Making an Airline Delay Prediction Model

Nick Van Bergen • DSI0927 • 23 December, 2021

Problem:

Airline delays cost the consumer time and money.

Solution:

Use A.I. to predict a flight will be delayed.

Challenges:

- Large volume of data.
- Many factors can contribute to a delay making analysis and predictions difficult.

Benefit:

Help a customer plan a trip or buy insurance!

Deployment:

An App! ...

Eventually!!!

Deliverables

Today: A model that works

- Used XGBoost to find and guess if a flight will be delayed.

Next: A beta version of an app.

- A web based app that can tell you if you are likely to be delayed.
- Stretch: a model and app that will predict a delay length (in bins of time) and suggest alternatives.

Building the prediction model .

Definitions.

1. Delay:

   a. Any flight that arrives 15 later than original scheduled arrival time.

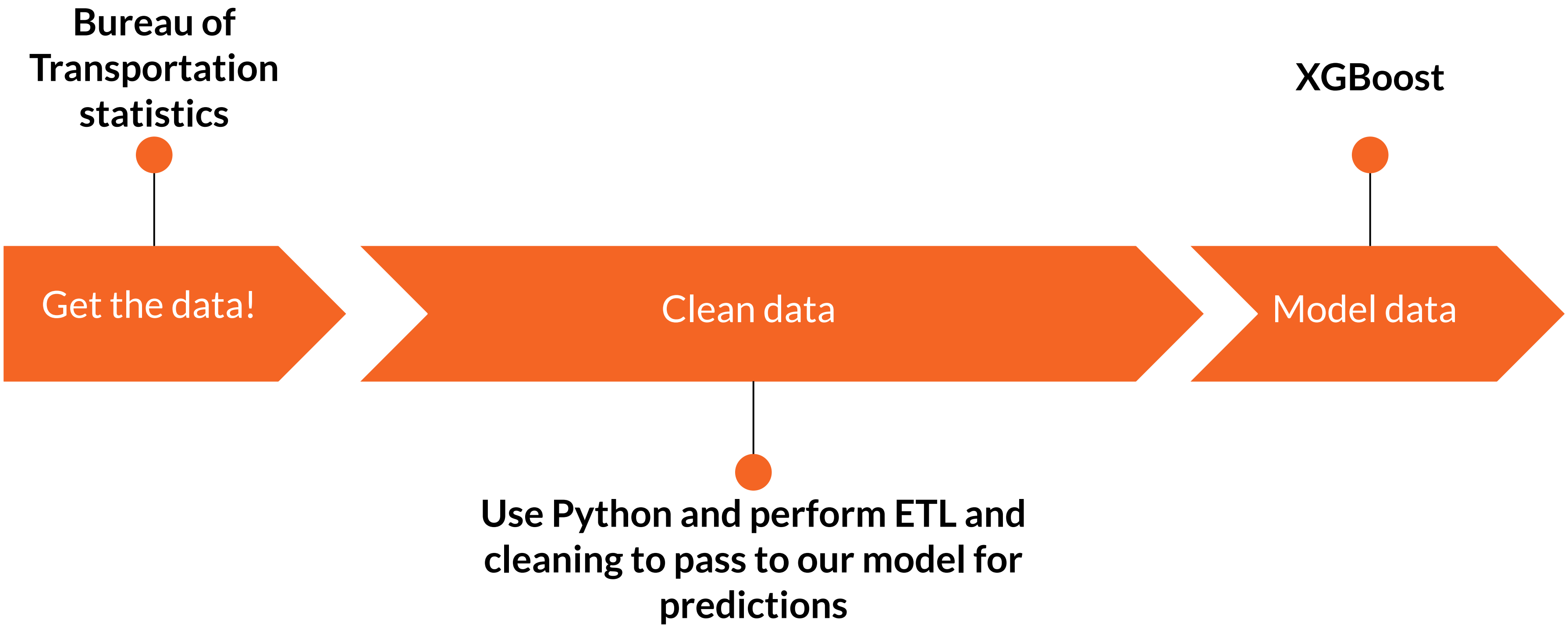
2. Arrival:

   a. Aircraft is parked, plugged into power, brakes are armed and the door is open.

What we want our model to do.

Predict if a flight will be delayed. Yes/No

Building a prediction model in three easy steps!



Getting the Data

Data set statistics.

Initial Scrape

- 6 years of flight history starting from August 2021 to January 2016
- 68 individual CSV’s. Each ~110mb files.
- 34 variable columns
- 34,409,230 observations

Cleaning subset

- 50/50 delayed vs on-time observations
- Random Sampling to 25% of original population
- 2, 798, 138 observations

Data exploration goals

Intuition vs Reality

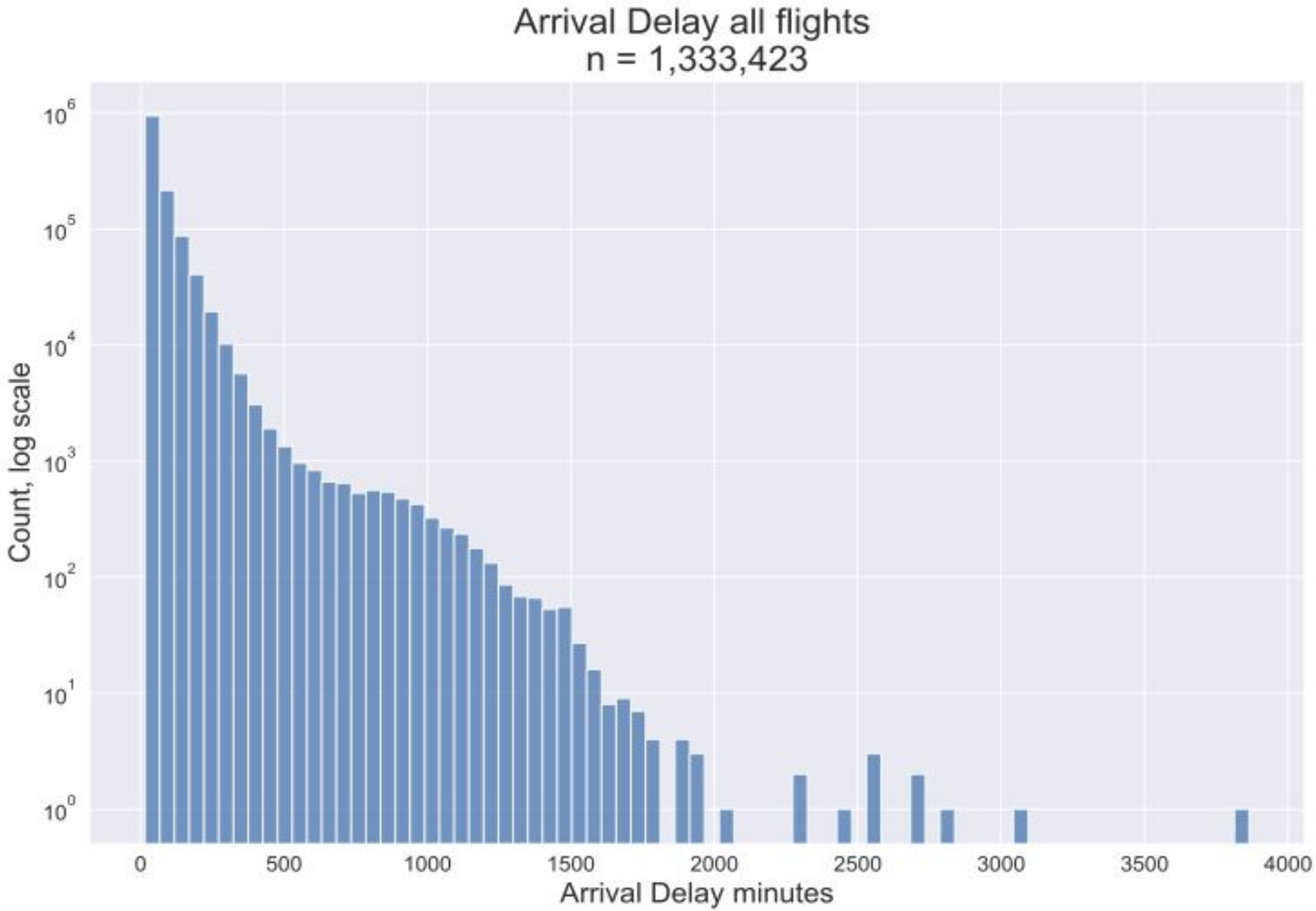
- See if our intuition holds up to the reported observations coming in.
- Larger airlines delayed more than smaller airlines.
- Busier airports delayed more than not busier airports.

