 'CRS_ELAPSED_TIME', 'DISTANCE', 'DEST_STATE', 'DEP_DELAY']
features = df[feature_list]
```

```
[ ] rf.score(test_features, test_labels)
```

```
↳ 0.0327838195349196
```

```
[ ] from sklearn.model_selection import cross_val_score
scores = cross_val_score(rf, features, labels, cv=3)
scores
```

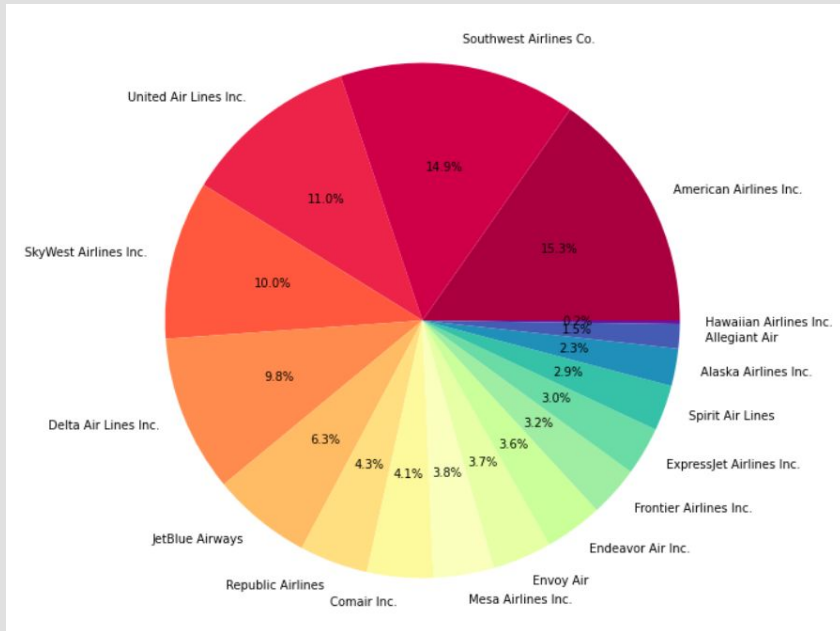
```
↳ array([-0.05612057, -0.04022531, -0.28383399])
```

```
↳ [('CRS_DEP_TIME', 0.25),
    ('CRS_ARR_TIME', 0.24),
    ('DISTANCE', 0.21),
    ('CRS_ELAPSED_TIME', 0.18),
    ('DEST_STATE', 0.12)]
```



# Results

## 50+ Minute Delays by Airline



## Takeaways...

1. We were able to accurately predict ARR\_DELAY and DEP\_DELAY but only if the other was a predictor

# Thank you!

...

