# Predicting Flight Departure Delay

James Gao / Nathan Nusaputra / Ankit Patel / Ryan Sawasaki

W261 Machine Learning at Scale

# Motivation and Problem Statement

- Flight delays have major financial implications
  - $8 billion per year for airlines
  - $18 billion per year for passenger
- Ability to predict flight delays can mitigate the disruption to operations and result in cost savings
- Goal: 2 hours prior to scheduled departure time, predict if a flight is going to be delayed by more than 15 minutes
- Prediction performance based on precision, recall, and accuracy

**Total Cost of Delay in the U.S. (dollars, billion)**

| | 2016 | 2017 | 2018 | 2019 |
|---|---|---|---|---|
| Airlines | 5.6 | 6.4 | 7.7 | 8.3 |
| Passengers | 13.3 | 14.8 | 16.4 | 18.1 |
| Lost Demand | 1.8 | 2.0 | 2.2 | 2.4 |
| Indirect | 3.0 | 3.4 | 3.9 | 4.2 |
| **Total** | **23.7** | **26.6** | **30.2** | **33.0** |

Cost of Delay Estimated 2019
https://www.faa.gov/data_research/aviation_data_statistics/media/cost_delay_estimates.pdf

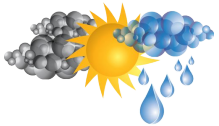
# Datasets

## Airlines

- US domestic flights from 2015 to 2019
- 31M+ records
- Arrival and departure schedules
- Flight delays, distance, duration
- Airport and airline carriers

- 82% flights are on time and 18% flights are delayed
- 1.5% flights are cancellations
- Dropped missing flight records

## Weather



- Weather from 2015 to 2019
- 630M+ weather records
- Temperature, wind speed, visibility

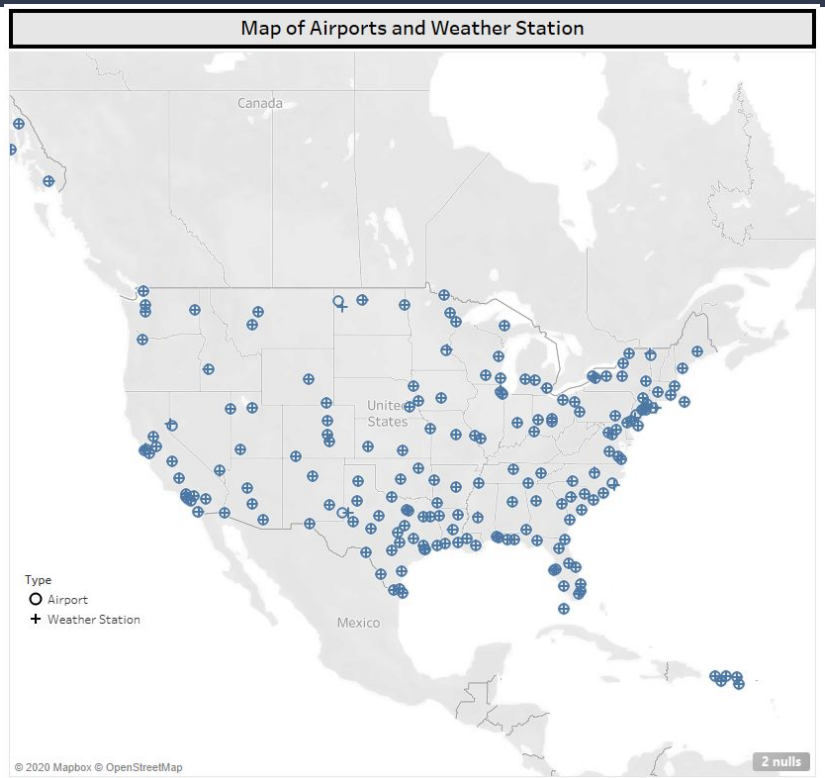
- 80% records are with missing values
- Records with NA, 9999, and no values are dropped
- Precipitation records are imputed with daily averages

