SAMSUNG

Together for Tomorrow! Enabling People

Education for Future Generations

Next-Gen Airline Analytics & Real-Time Monitoring

SIC 7

BD Course

G 7

Agenda

- **Team**
- Introduction
- Problem Statement
- Methodology
- Key Tasks
- Results and Insights
- Conclusion
- Q&A

Team Members

Nancy Amr

Yasmina Ayman

Nasser Abdelfattah

Agenda

- Team
- **Introduction**
- Problem Statement
- Methodology
- Key Tasks
- Results and Insights
- Conclusion
- Q&A

Introduction

- **The Mission:** This project establishes a modern, end-to-end data platform designed to ingest, warehouse, and analyze massive volumes of airline on-time performance data.

- **The Dataset:** We are utilizing the extensive "Airline On-Time Performance" dataset, spanning over two decades (1987–2008), which tracks millions of flight operations including delays, cancellations, and routing information.

- **The Approach:** A hybrid architecture that combines:
  - **Batch Processing:** For historical analysis and warehousing using AWS S3 and Snowflake.
  - **Speed Layer:** For real-time anomaly detection using Kafka and Spark Structured Streaming.

Agenda

Problem Statement

•**Data Latency:** Traditional reporting relies on end-of-day batch processing. **Operational Blind Spots:** Without real-time monitoring, high-impact events—specifically delays exceeding 60 minutes—can go unnoticed until it is too late to mitigate passenger impact.

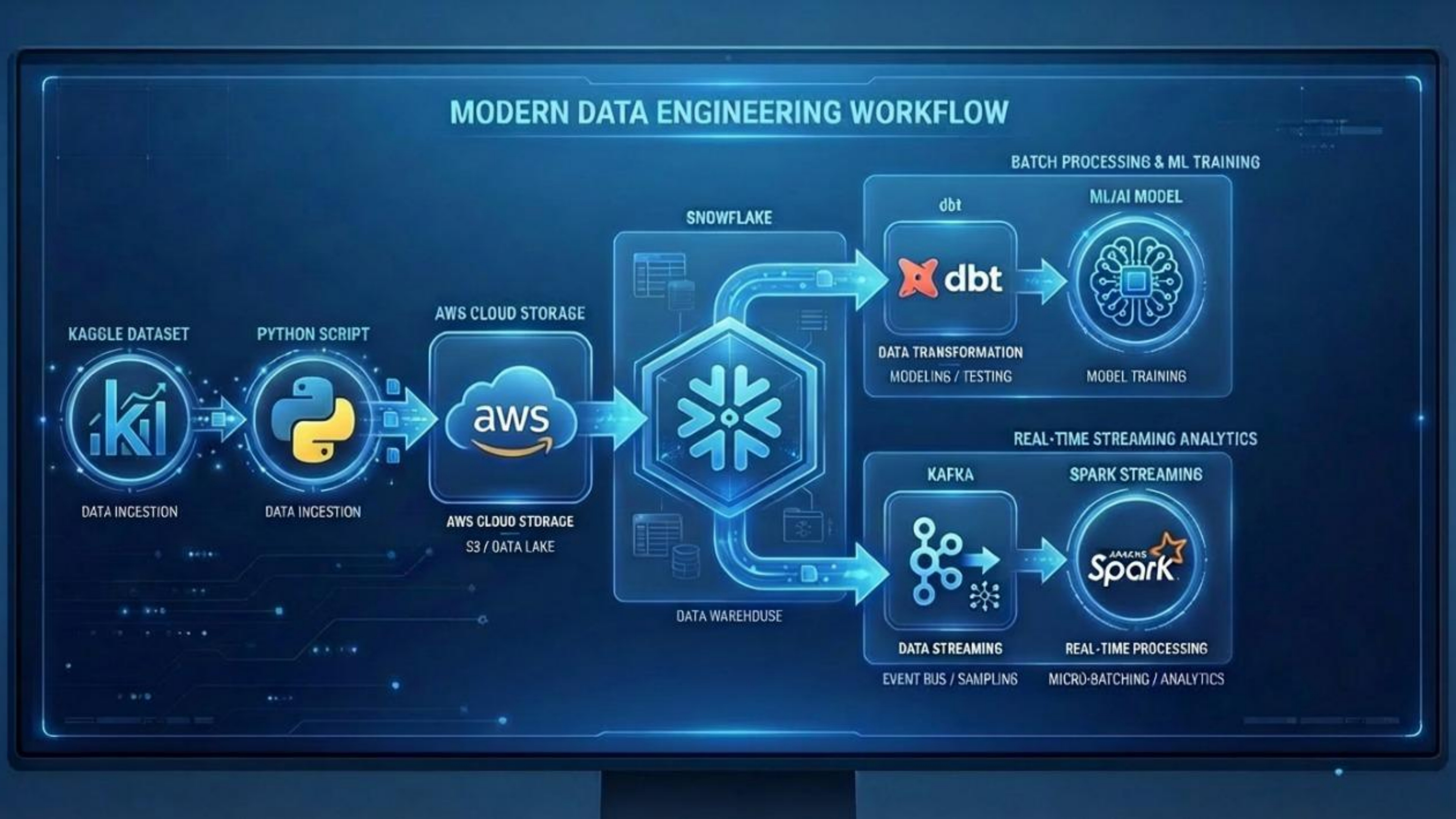
•**Data Accessibility:** Raw flight data is often locked in static files (CSV/Parquet) or external APIs (Kaggle), making it inaccessible for immediate SQL querying or machine learning applications without a robust extraction pipeline.

