Predicting Flight Delays for Traffic Management & Customer Sentiment Optimization

Team_1-1: Henry Dinh; Daniel Rhinehart; Yong Lim; Deva Empranthiri; Varun Val

DATA261 - Final Project (Phase2)

Date: Nov 18th, 2024

Agenda

- **Overview**
  - Approach
  - Team Members
  - Phase Leads & Work Breakdown
- **Exploratory Data Analysis**
  - Data Sets Utilized
  - EDA - Key Takeaways
  - Issues Identified
- **Feature Engineering**
  - Custom Data Join
  - Feature Engineering
  - Features Selected for Use
  - Feature Transformations
- **Modeling**
  - ML Approach
  - ML Pipeline
  - Metrics Utilized
  - Hyper-Tuning Param Results
  - Cross-Validation Training
- **Results and Recommendations**
  - Results
  - Next Steps - future models
  - Conclusion

Overview

Daniel Rhinehart

Approach

We are a startup commissioned by the FAA to develop and implement a machine learning solution which predicts the following:

- When a flight's departure time will be delayed (relative to its planned departure time)
- When this delay will be greater than or equal to 10 minutes

The FAA intends to use these predictions to:

- Adjust their inbound/outbound airport traffic loads accordingly
- Inform airlines so they can better enact mitigation measures to maintain positive customer sentiment (e.g. alternate plane use, pre-emptive adjustment of connecting flights)

We will use logistic regression on various datasets for this Phase of the project

Team

Henry Dinh

Dan
Rhinehart

Yong Lim

Deva
Empranthiri

Varun Val

Phase Leads and Work Breakdown

Phase Leadership

Phase	Description	Leader
Phase 1	Project Plan, describe datasets, joins, tasks, and metrics	Daniel, Henry
Phase 2	EDA, baseline pipeline, Scalability, Efficiency, Distributed/parallel Training, and Scoring Pipeline	Yong
Phase 3	Select the optimal algorithm, fine-tune and submit a final report	Varun, Deva

Work Breakdown

Phase	Task	LOE	Team Member
Phase 1	Set up Blob Storage for Team	M	dan
Phase 1	Define Project Approach, Timeline and Report	L	dan, henry, yong, varun
Phase 1	Initial EDA	L	henry, dan
Phase 1	Author Final Report (in Notebook)	M	dan
Phase 2	Data Pipeline: Ingestion, Feature Engineering, Feature Selection and Train/Test Split	L	yong
Phase 2	Baseline Model Implementation: Scalability, Efficiency, Distributed Training	M	yong
Phase 2	Scoring Pipeline Implementation: Metric Assessment, Performance Visuals	M	yong
Phase 2	Final DB Notebook (code)	M	yong
Phase 2	Mid-Point Presentation Creation	S	dan

Phase	Task	LOE	Team Member
Phase 3	Creation and EDA of a good graph based feature (eg: topic based page-rank by airline)	M	deva
Phase 3	Review and Selection of FInal Algorithm	S	varun
Phase 3	Refinement and Fine-Tuning of Neural Network experiments	M	varun
Phase 3	Refinement and Fine-Tuning of Decision Tree experiments	M	deva
Phase 3	Final DB Notebook Report Write-Up	M	varun, deva
Phase 3	Final Presentation Creation	M	varun