Glean any fast facts

- We want to know how data is being presented and see if we can make meaningful engineered features to improve our model's ability to make predictions.

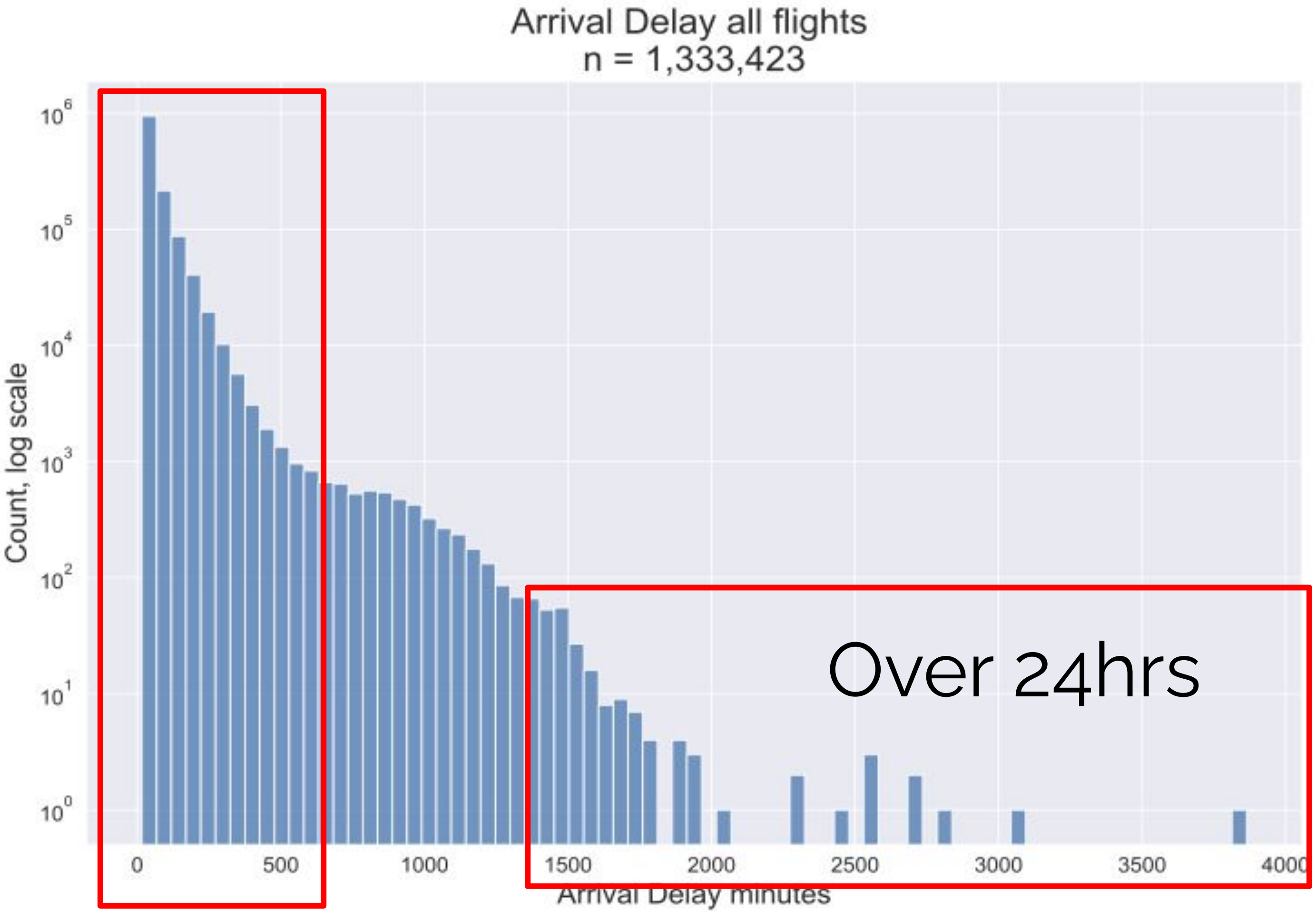
Exploratory Highlights

Exploratory Analysis Highlights



