Predicting Flight Delays

for United Airlines

August 8, 2023

By Jonathan Tran, Mamesa El, Sneha Narain, Sam Gupta
Section 2 Group 4 (Cluster 2-1)

Final Presentation

Predicting Flight Delays

for United Airlines

August 8, 2023

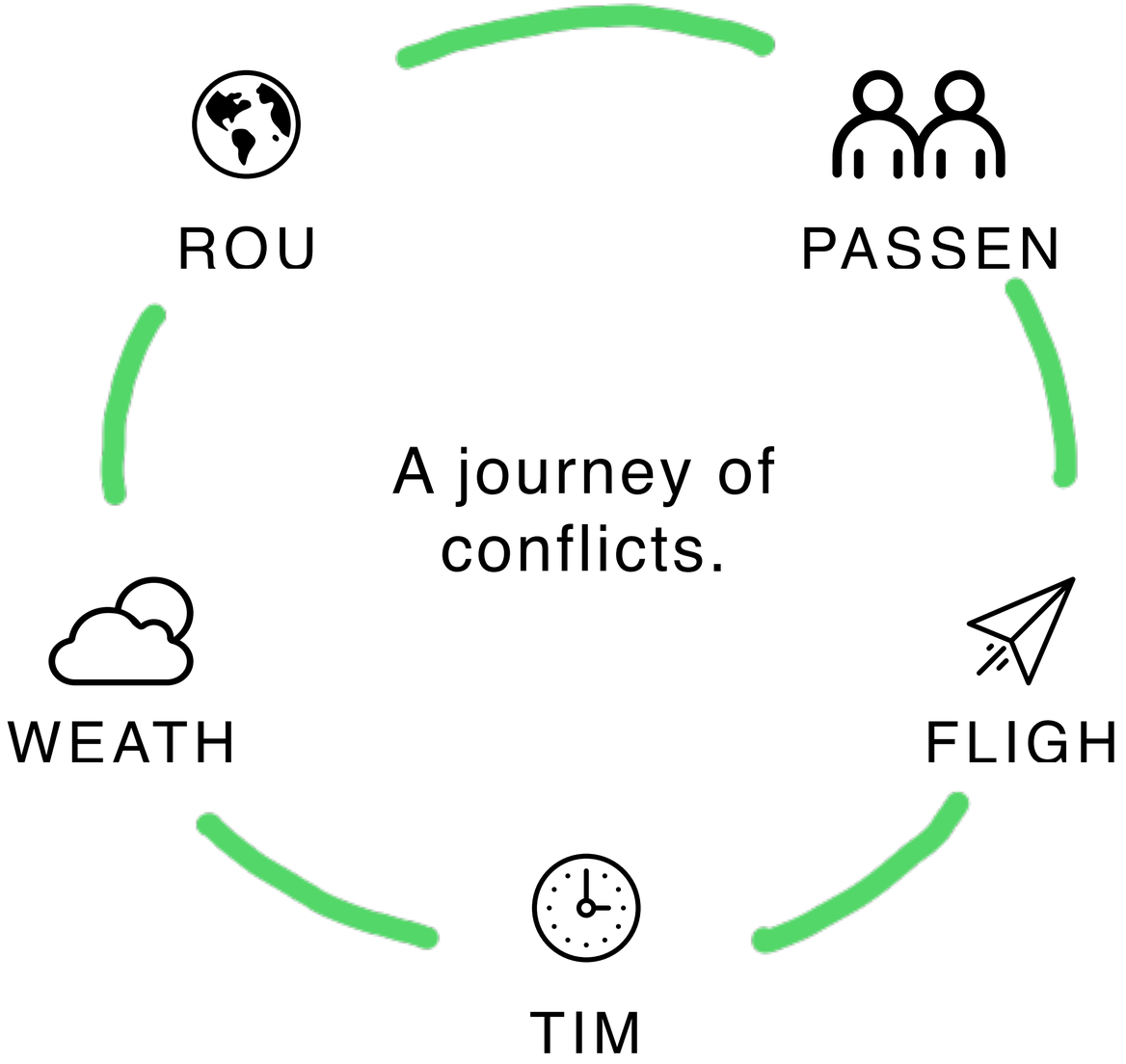
By Jonathan Tran, Mamesa El, Sneha Narain, Sam Gupta
Section 2 Group 4 (Cluster 2-1)

