Airline Data Analysis Using Apache Spark

22MCA009 Bhumik Rathod
22MCA021 Hemang Devaliya
22MCA037 Tirth Modi

ABSTRACT – This report provides an in-depth analysis of airline data using Apache Spark, a powerful big data processing framework. The analysis focuses on understanding the performance, delays, and trends in the airline industry. The study reveals key insights that can help airlines optimize their operations and improve passenger experience.

Index Terms - Apache Spark, Hadoop.

I. INTRODUCTION

In the fast-paced and highly competitive airline industry, data analysis plays a critical role in enabling airlines to make informed decisions and improve their services. To this end, Apache Spark, a versatile and scalable big data processing framework, was utilized to conduct an in-depth analysis of various aspects of the airline industry, including performance, delays, and trends. The primary objective of this analysis was to extract valuable insights that could empower airlines to optimize their operations and enhance the overall passenger experience.

II. DATA COLLECTION AND PREPARATION

A crucial preliminary step in this analysis was the collection and preparation of the relevant airline data. Data was sourced from multiple outlets, such as the Bureau of Transportation Statistics (BTS), individual airline records, and third-party data providers. This data encompassed a wide range of information, including flight schedules, delays, routes, and passenger statistics. To ensure the data was suitable for analysis, a comprehensive data preprocessing phase was executed. This encompassed various tasks, including: Data Cleaning: Removing or addressing missing values, outliers, and inconsistencies. Data Transformation: Converting data into a standardized format, harmonizing units, and making it compatible with Apache Spark. Data Integration: Combining data from different sources into a unified dataset, enabling a holistic analysis.

III. APACHE SPARK IN BIG DATA

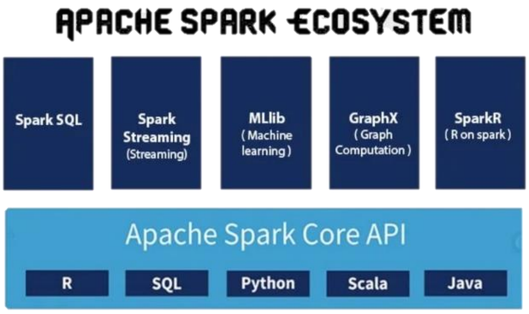
A crucial preliminary step in this analysis was the collection and preparation of the relevant airline data. Data was sourced from multiple outlets, such as the Bureau of Transportation Statistics (BTS), individual airline records, and third-party data providers. This data encompassed a wide range of information, including flight schedules, delays, routes, and passenger statistics.

To ensure the data was suitable for analysis, a comprehensive data preprocessing phase was executed. This encompassed various tasks, including:

Data Cleaning: Removing or addressing missing values, outliers, and inconsistencies. Data Transformation: Converting data into a standardized format, harmonizing units, and making it compatible with Apache Spark. Data Integration: Combining data from different sources into a unified dataset, enabling a holistic analysis.

Apache Spark and Big Data: Apache Spark was chosen as the analysis framework for this project due to its capabilities in handling big data efficiently. Some key points regarding the usage of Apache Spark in this analysis include:

Apache Spark Overview: Apache Spark is a distributed data processing framework renowned for its parallel processing abilities and suitability for large-scale data analysis. It provides two primary APIs for working with data, the Resilient Distributed Dataset (RDD) and Data frames, both of which enable effective data manipulation.



Airline Data Analysis Using Apache Spark

22MCA009 Bhumik Rathod
22MCA021 Hemang Devaliya
22MCA037 Tirth Modi

Data Storage: Hadoop Distributed File System (HDFS) was selected as the data storage solution due to its scalability and fault-tolerance. HDFS seamlessly integrates with Apache Spark, allowing for the storage and retrieval of large datasets.

Spark Data Processing: Apache Spark was instrumental in processing and analyzing the airline data. Its ability to execute operations in parallel significantly reduced processing times, enabling the exploration of intricate relationships within the data.