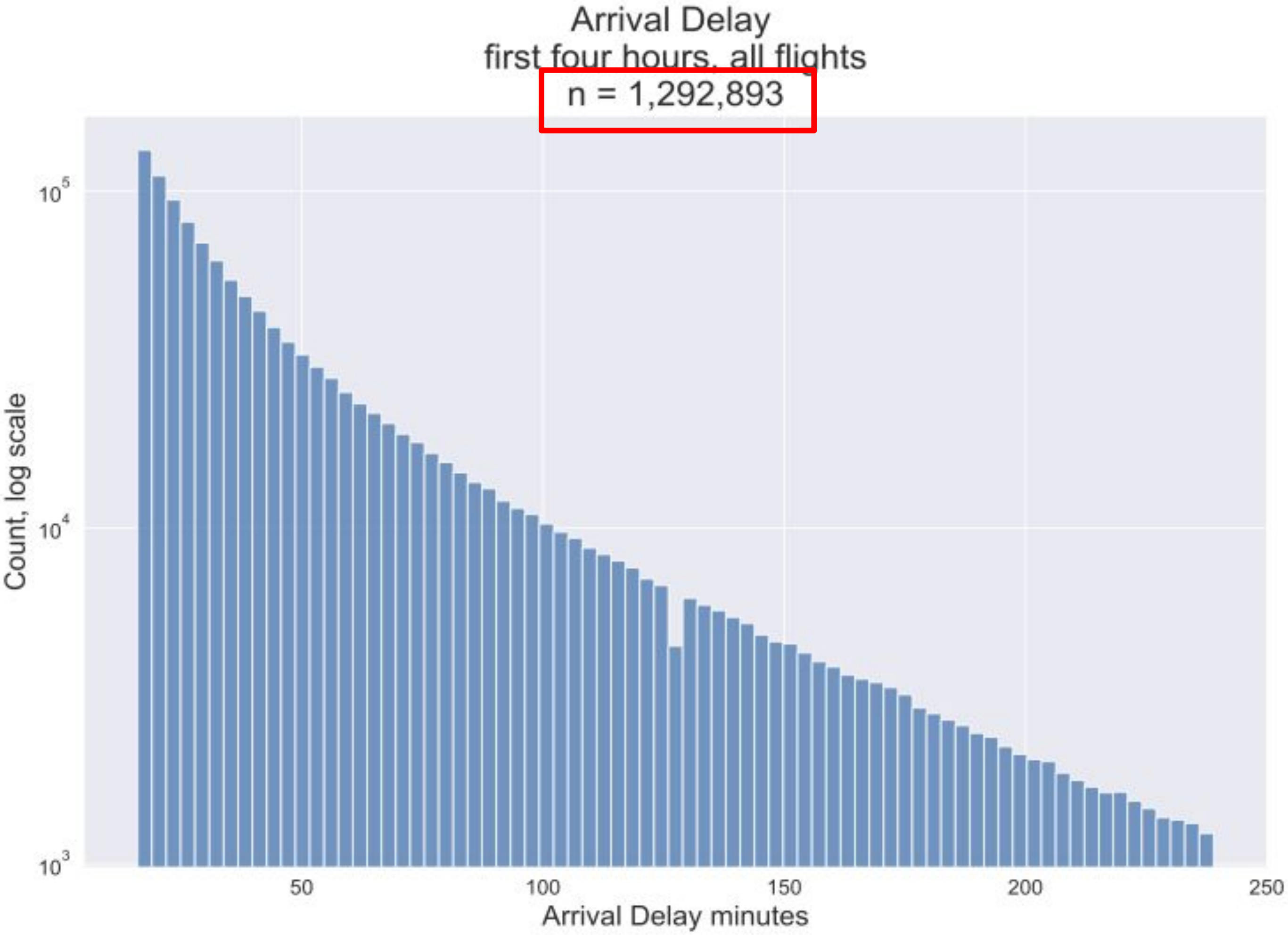
Exploratory Analysis Highlights

Mostly very short delays



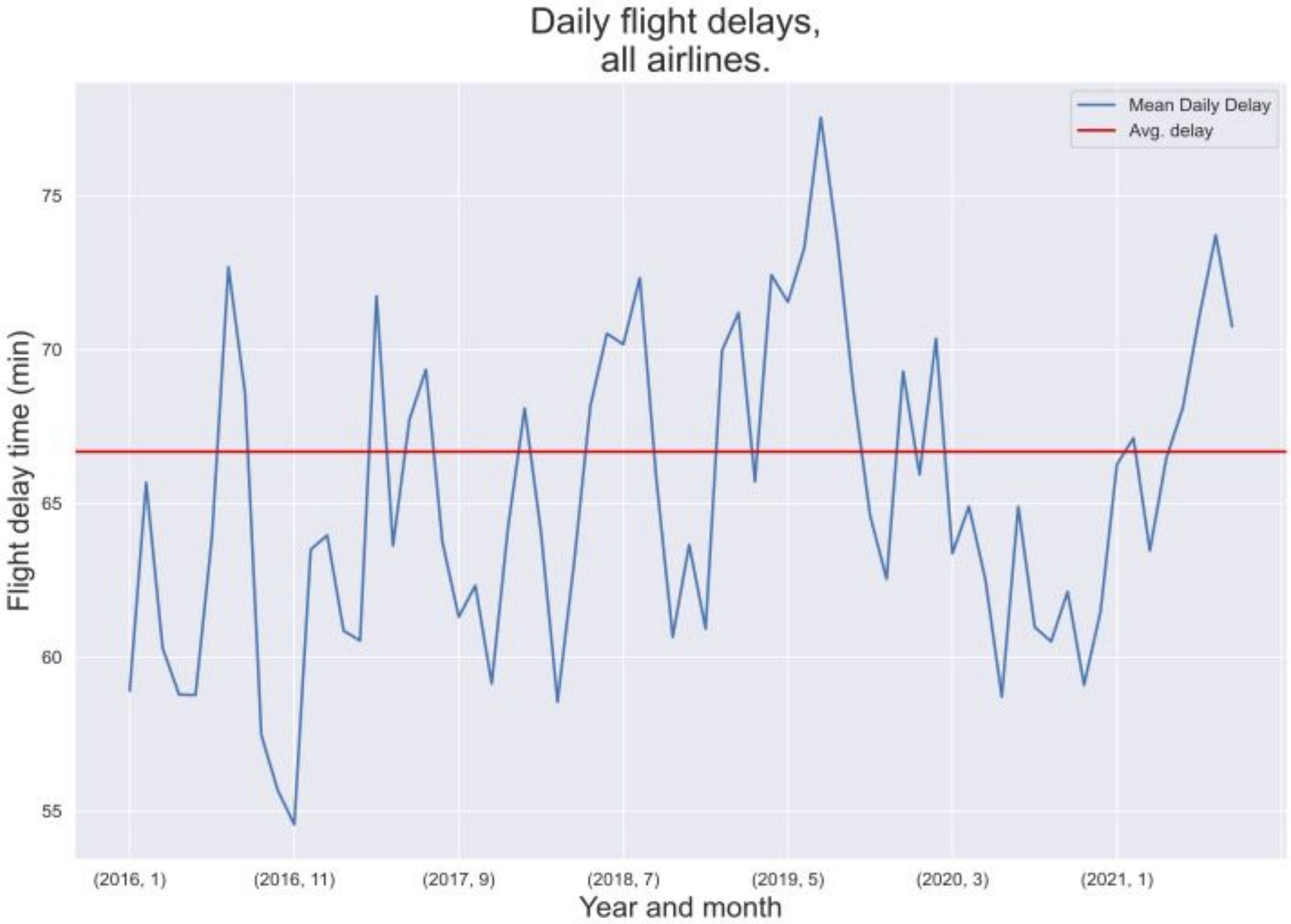
Exploratory Analysis Highlights

1.29M / 1.33M delayed up to 4 hours



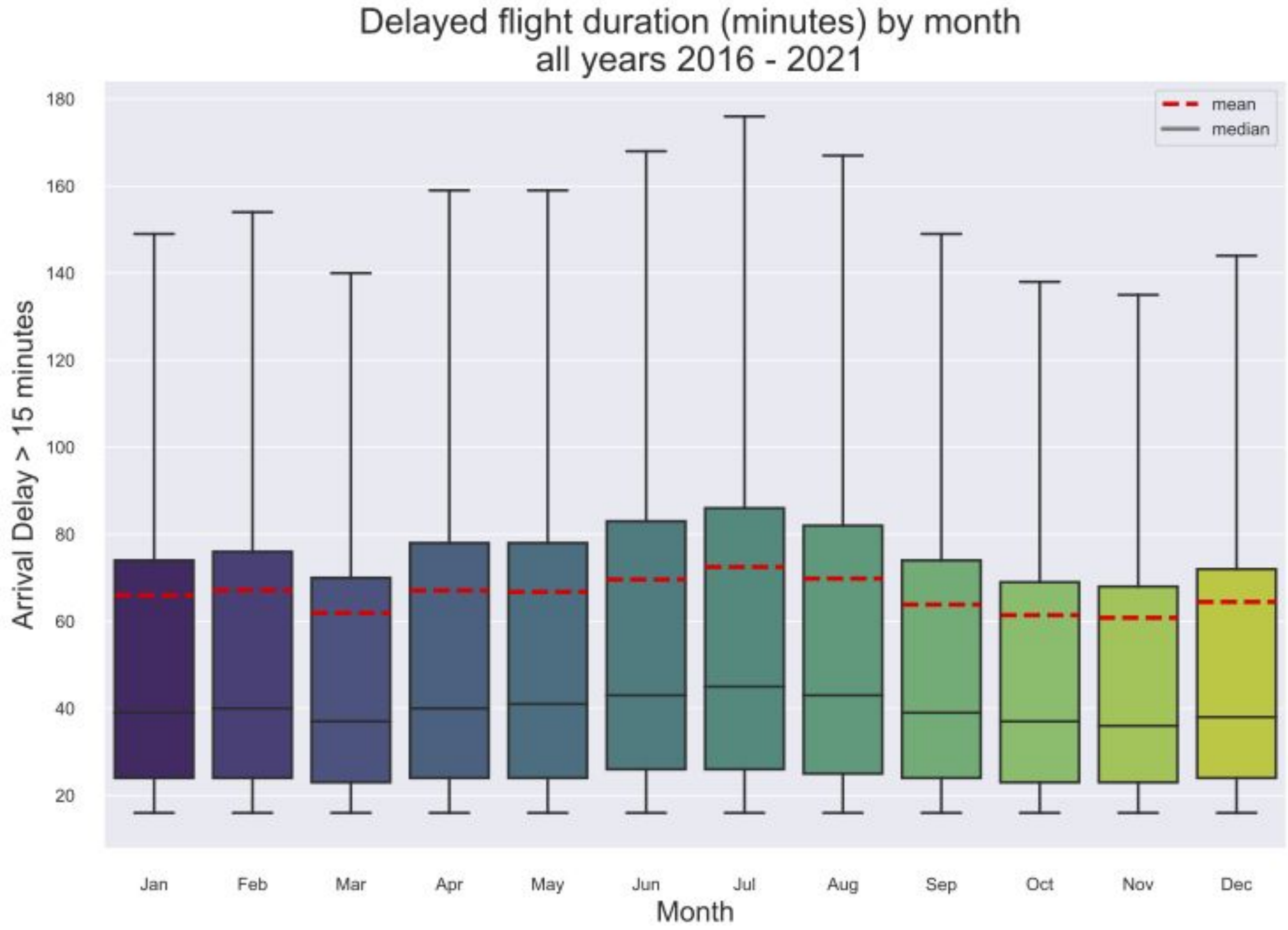