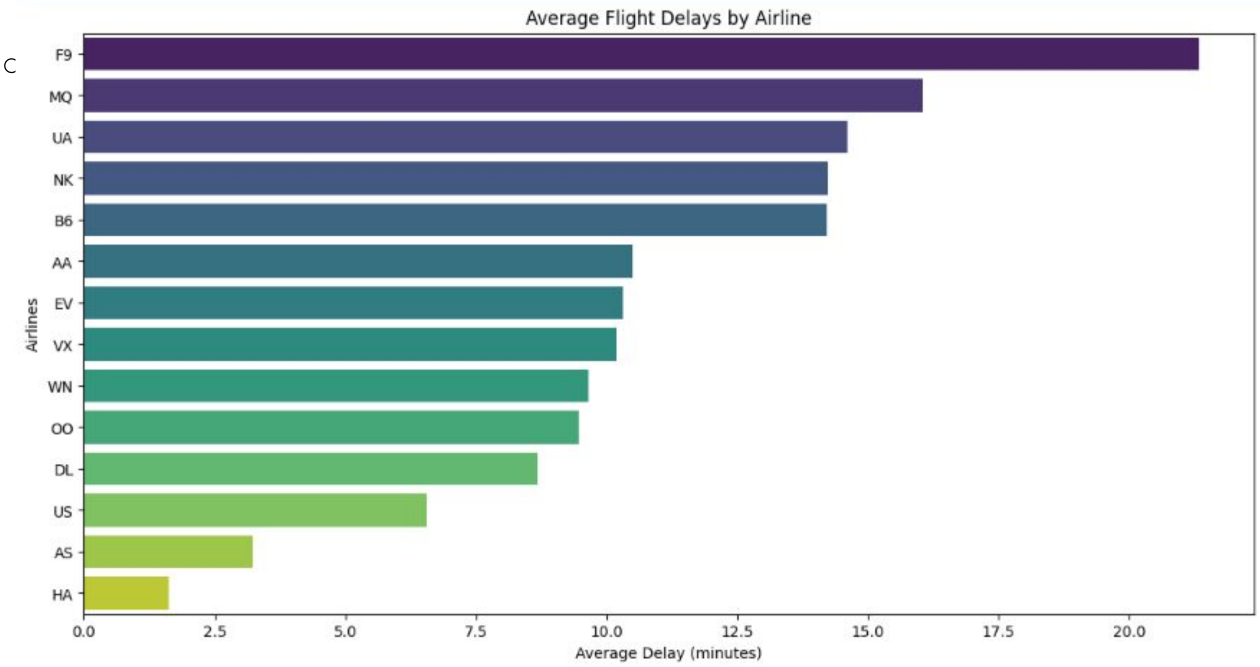
EDA

Henry Dinh

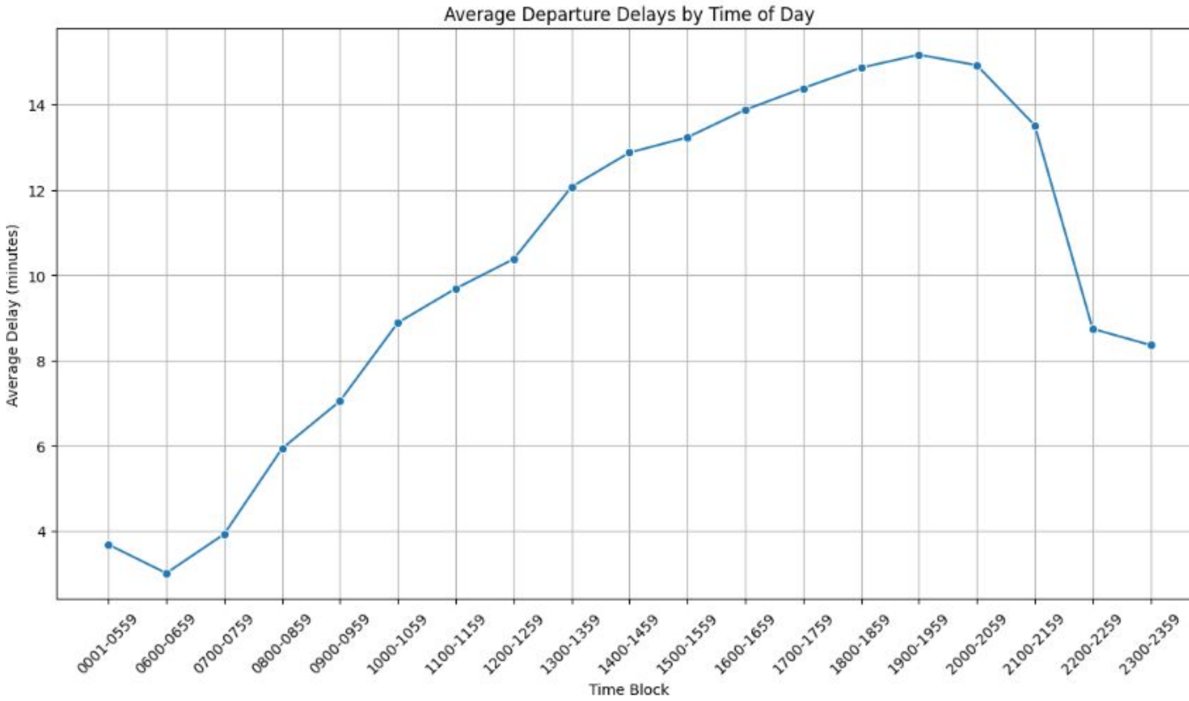
Data Sets Utilized

- US DOT Flights Data
  - **Description**: Passenger flights on-time performance data taken from the TranStats data collection available from the U.S. Department of Transportation
- NOA Weather Data
  - **Description**: Weather data corresponding to the origin and destination airports at the time of departure and arrival from the National Oceanic and Atmospheric Administration
- US DOT Airport Data
  - **Description**: Airport metadata from the US Department of Transportation.
- Airport Codes Data
  - **Description**: Airport codes consisting of either IATA three-letter codes (passenger reservation, ticketing, baggage-handling systems) or ICAO four-letter codes used by ATC systems and airports which do not have an IATA code