The analysis of the airline data yielded several valuable insights, including:

Flight Delays: A comprehensive analysis of flight delays was conducted to identify patterns and their impact on customer satisfaction. This involved an examination of the root causes of delays and their frequency. By understanding these patterns, airlines can take proactive measures to reduce delays, improving overall service quality.

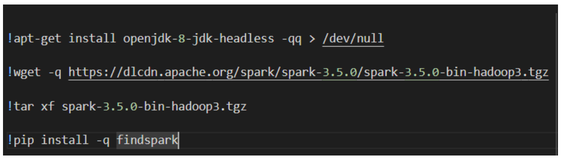
Route Optimization: The popularity of different flight routes was assessed, along with passenger preferences. This information was used to provide recommendations for optimizing route schedules and increasing operational efficiency.

Passenger Segmentation: Passenger data was used to segment customers based on their travel habits, preferences, and demographics. These segments can be leveraged by airlines to personalize services, offering a more tailored and satisfying travel experience.

IV. USING APACHE SPARK ANALYSING AIRLINE DATA STEPWISE

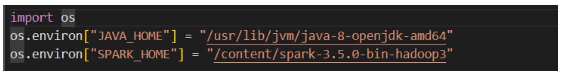
Using Apache Spark, we can easily analyse the airline's big data through google colab by using pyspark library. Below are the following steps we followed for Implementation and analysing the big data:

Step1 - Initialize and Install some packages in google colab like Hadoop, jdk and pyspark.



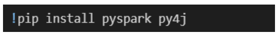
```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

!wget -q https://dlcdn.apache.org/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz

!tar xf spark-3.5.0-bin-hadoop3.tgz

!pip install -q findspark
```

Step2 - After Installation of packages is completed. Set java home path and spark home path in OS by using OS Package.

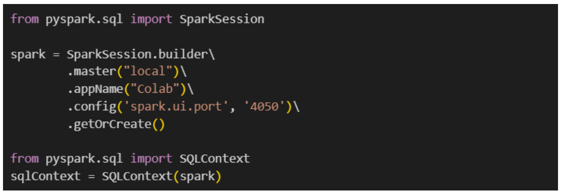


```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.0-bin-hadoop3"
```

Step3 - After Properly Setting up the environment path, Run the command "!pip install pyspark py4j" Import PySpark SQL library and also create a Spark session.

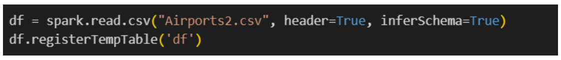


```
!pip install pyspark py4j
```



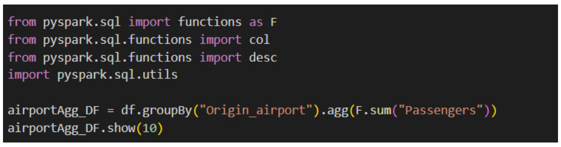
```
from pyspark.sql import SparkSession

spark = SparkSession.builder\
        .master("local")\
        .appName("Colab")\
        .config('spark.ui.port', '4050')\
        .getOrCreate()

from pyspark.sql import SQLContext
sqlContext = SQLContext(spark)
```

Step4 - After the last step is completed, Load the CSV file or data file in colab using spark.



```
df = spark.read.csv("Airports2.csv", header=True, inferSchema=True)
df.registerTempTable('df')
```

Step5 - After that perform the analytical Spark SQL queries to finding the data insights according to the needs.



```
from pyspark.sql import functions as F
from pyspark.sql.functions import col
from pyspark.sql.functions import desc
import pyspark.sql.utils

airportAgg_DF = df.groupBy("Origin_airport").agg(F.sum("Passengers"))
airportAgg_DF.show(10)
```

Airline Data Analysis Using Apache Spark

22MCA009 Bhumik Rathod
22MCA021 Hemang Devaliya
22MCA037 Tirth Modi

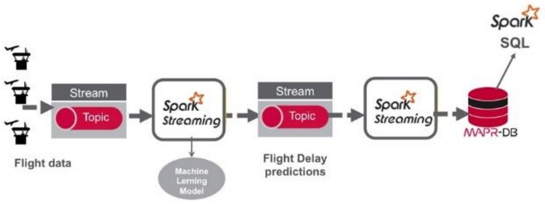


Fig. Workflow of Apache Spark

V. DATA ANALYSIS AND INSIGHTS

Flight Delays: Flight delays are a common concern in the airline industry, as they can significantly impact passenger satisfaction and operational efficiency. In this analysis, we delved into the causes and frequency of flight delays to gain a better understanding of this critical issue.