Exploratory Analysis Highlights

Avg delay
~66.5 minutes



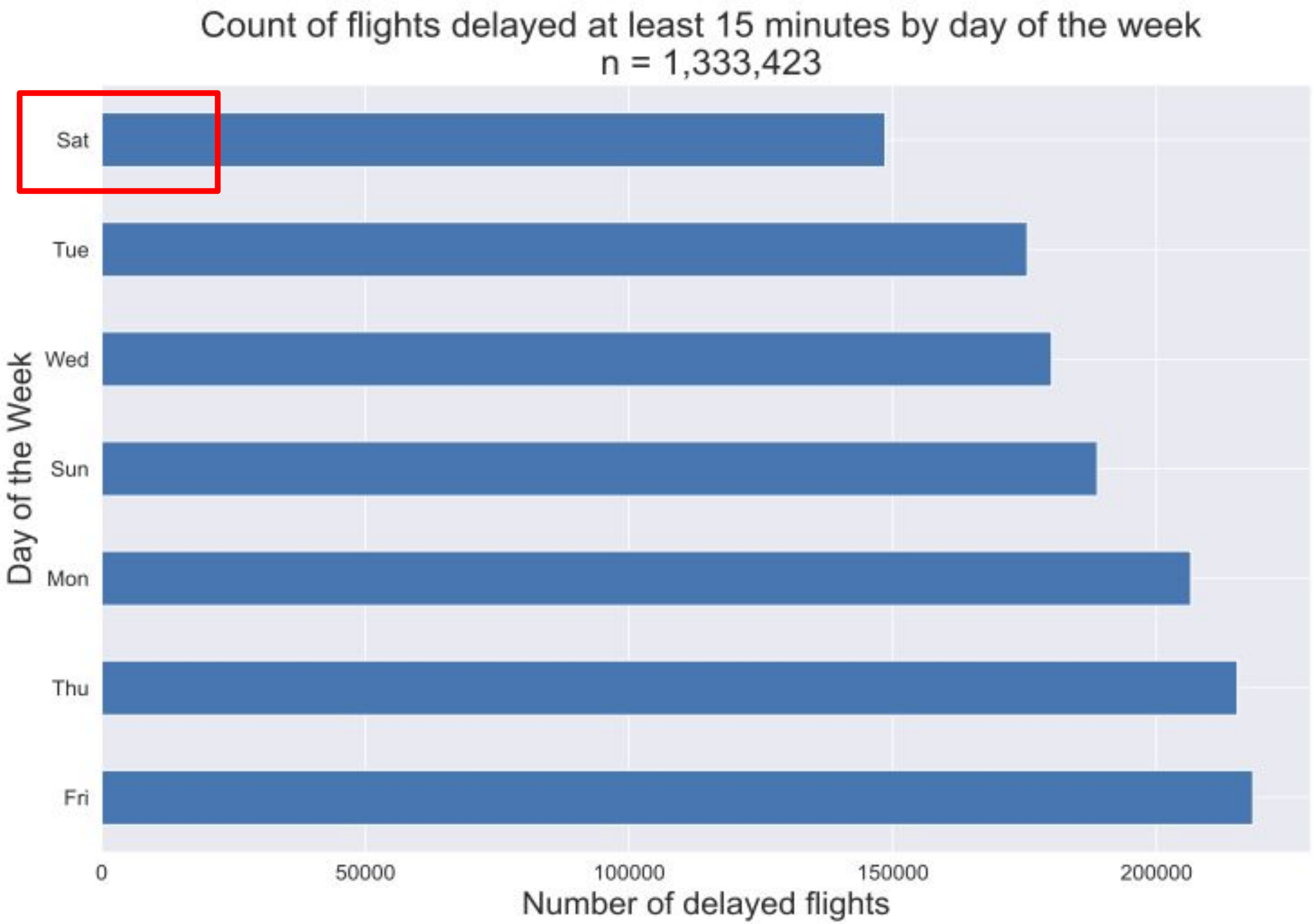
Exploratory Analysis Highlights

Summer travel period is the busiest



Exploratory Analysis Highlights

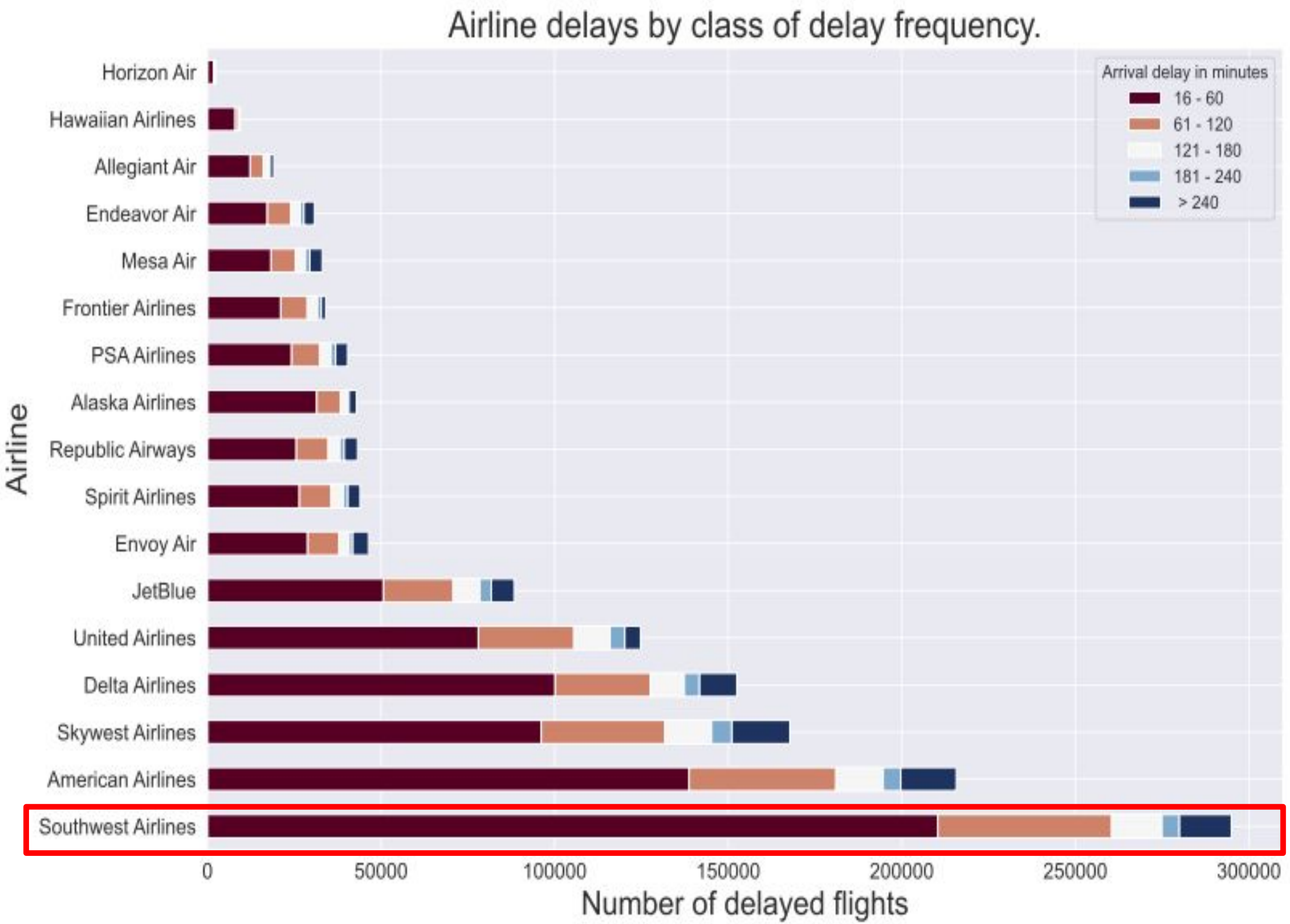
Saturdays had the fewest delays