# Joining Datasets

- Airlines Dataset Preparation
  - Joining Airline with external airport location dataset for longitude and latitude
  - Finding closest weather station to each airport using full cross join, haversine distance, and rank
- Weather Data Set Preparation
  - Visibility, temperature, horizon distance, precipitation
  - Filter out nulls, keep 1s and 5s in quality control columns

```
from pyspark.sql.functions import col, radians, asin, sin, sqrt, cos

airports_and_stations_exploded_with_distance = airports_and_stations_exploded.withColumn("dlon", radians(col("longitude_decimal_degrees")) - radians(col("LONGITUDE"))) \
                                                                             .withColumn("dlat", radians(col("latitude_decimal_degrees")) - radians(col("LATITUDE"))) \
                                                                             .withColumn("haversine_dist", asin(sqrt(
                                                                                                                    sin(col("dlat") / 2) ** 2 + cos(radians(col("LATITUDE")))
                                                                                                                    * cos(radians(col("latitude_decimal_degrees"))) * sin(col("dlon") / 2) **
2
                                                                                                                    )
                                                                                                                ) * 2 * 3963 * 5280) \
                                                                             .drop("dlon", "dlat")

airports_and_stations_exploded_with_distance.createOrReplaceTempView('airports_and_stations_exploded_with_distance')
```

# Airport and Weather Stations



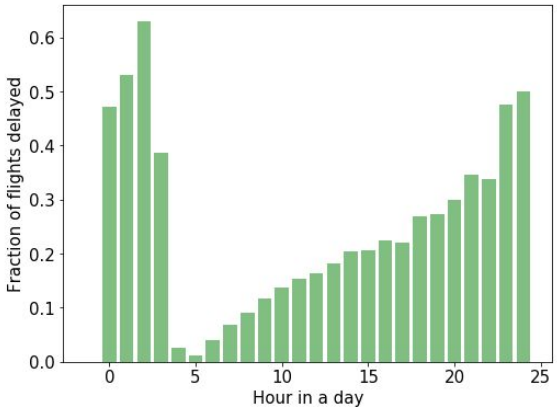
# Joining Datasets

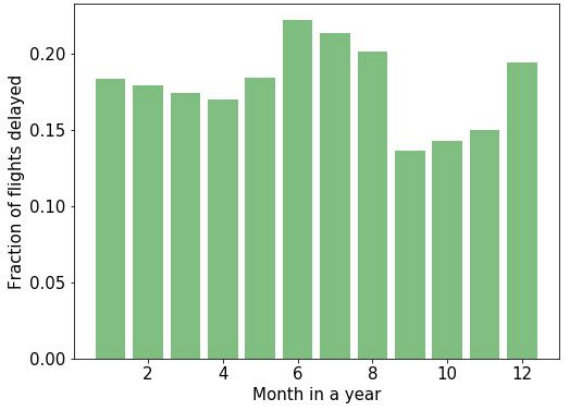
- Convert airlines data from local time to UTC
  - Used outside data source for timezone based on IATA codes
  - Converted to UTC with pytz library
- Join airlines and weather datasets
  - Subtract 2 hours from the departure time and round down
  - Join datasets on the two join keys: timestamp and station ID

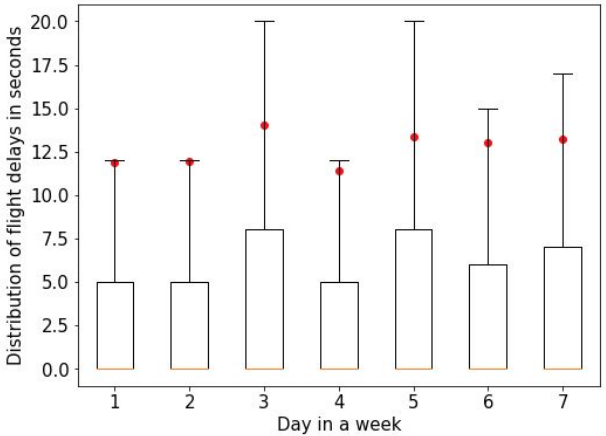
Cmd 22

```
# Convert the date and time in the airlines dataset to timestamps

# UDF for converting year, month, day to timestamps
def create_datetime_from_parts(year, month, day, local_time):
    time = str(local_time)
    num_str = time.zfill(4)
    hour = num_str[0:2]
    min = num_str[2:4]
    return f'{year}-{month}-{day} {hour}:{min}:0'

# Apply UDF to create column for local departure time
create_datetime_udf = f.udf(create_datetime_from_parts, types.StringType())
airlines_with_localtime =
airlines_with_weather_station.withColumn("LOCAL_DEP_TIME",
create_datetime_udf('YEAR', 'MONTH',
'DAY_OF_MONTH','CRS_DEP_TIME').cast(types.TimestampType()))

# join airlines dataset with timezone dataset. join based on departure airport
'ORIGIN' and airport 'IATA'
airlines_with_timezone = airlines_with_localtime.join(f.broadcast(airport_tz),
airport_tz.IATA==airlines_with_localtime.ORIGIN, 'inner')

# use the spark built-in function to convert the local time to UTC time based on
timezone
airlines_with_utc =
airlines_with_timezone.withColumn("UTC_DEP_TIME",to_utc_timestamp(col("LOCAL_DEP_T
IME"), col("timezone")))
```