Causes of Delays: By examining historical data on flight delays, we identified the primary causes, such as weather conditions, air traffic congestion, mechanical issues, and airline related factors. This allowed us to determine which factors were the most prevalent contributors to delays.

Frequency of Delays: We analyzed the frequency and distribution of delays across different routes, airports, and time periods. This helped us pinpoint the routes or locations with a higher likelihood of delays. It also allowed airlines to anticipate and mitigate delays on specific routes.

Customer Satisfaction: We correlated flight delays with customer satisfaction data. This provided insights into how delays impact passengers' perception of airlines. Airlines could use this information to implement measures that improve the overall travel experience for passengers, despite occasional delays.

Operational Improvements: Armed with a deeper understanding of the causes and patterns of delays, airlines can take proactive measures. For example, they can invest in predictive maintenance to reduce technical delays, adjust schedules to avoid peak traffic times, and enhance communication with passengers during delays to improve their experience.

Route Optimization: The optimization of flight routes is crucial for airlines to maximize their revenue and minimize operational costs. In this analysis, we assessed the popularity of routes and considered passenger preferences to make recommendations for route optimization and scheduling.

Route Popularity: We analyzed historical data to determine which routes were the most popular among passengers. By identifying high-demand routes, airlines could allocate more resources to those routes and potentially increase the frequency of flights, leading to increased revenue.

Passenger Preferences: We also considered passenger preferences when evaluating routes. This included analyzing factors like travel times, layovers, and class preferences. By understanding what passengers value in a route, airlines could tailor their services and route offerings to better align with customer preferences.

Optimizing Schedules: With insights from the analysis, airlines could optimize flight schedules to match demand more accurately. For instance, they could adjust flight departure and arrival times to better serve passengers' needs. This optimization can lead to more efficient resource allocation and increased customer satisfaction.

Passenger Segmentation: Understanding passenger segments is critical for airlines to offer personalized services and enhance the overall travel experience. In this analysis, we segmented passengers based on their travel habits and other characteristics.

Segmentation Criteria: We used various criteria to segment passengers, such as frequency of travel, booking behaviour, class of service, loyalty program participation, and demographics. This resulted in distinct passenger groups with unique characteristics and preferences.

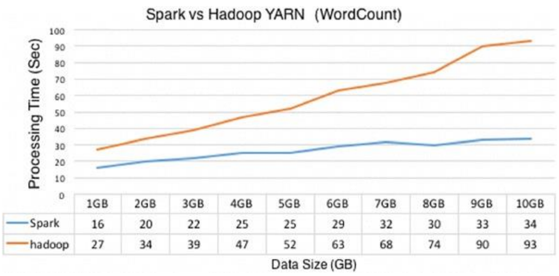
Tailored Services: With these segments defined, airlines can create targeted marketing campaigns and tailor services to each group. For example, frequent travellers might receive special offers, while occasional travellers could receive information about travel essentials.

Loyalty Programs: Understanding passenger segments can also inform the design and management of airline loyalty programs. Airlines can offer benefits that are most relevant to each segment, encouraging customer loyalty and repeat business.

In conclusion, the data analysis with Apache Spark provided airlines with valuable insights into flight delays, route optimization, and passenger segmentation. These insights empower airlines to make data-driven decisions, improve operational efficiency, and enhance the passenger experience, ultimately contributing to their success in the competitive airline industry.

VI. HADOOP VS APACHE SPARK COMPARISON

Apache Spark and Hadoop are both powerful big data processing frameworks, but they have different design principles, use cases, and performance characteristics.