Agenda

- Team
- Introduction
- Problem Statement
- **Methodology**
- Key Tasks
- Results and Insights
- Conclusion
- Q&A

Methodology

- We propose an automated pipeline that seamlessly moves data from raw sources to a cloud data warehouse for deep analysis, while simultaneously tapping into the data stream to generate instant alerts for critical anomalies.


MODERN DATA ENGINEERING WORKFLOW
KAGGLE DATASET
DATA INCESTION
PYTHON SCRIPT
DATA INCESTION
AWS CLOUD STORAGE
aws
AWS CLOUD STDRAGE
S3 / OATA LAKE
SNOWFLAKE
DATA WAREHDUSE
BATCH PROCESSING & ML TRAINING
dbt
dbt
DATA TRANSFORMATION
MODELIN6 / TESTING
ML/AI MODEL
MOBEL TRAINING
REAL-TIME STREAMING ANALYTICS
KAFKA
DATA STREAMING
EVENT BUS / SAMPLIN6
SPARK STREAMING
Spark
REAL-TIME PROCESSING
MICRO-BATCHING / ANALYTICS

Agenda

- Team
- Introduction
- Problem Statement
- Methodology
- **Key Tasks**
- Results and Insights
- Conclusion
- Q&A

SAMSUNG

Together for Tomorrow! Enabling People
Education for Future Generations

Data Ingestion

Objective:

- To establish a secure, automated pipeline that transfers raw flight data from external sources (Kaggle) to centralized cloud storage (AWS S3) for downstream processing.

Pipeline Workflow:

- **Source Authentication:** The system connects to the Kaggle API using stored credentials to access the "Airline On-Time Performance" dataset.
- **Automated Extraction:** The script downloads the dataset as a ZIP file and extracts it into a temporary directory to ensure memory-efficient file handling.
- **Cloud Upload:** Utilizing the `boto3` library, the pipeline recursively uploads the extracted Parquet files to the AWS S3 bucket (`s3://airline-dataset-1/raw-data/`), preserving the original directory structure.

Data Ingestion

Why AWS S3?

- **Compatibility:** We upload to S3 first because Snowflake supports loading from cloud storage (S3, GCS) but does not support direct API connections to Kaggle.
- **Staging:** S3 acts as a persistent raw data layer (Bronze) before data enters the warehouse.

Data Ingestion

Security & Validation:

- **Policy Verification:** A pre-flight validation script confirms that IAM policies grant necessary permissions (PutObject, GetObject) before the pipeline runs.
- **Error Handling:** The process tracks upload success/failure rates and validates file sizes in real-time to ensure data integrity.

Data Warehousing with Snowflake

Architecture & Setup:

- **Medallion Architecture:** We established a RAW schema within the `AIRLINE_PROJECT` database to serve as the "Bronze Layer," storing unaltered source data.
- **S3 Integration:** configured an **External Stage** (`airline_s3_stage`) to act as a secure virtual bridge, allowing Snowflake to access files directly in AWS S3 without physical data movement.
- **File Formatting:** Defined a custom PARQUET_FF file format to correctly interpret the Snappy-compressed Parquet data structure.

Data Warehousing with Snowflake

- **Raw Integrity:** Defined all columns as VARCHAR types to prevent loading failures and preserve the original data state.
- **Data Lineage:** Integrated `FileName` and `LoadTimestamp` metadata columns to track source files and ingestion time.
- **Bulk Loading:** Utilized the COPY INTO command to ingest data directly from the S3 external stage.
- **Performance:** Successfully loaded **6.7 million records** with 100% data integrity and zero parsing errors.

SAMSUNG

Together for Tomorrow!
Enabling People
Education for Future Generations

Data Transformation

Data Transformation

Role of dbt:

•Orchestrates the SQL transformation pipelines.

•Manages dependencies between the raw tables and the final analytical views.

Machine Learning Integration

Flight Delay Prediction (Random Forest)

Purpose:

To leverage the refined historical data stored in the warehouse for predictive analytics and operational optimization.