# Flight delay varies with time

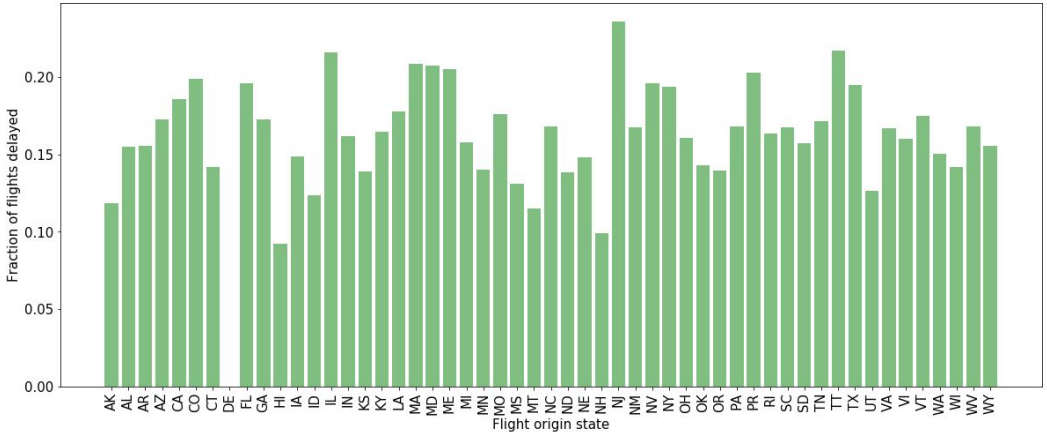


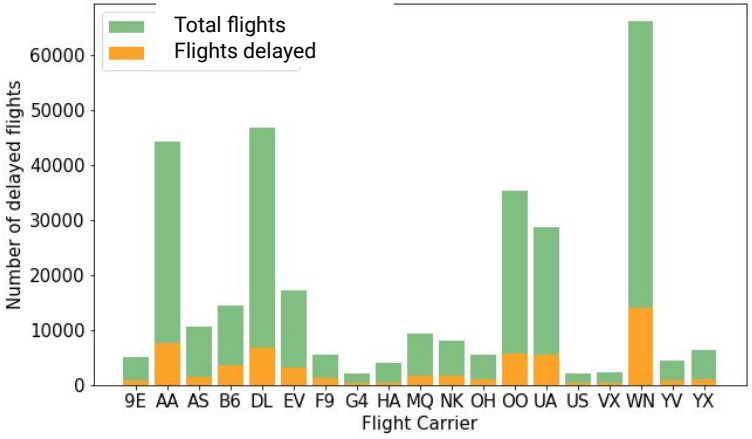




- Flights departing later evening or early morning tend to delay more than those departing during day time
- Seasonal effects in flight delay are observed over the month of year
- The variability in flight delay is marginal over the day of week and the day of month
- 15.4% flights are delayed during weekdays and 7.4% flights are delayed over weekends

