

Milestone -03

DataDev

4/25/2024

Team Members

- Sayendranadh Chowdary Devabahktieni
- Bharath Komineni
- Manoj
- Ashok Swarna

Project Implementation Steps

Step 1: **Data Preparation:**

- Obtained the airline fare dataset from a reliable source or API.
- Ensured that the dataset is in a suitable format for PySpark, such as CSV or Parquet.
- Preprocessed the dataset if necessary, handling missing values, encoding categorical variables, and ensuring uniformity in data types using PySpark DataFrame operations.

Step 2: **PySpark Cluster Setup:**

- Installed Apache Spark and set up the PySpark environment.
- Configured PySpark environment variables such as `SPARK_HOME` and `PYSPARK_PYTHON`.

Step 3: **PySpark Implementation:**

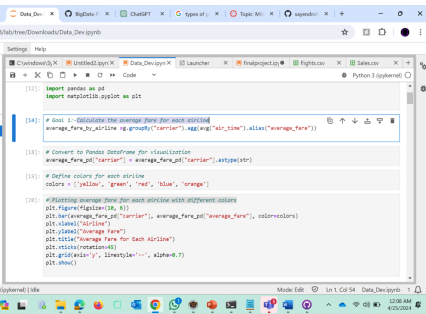
- Created a folder structure for the project.
- Designed and implemented PySpark jobs for each analysis goal, utilizing RDDs or DataFrames as appropriate.
- Developed PySpark transformations and actions for data processing and analysis.

- using `spark-submit` command or interactive shell and monitored the output.

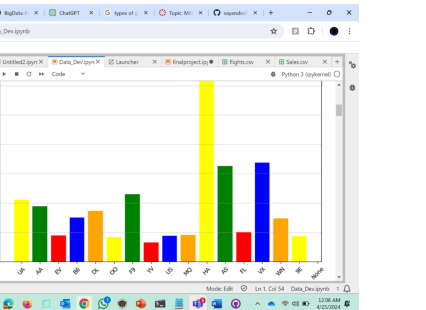
ation:

- generated by the PySpark jobs to evaluate our goal, utilizing PySpark DataFrame operations and compared them against industry research on airline fares.

average fare for each airline



- pycharm | ide Mode: Edit Ln 1, Col 54 Data_Descr.pyb 1



- Mode: Edit Ln 1, Col 54 Data_Decrypt.py 12:00 AM 4/25/2024

Goal 2: Identify the Cheapest Airlines and Fares

- Source Code:

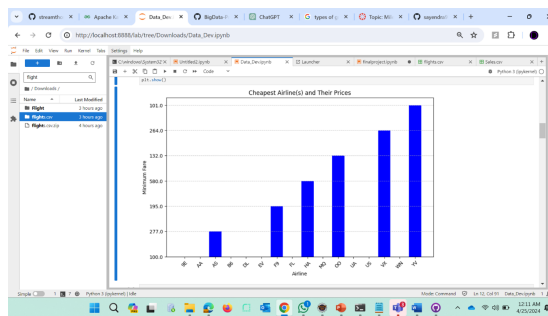
```

1 # Importing libraries
2 import pandas as pd
3 import numpy as np
4
5 # Importing data
6 df = pd.read_csv('airline_data.csv')
7
8 # Grouping data by airline and finding the minimum fare
9 df_grouped = df.groupby('airline').min('fare')
10
11 # Sorting the data by minimum fare
12 df_grouped = df_grouped.sort_values('fare')
13
14 # Displaying the result
15 df_grouped

```

Identifying the Cheapest Airlines and Fares

- Results:



Identifying the Cheapest Airlines and Fares

Goal 3: Find Minimum Fare for Eight Routes and Convert 'min_fare' Column

- Source Code:

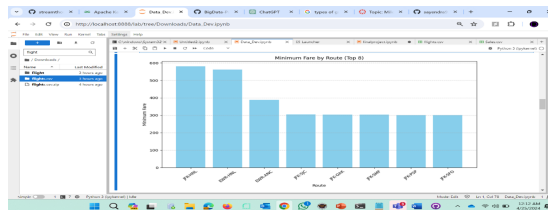
```

1 # Importing libraries
2 import pandas as pd
3 import numpy as np
4
5 # Importing data
6 df = pd.read_csv('airline_data.csv')
7
8 # Grouping data by route and finding the minimum fare
9 df_grouped = df.groupby('route').min('fare')
10
11 # Sorting the data by minimum fare
12 df_grouped = df_grouped.sort_values('fare')
13
14 # Displaying the result
15 df_grouped

```

Finding Minimum Fare for Eight Routes

- Results:



Finding Minimum Fare for Eight Routes

Goal 4: Line chart Trend of ticket purchases over time (Monthly)