Final Presentation

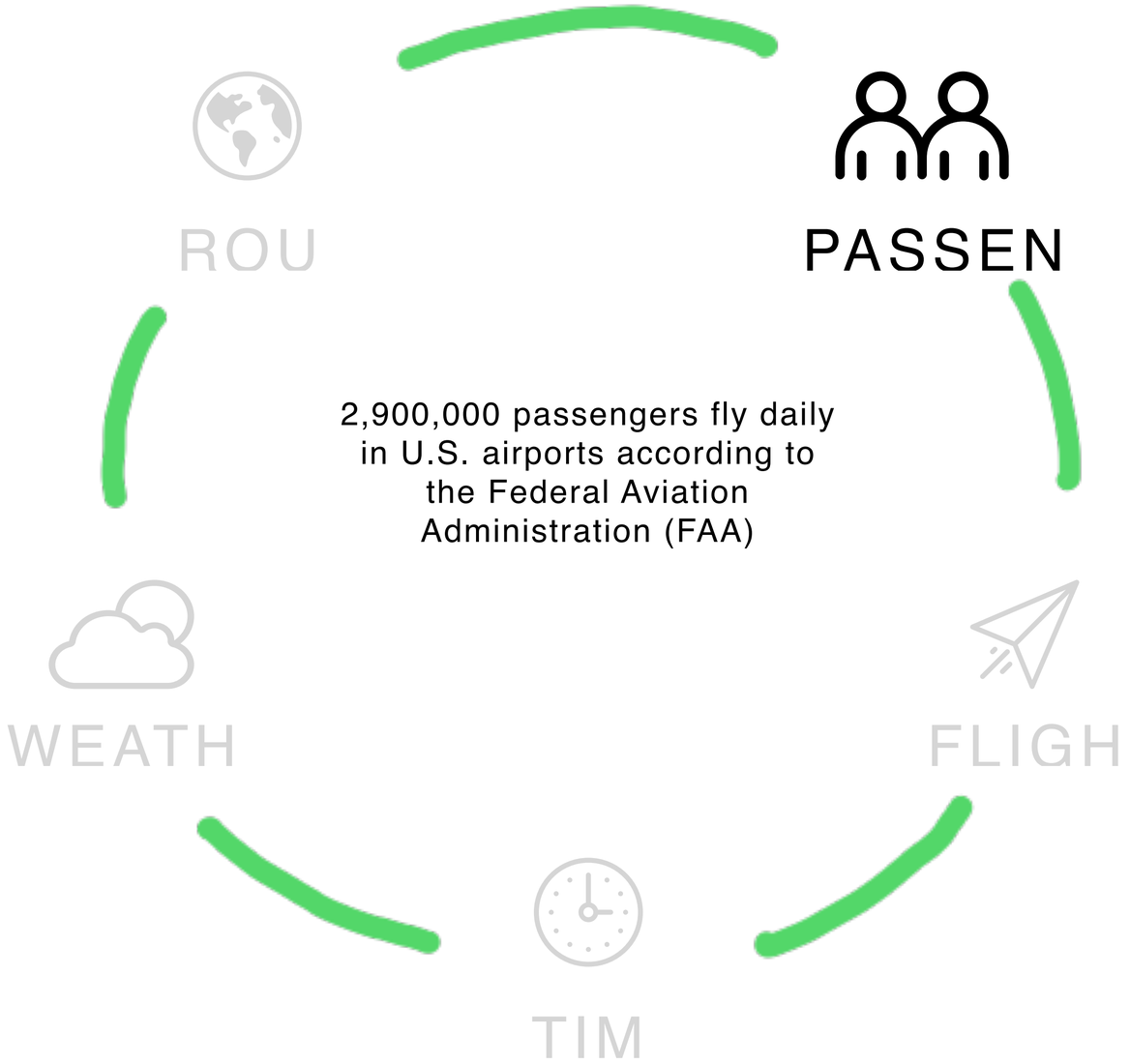
NAVIGATION

TABLE OF CONTENTS

UNDERSTANDING THE PROBLEM

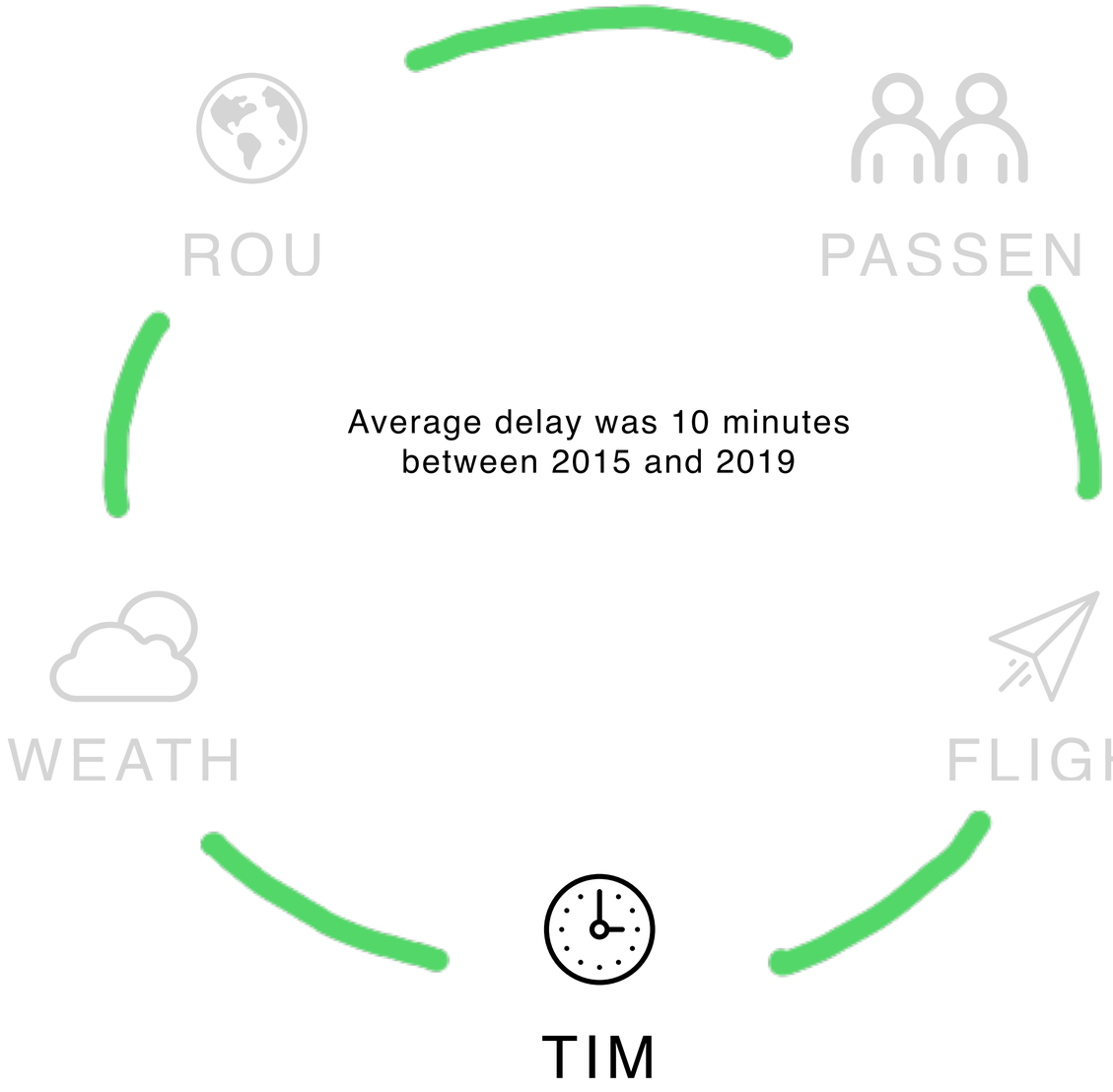


UNDERSTANDING THE PROBLEM

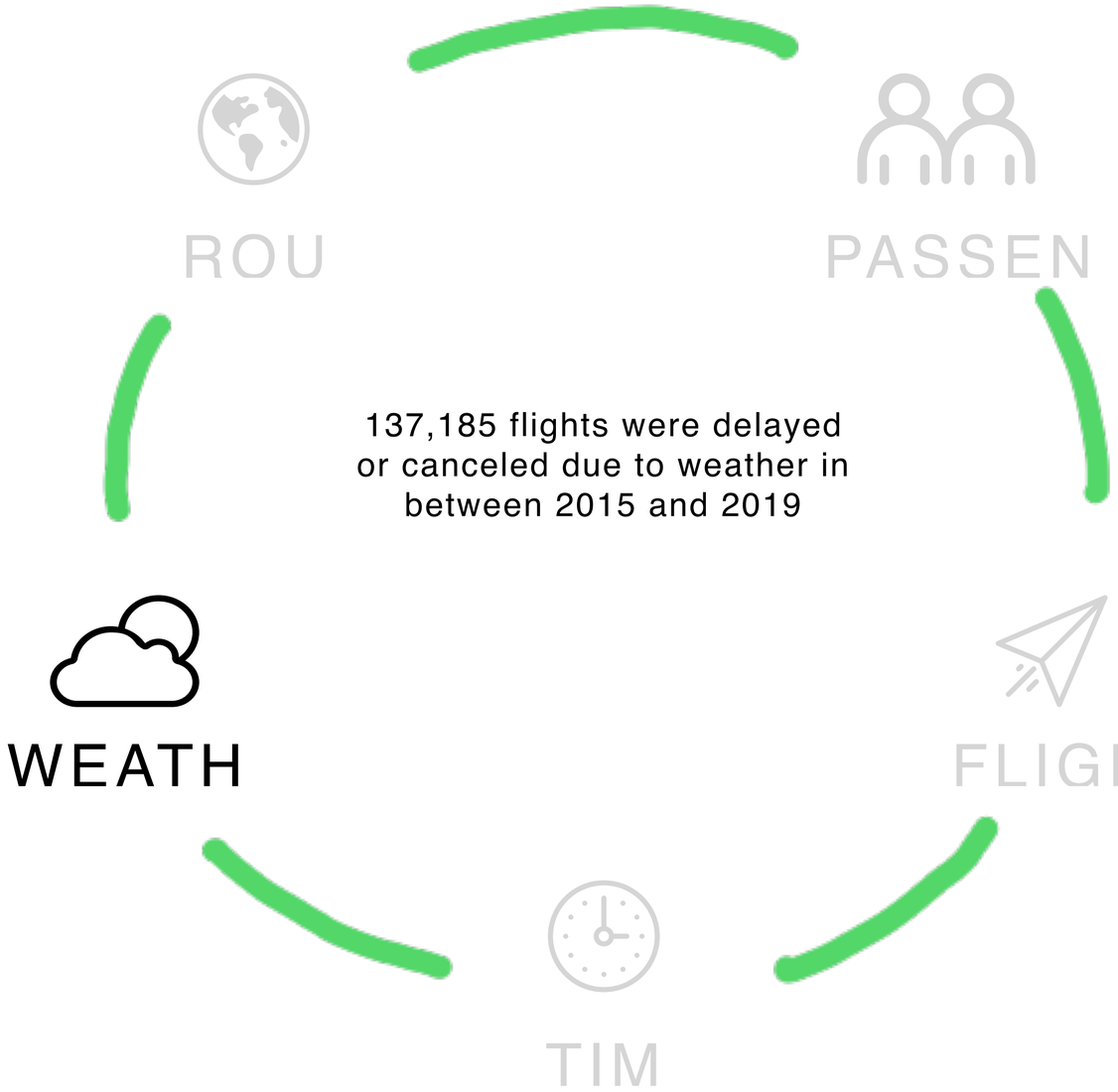


UNDERSTANDING THE PROBLEM

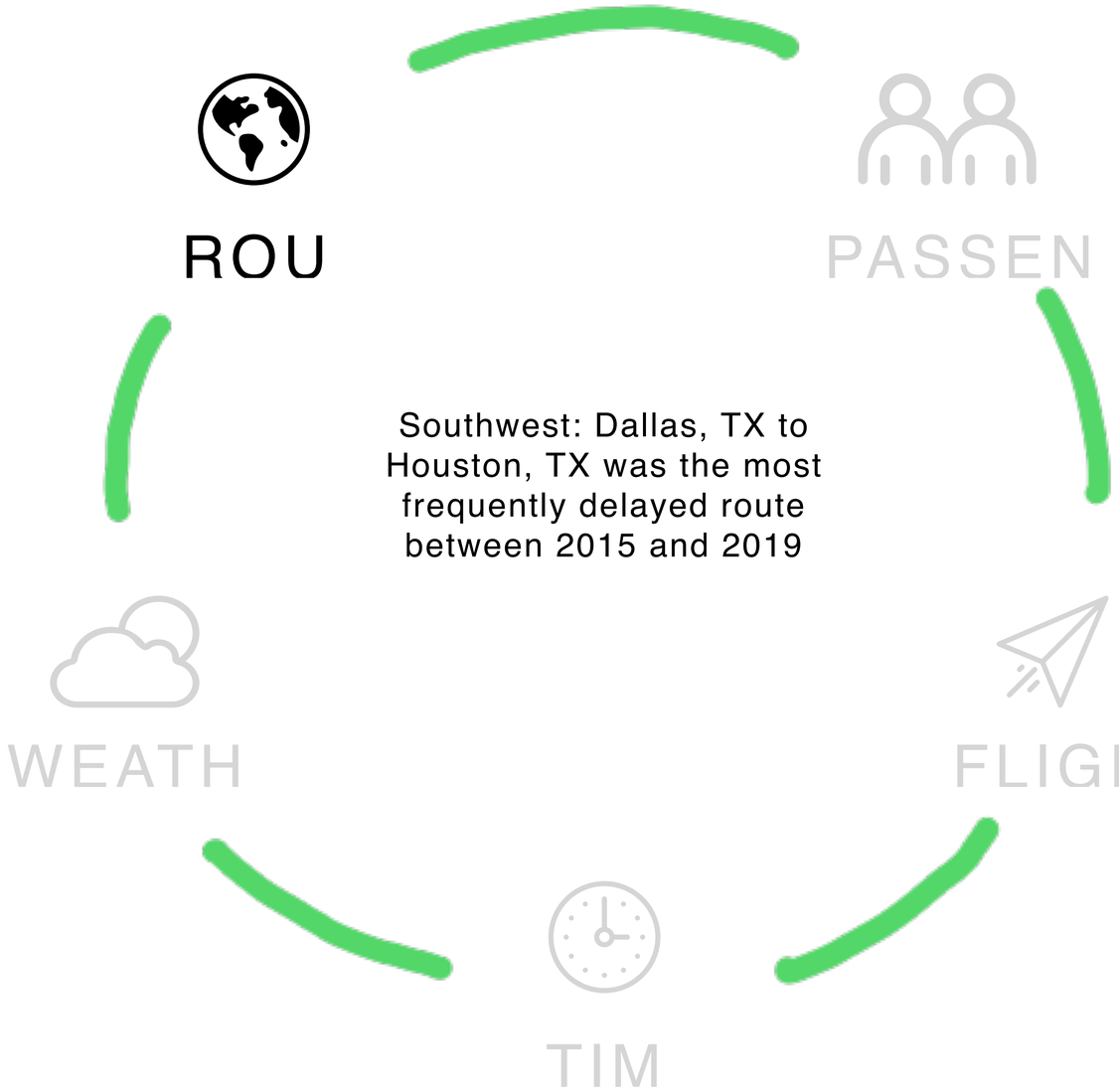
UNDERSTANDING THE PROBLEM



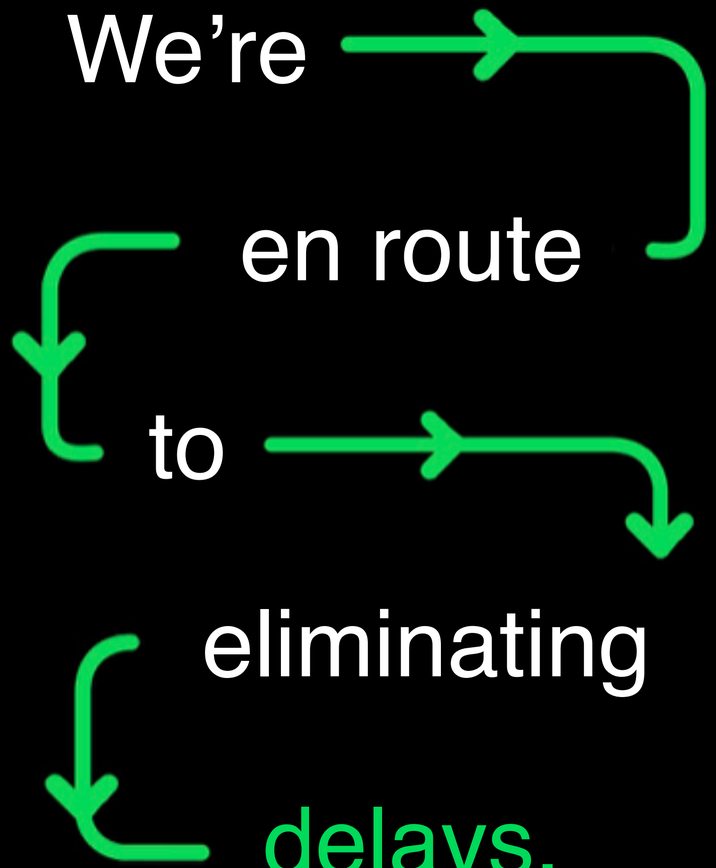
UNDERSTANDING THE PROBLEM



UNDERSTANDING THE PROBLEM



Southwest: Dallas, TX to Houston, TX was the most frequently delayed route between 2015 and 2019