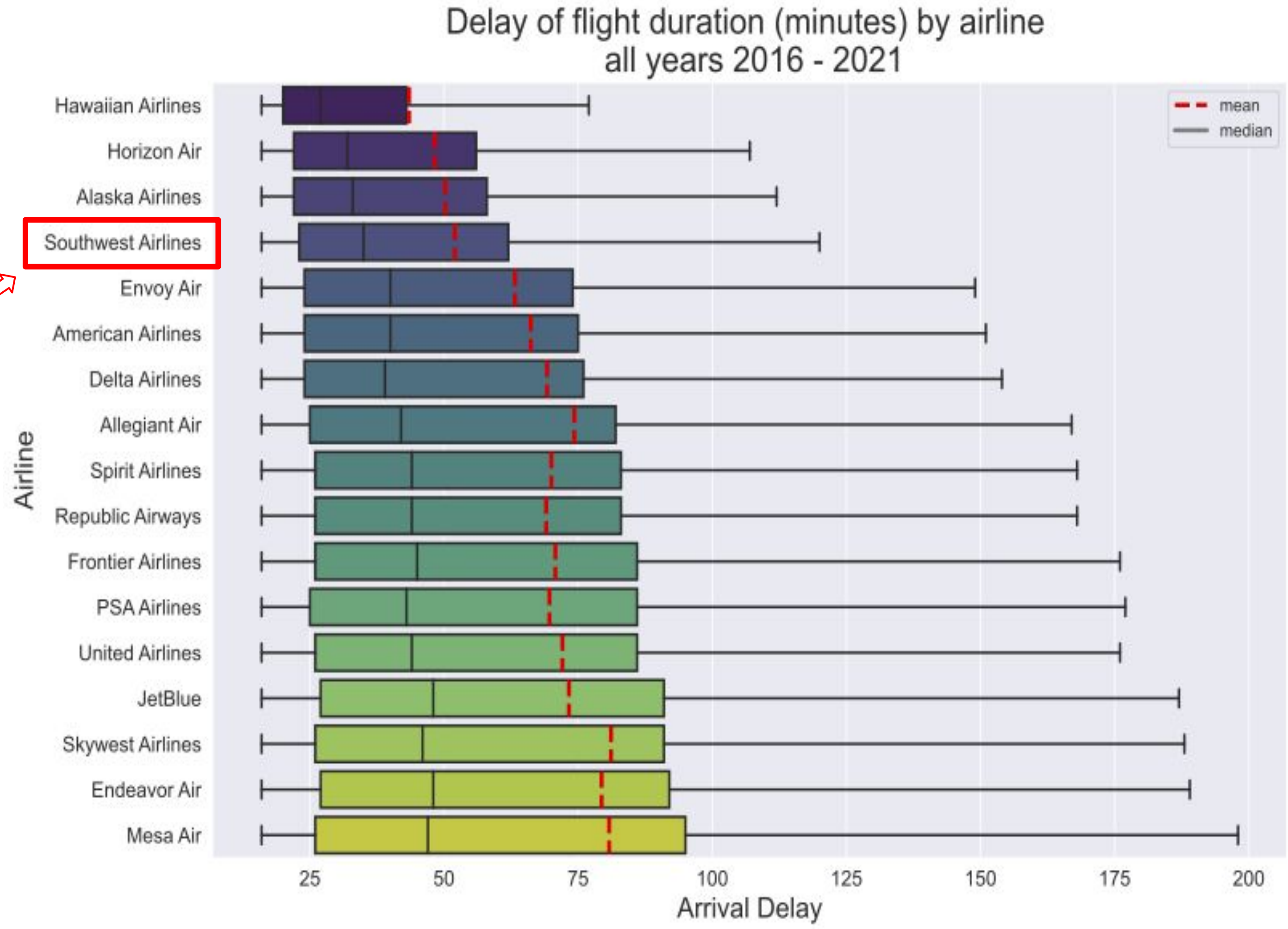
Exploratory Analysis Highlights

Airline delay ranges

Surprised at the quantity



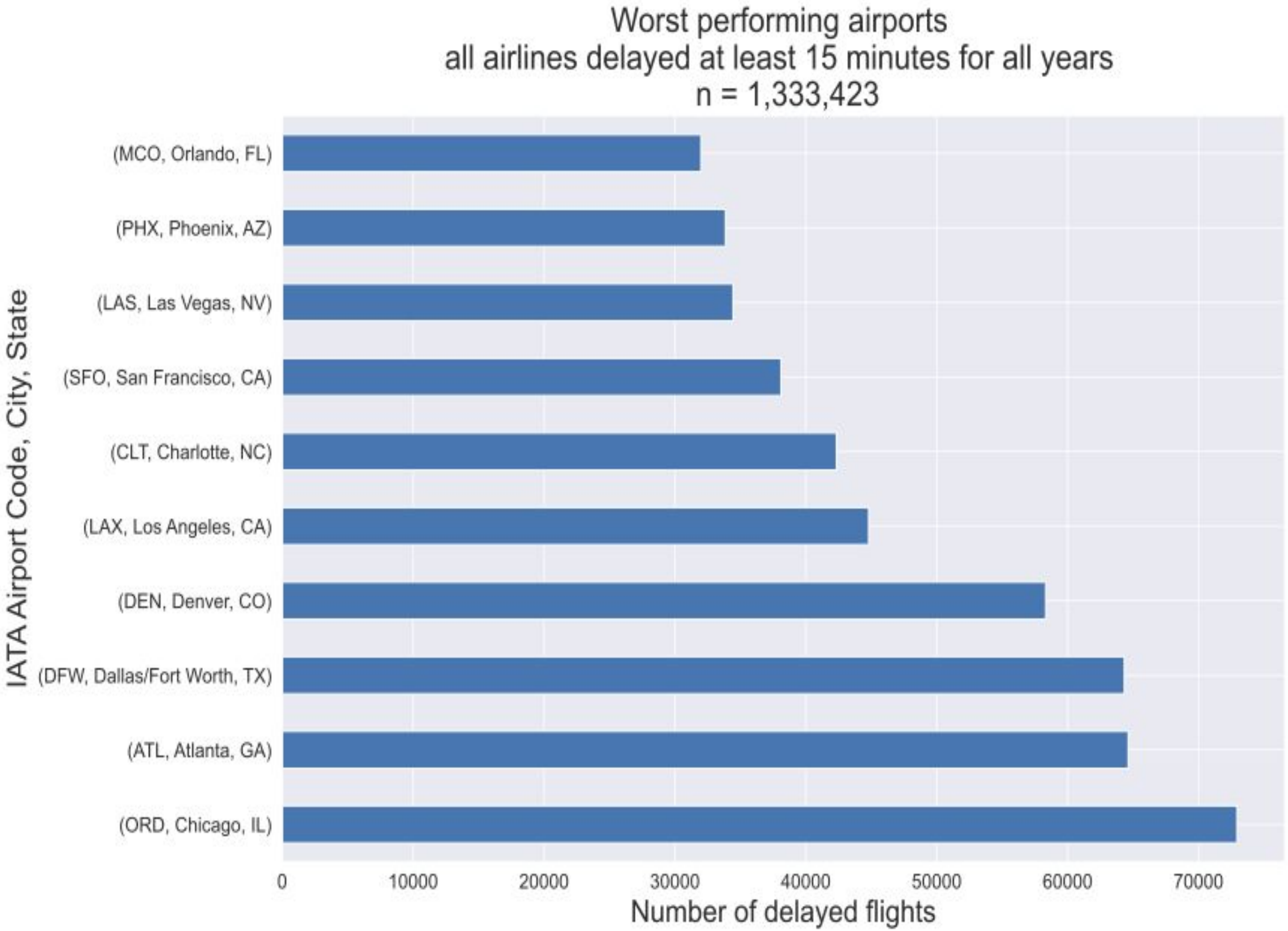
Exploratory Analysis Highlights



Though high absolute frequency, relative short duration.

