Flight Delay & Cancellation Analysis (2019–2023)

Group 5

Name	Student ID
Poojitha Jayareddygari	801426875
Sahit Ceeka	801424751
Navya Reddy Thadisana	801425759
Sai Kiran Jagini	801484665
Jeevith Gowda	801455831

1. Introduction

Flight delays and cancellations are significant issues for both airports and passengers, affecting operational efficiency, cost, and customer satisfaction. This project implements an **end-to-end big-data pipeline using Apache Spark** to analyze, model, and simulate delay patterns using **3+ million U.S. flight records (2019–2023)**.

The solution integrates:

- Large-scale data ingestion
- Cleaning & feature engineering
- Extended Exploratory Data Analysis
- Spark SQL analytics
- Predictive ML Modeling (classification + regression)
- Structured Streaming for real-time predictions
- Live visualization dashboards

The goal is to understand the factors behind flight delays and build a streaming-compatible ML system that predicts delays in real time.

2. Dataset Overview

2.1 Source

- Kaggle: U.S. Domestic Airline Flight Delays & Cancellations (2019–2023)
- Period: Jan 2019 – Dec 2023
- Size: ~3 million rows (~2.1GB CSV)

2.2 Key Features Used

Column	Meaning
FL_DATE	Flight date
OP_UNIQUE_CARRIER	Airline code
ORIGIN, DEST	Airport codes
DEP_DELAY, ARR_DELAY	Delays in minutes
CANCELLED, DIVERTED	Event indicators
DISTANCE	Mileage

2.3 Preprocessing

- Standardized column names
- Removed nulls for critical fields (DEP_DELAY, ARR_DELAY)
- Added derived fields:
    - year
    - month
    - day_of_week
- Categorical encoding for airline, origin, and destination
- Saved curated dataset as Parquet

3. Methodology

3.1 Data Ingestion & Cleaning

Scripts: 01_ingest_eda.py

- Loaded raw CSVs using Spark
- Handled types & missing data
- Generated curated Parquet files

Sample Output (arr_delay_summary.csv):

month,avg_arr_delay,total_flights
1,2.19,260785
6,10.06,254998
7,9.49,278911
12,6.67,209504

3.2 Extended Exploratory Data Analysis (EDA)

Script: 02_extended_eda.py

3.2.1 EDA Tables

Located in: outputs/tables/

Top Delayed Airlines

airline,avg_arr_delay,num_flights
Allegiant Air,13.28,50179
JetBlue Airways,12.28,109447
Frontier Airlines,11.10,62711

Monthly Delay Patterns

month,avg_arr_delay,num_flights
1,2.19,260785
6,10.06,254998
7,9.49,278911

Route Delay Summary

origin,dest,num_flights,on_time_rate
DFW,LAX,13904,0.612
DEN,PHX,12011,0.654
ORD,LGA,8789,0.589

3.2.2 Visualizations

Located in: outputs/plots/

Includes:

- arr_delay_histogram.png
- avg_delay_airline.png
- monthly_avg_delay.png
- dep_vs_arr_scatter.png
- top_routes.png
- distance_vs_delay.png
- cancellations_pie.png
- origin_delay.png
- dest_delay.png

Key EDA Insight:
Arrival delay is almost entirely driven by departure delay (correlation ≈ 0.95).

3.3 SQL Analysis

Script: 02_sql_analysis.py

Used Spark SQL to compute aggregated performance metrics.

Airport On-Time Performance

airport,num_flights,on_time_rate
ATL,128209,0.892
DFW,110982,0.866
DEN,102614,0.842

3.4 Predictive Modeling

Script: 03_predictive_model.py

Performed both classification and regression using MLlib and Sklearn.