We’re
en route
to
eliminating
delays.

FLIGHT INFORMATION

DATA SUMMARY

13M
ROWS OF DATA

DATA SIZE

3 Months: 1,401,363 rows

12 Months: 5,811,854 rows

60 Months: 12,926,912 rows

216 columns

29
FEATURES

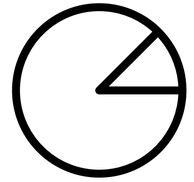
INSIGHT

Numeric: 6

Categorical: 6

17 Numeric -> 7 PCA

Data split by year where possible



60/20/20
SPLIT

TRAIN/TEST/VALID

60% Train (7,756,147 observations)

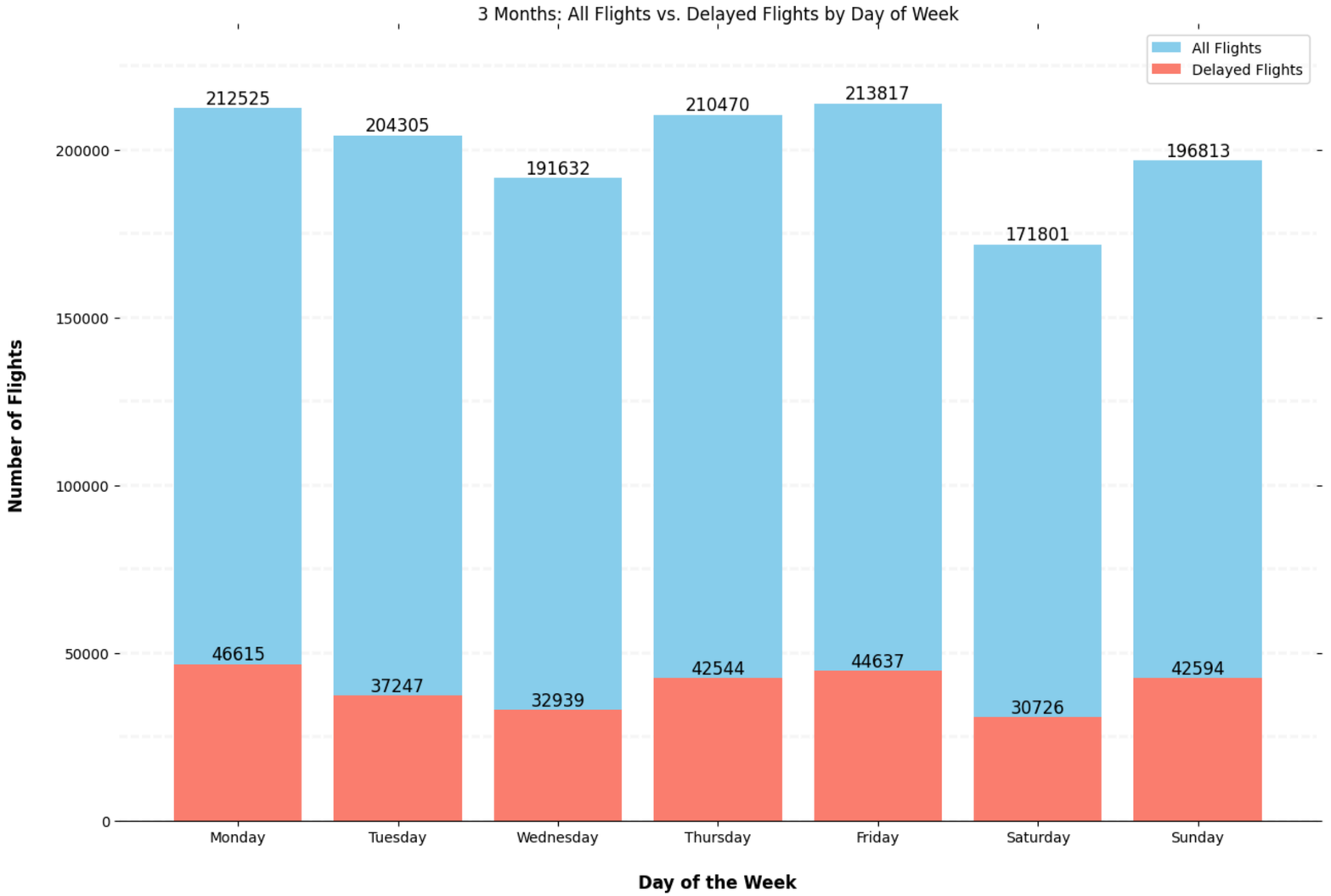
20% Test (2,585,382 observations)

20% Validation (2,585,382 observations)