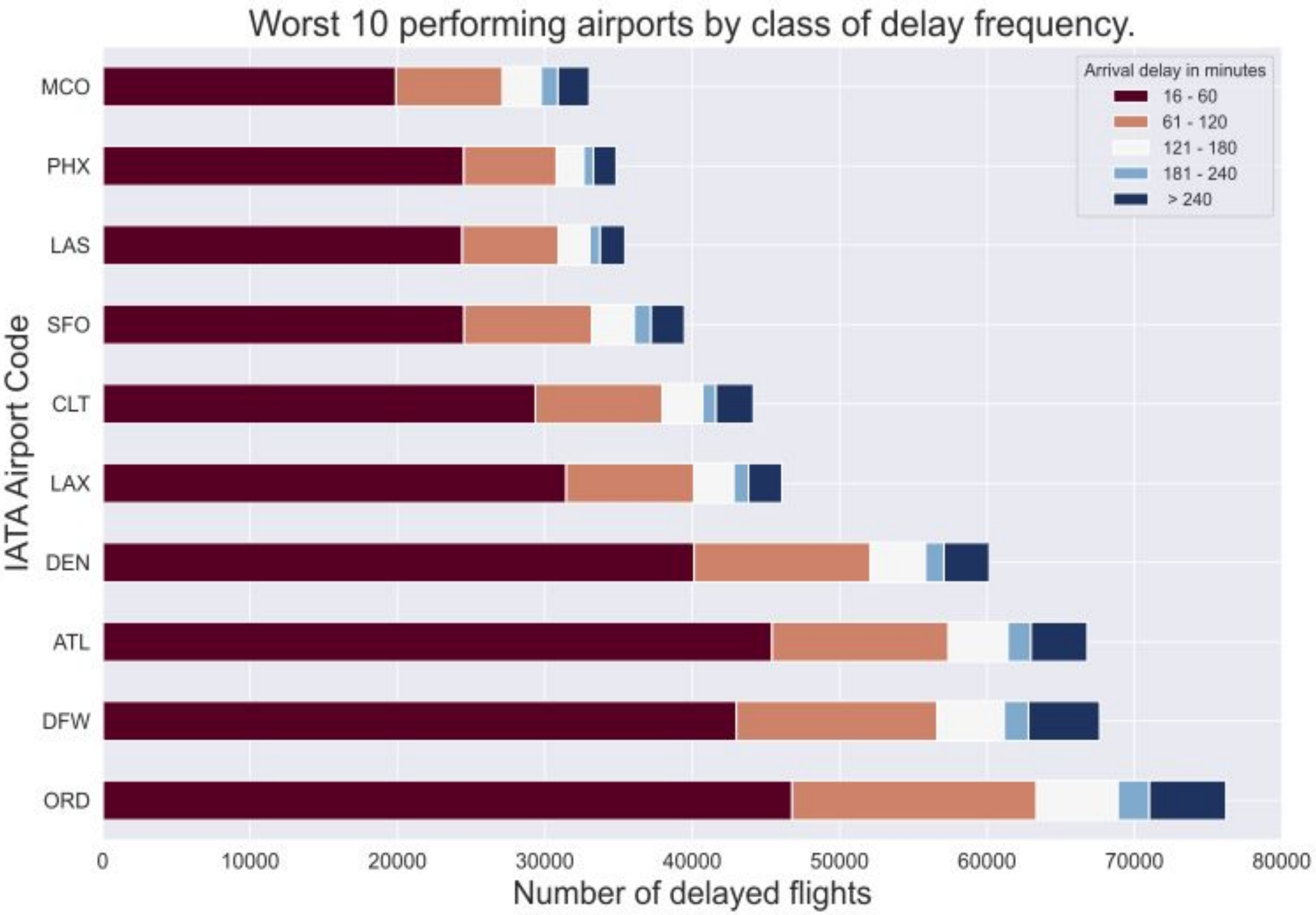
Exploratory Analysis Highlights

Where?



Exploratory Analysis Highlights

Delay durations.



These data are rich and dense.

Our intuition is good

- Generally our intuition informs us pretty well about where we are likely to have a delay.
    - Large, busy airports on large airlines tended to show delays.
- There were busy travel days and months throughout our data...but we are agnostic to time dimensions in this analysis.

Modeling

Modeling - approach

Features

Time columns and delay metrics ignored and removed from the feature set.

Continuous Variable

A sole continuous variable, Distance (miles).

Categorical Variables

Origin, *Destination*, *Day of Month*, *Day of Week*, *Month*, *Airline* resulting in 820 dummy columns.

Modeling - Task definition

Task

Scoped to binary classification: delay or no delay predictions.

Target Variable

1 delayed flight, 0 not-delayed

Modeling- Candidates

Model candidates

Classification species of Boosted Tree algorithms and a logistic regression.

Justification

Tabular, labeled, structured data.

Models

AdaBoost, XGBoost, Light GBM, and Logistic Regression.

Modeling- Selection: Results

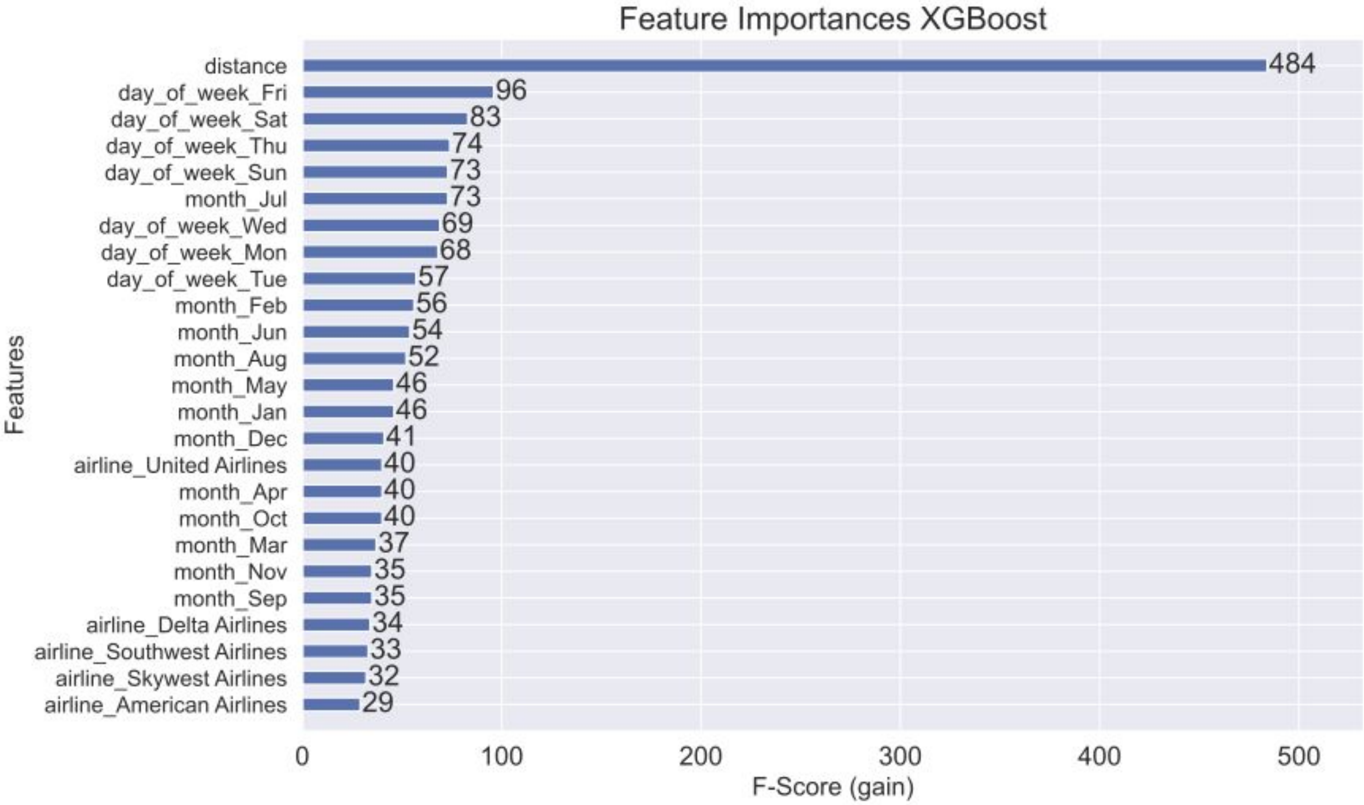
XGB speed-up:

tree_method = 'hist'

	LogReg_train	AdaBoost_train	XGB_train	LGBM_train
fit_time	64.059811	149.218789	39.787447	11.307615
score_time	1.031001	15.405630	2.508643	2.082145
test_accuracy	0.573486	0.572261	0.585975	0.588946
test_precision	0.575740	0.573699	0.588334	0.590858
test_recall	0.607307	0.612711	0.614092	0.618605
test_f1	0.591070	0.592531	0.600932	0.604405
test_roc_auc	0.601619	0.600139	0.619913	0.624587