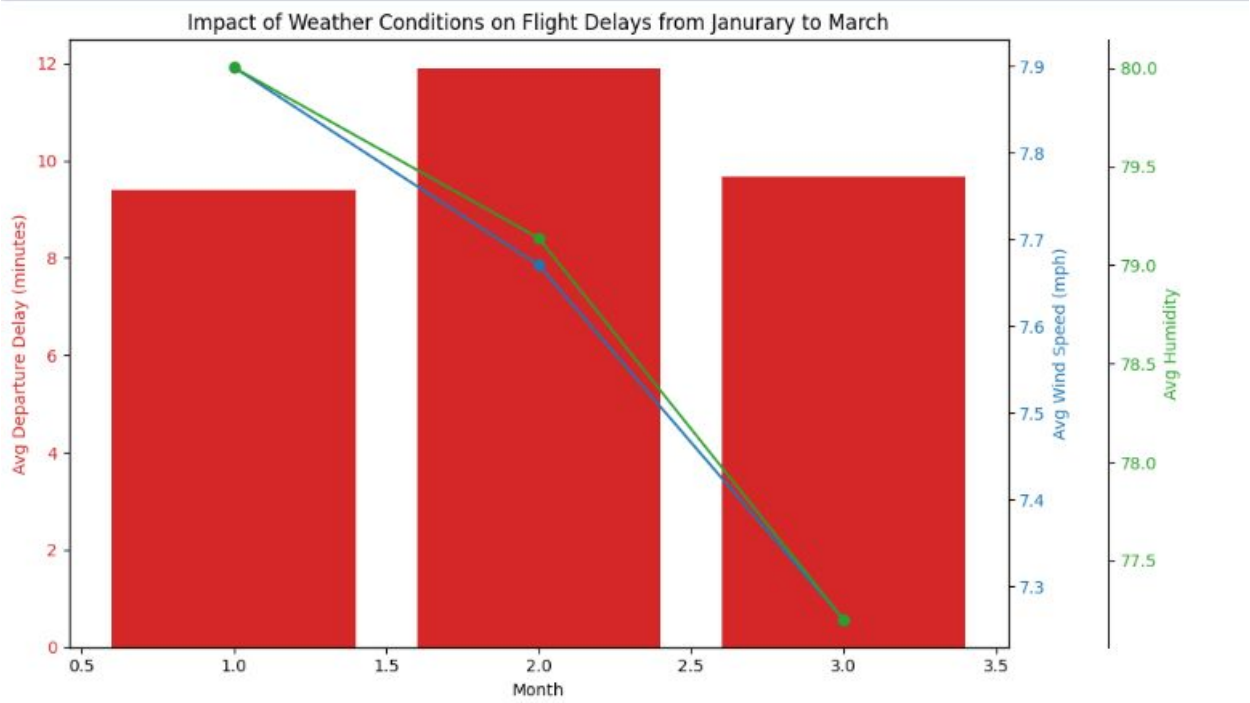
Average Flight Delays by Airline



EDA - Average Departure Delays by Time of Day



EDA - Impact of Weather Conditions on Flight Delays



EDA - Issues Identified

Data Quality

- Missing Values
    - The datasets exhibited a significant proportion of missing values in weather-related features and some delay categories.
    - Weather data:
        - There are 124 columns in total but only the following columns have less than 20% missing values
        - [YEAR, REPORT_TYPE, SOURCE, DATE, STATION, NAME, ELEVATION, LONGITUDE, LATITUDE, HourlyDryBulbTemperature, REM, HourlyWindSpeed, HourlyWindDirection, HourlyDewPointTemperature, HourlyRelativeHumidity]
    - Airlines data:
        - There are 102 columns in total but we decide to choose the following columns
        - ['CARRIER_DELAY', 'WEATHER_DELAY', 'NAS_DELAY', 'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY', 'ARR_DELAY', 'ARR_DEL15', 'ARR_DELAY_NEW', 'DEP_DELAY', 'DEP_DEL15', 'DEP_DELAY_NEW', 'DISTANCE', 'TAXI_OUT', 'TAXI_IN', 'AIR_TIME', 'CRS_ELAPSED_TIME', 'ACTUAL_ELAPSED_TIME']

