* 11/15 10am Meeting
* **11/20 Proj. Workshop II**
* 11/20 Start Presentation
* **12/2, 12/4 Presentations**

**1. Data Cleaning and Preparation**

Starting from the raw flight delay dataset, our goal was to remove noise, prevent leakage, and create a streamlined dataset for predicting arrival delays. Here’s a breakdown of each data cleaning step and the rationale:

#### **Filtering for Major Airports**

* **Step**: We restricted the dataset to flights that both originate and land at the 30 largest U.S. airports (e.g., ATL, LAX, ORD).
* **Rationale**: Large airports account for significant traffic volume and exhibit more consistent patterns due to their size and operational scope. Focusing on these high-traffic hubs reduces variability, allowing the model to focus on common, recurring factors that impact flight delays.

#### **Removing Canceled Flights**

* **Step**: We removed all rows where CANCELLED was marked as true.
* **Rationale**: The objective is to predict arrival delays, not cancellations, which involve separate factors. Including canceled flights would introduce noise because these flights don’t follow the same delay dynamics as completed flights.

#### **Removing Extreme Delays**

* **Step**: We excluded flights with ARR\_DELAY greater than 8 hours (480 minutes).
* **Rationale**: Extreme delays (often over 8 hours) tend to occur due to rare events, which can skew predictions if included. These outliers, while informative in some contexts, are hard to predict and don’t represent typical delay conditions. Removing them narrows the focus to regular delay patterns.

#### **Excluding Data from 2020**

* **Step**: All flights from 2020 were removed from the dataset.
* **Rationale**: The COVID-19 pandemic led to atypical flight patterns, including long delays and sudden cancellations, due to global lockdowns and operational constraints. Removing this year ensures the data reflects typical conditions, making the model more applicable to regular operations.

#### **Dropping Columns to Prevent Leakage and Redundancy**

* **Step**: We removed columns that could either cause leakage (directly revealing delay information) or were redundant for prediction.
  + **Leaking Columns**: DEP\_DELAY, ARR\_TIME, CRS\_DEP\_TIME, and CRS\_ARR\_TIME, as these could directly or indirectly reveal the target delay outcome.
  + **Operational Details**: Columns like TAXI\_IN, TAXI\_OUT, WHEELS\_ON, and WHEELS\_OFF provide granular operational data, adding complexity without aiding prediction.
  + **Irrelevant Columns**: Categorical data related to flight diversions and cancellations, such as DIVERTED and CANCELLATION\_CODE, were removed, as these are irrelevant to our goal.

### **2. Feature Engineering: Designing Predictive Features**

Following data cleaning, we developed features aimed at capturing patterns affecting delays while avoiding leakage. Here’s the breakdown of our engineered features:

#### **Date-Based Features**

* **MONTH**: Extracted as a numerical feature from FL\_DATE.
  + **Rationale**: Flight delays vary seasonally due to weather and holiday travel peaks. This feature allows the model to account for such seasonality, capturing the month’s influence on delays.
* **DAY\_OF\_WEEK**: Extracted from FL\_DATE (0=Monday, 6=Sunday).
  + **Rationale**: Weekly travel patterns differ, with Monday through Thursday generally reflecting business travel peaks and weekends showing more leisure travel. This feature helps capture those cycles.
* **DEP\_HOUR**: Extracted as the hour of the scheduled departure from CRS\_DEP\_TIME.
  + **Rationale**: Delays often accumulate throughout the day, particularly during peak times. By including DEP\_HOUR, the model can recognize these patterns, helping to capture typical delays by time of day.

#### **Flight and Route Characteristics**

* **DISTANCE**: Retained as a numerical feature, representing the distance between origin and destination airports.
  + **Rationale**: Flight distance affects delays, as longer routes face more potential disruptions and logistical complexities.
* **AIR\_TIME**: Actual in-air flight time.
  + **Rationale**: This captures time spent in the air and complements DISTANCE, as certain delays depend on flight duration rather than distance alone.
* **ELAPSED\_TIME**: Scheduled flight duration.
  + **Rationale**: This provides the planned time for each route, allowing the model to differentiate between short-haul and long-haul flights, which have different delay patterns.

#### **Average Weather Delay**

* **AVG\_WEATHER\_DELAY**:
  + **Description**: Calculated as the monthly average delay attributed to weather.
  + **Rationale**: Weather is a major delay factor. Rather than using real-time weather data (which would leak information), we averaged delays by month, capturing typical weather patterns’ effects on delays.

