1. Introduction

Goal:
To build interpretable and operationally actionable machine learning models that predict flight delays, focusing on **controllable causes** such as airline inefficiencies or scheduling failures.
This enables airlines to proactively mitigate delay causes that are **within their control**, reducing downstream delays, penalties, and passenger dissatisfaction.

Key Sub-Goals:

- Identify whether delays will occur for a route/month (classification)
- Quantify expected delay in minutes per flight (regression)
- Predict a custom **OAI** score to reflect internal inefficiency level
- Interpret top features using SHAP for transparent decisions

---

2. Methodology & Pipeline

Data Source & Structure:

- 2015–2023 U.S. DOT monthly airline performance dataset
- ~180,000 rows aggregated to (year, month, carrier, airport)
- Target variables derived from:
    - arr_delay (total delay in minutes)
    - arr_del15 (number of delayed flights)
    - OAI (custom weighted controllable delay score)

Key Steps:

1. **Data Cleaning**
    - Removed rows with null values in delay-related columns
    - Encoded categorical variables (carrier, airport)
2. **Aggregation & Time-Aware Processing**
    - Created a monthly_df by grouping (year, month, carrier, airport)
    - Added lag features using .shift(1)
    - Used .rolling(3) for smoothed 3-month averages
3. **Feature Engineering**
    - Lagged features (previous month delay, counts)
    - Rolling means (trend stability)
    - Domain-derived ratios:

        - Delay per flight
        - Cancellation rate
    - Custom target : OAI

4. **Modeling**
    - **Classification**: XGBoostClassifier with scale_pos_weight for imbalance
    - **Regression**: XGBoostRegressor on arr_delay and OAI
    - Used log loss transform where necessary
5. **Interpretability**
    - Used SHAP to rank features by importance
    - Identified and selected top-K most impactful features

---

3. Key Findings from EDA

🔹 **Delay Trends**

- Delay rates increase during summer months (June–August) and end-of-year travel (Nov–Dec)
- Flights in March/April tend to show higher reliability

🔹 **Airport & Carrier Insights**

- ATL, ORD, DFW are high-volume airports with variable performance
- Carriers like AA, DL, UA have the most total delay minutes due to volume, but not always the worst delay rate

🔹 **Delay Distribution**

- arr_delay shows a **long tail**, with a small number of months causing massive delays
- arr_del15 / arr_flights is a better normalized signal for classification

🔹 **Correlation Findings**

- Strong correlation between carrier_ct, carrier_delay and future delays
- NAS and weather-related delays are unpredictable, weakly correlated with future delays