Feature Engineering

Yong Lim

Custom Data Join

We opted to do a custom data join as follows:

- Data Sets Joined: Airline (time series), Airport, weather station, and weather (time series)
- Joining Logic
  - Join Airline Data and Airport Data (Origin airport & Prior airport)
  - Join Airline-Airport Data with Station Data (Origin airport & Prior airport)
  - Join Joint Data with Weather Data (Origin airport & Prior airport)

Dataset	3 month		1 Year	
	data size	load/join time (sec)	data size	load/join time (sec)
Original airline dat	2806942	12	14844074	23
Cleaned airline dat	1356814	1	7268232	1
Join airport data	1356814	0.23	7268232	0.26
Join station data	1350012	0.19	7238779	0.24
Join weather data	1349914	2.48	7238107	2.23

Doing this custom join yielding the following benefits:

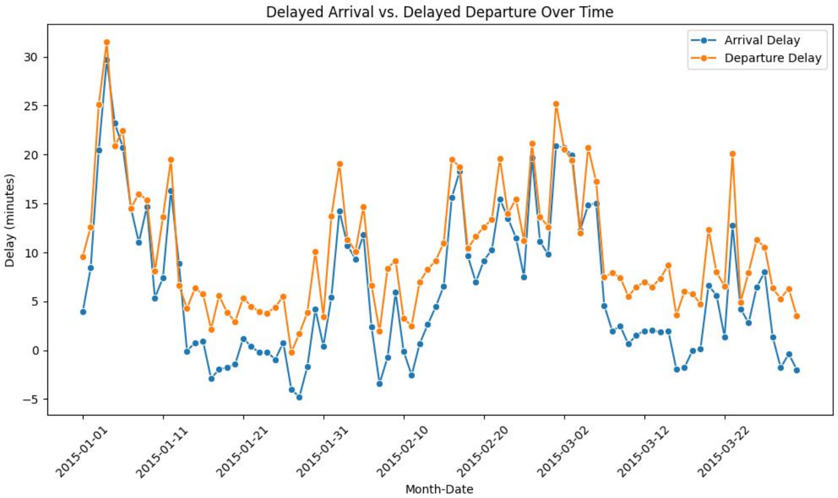
- Combining dataset provides a more comprehensive view, which enriches the analysis and insights derived from the dataset.
- Custom join can highlight discrepancies and errors, improving overall data quality and reliability