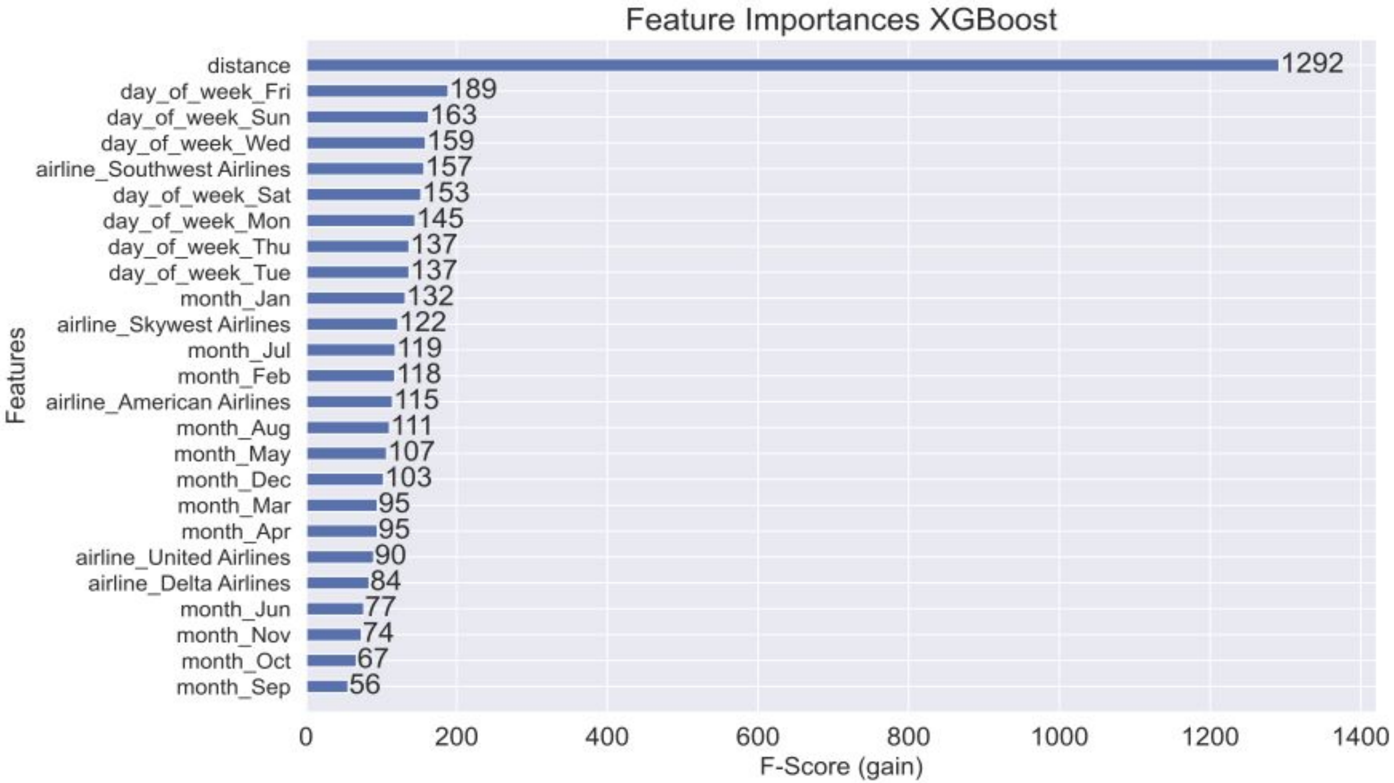
Modeling- Baseline Results

Distance mattered most to our model. Followed by temporal descriptions



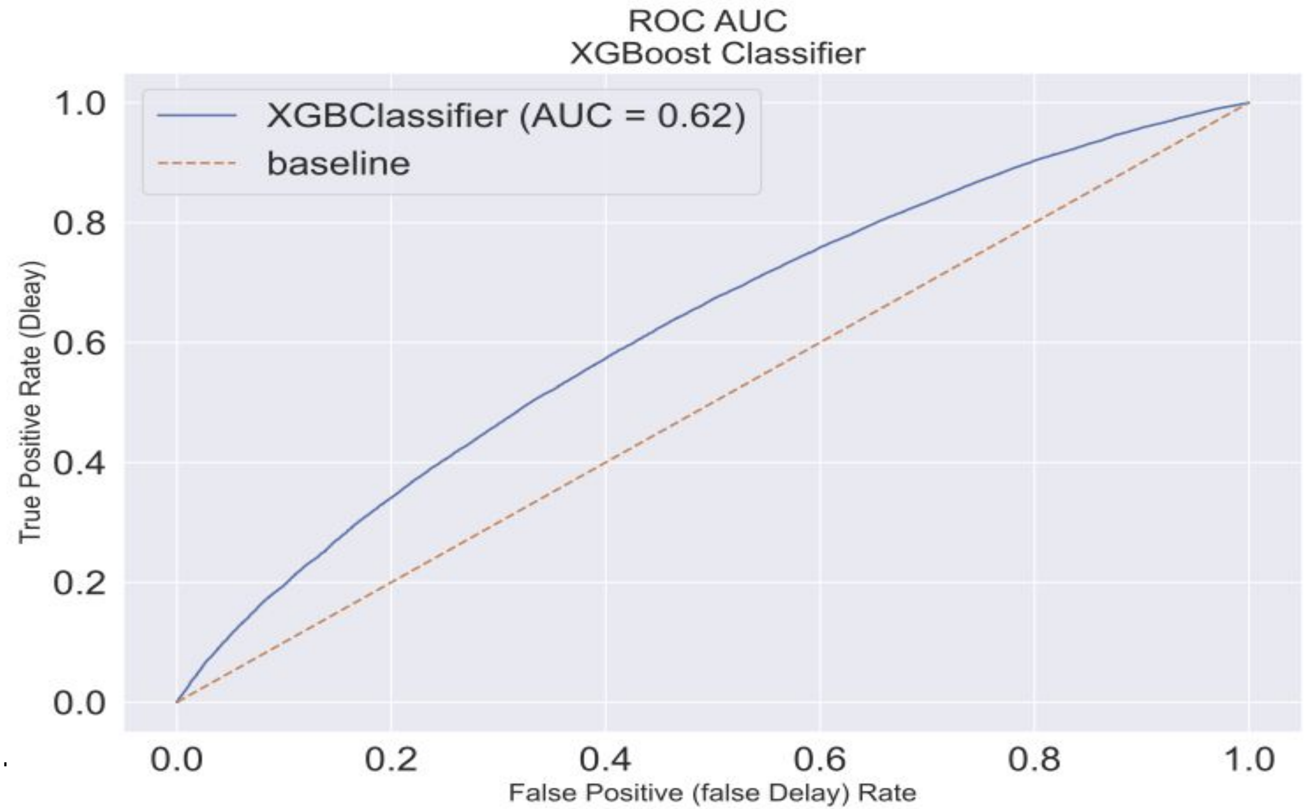
Modeling- Tuning: Test Results

Distance mattered most to our model. Followed by temporal descriptions

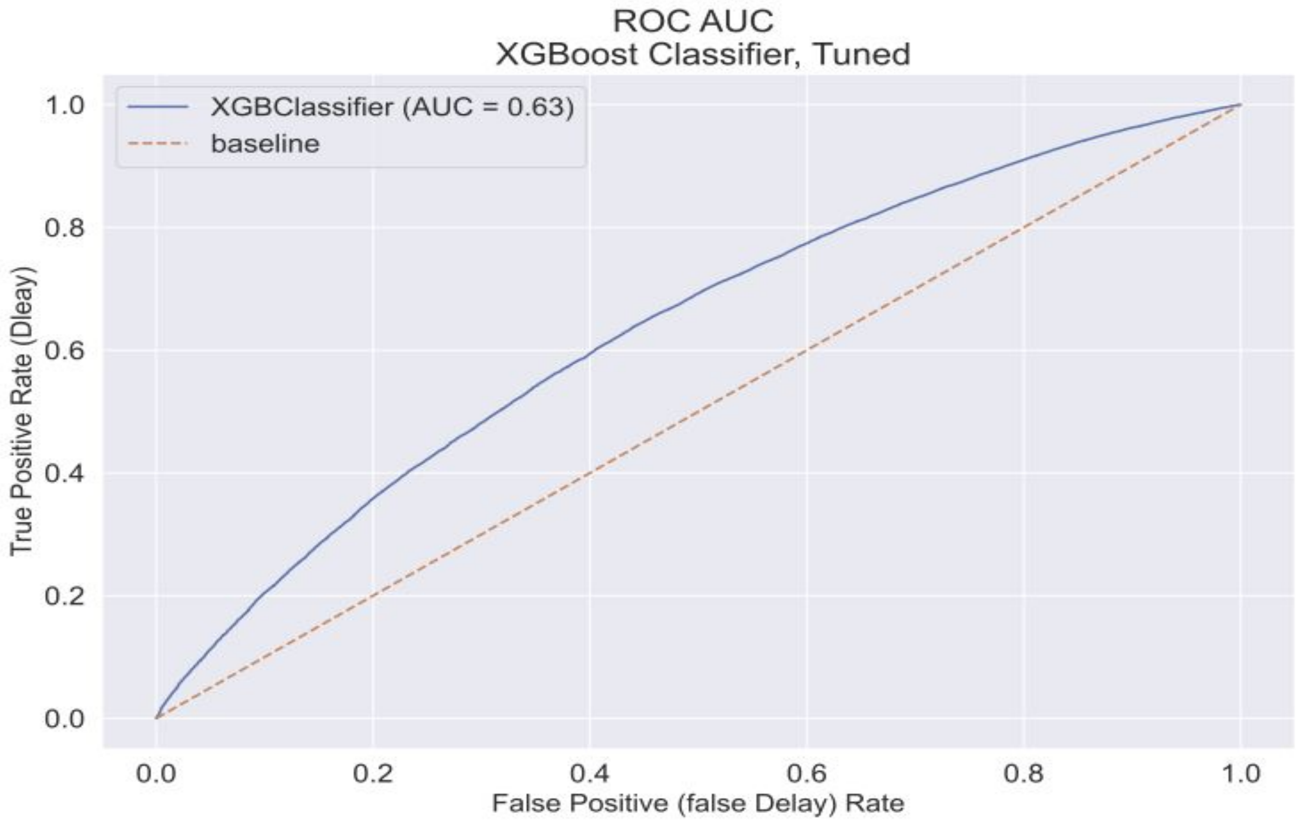


Modeling- Baseline Results

Distance mattered most to our model. Followed by temporal descriptions

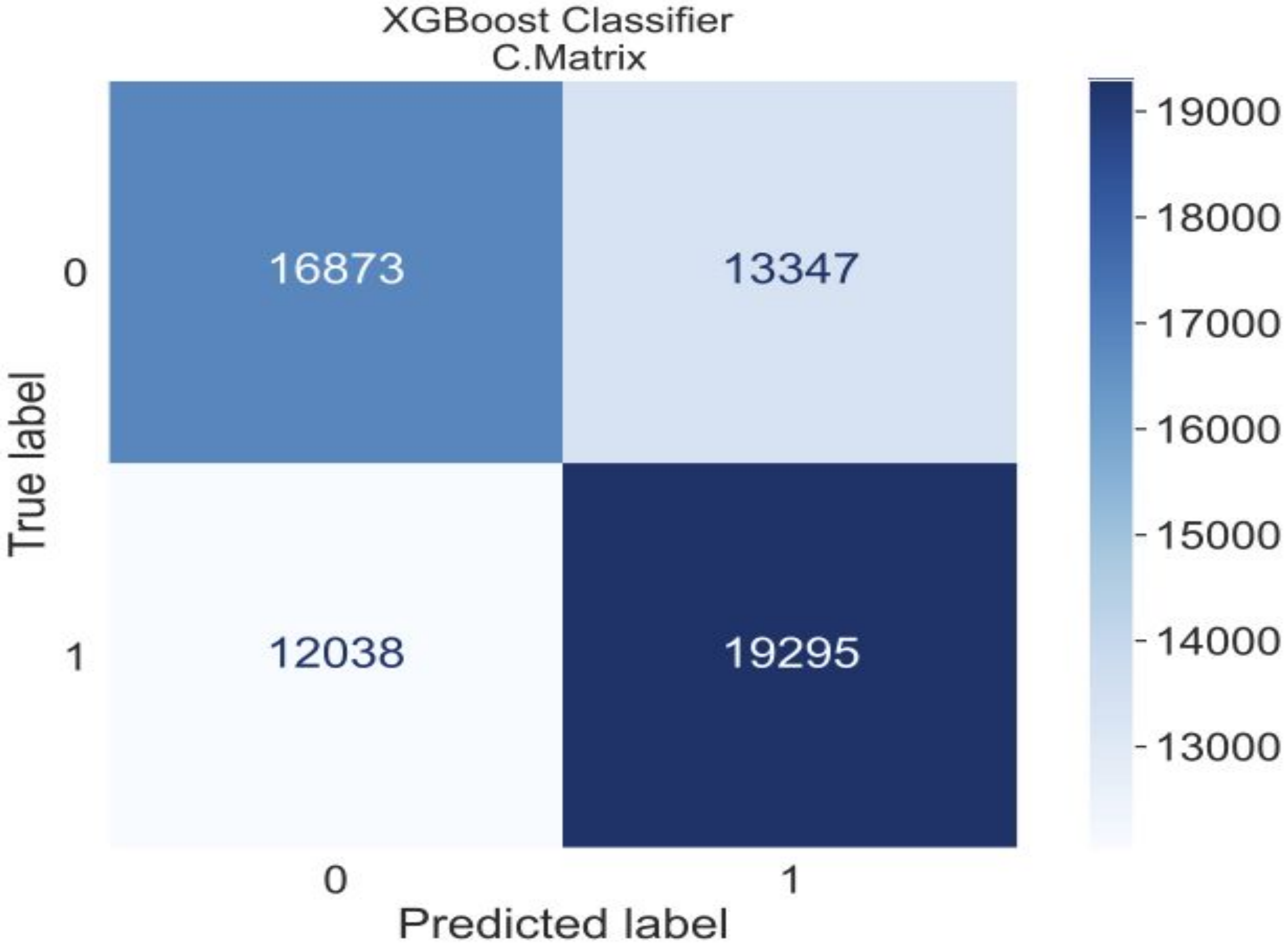


