SUMMARY

1 WORFLOW SUMMARY

1.1 DATA COLLECTION, STAGING AND VALIDATION

- Collect flight tracking data from sources like AWS, Google Cloud, Lambda, SQL Server.
- Store raw data in a database.
- Perform initial data cleansing (filter irrelevant or incorrect data).
- Run ML validation (TensorFlow) to check for accuracy, business logic, and issues.
- If no issues → Move to next step.
- If issues found → Re-process data.

1.2 ERROR HANDLING

A few years ago, the Harvard Business Review found that on average, 47% of newly-created data records have at least one critical error. And one error is all it takes to create an anomaly in the analysis.[1]

- Group related actions into scopes and use scopes to handle errors collectively.
- Implement a try-catch pattern using scopes to manage errors.
- Create a "Try" scope for the main actions and a "Catch" scope for error handling. If an error occurs in the main scope, the error handling scope can be triggered to manage the error.
- Configure the "Catch" scope to run if the "Try" scope fails.
- By implementing an exponential retry policy, you can effectively manage transient failures and enhance the reliability of your automated workflows.
- The Retry policy is designed to handle transient failures, which are temporary issues that can often be resolved by retrying the action.
- **Exponential retry policy:** starts with a short retry interval and gradually increases the interval between retries. This approach helps to avoid overwhelming the system with frequent retries and allows more time for the issue to resolve.
- Implement logging and notifications to monitor and track errors.
- Use actions to log error details to a database, SharePoint list, or other storage solutions.
- Set up notifications to alert relevant stakeholders when an error occurs. For example, if an error is detected, log the error details and send an email notification to the support team for immediate attention.

1.3 DATA PREPROCESSING & ALERTS

- Further data cleansing and validation.
- Store in DWH (Data Warehouse) and filter using SQL stored procedures.
- Ingest processed data into Power BI for reporting and real-time analysis.

1.4 TRIGGERING THE ALERTING SYSTEM

- Power BI detects flight delays and triggers an alert.
- Alert is sent to an email service provider (SendGrid).
- Notification is also forwarded for multichannel notifications (Zapier, Twilio for SMS, etc.).

[1] Harvard Business Review: https://hbr.org/2017/09/only-3-of-companies-data-meets-basic-quality-standards

1.5 Email Composition & Delivery

- Prepare standard copies and make them dynamic on specific fields; auto-fill delay details, estimated new departure time, and possible compensations.
- Format emails using Power Automate (or a similar workflow automation tool).
- Send emails via SendGrid (or another SMTP service).

1.6 Customer Interaction & Feedback

Include options for:

- Checking real-time updates (Airline's website/app).
- Speaking to customer support (email, phone, chatbot).
- Collect feedback from customers about the experience.

1.7 Compensation Handling (If Applicable)

If delay exceeds threshold (e.g., 3+ hours), trigger:

- Loyalty programs (discounts, in-app points)
- Refund processing
- Rebooking links
- Provide links for compensation claims & support requests.