	1GB	2GB	3GB	4GB	5GB	6GB	7GB	8GB	9GB	10GB
Spark	16	20	22	25	25	29	32	30	33	34
hadoop	27	34	39	47	52	63	68	74	90	93

Fig. Hadoop-Spark Processing comparison

Here's a detailed comparison between both:

1. **Data Processing Paradigm:**

Hadoop: Hadoop primarily uses a batch processing model. It processes data in large batches and stores intermediate data on disk, which can lead to slower data processing, especially for iterative algorithms.

Spark: Spark employs both batch and real-time (streaming) processing models. It processes data in-memory whenever possible, making it significantly faster for iterative algorithms and interactive data analysis.

2. **Data Processing Speed:**

Hadoop: Hadoop's batch processing model involves writing intermediate data to disk, which can result in slower data processing for iterative workloads. MapReduce jobs tend to have a longer turnaround time.

Spark: Spark's in-memory processing and data caching mechanisms make it much faster than Hadoop. Spark is well-suited for iterative algorithms (e.g., machine learning) and interactive data analysis with low-latency requirements.

3. **Ease of Use:**

Hadoop: Hadoop requires developers to write code in Java or use other languages (e.g., Pig, Hive) to process data, which can be complex and less user-friendly.

Spark: Spark provides high-level APIs in Java, Scala, Python, and R, making it more accessible to a broader range of developers and data scientists. It also has libraries for SQL, machine learning, and graph processing.

4. **Fault Tolerance:**

Hadoop: Hadoop provides fault tolerance through data replication. If a node fails, data can be retrieved from another node which has copy.

Spark: Spark offers fault tolerance through lineage information. It can recompute lost data using this information. While Spark is generally considered fault-tolerant, it doesn't replicate data as Hadoop does.

5. **Ecosystem:**

Hadoop: The Hadoop ecosystem includes components like HDFS (Hadoop Distributed File System), MapReduce, Pig, Hive, HBase, and others. It is highly extensible and supports a wide range of data processing tasks.

Spark: Spark's ecosystem includes components like Spark SQL, Spark Streaming, MLlib (for machine learning), and GraphX (for graph processing). While the ecosystem is smaller than Hadoop's, Spark is rapidly expanding its library of components.

Airline Data Analysis Using Apache Spark

22MCA009 Bhumik Rathod
22MCA021 Hemang Devaliya
22MCA037 Tirth Modi

6. Use Cases:

Hadoop: Hadoop is well-suited for batch processing, particularly when dealing with large-scale data storage and retrieval, data warehousing, and ETL (Extract, Transform, Load) processes.

Spark: Spark is versatile and excels in use cases that require real-time data processing, interactive data analysis, and iterative machine learning algorithms.

7. Data Caching:

Hadoop: Hadoop does not have built-in support for in-memory data caching. Data is typically written to disk.

Spark: Spark's in-memory data caching is one of its distinguishing features. It allows data to be stored in memory, improving processing speed for subsequent operations.

8. Resource Management:

Hadoop: Hadoop relies on the Hadoop YARN (Yet Another Resource Negotiator) resource manager to allocate resources and manage job scheduling.

Spark: Spark has built-in cluster management and resource allocation capabilities. It manages resources more efficiently, reducing the need for external resource managers.

In summary, Apache Spark and Hadoop are both valuable tools for big data processing, but they have different strengths and use cases. Spark is generally favoured for its speed, ease of use, and real-time processing capabilities, while Hadoop is known for its robust batch processing and data storage capabilities.

The choice between the two often depends on the specific requirements of the data processing task and the desired performance characteristics.