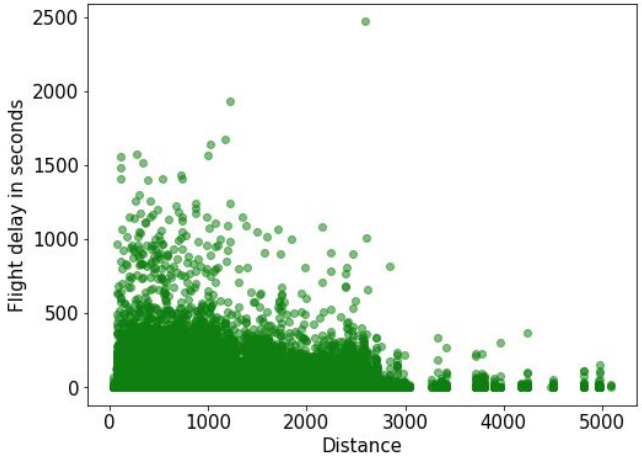
# Flight delay varies with carrier and location

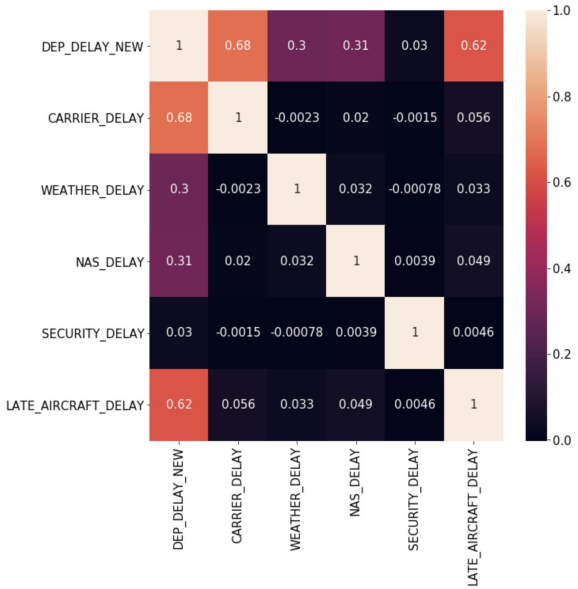



- Fraction of flights delayed varies significantly based on the origin and destination states
- A large fraction of flights delayed is operated by the large carriers compared to the small carriers

# Flight delay varies with distance and its prior status



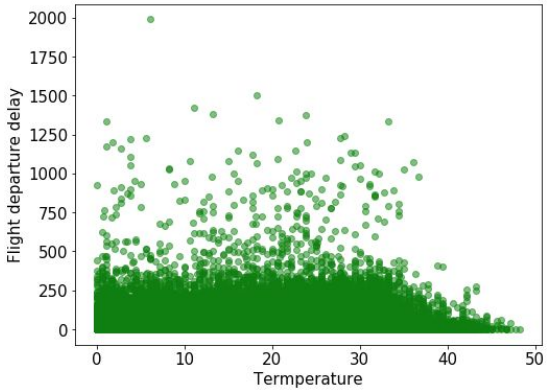



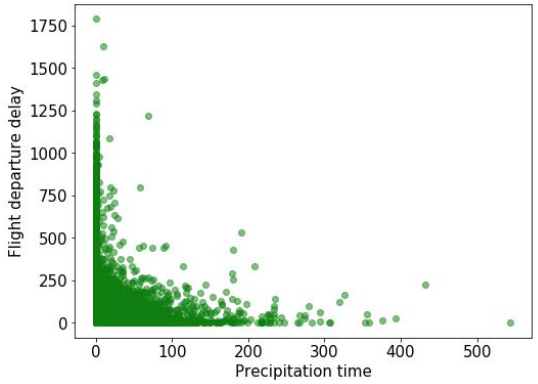

| | Ontime | Delay |
|---|---|---|
| **Past ontime** | 78.06% | 11.83% |
| **Past delayed** | 3.69% | 6.39% |

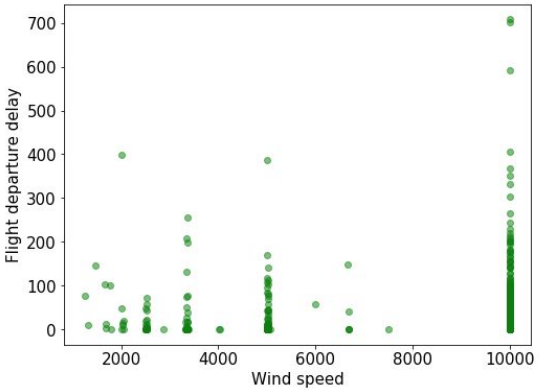
Delay propagation feature

- Flight departure delay is negatively correlated with flight distance
- Flight departure delay is strongly correlated to carrier, NAS, weather, and late aircraft delays
- Probability of flight status remain the same as its past status in a day is 0.84 and the status changes with probability 0.16