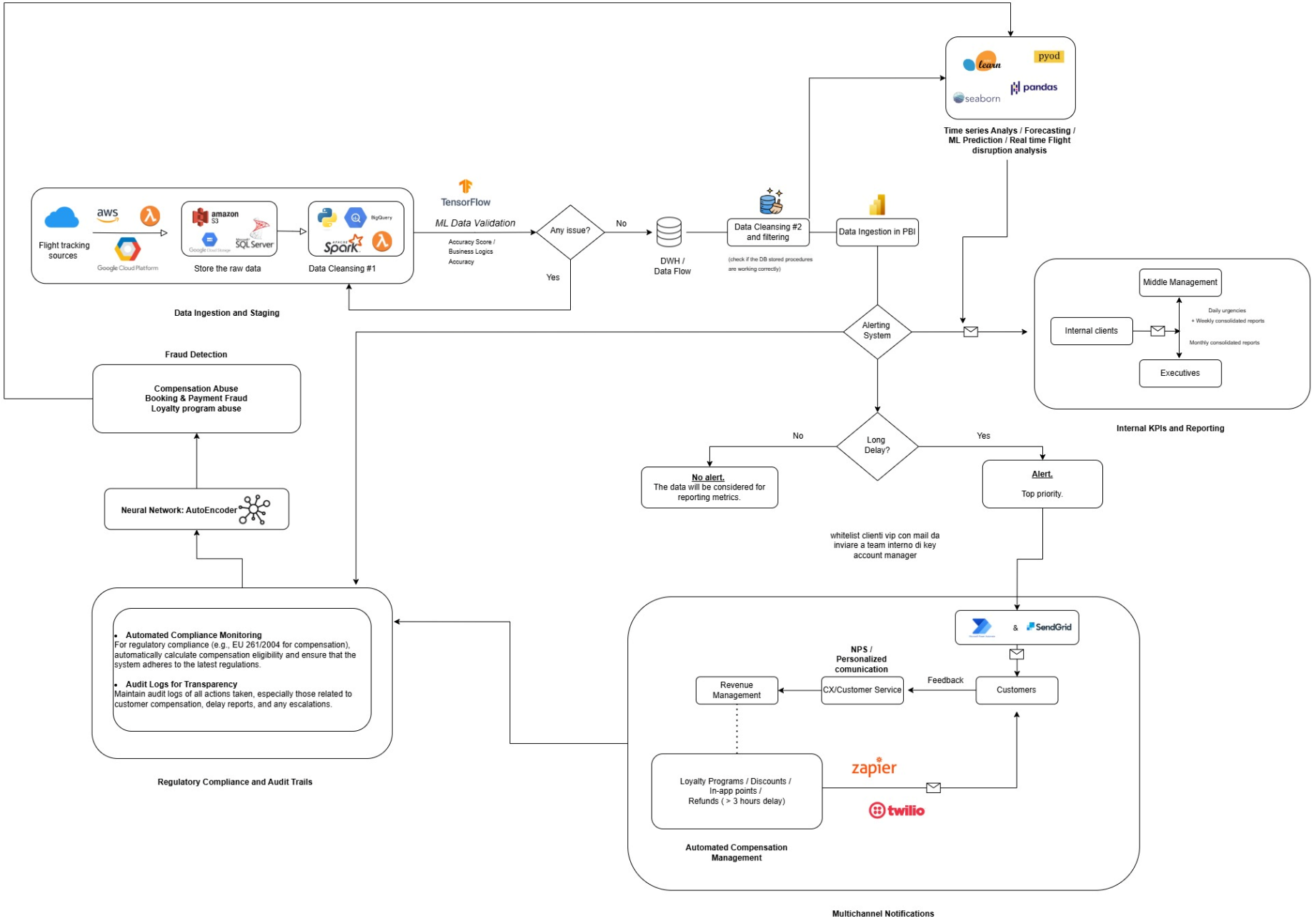
1.8 Compliance & Reporting

- Ensure regulatory compliance (e.g., EU 261/2004 for compensation).
- Maintain audit logs of actions taken (e.g., customer emails sent, refunds issued).
- Generate reports for middle management & executives.

1.9 Fraud Detection Use Cases in Airline Data Pipelines

- **Compensation Abuse**
  - Repeated or false claims for delayed flights
  - Duplicate voucher/redemption requests
  - Fraud detection model flags suspicious claim patterns
- **Booking & Payment Fraud**
  - Use of stolen credit cards or bots making fake bookings
  - Anomalies in payment location, timing, or device info
  - Apply fraud scoring at the booking ingestion layer
- **Loyalty Program Abuse**
  - Exploitation of frequent flyer benefits
  - Shared accounts, excessive redemptions, or repeated ID usage
  - Behavioral modeling to detect misuse

1.10 Flux Diagram – from data gathering to CX. A comprehensive approach to flight data tracking and email automation



2 FULL WORKFLOW LOGIC AND IMPROVEMENTS

This section will take a deep dive into the summarized and schematic Chapter 1.

This is an in-depth analysis.

2.1 DETAILED DESCRIPTION: DATA COLLECTION, STAGING AND VALIDATION

The first step in the process is ensuring that flight tracking data is consistently collected from reliable sources. Whether the data is obtained from APIs, databases, or flat files, it is crucial that it is regularly updated to **maintain accuracy and relevance**.

This step ensures that the flight data is current and available for further processing.

Furthermore, **data integrity** must be verified before moving to subsequent stages.

Incomplete or corrupted data can lead to inaccuracies, especially in critical fields such as **DEPARTURE DELAY**, **ARRIVAL DELAY**, **FLIGHT NUMBER**, and **DATE**.

Ensuring that these fields are populated correctly will be essential for making informed decisions based on the data.