Feature Engineering

Feature Creations

- DEP_DEL10: 10 minutes departure delay for our target value
- PREV_ORIGIN: Flight prior airport code
- PREV_**: Flight's prior airport weather conditions
- DEP_**: Flight's origin airport weather conditions

	DEP_DELAY
count	2722232.00
mean	10.36
std	37.86
min	-61.00
25%	-5.00
50%	-1.00
75%	9.00
max	1988.00



Features Selected for Use

- **Numeric Features**
  - PREV_ARR_DELAY: Flight's arrival delay in origin airport in minutes.
  - CARRIER_DELAY: Delay caused by the carrier.
  - WEATHER_DELAY: Delay caused by weather conditions.
  - SECURITY_DELAY: Delay due to security reasons.
  - NAS_DELAY: Delay due to National Airspace System issues.
  - DEP_W_HourlyWindSpeed: Departure airport hourly wind speed.
  - DEP_W_HourlyDewPointTemperature: Departure airport hourly dew point temperature.
  - DEP_W_HourlyRelativeHumidity: Departure airport hourly relative humidity.
  - PREV_W_HourlyWindSpeed: Flight’s prior airport hourly wind speed.
  - PREV_W_HourlyDewPointTemperature: Flight’s prior airport hourly dew point temperature.
  - PREV_W_HourlyRelativeHumidity: Flight’s prior airport hourly relative humidity.
- **Categorical Features**
  - QUARTER, MONTH, DAY_OF_MONTH, DAY_OF_WEEK: Time related features
  - OP_CARRIER: The operating carrier.
  - ORIGIN: The origin airport.
  - PREV_ORIGIN: The flight’s prior airport.

Feature Transformations

- **Missing Values:**
  - Fill null values in median values since the distribution was right skewed.
- **Encoding Categorical Variables:**
  - Utilize StringIndexer to assign indices to the categorical values and transform these values into binary vectors using OneHotEncoder
- **Scaling and Normalization:**
  - Standardize features to have a mean of 0 and a standard deviation of 1 using StandardScaler.

Modeling

Varun Val

ML Approach

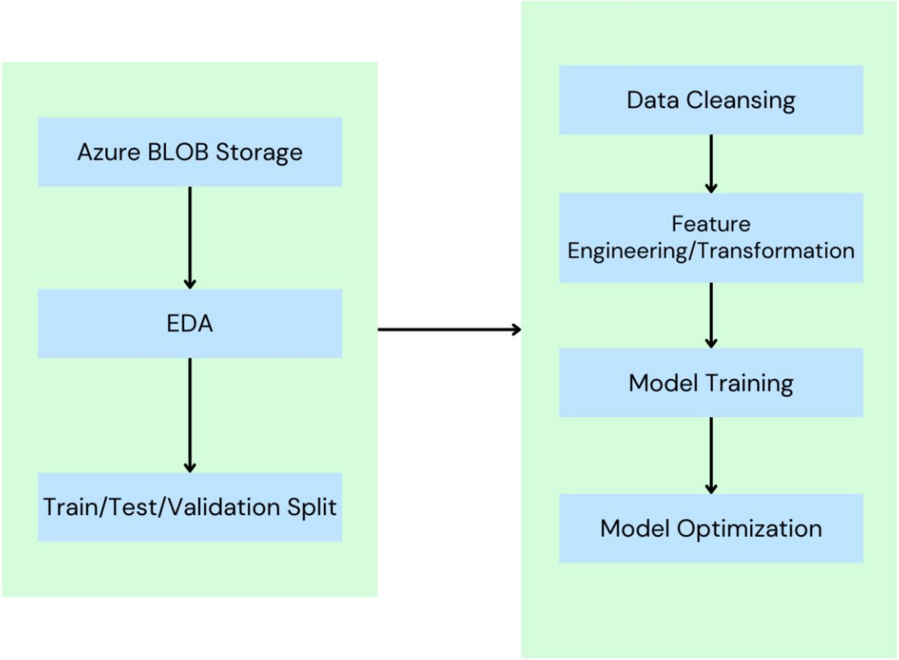
- For this phase we worked on constructing our baseline model which is a simple *Logistic Regression* Model
  - Since a logistic regression model cannot handle missing or categorical values:
    - Numerical features: Imputed as previously mentioned
    - Categorical features: One-hot encoded as part of ML pipeline