Model Application:

Random Forest Regressor - deployed directly in Snowflake using Python stored procedures

Training Dataset:

- **Source**: Processed historical flight data (ML_FEATURES_TRAIN) (1987–2008)
- **Features Used:** Selected key features from carrier, route, airport, time, and base flight attributes (distance, day, hour, weekend/rush indicators)
- **Dataset Size:** Representative sample of the *full dataset* used to speed up model development and proof of concept. *(Full-data training planned in next iteration)*
- **Data Quality**: Trained on cleaned "Silver/Gold" layer data with outlier removal (-60 to +300 minutes)

Metrics (Sample-Based Training):

- **MAE (Mean Absolute Error)**: 18.76 minutes
- **RMSE (Root Mean Squared Error)**: 33.11 minutes
- **R^2 Score**: Strong correlation with actual delays

SAMSUNG

Together for Tomorrow! Enabling People

Education for Future Generations

Operational Dashboarding

Key Metrics Visualized:

Historical Analysis: Trends in on-time performance over the last two decades.

Carrier Performance: Comparison of delay statistics across different airlines.

Route Efficiency: Identification of the most and least reliable flight routes based on the "Gold" layer aggregations.

SAMSUNG

Together for Tomorrow! Enabling People
Education for Future Generations

The "Analytical Breakdown"

The dashboard breaks down airline operations into four key views:

1. **Growth:** Flight volume steadily increased from 1987 to 2005.
2. **Competition:** Delta Air Lines leads in total volume, followed by Southwest and American.
3. **Geography:** Traffic is heavily concentrated in major hubs like Atlanta, Chicago, and Dallas.
4. **Seasonality:** Flight volume is cyclical, with clear dips in February and peaks during Summer and Winter holidays.

The "Operational Focus" (Focus on efficiency)

Despite managing a massive volume of 116 million flights, the industry maintained a solid On-Time performance of nearly 81%. The average arrival delay was kept to just over 7 minutes. The data reveals that operational load is heaviest during the Summer months and December, requiring peak resource allocation during those times.

Data Streaming

Real-Time Flight Delay Anomaly Detection

- Why build a streaming layer?

The Objective is to detect and alert on "High Delay Anomalies" (delays greater than 60 minutes) the moment they occur.

It is a real-time pipeline that simulates a live feed of flight data while traditional batch processing provides insights hours or days late

Data Streaming

How the data flows through the system ?

Snowflake (Source) → Python Producer → Apache Kafka (Broker) → Spark (Processor).

Ingestion & Simulation:

- A Producer Script in Python is used in the connection to Snowflake and queries historical flight data from the year 2008 (2.3 million flight records).
- Each row is converted into a JSON message.

Data Streaming

Apache Kafka (Broker) :

- The producer publishes messages to a specific Kafka Topic while it holds the data until the consumer is ready to process it.
- Data integrity is verified by using a Console Consumer, confirming that raw JSON flight data (Arrival Time, Carrier, Delay) was successfully landing in the pipeline.

Real-Time Processing (Spark):

- A PySpark application is deployed to consume the Kafka stream.
- The system continuously monitors the ARRIVAL_DELAY column and it solates only those flights where the delay exceeds **60 minutes**.

Data Streaming

Anomaly Detection & Output:

- **Tagging:** When a high-delay flight is identified, Spark appends a new column: `ALERT_MESSAGE`.
- **The Message:** "HIGH DELAY DETECTED (>60m)" is attached to the specific flight record.
- The output clearly shows the ALERT_MESSAGE column appended to the data, serving as a flag for downstream alerting systems

Feature Work & Future Model Improvements

1. Scaling Up

- Current training: **Representative sample** for POC.
- **Next step:** Train model on the **full dataset (~118M rows)** for higher accuracy.
- Predict delays for **all upcoming flights** and estimate potential cancellations.

2. Streaming & Real-Time Integration

- Plan to integrate with **real-time streaming data** from **Kafka**.
- Enables:
  - Instant predictions for flights as data arrives
  - Dynamic operational decision-making
  - Improved airline scheduling, customer alerts, and congestion management

SAMSUNG

Together for Tomorrow! Enabling People
Education for Future Generations

Business Recommendations

- Plan flights and assign crew based on busy seasons (**summer & December**) and relax resources in slow months (**February**).
- Focus on improving operations at big airports (**Atlanta, Chicago, Dallas**) to avoid congestion.
- Send automatic passenger notifications when a delay is expected or detected.
- Fix or reschedule routes that are frequently delayed.
- Add weather data later to help predict delays better.
- Show delay expectations in the booking app to improve customer experience.

SAMSUNG

Together for Tomorrow!

Enabling People

Education for Future Generations