2.2 DETAILED DESCRIPTION: ERROR HANDLING

Error Detection and Recovery in Workflow Steps

- **Handling Data Ingestion:** Implement strict data validation as soon as data is ingested. For example, verify that flight records contain all required fields (flight number, delay duration, passenger contacts) and meet expected formats/ranges before further processing. **This prevents bad data from propagating and causing downstream errors.**
- **Handling Data Preprocessing**: After ingestion, handle errors in transformation or enrichment logic gracefully. Use **defensive programming** in any code or formulas (e.g., check for division by zero in delay calculations, handle null values).
- **Handling Alert Triggering**: Ensure the logic that triggers alerts (e.g., a scheduled check or event-driven trigger when delay data updates) is robust. If using scheduling, monitor that the trigger runs on time; if an external event triggers the flow, handle the case where the event might not arrive.
- Incorporate **retry mechanisms** for transient failures.

2.3 DETAILED DESCRIPTION: FRAUD DETECTION

2.3.1 Compensation Abuse / Overclaim Detection

- **Context:** Airlines often compensate passengers for delays with vouchers, refunds, or loyalty points.
- **Problem:** Some users might **exploit the system**, submitting **false claims**, **duplicate requests**, or **manipulating ticket rebookings** to trigger multiple compensations.

- **Fraud Model Use:**
    - Train models to detect **suspicious claim patterns**
    - Flag **unusual frequency or value of claims per passenger**
    - Combine flight logs, delay data, and compensation logs

Where to implement: After the **delay classification**, in the **customer notification & compensation** module.

2.3.2 Flight Booking & Payment Fraud

- **Context:** Fake bookings, stolen credit card use, or multiple reservations by bots.
- **Problem:** Revenue loss, seat blocking, and identity misuse.
- **Fraud Model Use:**
  Use historical booking and payment data to train a **supervised ML model.**
    - Features: number of recent bookings, country mismatch, payment type, device info, time of booking
    - Model output: fraud probability → flag for manual review or auto-block

Where to implement: In the **pre-ingestion layer**, during or right after **real-time booking feed ingestion**.

2.3.3 Loyalty Program Abuse

- **Context:** Frequent flyer miles, upgrade credits, or partner benefits are sometimes **manipulated or claimed multiple times**.
- **Fraud Model Use:**
    - Detect accounts with **suspicious redemption patterns.**
    - Identify shared accounts, repeated usage of the same ID, or bot-driven activity

Where to implement: In the **customer profile enrichment layer** or during **reward processing**.

2.3.4 How to Build & Integrate the Model

Step	Description
1. Data Collection	Collect user actions, delay logs, compensation requests, booking logs, loyalty data
2. Feature Engineering	Count claims per user, time since last flight, number of overrides, origin-destination anomaly, etc.
3. Label Data (if available)	Use historical known fraud cases or flagged suspicious behavior
4. Model Training	Use supervised models (Random Forest, XGBoost, Logistic Regression) or unsupervised (Isolation Forest, Autoencoders for anomaly detection)
5. Scoring Pipeline	Run fraud detection models post-cleaning, pre-compensation
6. Alerting	If probability of fraud > threshold → flag, log, or block action.

2.3.5 Tools

Tool	Use
BigQuery ML	Fast, SQL-based fraud scoring directly on large datasets
Vertex AI	For training and deploying fraud models at scale
Cloud Functions	Lightweight fraud check triggers on new claims or booking
Cloud Pub/Sub + Dataflow	Real-time detection on streaming data
Looker / Power BI	For fraud investigation dashboards

2.4 Detailed Description: Data Preprocessing

Once the data has been collected, the next stage is data preprocessing, which focuses on transforming raw data into a structured format suitable for analysis.

This involves **cleansing the data and handling any inconsistencies**.

Key tasks include:

- Data Transformation
- Handling Missing Data: Using methods like interpolation to estimate delay times where data is missing or filling null values with appropriate defaults.

Additionally, **feature engineering** can be performed to enhance insights from the data:

- Time of Day: Categorize flights based on the scheduled departure time (e.g., Morning, Afternoon, Evening) to **analyze flight patterns** at different times of day.
- Delay Reason: Aggregate delay reasons such as AIRLINE DELAY, WEATHER DELAY, and SECURITY DELAY to provide a clearer picture of the causes of delays.

2.5 Data Pipeline Automation

Once the data is ready for analysis, automation plays a critical role in ensuring **timely updates**. To streamline the process, it is important to set up a scheduled job that periodically pulls the data from the source and updates the Power BI dashboard.