# Flight delay varies with the weather parameters




- Flight departure delay is negatively correlated with temperature
- Flight departure delay decreases as the precipitation time increases
- Flight departure delay is positively correlated with wind speed

# Feature Engineering

- Account for delay propagation by creating feature that indicates if the previous flight leg is delayed
  - Partition planes by tail number, sort and calculate the time difference between flight departures
  - For time difference between 2 and 7.5 hours, record the previous flight departure status
- Preparation for algorithm exploration
  - String Indexer and One Hot Encoder for categorical variables
  - Vector Assembler for combining the numerical and categorical features

| | TAIL_NUM | UTC_DEP_TIME | PREV_UTC_DEP_TIME | DEP_TIME_DIFF | DEP_DEL15 | PREV_DEP_DEL15 |
|---|---|---|---|---|---|---|
| 37 | N022AA | 2015-01-17T02:40:00.000+0000 | 2015-01-15T21:54:00.000+0000 | 28.766666666666666 | 0 | 0 |
| 38 | N022AA | 2015-01-17T14:13:00.000+0000 | 2015-01-17T02:40:00.000+0000 | 11.55 | 0 | 0 |
| 39 | N022AA | 2015-01-17T16:45:00.000+0000 | 2015-01-17T14:13:00.000+0000 | 2.533333333333333 | 1 | 0 |
| 40 | N022AA | 2015-01-17T20:20:00.000+0000 | 2015-01-17T16:45:00.000+0000 | 3.5833333333333335 | 1 | 1 |
| 41 | N022AA | 2015-01-18T00:15:00.000+0000 | 2015-01-17T20:20:00.000+0000 | 3.9166666666666665 | 0 | 1 |
| 42 | N022AA | 2015-01-18T13:15:00.000+0000 | 2015-01-18T00:15:00.000+0000 | 13 | 0 | 0 |
| 43 | N022AA | 2015-01-18T21:40:00.000+0000 | 2015-01-18T13:15:00.000+0000 | 8.416666666666666 | 0 | 0 |
| 44 | N022AA | 2015-01-19T01:35:00.000+0000 | 2015-01-18T21:40:00.000+0000 | 3.9166666666666665 | 0 | 0 |