For 60 Month Dataset

3 MONTH

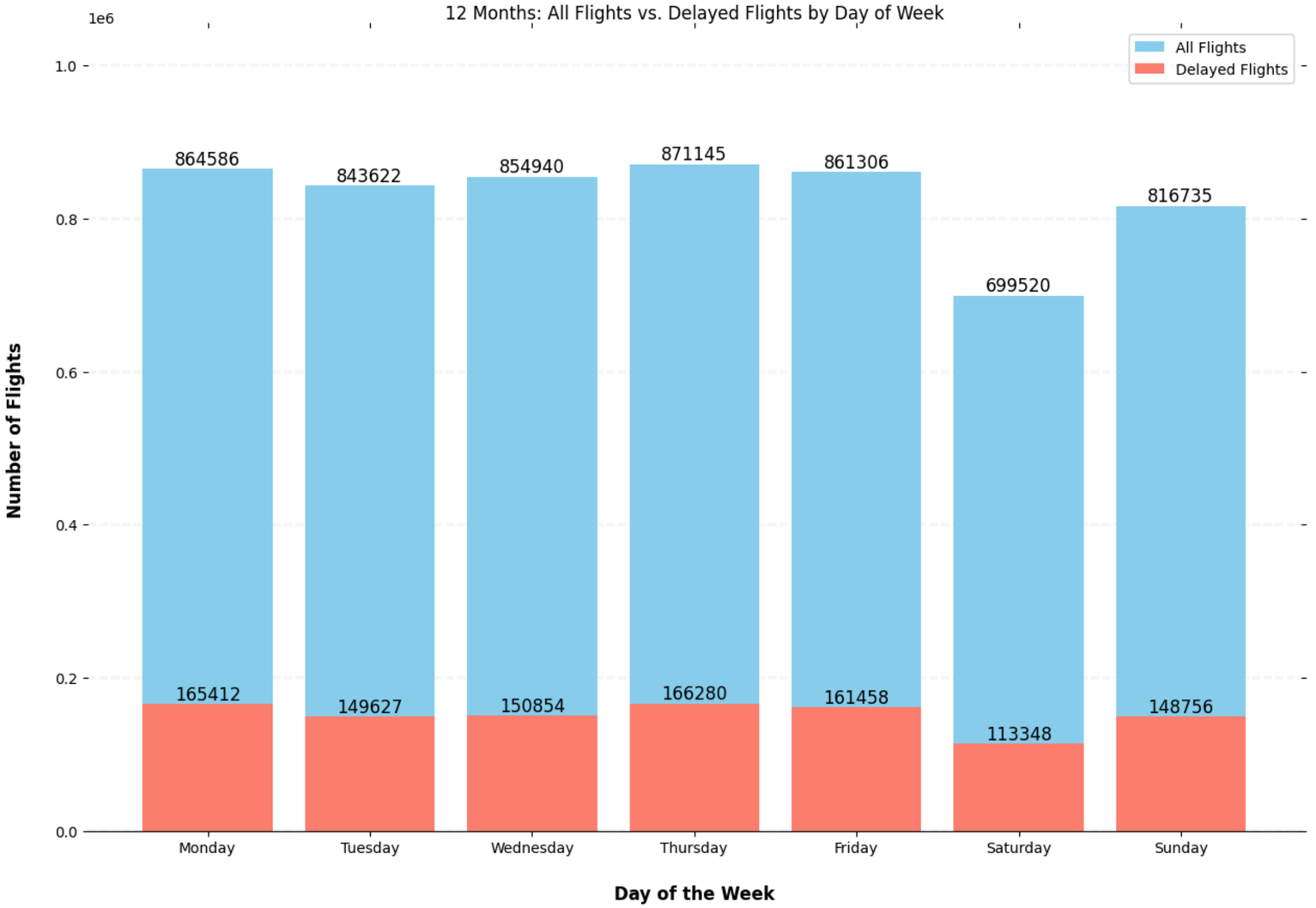
EDA



ALL FLIGHTS VS. DELAYED FLIGHTS BY DAY OF WEEK

1 2 M O N T H

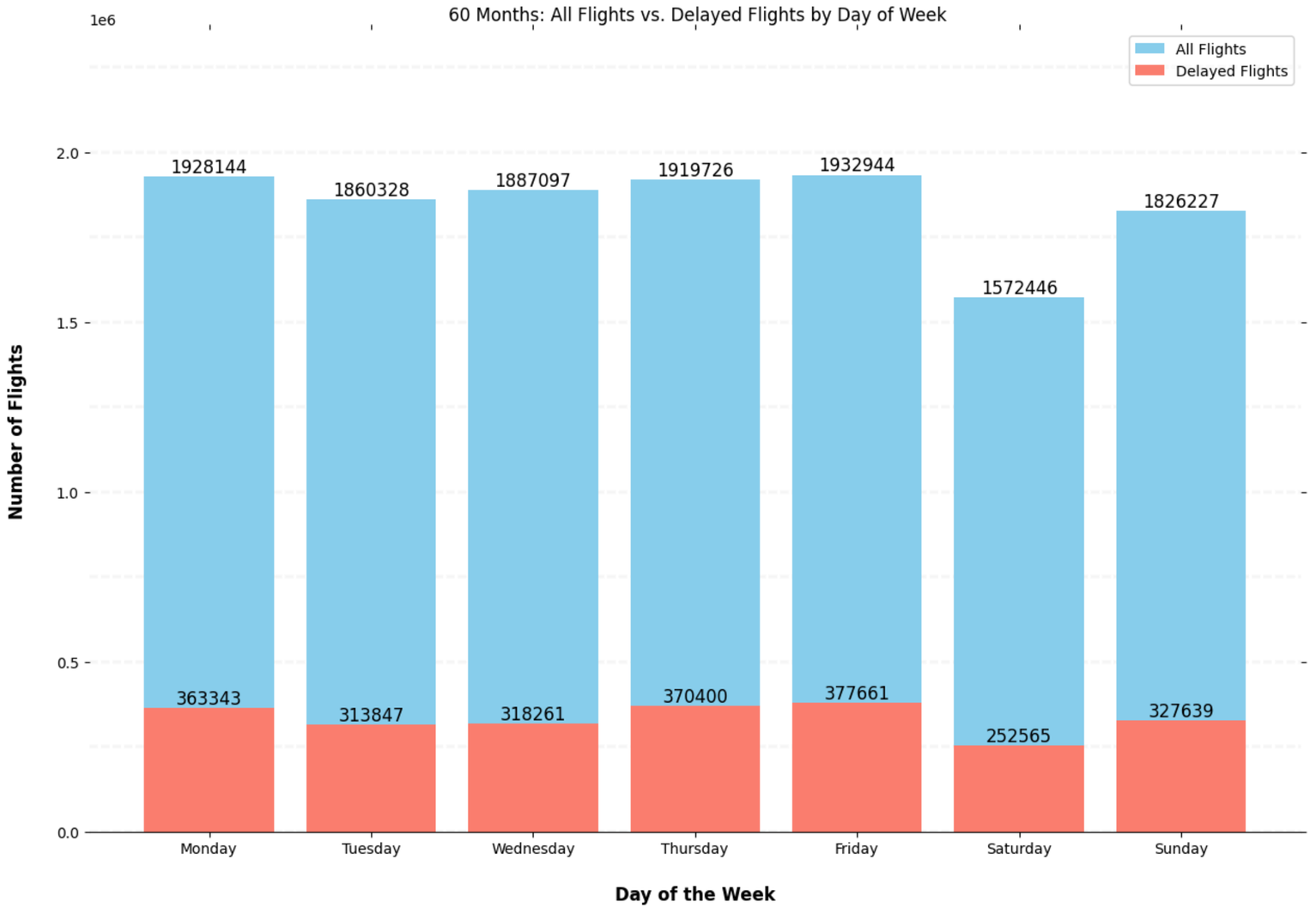
EDA



ALL FLIGHTS VS. DELAYED FLIGHTS BY DAY OF WEEK

6 0 M O N T H

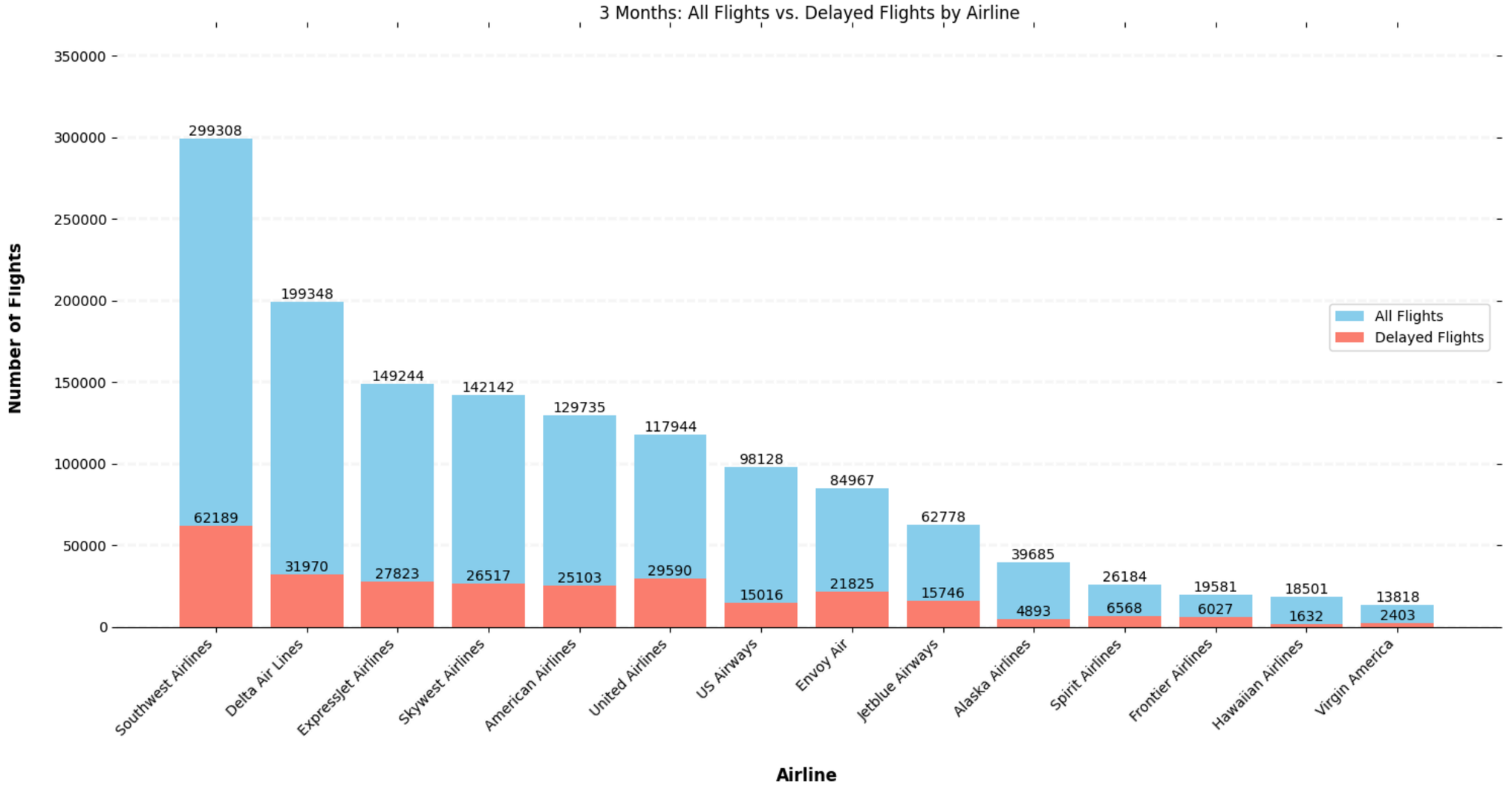
EDA



ALL FLIGHTS VS. DELAYED FLIGHTS BY DAY OF WEEK

3 MONTH

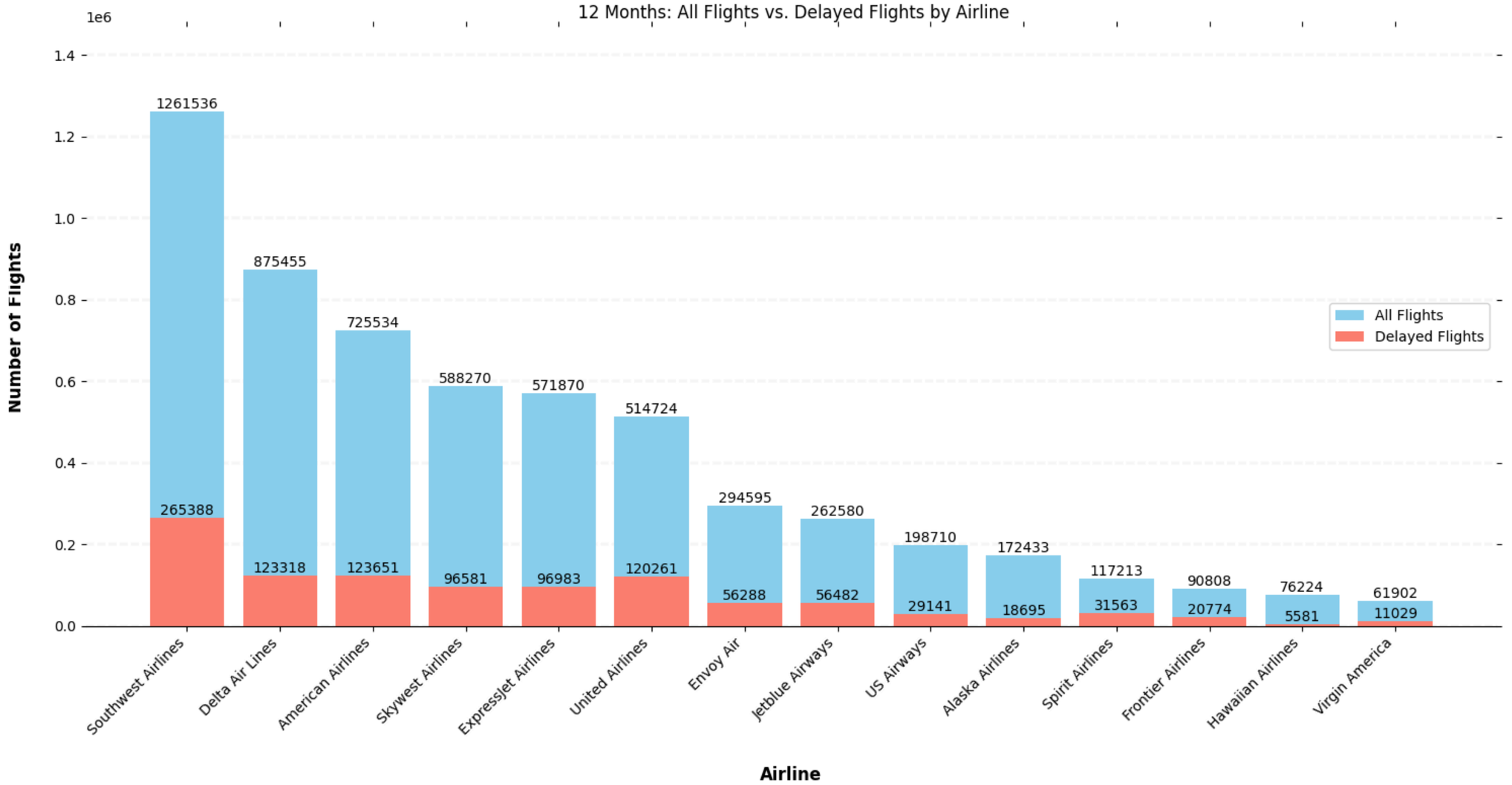
EDA



ALL FLIGHTS VS. DELAYED FLIGHTS BY AIRLINE

1 2 M O N T H

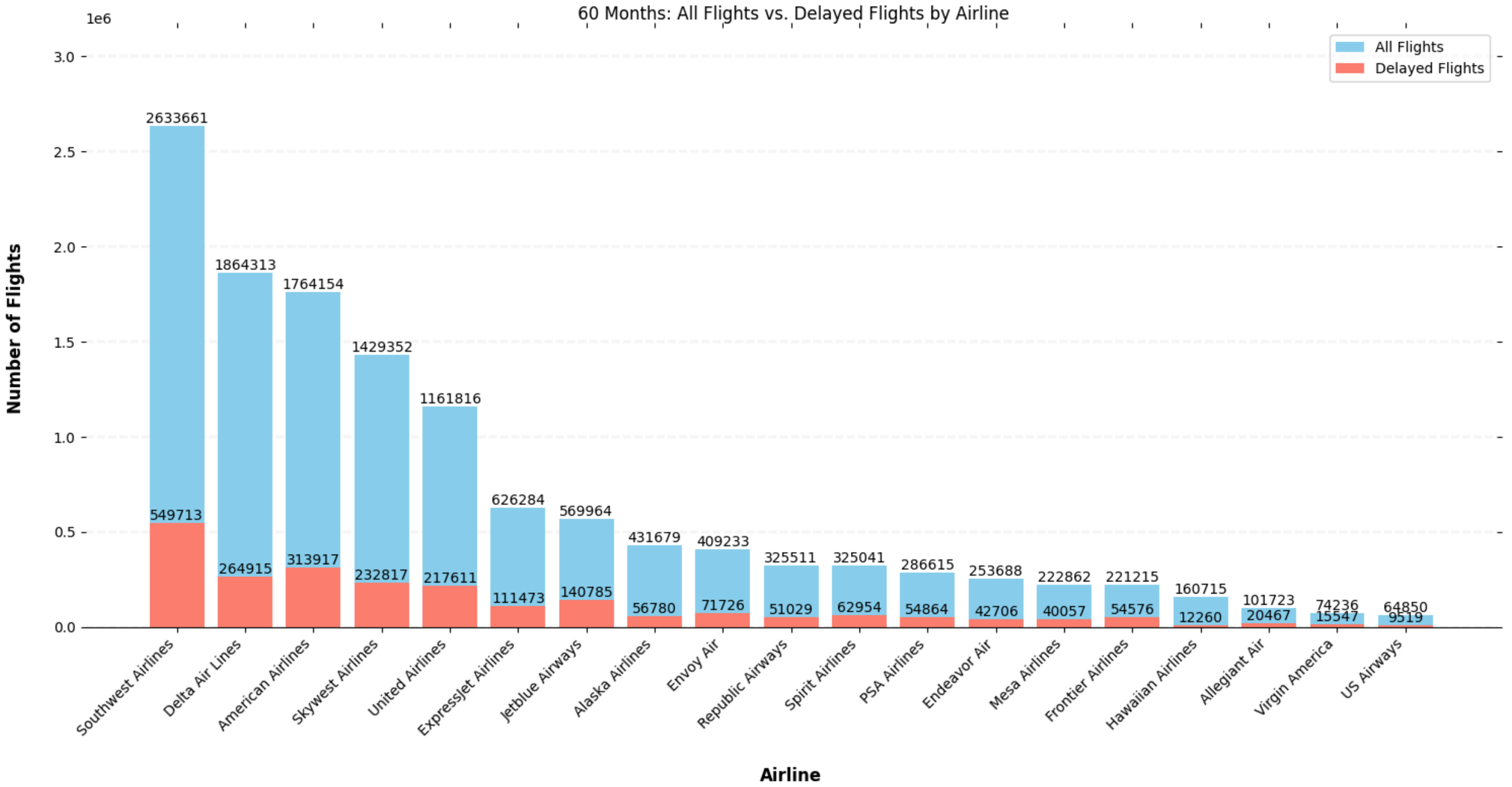
EDA



ALL FLIGHTS VS. DELAYED FLIGHTS BY AIRLINE

6 0 M O N T H

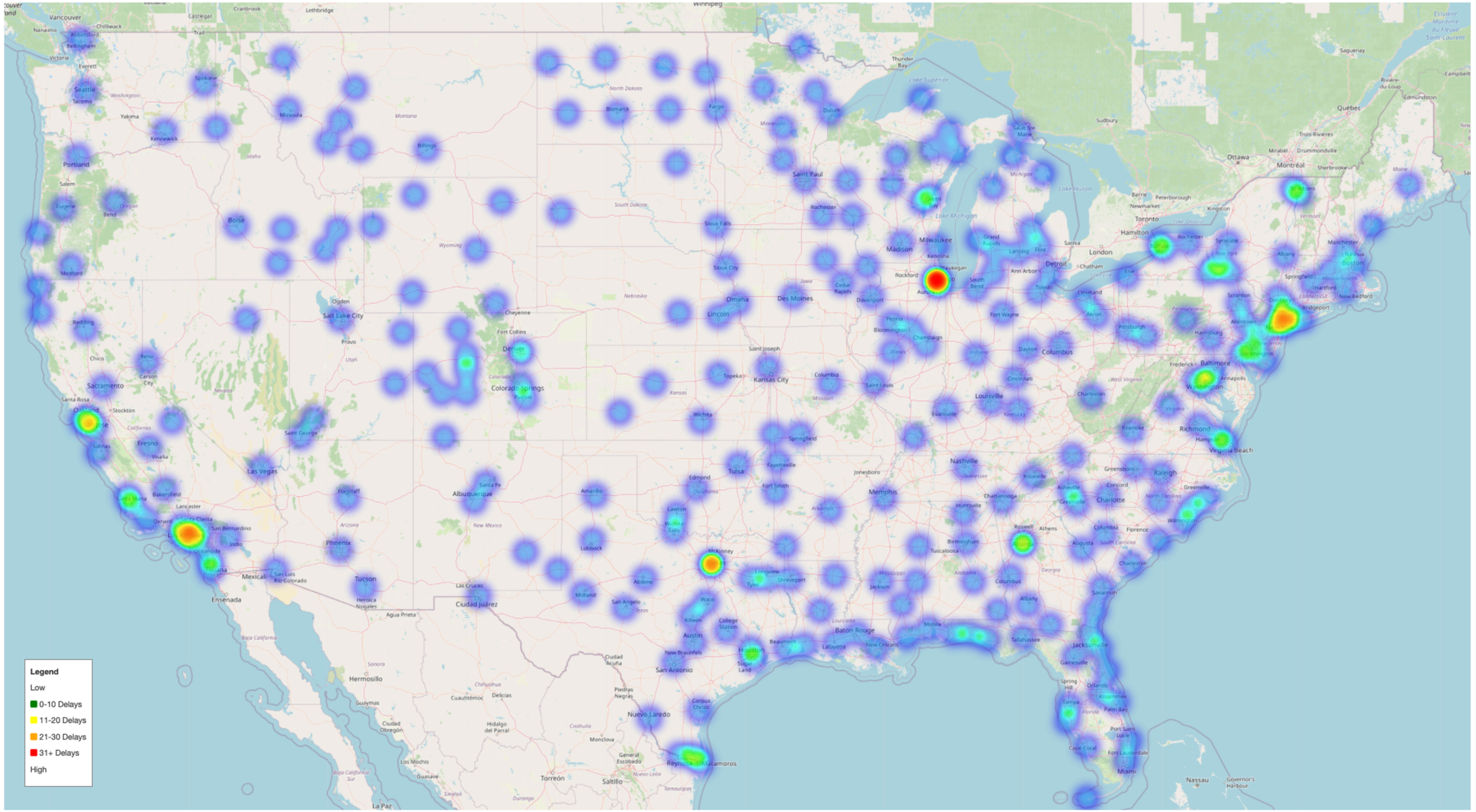
EDA



ALL FLIGHTS VS. DELAYED FLIGHTS BY AIRLINE

3 MONTH

EDA



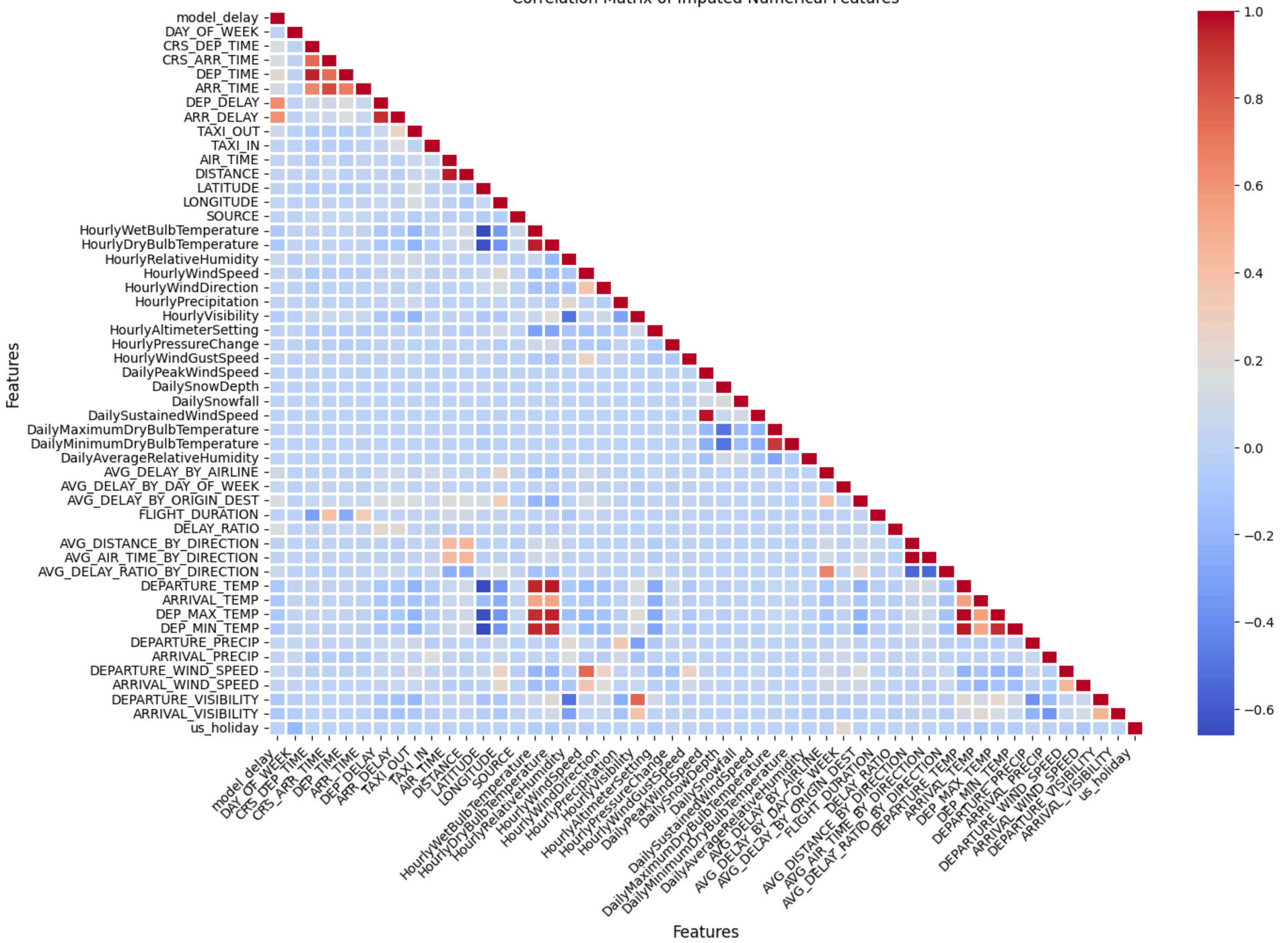
FLIGHT DEPARTURE DELAYS BY AIRPORT HEATMAP

EDA

Top 10 Delayed Routes over 60 Months

Airline	Route	Aircraft Model	Passenger Cap	Total Delays	Avg Delay
Southwest	Dallas to Houston, TX	Boeing 737-700 NG	143	6593	23 min