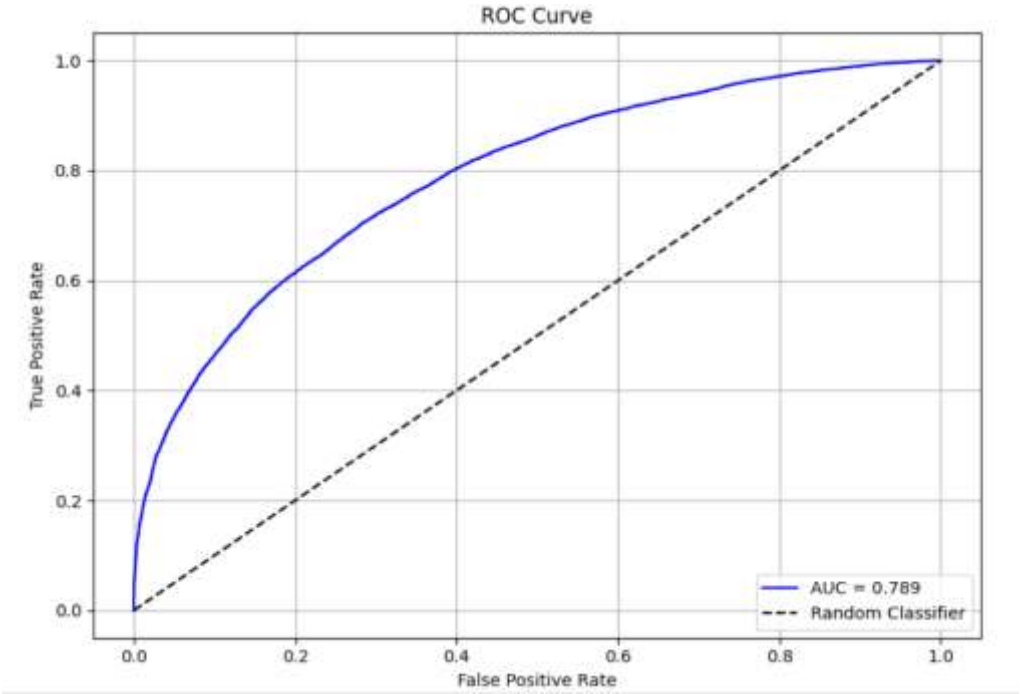
---

4. Model Performance

🔹 **Classification**

- Target: 1 if arr_del15 / arr_flights ≥ 0.2
- Model: XGBoostClassifier with SHAP-based feature pruning
- Metrics:

- **F1 Score**: 0.71
- **Precision**: 0.65
- **Recall**: 0.77



- SHAP reveals prev_carrier_delay, prev_arr_del15, and rolling_OAI as top predictors

🔹 **Delay Regression**

- Target: arr_delay / arr_flights
- Metrics:
  - **MAE**: 5.4 minutes
  - **RMSE**: 110.1 minutes
  - **R^2**: 0.17

🔹 **OAI Regression**

- Custom target focusing on controllable delay types
- Metrics:
  - **MAE**: 4.1
  - **RMSE**: 73.5
  - **R^2**: 0.16
- More stable than raw delay regression

---

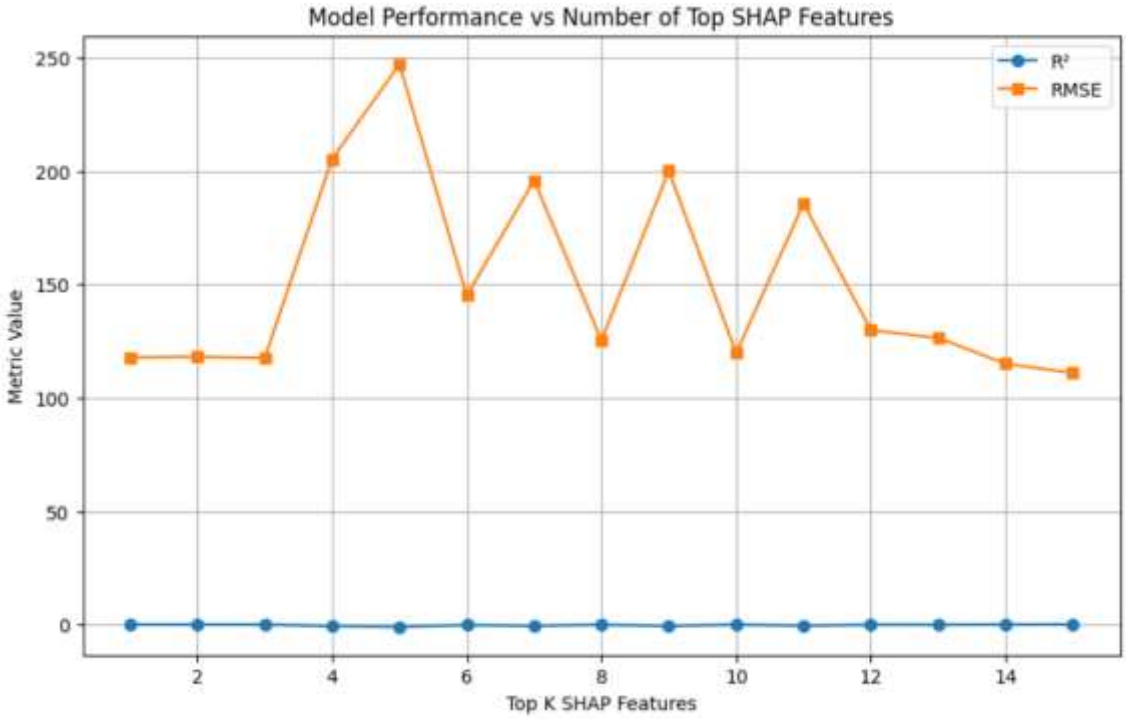
5. SHAP Interpretability

Top features (based on mean absolute SHAP values):

- prev_carrier_delay
- rolling_3mo_delay
- carrier_ct
- late_aircraft_ct
- prev_month_flights
- rolling_OAI
- cancellation_rate

Insights:

- Carrier-driven delays dominate importance
- Weather & NAS delays have near-zero SHAP values → not useful for predictions
- SHAP plots help trace each prediction back to its feature contributions



---

7. Why XGBoost?

XGBoost (Extreme Gradient Boosting) was selected because it strikes the best balance between performance, interpretability, and flexibility — especially for tabular structured data like this airline dataset.