The job can be automated using tools such as Airflow, AWS Lambda, or Azure Functions.

Key actions within this step include:

- **Data Pulling**: Set up an automated task to retrieve the flight data at regular intervals (e.g., every 10 to 30 minutes) to **ensure that the dashboard reflects the latest information**.
- **Real-Time Updates**: Process the collected data and update the dashboard in real-time to provide the most up-to-date insights.

- **Notification Triggers**: Automatically generate notifications for flights falling into delayed categories. These notifications can be sent via email, SMS, or other messaging platforms (eg. the airline company mobile app).

2.6 Maintenance & Monitoring

Maintaining the health of the data pipeline is essential to ensure its continued success. Regular monitoring is necessary to detect and resolve any issues that may arise during data collection, transformation, or processing.

Key aspects to focus on:

- **Automated Checks**: Implement automated checks for data inconsistencies or pipeline failures. This helps in identifying problems early and ensuring that the system remains reliable.
- **Audit Logs**: Maintain logs of notifications and actions for auditing purposes, ensuring transparency and accountability in the process.
- **Manual Review**: While automation handles most of the tasks, it's important to allow for manual review in cases where the system cannot categorize or process certain flights accurately. This provides an opportunity for human intervention when necessary.

2.7 Further Improvements

The dashboard provides spot, punctual, and objective data, which is crucial for alerting systems and preventive controls to ensure data ingestion accuracy. **However, to add further value, we should consider building monitoring metrics that serve not only the operational side but also provide reports for middle managers and executives. This allows for a comprehensive understanding of the system's performance and enables better decision-making across all levels of the organization**.

By implementing Row-Level Security (RLS) in Power BI, you can create a single point of truth for your data, while ensuring that different users see only the data relevant to their role

- **Purpose**: RLS allows you to define roles and restrict access to specific data based on user credentials. For example, managers can access performance data only for their specific teams or regions, while executives can see a broader view
  of
  the entire system.
- **Implementation**: You can set up roles in Power BI where the data displayed is filtered based on the user's role or permissions. This ensures that users have access to only the data relevant to them, making the dashboard more secure and tailored to individual needs.

2.7.1 Middle Management & Executive Reporting

For middle managers and executives, having access to actionable, high-level insights is crucial for driving performance and aligning operational activities with broader organizational goals. To support decision-making, the reporting system must provide a comprehensive, yet concise, view of key metrics, performance trends, and forecasts that inform strategic planning.

Customizable views enable managers to focus on the metrics that are most relevant to their function, whether it's operational efficiency, cost management, or resource allocation.

Based on the fields in the ***"Flight Delays"*** Power BI dashboard, we can implement a variety of valuable metrics to help track operational efficiency

2.7.2 Additional Metrics

2.7.2.1 Flight Delay Metrics

These metrics help track the performance of flights, providing insights into delays and operational efficiency.

- **Average Departure Delay**:
  - **Metric**: The average delay in the DEPARTURE DELAY field.
  - **Purpose**: Helps track how timely flights are departing, highlighting areas for improvement in scheduling or ground operations.
- **Average Arrival Delay**:
  - **Metric**: The average delay in the ARRIVAL DELAY field.
  - **Purpose**: Tracks how well the arrival operations are running, helping to pinpoint areas for improvement in the arrival process (e.g., weather, air system delays).
- **On-time Performance Rate**:
  - **Metric**: Percentage of flights that have no delay (DEPARTURE DELAY ≤ 0).
  - **Purpose**: Measures operational efficiency and customer service quality. A higher percentage indicates good operational performance.

2.7.2.2 Flight and Operational Efficiency Metrics

These metrics help gauge the operational efficiency of the airline.

- **Average Taxi Time**:
  - **Metric**: Average **TAXI OUT** and **TAXI IN** times.
  - **Purpose**: Helps evaluate the efficiency of ground operations, including how long it takes for an aircraft to taxi out to the runway and taxi back to the gate after landing.
- **Air Time Efficiency**:
  - **Metric**: **AIR TIME** / **ELAPSED TIME**.
  - **Purpose**: Measures how much of the total flight time is spent in the air versus other operational factors like taxiing, delays, etc. A higher ratio indicates more efficient air operations.