Southwest	Houston to Dallas, TX	Boeing 737-700 NG	143	6002	25 min
Hawaiian	Kahului (Maui) to Honolulu, HI	Boeing 717-200	128	5599	13 min
American	Dallas/Fort Worth, TX to Los Angeles, CA	Airbus 321neo	196	5002	32 min
Delta	Atlanta, GA to LaGuardia (Queens), NY	Airbus 321-200	191	4698	37 min
United	Newark, NJ to San Francisco, CA	Boeing 737-800 NG Boeing 737 MAX 8 Airbus 319-100 Airbus 320-200	166 166 126 150	4551	45 min
United	Chicago, IL to San Francisco, CA	Boeing 737-800 NG Boeing 737-900 NG Boeing 737 MAX 9 Boeing 757-300 Boeing 777-200 Airbus 320-200	166 179 179 234 276 150	4526	43 min
Southwest	Oakland to Los Angeles, CA	Boeing 737-800 NG Boeing 737 MAX 8	175 175	4492	24 min
Southwest	Los Angeles to Oakland, CA	Boeing 737-800 NG Boeing 737 MAX 8	175 175	4467	27 min
American	Dallas/Fort Worth, TX to Chicago, IL	Airbus 321-200	190	4446	36 min

TOP 10 DELAYED ROUTES


Correlation Matrix of Imputed Numerical Features
