**Technical Analysis — Predicting Flight Delays**

**1. Data Acquisition and Structure**

The dataset was sourced from the **U.S. Bureau of Transportation Statistics (BTS)**, covering **five years (2014–2018)** of flight on-time performance records.  
After filtering the **top 9 busiest airports** and **top 5 airlines**, the merged dataset contained **~1.66 million rows × 20 columns**.  
The **target variable** is\_delay indicates whether the arrival delay exceeded 15 minutes (1 = delayed, 0 = on time).

Due to the **80/20 imbalance** (on-time vs delayed), evaluation metrics such as **recall**, **precision**, **F1-score**, and **ROC AUC** were prioritized over accuracy to better capture model sensitivity to delayed flights.

**2. Exploratory Data Analysis (EDA)**

Key insights emerged from EDA:

* **Temporal patterns:**
  + Highest delays: **June–August**
  + Lowest delays: **September**
* **Daily patterns:**
  + Delay rates peak between **3 PM–8 PM**
  + Early flights (before 8 AM) have low delay probability
* **Weekly trends:**
  + **Thursdays** experience more delays; **Tuesdays** the least.
* **Airline & airport patterns:**
  + **WN (Southwest)** and **OO (SkyWest)** show higher delay ratios.
  + **ORD** (Chicago) and **SFO** (San Francisco) are major delay hubs.

These observations guided the **feature design** and **risk-based alerting strategy**, ensuring that temporal and operational variability were encoded as predictive inputs.

**3. Feature Engineering**

Two rounds of feature expansion were applied:

1. **Iteration I – Time-based features:**
   * Extracted Month, Day\_of\_Week, and Dep\_Hour
   * Encoded airline and airport identifiers as categorical variables
   * Added Distance as a numerical feature

Logistic Regression achieved **accuracy ≈ 0.79**, but **recall ≈ 0**, indicating poor detection of delayed flights due to class imbalance.

1. **Iteration II – Enriched contextual features:**
   * **Holiday flag:** U.S. federal holidays
   * **Weather data:** mean temperature (TAVG), wind speed (AWND), precipitation (PRCP), and snow depth (SNOW)
   * Missing weather values were imputed by *station × month averages*
   * Resulting dataset expanded to **102 columns** (combined\_csv\_v2.csv)

These enhancements slightly improved model robustness but still failed to meet recall expectations.

**4. Modeling and Cloud Deployment**

**(a) Baseline – Logistic Regression**

* Framework: Local Scikit-Learn
* Accuracy: **0.79**, Recall: **≈0.00**
* Issue: Strong bias toward majority class ("on time")

**(b) Linear Learner (AWS SageMaker)**

* Scalable binary classifier trained via **Batch Transform**
* Key results:
  + Accuracy: **0.77**
  + Precision: **0.63**
  + Recall: **0.15**
  + F1 Score: **0.23**
  + ROC AUC: **0.70**
* Limitation: still underperforming on minority (delay) class.

**(c) XGBoost Ensemble**

* Objective: binary:logistic
* Parameters: 75 rounds, max\_depth = 5, eta = 0.15, subsample = 0.7
* Results:
  + Accuracy: **0.79**
  + Precision: **0.68**
  + Recall: **0.24**
  + F1 Score: **0.35**
  + ROC AUC: **0.75**

The **XGBoost model** demonstrated superior recall and balanced predictions, making it suitable for **production deployment** within a travel-booking alert system.

**5. Model Evaluation and Insights**

| **Metric** | **Logistic Regression** | **Linear Learner** | **XGBoost** |
| --- | --- | --- | --- |
| Accuracy | 0.79 | 0.77 | **0.79** |
| Precision | 0.60 | 0.63 | **0.68** |
| Recall | ~0 | 0.15 | **0.24** |
| F1-Score | 0.05 | 0.23 | **0.35** |
| ROC AUC | 0.60 | 0.70 | **0.75** |

* The **recall improvement (0.15 → 0.24)** shows better detection of delayed flights.
* **XGBoost** learns nonlinear dependencies between weather, schedule, and operational factors.
* The modest drop in speed is offset by its **higher reliability and interpretability** in delay prediction.

**6. Future Enhancements**

* Apply **SMOTE** or **class weighting** to mitigate class imbalance.
* Engineer **aggregate weather severity** or **airport congestion indices**.
* Explore **deep learning (LSTM, TabTransformer)** for temporal dependency modeling.
* Integrate **real-time weather API pipelines** for dynamic inference on live flight data.

**Summary**

This project successfully transitioned from **local prototyping (Logistic Regression)** to **cloud-based scalable modeling (AWS SageMaker)**.  
Through iterative feature enrichment and ensemble learning, it achieved a **balanced, deployable model** for predicting flight delays, meeting the core business goal of enhancing **customer experience through predictive alerts**.