- **Flight Distance Efficiency**:
  - **Metric**: **DISTANCE** / **ELAPSED TIME**.
  - **Purpose**: Helps measure the efficiency of a flight's travel distance relative to its elapsed time. A longer travel time relative to distance could indicate operational inefficiencies.
- **Scheduled vs Actual Departure Time**:
  - **Metric**: Difference between **SCHEDULED DEPARTURE** and **DEPARTURE TIME**.
  - **Purpose**: Measures the accuracy of flight scheduling, with the goal of minimizing discrepancies between scheduled and actual departure times.

2.7.2.3 Customer Experience Metrics

These metrics help assess customer satisfaction and reliability.

- **Flight Cancellation Rate**:
  - **Metric**: Percentage of flights where **CANCELLED** is TRUE.
  - **Purpose**: Helps measure the reliability of the airline. A higher rate of cancellations could signal operational or scheduling issues that need to be addressed.

- **Diversion Rate**:
    - **Metric**: Percentage of flights that were **DIVERTED**.
    - **Purpose**: Tracks how often flights are diverted, which may be caused by operational issues, weather, or other external factors.
- **Flight Completion Rate**:
    - **Metric**: Percentage of flights that were completed successfully without cancellations or diversions.
    - **Purpose**: Measures the success rate of flights being completed as scheduled.

2.7.2.4 Delay Cause Metrics

These metrics track the various factors contributing to delays, allowing better identification of operational bottlenecks.

- **Delay by Reason**:
    - **Metric**: Breakdown of delays based on categories such as **AIRLINE DELAY**, **WEATHER DELAY**, **SECURITY DELAY**, **LATE AIRCRAFT DELAY**, etc.
    - **Purpose**: Helps identify the root causes of delays, allowing for focused improvements in specific areas (e.g., weather-related delays, operational issues).
- **Weather Delay Impact**:
    - **Metric**: Percentage of total delays attributed to **WEATHER DELAY**.
    - **Purpose**: Helps quantify the impact of weather on operations and may inform future decisions on managing scheduling during adverse weather conditions.
- **Late Aircraft Delay Impact**:
    - **Metric**: Percentage of total delays attributed to **LATE AIRCRAFT DELAY**.
    - **Purpose**: Measures how often delays from previous flights are impacting the current flight schedule, providing insight into how turnaround times can be improved.

2.7.2.5 Operational Forecasting and Performance Trends

These metrics are more forward-looking and help both middle management and executives plan for future performance.

- **Flight Performance Forecasting**:
    - **Metric**: Forecasted **DEPARTURE DELAY** and **ARRIVAL DELAY** based on historical trends and real-time data.
    - **Purpose**: Helps predict potential delays for upcoming flights, allowing for proactive planning and customer communication.
- **Historical Performance Trends**:
    - **Metric**: Year-over-year or month-over-month comparisons of **DEPARTURE DELAY** and **ARRIVAL DELAY** averages.
    - **Purpose**: Tracks improvements or declines in operational performance over time, giving management the data they need to understand seasonal trends and make strategic adjustments.

2.7.2.6 Cost-Related Metrics

These metrics help evaluate the financial impact of flight operations.

- **Operational Cost per Delayed Flight**:
    - **Metric**: Average operational cost (fuel, crew time, etc.) per flight that experiences a delay.
    - **Purpose**: Helps assess the financial impact of delays and identify areas for cost reduction in operations.
- **Cost of Cancellations**:

- **Metric**: Average cost per canceled flight.
- **Purpose**: Measures the financial consequences of cancellations, including compensation, rebooking, and other associated costs.

2.7.2.7 Compliance and Regulatory Metrics

These metrics are important for ensuring that the airline is complying with regulations.

- **Regulatory Compliance on Scheduled Arrival Times**:
  - **Metric**: Percentage of flights arriving within scheduled time windows (as per regulatory standards).
  - **Purpose**: Helps ensure that the airline adheres to industry regulations regarding punctuality.

Incorporating additional data, such as total flight distance, engine type, fuel capacity and historical trends, can significantly enhance the optimization process for flight operations. These data points provide a deeper layer of insight into the factors that influence flight performance, efficiency, and overall operational success. By leveraging this extra information, airlines can make more informed decisions and take proactive measures to improve various aspects of their operations.

As an analytical business partner supporting airlines, we can position ourselves as a key enabler of operational optimization, cost reduction, and performance enhancement by leveraging advanced data analytics. Our approach would focus on helping airlines use their data more effectively, offering actionable insights to improve decision-making across various aspects of their operations.