Features
model_delay
DAY_OF_WEEK
CRS_DEP_TIME
CRS_ARR_TIME
DEP_TIME
ARR_TIME
DEP_DELAY
ARR_DELAY
TAXI_OUT
TAXI_IN
AIR_TIME
DISTANCE
LATITUDE
LONGITUDE
SOURCE
HourlyWetBulbTemperature
HourlyDryBulbTemperature
HourlyRelativeHumidity
HourlyWindSpeed
HourlyWindDirection
HourlyPrecipitation
HourlyVisibility
HourlyAltimeterSetting
HourlyPressureChange
HourlyWindGustSpeed
DailyPeakWindSpeed
DailySnowDepth
DailySnowfall
DailySustainedWindSpeed
DailyMaximumDryBulbTemperature
DailyMinimumDryBulbTemperature
DailyAverageRelativeHumidity
AVG_DELAY_BY_AIRLINE
AVG_DELAY_BY_DAY_OF_WEEK
AVG_DELAY_BY_ORIGIN_DEST
FLIGHT_DURATION
DELAY_RATIO
AVG_DISTANCE_BY_DIRECTION
AVG_AIR_TIME_BY_DIRECTION
AVG_DELAY_RATIO_BY_DIRECTION
DEPARTURE_TEMP
ARRIVAL_TEMP
DEP_MAX_TEMP
DEP_MIN_TEMP
DEPARTURE_PRECIP
ARRIVAL_PRECIP
DEPARTURE_WIND_SPEED
ARRIVAL_WIND_SPEED
DEPARTURE_VISIBILITY
ARRIVAL_VISIBILITY
us_holiday
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