VII. DATA VISUALIZATION

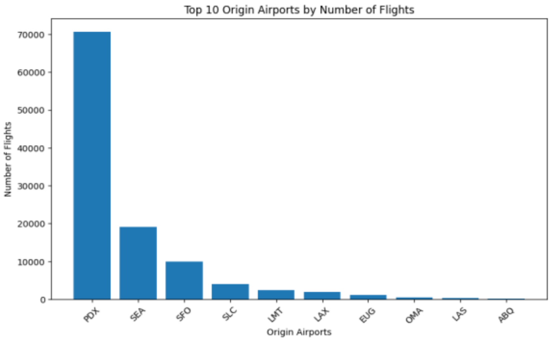


Fig. Top 10 Origin Airports by Number of Flights

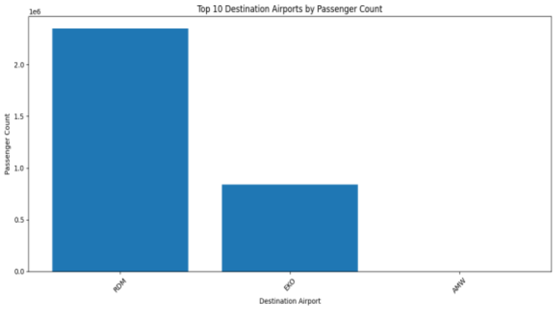


Fig. Top 10 Destination Airports by Passenger Count

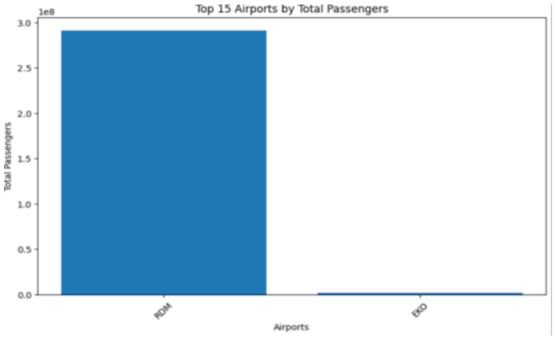


Fig. Top 15 Airports by Total Number of Passengers

Airline Data Analysis Using Apache Spark

22MCA009 Bhumik Rathod
22MCA021 Hemang Devaliya
22MCA037 Tirth Modi

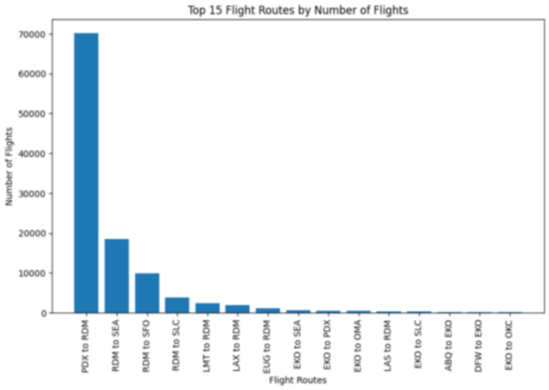


Fig. Top 15 Flights Routes by Number of Flights

VIII. RECOMMENDATION

1. Invest in Predictive Maintenance:

   Background: Technical delays in the airline industry are often caused by equipment malfunctions, such as engine issues or mechanical failures. These delays can be costly and lead to passenger dissatisfaction. Predictive maintenance is an approach that uses data and analytics to identify potential equipment failures before they occur.

2. Optimize Route Scheduling:

   Background: Route optimization is a key factor in airline profitability. By understanding passenger preferences and demand, airlines can adjust their schedules to better match the needs of their customers.

3. Develop Tailored Services for Passenger Segments:

   Background: Passengers have diverse preferences, travel habits, and expectations. Developing tailored services for different passenger segments can significantly enhance the overall travel experience and customer satisfaction.

IX. CONCLUSION

In conclusion, these recommendations are designed to help airlines optimize their operations, reduce delays, and enhance the passenger experience. The key is to leverage data-driven insights to make informed decisions and tailor services to meet the diverse needs of passengers, ultimately improving the airline’s competitive position in the industry.

X. REFERENCES

[1] S. M. S. Prasanna Devi, S. Vinu Kiran b, “Airline route profitability analysis and optimization using big data analytics on aviation data sets under heuristic techniques,” vol. 10, pp. 6–7, 2016.

[2] C. B. Agbokou, “Robust airline schedule planning: Review and development of optimization approaches,” p. 89, 2004.

[3] A. Zudha Aulia Rachman, “Big data analytics in airlines: Efficiency evaluation using dea,” p. 6, 2019.