Modeling- Tuning: Test Results



Distance mattered most to our model. Followed by temporal descriptions

Model Baseline Results

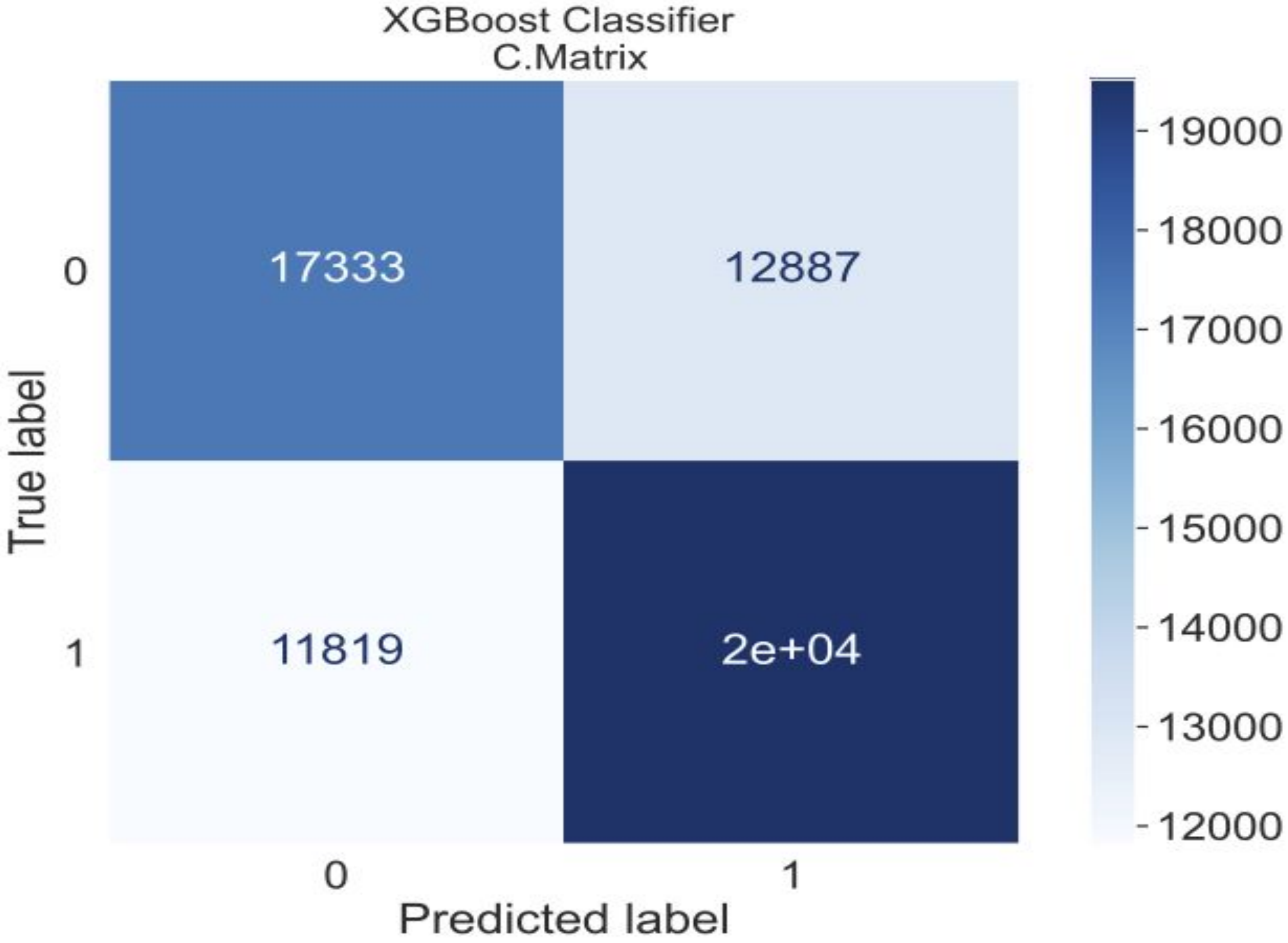


Precision: *0.58*

Recall: *0.61*

F1: *0.60*

Model Tuning Results



Precision: *0.62*

Recall: *0.66*

F1: *0.64*

Goals for next version

1. Continue to tune model until desired metrics are met. EG accuracy >=85%
2. Engineer more features and address overfit with more regularization.
3. Build the app.

THANK YOU