BEHIND THE SCENES

FEATURE ENGINEERING

Feature Count	Feature Type	Feature Name	Description
1	Categorical	DISTANCE_BIN	Categorize flight distance into bins (short, medium, and long-haul)
2	Categorical	DIRECTION	Categorize flig direction based on origin and destination airport coordinates
3	Numeric	AVG_DELAY_BY_AIRLINE	Calculate average delay by airline
4	Numeric	AVG_DELAY_BY_DAY_OF_WEEK	Calculate average delay by day of the week
5	Numeric	AVG_DELAY_BY_ORIGIN_DEST	Calculate delay by route
6	Numeric	CARRIER_ORIGIN_AIRPORT_FLIGHT_FREQUENCY	Calculate flight frequency by airline and airport
7	Numeric	FLIGHT_COUNT_BY_ORIGIN_DEST	Count number of flights by origin and destination
8	Numeric	DELAY_RATIO	Calculate delay ratio
9	Numeric	FLIGHT_FREQUENCY_BY_DAY	Calculate flight frequency by day of the week
10	Numeric	FLIGHTS_BY_DIRECTION	Calculate the number of flights for each carrier in each direction
11	Numeric	AVG_DISTANCE_BY_DIRECTION	Calculate the average distance of flights for each carrier in each direction
12	Numeric	AVG_AIR_TIME_BY_DIRECTION	Calculate the average air time of flights for each carrier in each direction
13	Numeric	AVG_DELAY_RATIO_BY_DIRECTION	Calculate the average delay ratio for each carrier in each direction
14	Numeric	DEPARTURE_TEMP, ARRIVAL_TEMP, DEP_MAX_TEMP, DEP_MIN_TEMP	Average, maximum, and minimum temperature during departure and arrival times
15	Numeric	DEPARTURE_PRECIP, ARRIVAL_PRECIP	Total precipitation amount during departure and arrival times
16	Numeric	DEPARTURE_WIND_SPEED, ARRIVAL_WIND_SPEED	Average wind speed during departure and arrival times
17	Numeric	DEPARTURE_VISIBILITY, ARRIVAL_VISIBILITY	Average visibility during departure and arrival times
18	Categorical	flight_time_category	Categorize time of day for flight departures
19	Graph	AIRPORT_CONNECTIVITY	Number of direct flights from each airport to other airports
20	Graph	AIRPORT_DEGREE_CENTRALITY	Measure the importance of an airport based on the number of connections it has
21	Graph	FLIGHT_DENSITY	Number of flights at each airport
22	Graph	AVG_FLIGHT_DISTANCE	Average flight distance from each airport
23	Graph	FLIGHT_FREQUENCY_BY_AIRLINE	Calculate the frequency of flights for each airline
24	Numeric	previous_flightnum_delay_ct	Calculate the number of delays of each flight number over a rolling window
25	Categorical	us_holiday	Categorize if the flight took place during a U.S. Holiday

DATA TYPE CONVERSION

- The features in the original joint table are string type
- Feature types are manually cast based on the original flight, weather, and station data.

FEATURE CREATION

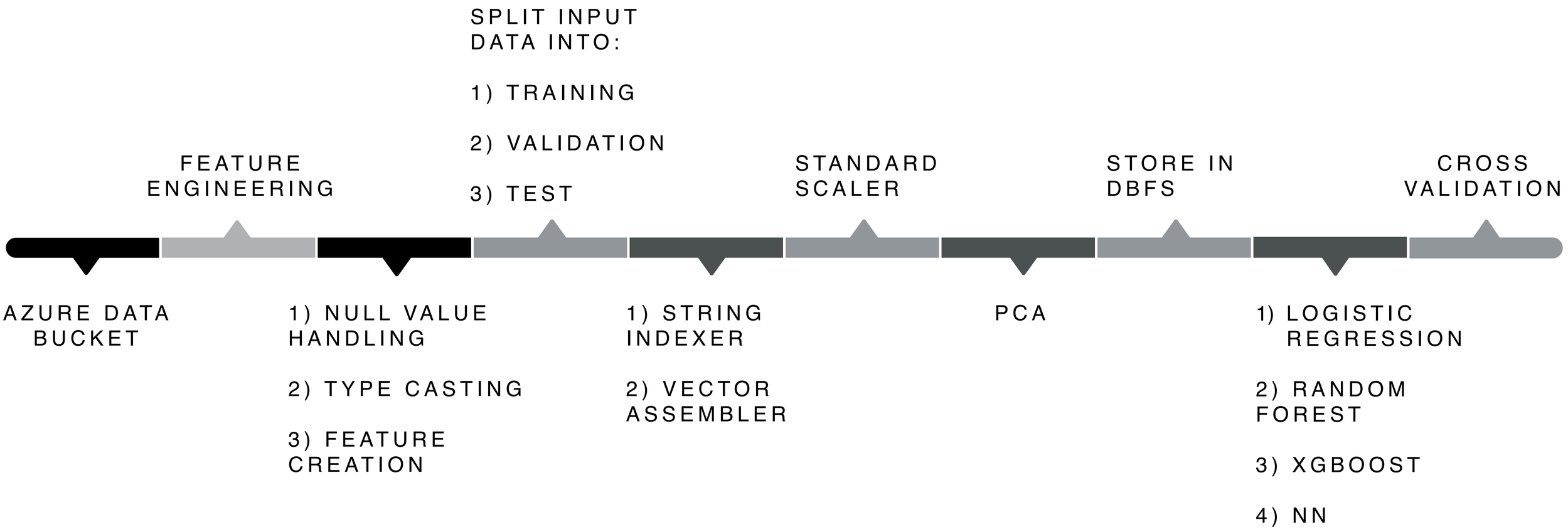
- Created 25 features
- 5 Categorical, 5 Graph, 15 Numeric

NULL VALUE HANDLING

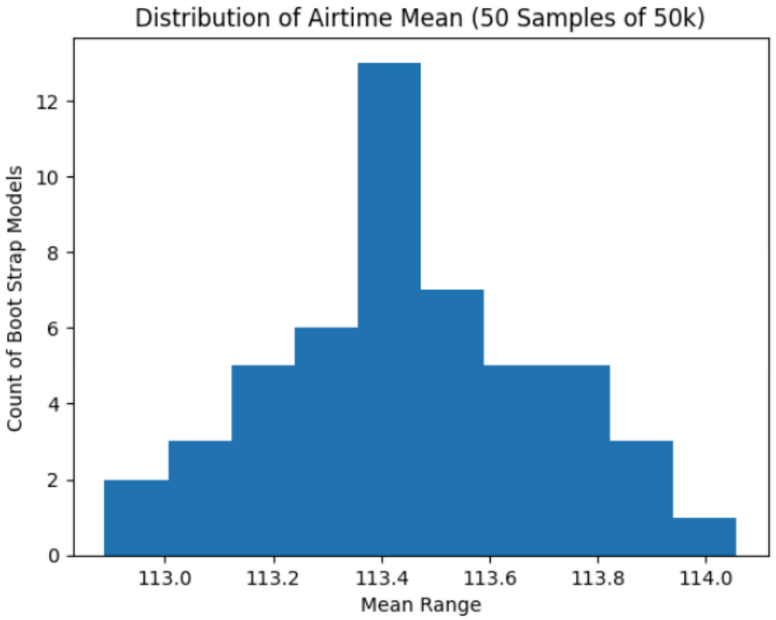
- Drop any features with a null count over 1 million
- Impute the remaining features will low null counts
- The 60 months final dataset has a total of 12.9 million rows and 29 columns