# Algorithm Exploration

## Algorithm Requirements

1. Classification problem
2. Compatible with distributed systems (and large datasets)

Baseline

### Logistic Regression

Easy to Interpret

Robust to Noise

Decision Threshold to be set

Parameters
- Max Iterations

*Accuracy:* *78%*
*Recall:* *30%*
*Precision:* *63%*

Explored

### Decision Tree

Easy to Interpret

Affected by Noise

Automatically Classifies

Parameters
- Max Depth
- Minimum Instances per Node

*Accuracy:* *84%*
*Recall:* *36%*
*Precision:* *63%*

Explored

### Random Forest

Difficult to Interpret

Majorly Affected by Noise

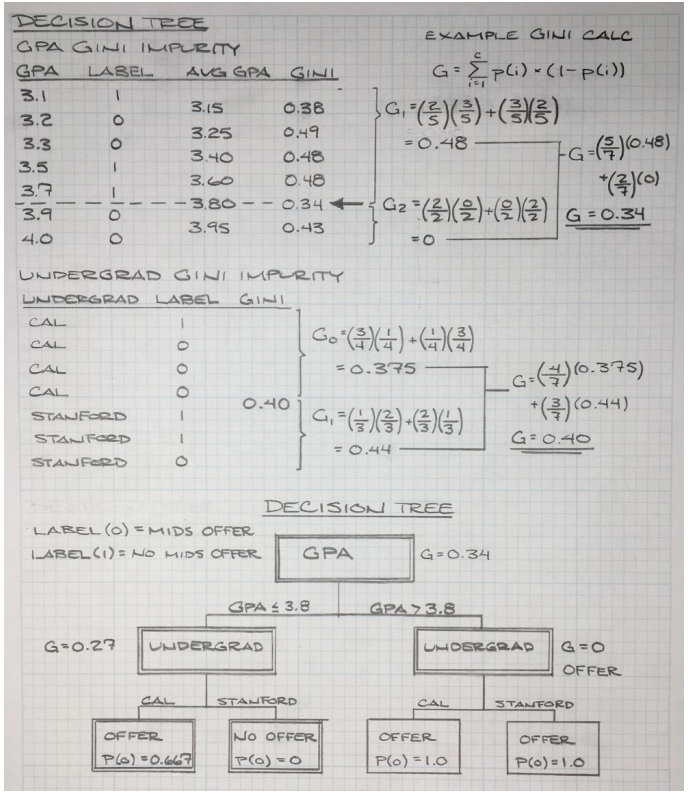
Automatically Classifies

Parameters
- Number of Trees
- Max Depth

*Accuracy:* *84%*
*Recall:* *35%*
*Precision:* *62%*

# Algorithm Implementation

- Toy example for implementation of decision tree
- Predicts if an applicant receives an admissions offer from MIDS
- Features
  - Applicant's undergrad school (Categorical: Cal or Stanford)
  - Applicant's GPA (Numerical: 0 to 4.0)
- Gini Impurity $G = \sum p(i) \times (1 - p(i))$



# Model Performance

Final Model

**Decision Tree**

Easy to Interpret

Affected by Noise

Automatically Classifies

Parameters
- Max Depth
- Minimum Instances per Node

**Parameters**
- maxDepth = 10
- minInstancesPerNode = 1000

**Features**

- Month of Year (cv)
- Day of Month (cv)
- Day of Week (cv)
- Weekend (nd)

- Carrier (cv)
- *Distance of Flight (nd)
- *Previous Departure Delay (nd)

- *Thunder (nd)
- Temperature (nd)
- Wind Speed (nd)
- *Horizon Distance (nd)
- Precipitation Time (nd)

c - categorical  v - vectorized
n - numerical  d - double

*Most Important Feature

**Performance**
- Accuracy: 84%
- **Recall: 36%**
- Precision: 63%

**Confusion Matrix**

| | | Actual: Delay | Actual: No Delay |
|---|---|---|---|
| n = 2,721,352 | | | |
| Predicted | **Delay** | TP **173,527** (36%) | FP **100,638** (5%) |
| Predicted | **No Delay** | FN **325,528** (64%) | TN **2,121,659** (95%) |

*Percentages are along Actuals axis

# Implications

## Practical Implications

- In 2017, **United** and United Express operated more than **1.6 million flights** carrying more than 148 million customers
- Assuming a similar delay rate as our dataset (~20%), about **320,000 of these flights experienced a delay**
- Using our model, we would've **correctly predicted 112,000 of these flights** to be delayed ahead of time
- With proper care taken for a predicted delay (contact customers, make schedule adjustments, etc.), **United can potentially be saving $132M**

Estimated Delay Costs/Flight

| Airlines | 2017 | 2018 |
|---|---|---|
| Alaska Airlines Inc. | $636 | $631 |
| Allegiant Air | | $1,327 |
| American Airlines Inc. | $1,227 | $1,381 |
| Delta Air Lines Inc. | $1,075 | $1,018 |
| Frontier Airlines Inc. | $1,094 | $1,608 |
| Hawaiian Airlines Inc. | $863 | $809 |
| JetBlue Airways | $1,429 | $1,413 |
| Southwest Airlines Co. | $806 | $806 |
| Spirit Air Lines | $1,025 | $980 |
| United Air Lines Inc. | $1,186 | $1,369 |

source: https://airinsight.com/more-on-us-airline-delay-costs/

# Improvements

## Future Improvements

- **Validating With Industry Expert**
  The airlines industry is a gigantic one. Professionals who have worked in the industry for decades continue to learn more about it. We can't expect ourselves to hold all the answers to predicting delays. Inviting industry experts (pilots, ATC controllers, airlines executives, etc.) in to discuss current and new features can have a huge impact.

- **Engineering New Features**
  New features that we've engineered for this project seemed to have a positive impact in training our model. We can continue to explore other features that might provide other information.

- **Addressing Variable Imbalance**
  Our dataset had an imbalance in the target variable (only about 20% of all rows had a delay flag). In future iterations, we could implement variable balancing techniques before training the model to alleviate some of the problems caused by this issue.

# Thank you!