- Downsampling was conducted on the training dataset to help deal with the class imbalance:
  - Before downsampling: 5400497 rows
  - After downsampling: 1382784 rows

- Cross-validation was carried out using block cross-validation to prevent leakage

ML Pipeline

- Our Machine Learning Pipeline is noted below:



Metrics Utilized

- We used the following metrics to measure success:
  - Accuracy: Helps in understanding the overall effectiveness of the model.
    - Accuracy = (TP + TN) / (TP + TN + FP + FN)
  - Precision: Useful for understanding the accuracy of the delay predictions.
    - Precision = TP / (TP + FP)
  - Recall (Sensitivity): Important for capturing the actual delays correctly. High recall means fewer missed delays.
    - Recall = TP / (TP + FN)
  - **F1 Score:** Provides a balance between precision and recall, especially valuable when you need a single metric to gauge model's performance.
    - F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Hyperparameter Tuning

- Hyperparameter tuning was carried out using Hyperopt:
  - elasticNetParam: uniform distribution 0.0 to 1.0
  - regParam: uniform distribution 0.0 to 1.0
  - maxIter: uniform integer distribution from 10 to 20

- To increase performance, cross-validation was not conducted within hyperopt
  - Existing training data was 80/20 split for a separate training/validation set

- F1 score was being maximized during the tuning

- Results after 30 trials:

elasticNetParam	0.3866
regParam	0.0184
maxIter	13
Avg time/trial	274.85 s

Cross-Validation Training

- 5 fold cross-validation was carried out on the dataset with the following results for each fold:
  - *Note: The below metrics are the weighted versions*

	F1 score	Accuracy	Precision	Recall
Block 1	0.710698	0.704104	0.729506	0.704104
Block 2	0.716105	0.731306	0.743264	0.717575
Block 3	0.727474	0.727295	0.734748	0.725914
Block 4	0.718161	0.715858	0.718084	0.716884
Block 5	0.720371	0.730429	0.729057	0.740336

- Training time over 2 worker nodes: 10.92 min

Results and Recommendations

Deva Empranthiri

Results

Our best model achieved **73.2% accuracy** on our test dataset using the logistic regression algorithm. This means that of the 1,837,610 test examples, it correctly predicted a late flight 132,494 out of 196,172 times.

Test set metrics:

	Value	Percentage
Accuracy	0.732	73.25%
Precision	0.236	23.64%
Recall	0.675	67.54%
F1 Score	0.350	35.02%

Results

Confusion matrix for test set

Next Steps

- **Run alternate Model Variants:**
    - Fully Connected Neural Network
        - Can learn more complicated patterns
        - Can be used to find non-linear relationships between weather data and delays.
    - Random Forest
        - Useful to find general patterns for airline delays
        - Minimizes overfitting by averaging predictions across trees
    - XGBoost Decision Trees
        - Able to deliver accurate predictions while using regularization to prevent overfitting
        - Can handle imbalanced datasets well

- **Run the best-performing model on the 5-year dataset to generate the final model.**

Conclusion

- Our current model based on logistic regression is performing as follows:
    - **Accuracy**: 0.732 (73.25%)
    - **Precision**: 0.236 (23.64%)
    - **Recall**: 0.675 (67.54%)
    - **F1 Score**: 0.350 (35.02%)
    - This translates into predicting 67.5 out of 100 late flights correctly...

- Though our current model is effective in predicting most late flights, we will spend the next few weeks:
    - **Training Different Models:** We will try different model architectures
    - **Hyperparameter Tuning:** We will tune hyperparameters to get the best accuracy on our dataset
    - **Identify our best model:** Create a final model with our best hyperparameters, and train it on our dataset.

Appendix

Appendix

- [Colab Link of Analysis]