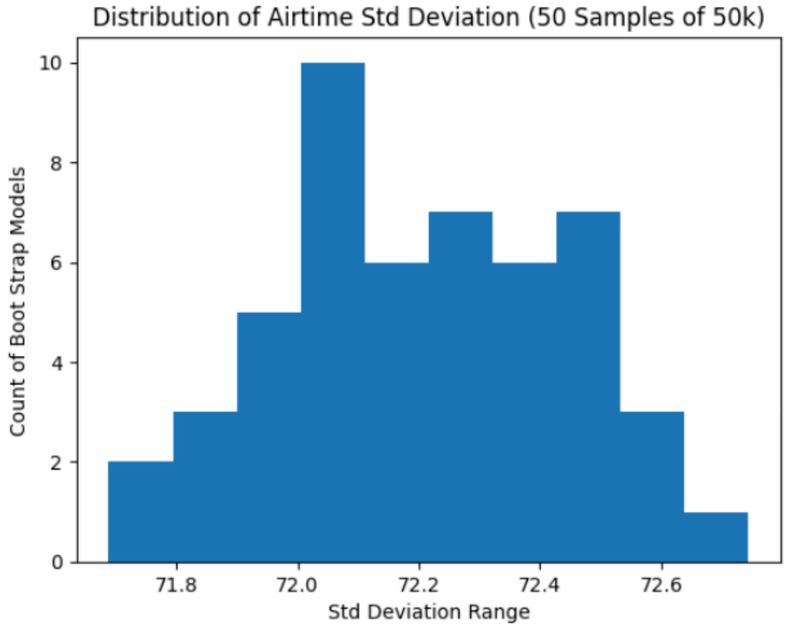
HOW IT WORKS

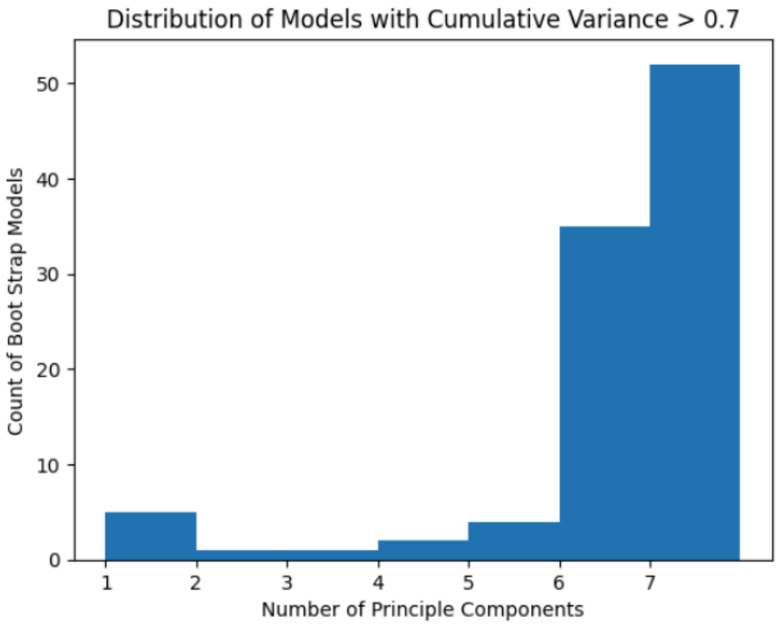
PIPELINE



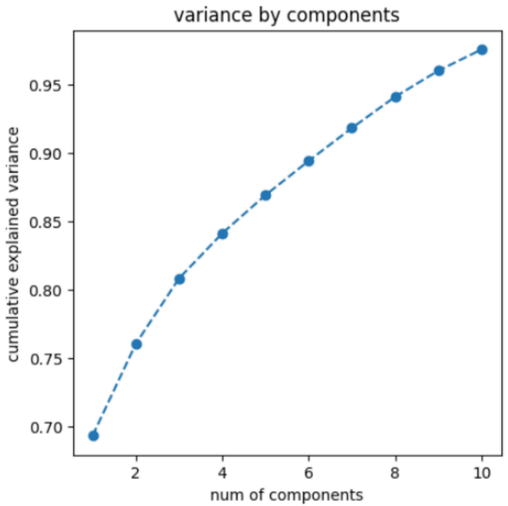
BOOTSTRAPPING METHODOLOGY







- PRESCALE DATA OUTSIDE FOLDS TO IMPROVE RUNTIME
- DISTRIBUTIONS ARE SIMILAR ACROSS MANY SAMPLES
- SIMILAR METHODOLOGY TO APPLY PCA ELBOW METHOD



BOOTSTRAPPING CODE

```python
# SQL loop to generate query for each numeric column
def get_mean_stddev(input_df, input_columns, sample_size, iteration_num = None, seed_num = None):
    sql_str = ''
    for i in range(len(input_columns)):
        # sql_str = "(select "
        sql_str = sql_str + "(select "
        if iteration_num:
            sql_str =  sql_str + str(iteration_num) + " as iteration_number, "
        if seed_num:
            sql_str =  sql_str + str(seed_num) + " as Random_seed, "
        sql_str =  sql_str + str(sample_size) + " as sample_size, "
        sql_str = sql_str + "'"+ input_columns[i] + "'" + ' as field_name, '
        sql_str = sql_str + 'avg('+ input_columns[i] + ') as field_avg, '
        sql_str = sql_str + 'stddev('+ input_columns[i] + ') as field_stddev'
        sql_str = sql_str + " from tmp_main group by field_name) UNION "

    sql_str = sql_str[:-7]
    input_df.createOrReplaceTempView("tmp_main")
    single_iteration_stats = spark.sql(sql_str)
    pandasDF = single_iteration_stats.toPandas()

    return pandasDF
```

```python
def return_stats(input_df, input_df_length, sample_size, column_names, iteration_num = None):

    # generate a bootstrap sample
    random_value = random.randint(1,10000)
    bootstrap_df = build_bootstrap_df(input_df, input_df_length, sample_size, input_seed = random_value).cache()

    # return a df of mean and standard deviaion
    stats_df = get_mean_stddev(bootstrap_df, column_names,sample_size, iteration_num)
    return stats_df


bootstrap_stat_collection = pd.DataFrame()
output_name = 'bootstrap_numerics'
# loop through iterations of models
for i in range(50):

    # generate a bootstrap df and append to holder
    tmp_df = return_stats(model_df,model_df_ct.value,50000,non_pca_num,str(i))
    bootstrap_stat_collection = pd.concat([bootstrap_stat_collection, tmp_df])
    # save to storage
    spark.createDataFrame(bootstrap_stat_collection).write.mode("overwrite").saveAsTable(output_name)
```

Logistic Regression

50%

AUC

15%

PR

12

```
6  search_space = {
7      "maxIter": hp.quniform("maxIter", 10,200,10),
8      "regParam": hp.quniform("regParam", 0.1,0.5,0.01,),
9      "elasticNetParam": hp.quniform("elasticNetParam", 0.0,1,0.1)
10 }
```

100%

Recall

85%

Precision

78%

F1 Score

85%

Accuracy

RANDOM FOREST

66%

AUC

25%

PR

12

```
search_space = {
    "numTrees": hp.quniform("numTrees", 5, 15, 1),
    "maxDepth": hp.quniform("maxDepth", 15, 25, 1),
    "maxBins": hp.quniform("maxBins", 32, 96, 8),
    "impurity": hp.choice("impurity", ["entropy"])
}
```

99%

Recall

85%

Precision

79%

F1 Score

83%

Accuracy

XGBOOST

70%

AUC

33%

PR

12

```
search_space = {
    "num_workers": hp.quniform("num_workers", 3,4,1),
    "n_estimators": hp.quniform("n_estimators", 90,120.0,10.0),
    "max_bin": hp.quniform("max_bin", 20.0,30.0,3.0),
    "max_depth": hp.quniform("max_depth", 6,16,2),
    "learning_rate": hp.quniform("learning_rate", 0.2,0.6,0.1),
    "max_leaves": hp.quniform("max_leaves", 6,10,1),
    "gamma": hp.quniform("gamma", 0,20,2),
    "scale_pos_weight": hp.quniform("scale_pos_weight", 1.2,1.4,0.02)
}
```

94%

Recall

84%

Precision

77%

F1 Score

80%

Accuracy

Multilayer Perceptron (NN)

67%

AUC

27%

PR


```
search_space = {
    "maxIter": hp.quniform("maxIter", 50, 200, 25),
    "layers": hp.choice('layers', [[feature_size, 4, 2], [feature_size, 4, 2, 2]]),
    "stepSize": hp.quniform("stepSize", 0.1, 1, 0.1),
    "solver": hp.choice('solver', ['gd', 'l-bfgs']),
    'blockSize': hp.quniform('blockSize', 16, 128, 16)
}
```

97%

Recall

86%

Precision

80%

F1 Score

84%

Accuracy

3 MONTH DATASET

PERFORMANCE METRICS

Model	3 Month	12 Month	60 Month	60 Month Heldout
Logistic Regression	Accuracy: 80% Recall: 95.5% Precision: 82.6% F1 Score: 75.7% AUC: 68.6% PR: 32.3%	Accuracy: 84.9% Recall: 100% Precision: 84.9% F1 Score: 78% AUC: 50% PR: 15.1%	-	-
Random Forest	Accuracy: 79.19% Recall: 99.52% Precision: 82.81% F1 Score: 75.71% AUC: 67.26% PR: 31.29%	Accuracy: 83.29% Recall: 99.60% Precision: 85.88% F1 Score: 79.54% AUC: 66.17% PR: 25.17%	-	-
XGBoost	Accuracy: 75.6% Recall: 91.0% Precision: 84.6% F1 Score: 75.8% AUC: 70.1% PR: 34.4%	Accuracy: 80.5% Recall: 93.8% Precision: 84.2% F1 Score: 80.0% AUC: 69.6% PR: 33.6%	Accuracy: 80.7% Recall: 93.1% Precision: 84.8% F1 Score: 77.6% AUC: 71.1% PR: 35.1%	Accuracy: 80.2% Recall: 95.2% Precision: 84.2% F1 Score: 77.2% AUC: 69.2% PR: 33.6%
Multilayer Perceptron (NN)	Accuracy: - Recall: - Precision: - F1 Score: 75.6% AUC: - PR: -	Accuracy: 83.8% Recall: 96.9% Precision: 85.9% F1 Score: 79.7% AUC: 67.4% PR: 26.5%	Accuracy: 82% Recall: 99.4% Precision: 82.3% F1 Score: 74.9% AUC: 69% PR: 33.1%	-

ACCURACY

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

RECALL

$$Recall = \frac{TP}{TP + FN}$$

PRECISION

$$Precision = \frac{TP}{TP + FP}$$

F1 SCORE

$$F1 = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$

NEXT STEPS

CONCLUSION

- XG BOOST PERFORMED BEST: F1 AND RUNTIME
- IMBALANCED DATA HANDLING - RESAMPLING, PARAMETER TUNING
- PRESCALING BASED ON BOOTSTRAP SUCCESSFUL
- CONSIDER ADDITIONAL EVALUATION METRICS - PR CURVES
- UNITED CAN USE MODEL TO IDENTIFY TRUE POSITIVES

THANK YOU

APPENDIX

BEHIND THE SCENES

FEATURE IMPORTANCE