3.4.1 Feature Engineering

Features used:

dep_delay,
distance,
month,
day_of_week,
airline_idx,
origin_idx,
dest_idx

3.4.2 Models Applied

Task	Model
Classification	Logistic Regression, Random Forest
Regression	Linear Regression, Random Forest Regressor

3.4.3 Performance Metrics (metrics.txt)

TASK=classify, ALGO=lr, THRESH=15.0, ROC-AUC=0.9334
TASK=classify, ALGO=rf, THRESH=15.0, ROC-AUC=0.9108
TASK=regress, ALGO=rfreg, RMSE=22.5989, R2=0.8139

3.4.4 Confusion Matrix

y,yhat,count
0,0,780073
0,1,12576
1,0,42413
1,1,112058

Accuracy: ~89%

3.4.5 Feature Importance (Random Forest)

dep_delay,0.9109
year,0.0047
distance,0.0015
month,0.0007
airline_idx,0.00055
origin_idx,0.00050
day_of_week_idx,0.00038
dest_idx,0.00028

Conclusion:
Departure delay dominates prediction — if a flight departs late, it arrives late.

3.5 Streaming Simulation & Real-Time Prediction

3.5.1 Stream Batch Generation

Script: 04a_make_stream_batches.py
Creates micro-batches like:

data/stream/batch_0001.csv
data/stream/batch_0002.csv
...

3.5.2 Structured Streaming + ML Prediction

Script: 04_stream_predict.py

Each micro-batch triggers:

- Model loading
- Feature transformation
- Real-time prediction

Writing outputs to: outputs/stream_out/predictions/

3.5.3 Real Streaming Output Example

airline,avg_arr_delay
JetBlue Airways,31.4
Frontier Airlines,30.9
Spirit Airlines,25.4
Republic Airways,5.0

3.5.4 Live Visualization

Script: viz_stream_live.py
Shows evolving delay patterns as batches arrive.

4. Results

4.1 Key EDA Findings

- Summer months (June–July) have the highest average delays

- Low-cost airlines (Frontier, Allegiant, Spirit) show higher delay frequencies
- Major hubs (DFW, ORD, ATL) experience heavy congestion
- Departure delay almost perfectly predicts arrival delay

4.2 ML Modeling Results

- Logistic Regression performs best for classification (AUC = 0.9334)
- Random Forest gives strong regression performance ($R^2 = 0.81$)
- Classification accuracy reaches 89%
- Feature importance shows dep_delay contributes ~91% to prediction power

4.3 Streaming Results

- Real-time delay predictions generated successfully per batch
- Live dashboard visualizes:
    - Airline average delay over time
    - Prediction probabilities
    - Trend shifts across micro-batches

5. Limitations

Data Limitations

- Weather data absent (major predictor of delays)
- Some airports/airlines underrepresented
- Cancellation reasons not always populated

Model Limitations

- Heavy dependence on departure delay feature
- Cannot predict early delays without departure information

Streaming Limitations

- Simulated streaming (not real airline API feed)
- Spark on Codespaces experiences resource bottleneck=

System Limitations

- Random Forest model is relatively large
- Shuffle operations slow due to dataset size

6. Reproduction Guide

Step 1 — Clone Repository

git clone https://github.com/<your-repo>/ITCS-6190-Course-Project.git
cd ITCS-6190-Course-Project

Step 2 — Create Virtual Environment

python3 -m venv .venv
source .venv/bin/activate

Step 3 — Install Requirements

pip install -r requirements.txt

Step 4 — Add Raw Data

Place Kaggle CSVs into:

data/raw/

Step 5 — Run Full Pipeline

```
chmod +x run.sh
./run.sh
```

Step 6 — Run Streaming

```
chmod +x run_stream_predict.sh
./run_stream_predict.sh
```

Step 7 — Launch Live Dashboard

```
python src/viz_stream_live.py
```

7. Conclusion

This project successfully demonstrates a full big-data analytics and ML pipeline for flight delay analysis, including:

- Large-scale ingestion & transformation
- Comprehensive EDA insights
- Predictive modeling with strong accuracy (AUC ~0.93)
- Real-time ML predictions using PySpark Structured Streaming
- Reproducible scripts and automation via run.sh

The end-to-end system reveals clear operational trends and shows how Spark can be used not only for historical analysis but also for real-time prediction in airline operations.