Key Advantages of XGBoost:

Reason	Justification
Handles **imbalanced classes**	Used scale_pos_weight to address class imbalance in classification (95% non-delayed).
Naturally handles **missing values**	Useful in large aviation datasets where cancellations/diversions may introduce nulls.
High accuracy on tabular data	Regularized boosting helps avoid overfitting.
Compatible with **SHAP**	Makes model transparent and explainable → vital for operational decision-making.
Efficient & scalable	Trained quickly on 100k+ monthly records.
Tree-based → no need for scaling	Input features like delays, counts used as-is.

Alternatives Considered (but not chosen):

- **Logistic Regression**: Simple but underfit; can't capture non-linear time/delay trends.
- **Random Forest**: Comparable performance but slower and harder to interpret with SHAP.
- **Neural Networks**: Overkill; less interpretable, and less effective on structured tabular data.

---

8. Why These Targets?

Classification Target

- **Target:** `arr_del15 / arr_flights ≥ 0.2` → Binary: 1 if ≥20% of flights delayed, else 0
- **Why this was chosen:**

Reason	Explanation
Normalized Delay Signal	Raw delay counts (`arr_del15`) vary with traffic volume. Normalizing by `arr_flights` gives a fair indicator of delay severity across airports and carriers.
Meaningful Threshold (20%)	A 20% threshold defines "high delay" months — enough to be operationally concerning but not overly rare.
Operational Relevance	Helps identify months/routes that require intervention for planning and scheduling.
Simplified Binary Target	Allows for efficient classification modeling and delay-risk flagging.
Compatible with Lag Features	Works well with time-aware inputs like previous month's delay rate.

Regression Target 1

- **Target:** `arr_delay / arr_flights` → Delay in minutes per flight
- **Why this was chosen:**

Reason	Explanation
Operational Usefulness	Airlines benefit from knowing how late the average flight is, not just how many are delayed.
Volume Normalization	Prevents airports with high traffic from appearing more delayed due to flight count alone.
Continuous Value for Modeling	Enables regression modeling and fine-grained predictions.
Smoothed Target	Dividing by `arr_flights` reduces variance and stabilizes predictions.
Planning and Efficiency	Helps measure delay intensity — useful for performance benchmarking.

Regression Target 2

- **Target:** `OAI_per_flight` — a custom metric focusing on controllable delays
- **Why this was chosen:**

Reason	Explanation
Aligned with Business Goal	Focuses on delays that airlines can control (carrier-related).
Weighted Delay Components	Incorporates both magnitude (`carrier_delay`) and frequency (`carrier_ct`).
Forecastable and Actionable	Helps anticipate operational inefficiencies in advance.
Excludes Uncontrollable Factors	Avoids overemphasis on weather or airspace system delays.
Clear Strategic Use	Supports prioritization in resource planning and delay mitigation.

- **Summary Table**

Target	Type	Purpose
`arr_del15 / arr_flights ≥ 0.2`	Classification	Predict high-delay months (risk flag)
`arr_delay / arr_flights`	Regression	Estimate average delay per flight
`OAI_per_flight`	Custom Regression	Predict operationally controllable delay level

---

6. Recommendations

- **Deploy delay-alert models**: Predict high-risk months for each airport/carrier
- **Resource allocation**: Flag routes with rising OAI for crew, aircraft buffer planning
- **Feature monitoring**: Watch SHAP impact of carrier_ct, late_aircraft_ct to identify internal inefficiencies
- **Model extension**:

- Add weather forecast API data
- Add holiday + airport event calendar
- Use external data to predict "uncontrollable" components separately

- **Two-Stage Strategy**:

1. Classification to predict delay-risk months
2. Regression to estimate severity (OAI, delay mins)

Thank You