### **3. Final Dataset Structure**

The final cleaned and engineered dataset included:

* **Categorical Features**: AIRLINE, DEST, ORIGIN
* **Numerical Features**: DAY\_OF\_WEEK, MONTH, DEP\_HOUR, DISTANCE, AIR\_TIME, ELAPSED\_TIME, AVG\_WEATHER\_DELAY
* **Target Variable**: ARR\_DELAY

This feature set balanced essential predictors with reduced complexity, capturing the most relevant factors without introducing leakage.

### **4. Initial Model Exploration**

#### **Purpose of Initial Models**

To assess initial relationships between features and ARR\_DELAY, we employed a simple linear regression model. The goal of this step was not to finalize the prediction model but to understand correlations within the data and establish a benchmark for future, more sophisticated models.

#### **Baseline Model**

* **Description**: The baseline model predicted the mean delay for all flights.
* **Rationale**: The baseline provided a minimum benchmark, allowing us to assess improvements as we experimented with more complex models. If our final model’s performance is significantly better than the baseline, it indicates that our features are informative for predicting delays.
* **MSE: 983.42**

#### **Simple Linear Regression**

* **Description**: We applied a simple linear regression model to assess the strength and direction of linear relationships between features and ARR\_DELAY.
* **Rationale**: Linear regression is straightforward, making it a useful tool for initial correlation testing. By examining the model’s performance, we gauged whether delays could be modeled through linear relationships or if non-linear factors likely play a significant role.
* **Insights**:
  + The linear model demonstrated some correlation between features like DEP\_HOUR, DISTANCE, and ELAPSED\_TIME with delays, but it fell short in capturing the full complexity. The presence of non-linear relationships between factors like weather patterns and delay trends suggests that simple linear models are insufficient, pointing us toward more complex models for capturing these patterns effectively.
* **MSE: 773.75**

### **5. Future Directions**

With initial explorations pointing to non-linear relationships, the next steps involve:

1. **Testing Non-Linear Models**:
   * We aim to explore models that can capture non-linear relationships, such as K-Nearest Neighbors (KNN) or tree-based models. These models are better suited for complex, non-linear datasets and can reveal deeper patterns in delay tendencies.
2. **Further Feature Refinement**:
   * Interactions or transformations between variables (e.g., interactions between DEP\_HOUR and DAY\_OF\_WEEK) may improve model performance by capturing latent patterns in delay data.
3. **Using Cross-Validation**:
   * As we build and test more sophisticated models, cross-validation will help ensure our model generalizes well across different subsets of data, improving reliability.
4. New features: prone to weather delay, prone to airport route delay

### **6. Neural Network**

* Attempted with various activation functions (Logistic = 767.87, Tanh = 770, ReLu = 769)
* Logistic, with CV, L2 regression alpha = 0.001, MSE 767.55

### **7. Lasso/Ridge**

* Best Lasso MSE: 766.54
* Best Lasso Lambda (alpha): 0.0121
* Best Ridge MSE: 767.16
* Best Ridge Lambda (alpha): 2.6367

**11/1 Workshop Notes**

Questions

* How to reduce MSE
* Average 20-25 minutes
* MAE Or RMSE?
* Should we remove negative delays?
* Picking a threshold - can we set it to 50
  + Or is there a way to make it an indicator problem of magnitude of delay on a scale of 0 to

Notes:

* More of a business question than an ML question
* Predict outliers well? Or predict normal delays well?
* Be consistent with using MAE or MSE
* Model works a lot worse for outliers - right approach
* \* do this analysis >< 50 min - for ALL models
* Recommendation as analysts - do airport managers (or consumers) care about outliers? If so, use a model that is more susceptible to outliers.
* Everything is up to us, think about the problem we’re trying to solve
* Another thing we can think about: how to solve the outlier problem? - Albert
* It’s more about exploring
* Ex: after 10 min charged parking fee

1. Get everything running - tune methods, get 1 thing down, then try experimenting

15 minutes: threshold for FAA and USDOT

<https://aspm.faa.gov/aspmhelp/index/Types_of_Delay.html>

* “Flight delays of less than 15 minutes are not reported in OPSNET” - operations network

45 minutes threshold

Uriel: Lasso, Ridge

Francisco: Decision tree, random forest

Vivian: KNN

Albert: Boosting, Linear Regression

Zac: Neural Network

2 datasets:

1. ARR\_delay: 0 or 1
2. 0 to 1 probability of being delayed (>15 minutes)
3. <15: 0 (no delay), >15: delayed
   1. Convert all <15min delays to 0s (not delayed)

Regular dataset with minutes

Transform output to 0-1

**Presentation: 12 minutes + 3-5 min Q&A**