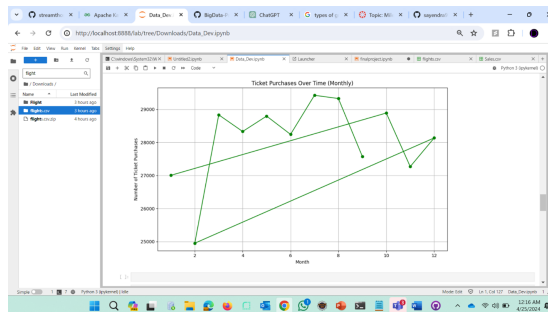
```

1 # Importing the dataset
2 import pandas as pd
3 import numpy as np
4
5 # Importing the dataset
6 df = pd.read_csv('TicketPurchases.csv')
7
8 # Grouping the data by month
9 df['Month'] = df['Date'].dt.month
10
11 # Grouping the data by month and summing the ticket purchases
12 df_grouped = df.groupby('Month').sum()
13
14 # Resetting the index
15 df_grouped.reset_index(inplace=True)
16
17 # Plotting the trend of ticket purchases over time
18 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
19
20 # Saving the plot as a file
21 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
22
23 # Displaying the plot
24 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
25
26 # Saving the plot as a file
27 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
28
29 # Displaying the plot
30 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
31
32 # Saving the plot as a file
33 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
34
35 # Displaying the plot
36 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
37
38 # Saving the plot as a file
39 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
40
41 # Displaying the plot
42 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
43
44 # Saving the plot as a file
45 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
46
47 # Displaying the plot
48 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
49
50 # Saving the plot as a file
51 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
52
53 # Displaying the plot
54 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
55
56 # Saving the plot as a file
57 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
58
59 # Displaying the plot
60 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
61
62 # Saving the plot as a file
63 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
64
65 # Displaying the plot
66 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
67
68 # Saving the plot as a file
69 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
70
71 # Displaying the plot
72 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
73
74 # Saving the plot as a file
75 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
76
77 # Displaying the plot
78 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
79
80 # Saving the plot as a file
81 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
82
83 # Displaying the plot
84 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
85
86 # Saving the plot as a file
87 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
88
89 # Displaying the plot
90 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
91
92 # Saving the plot as a file
93 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
94
95 # Displaying the plot
96 df_grouped.plot(x='Month', y='TicketPurchases', style='line')
97
98 # Saving the plot as a file
99 df_grouped.plot(x='Month', y='TicketPurchases', style='line', savefig='TicketPurchasesTrend.png')
100

```

- Source Code:

Ensure accurate and comprehensive ticket purchase data, including dates and quantities.



- Results:

Validate trends depicted in the line chart against historical data for accuracy and reliability.

Goal 5: Top 5 Trending Airlines

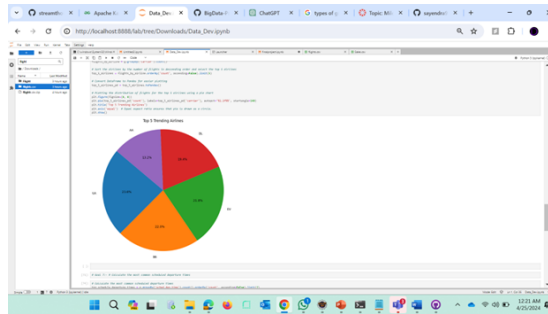
```

1 # Importing the dataset
2 import pandas as pd
3 import numpy as np
4
5 # Importing the dataset
6 df = pd.read_csv('FlightData.csv')
7
8 # Grouping the data by airline
9 df_grouped = df.groupby('Airline').sum()
10
11 # Resetting the index
12 df_grouped.reset_index(inplace=True)
13
14 # Sorting the data by total ticket purchases
15 df_grouped.sort_values(by='TicketPurchases', ascending=False, inplace=True)
16
17 # Selecting the top 5 airlines
18 top_5_airlines = df_grouped.head(5)
19
20 # Plotting the top 5 trending airlines
21 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
22
23 # Saving the plot as a file
24 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
25
26 # Displaying the plot
27 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
28
29 # Saving the plot as a file
30 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
31
32 # Displaying the plot
33 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
34
35 # Saving the plot as a file
36 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
37
38 # Displaying the plot
39 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
40
41 # Saving the plot as a file
42 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
43
44 # Displaying the plot
45 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
46
47 # Saving the plot as a file
48 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
49
50 # Displaying the plot
51 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
52
53 # Saving the plot as a file
54 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
55
56 # Displaying the plot
57 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
58
59 # Saving the plot as a file
60 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
61
62 # Displaying the plot
63 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
64
65 # Saving the plot as a file
66 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
67
68 # Displaying the plot
69 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
70
71 # Saving the plot as a file
72 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
73
74 # Displaying the plot
75 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
76
77 # Saving the plot as a file
78 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
79
80 # Displaying the plot
81 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
82
83 # Saving the plot as a file
84 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
85
86 # Displaying the plot
87 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
88
89 # Saving the plot as a file
90 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
91
92 # Displaying the plot
93 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
94
95 # Saving the plot as a file
96 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar', savefig='Top5TrendingAirlines.png')
97
98 # Displaying the plot
99 top_5_airlines.plot(x='Airline', y='TicketPurchases', style='bar')
100

```

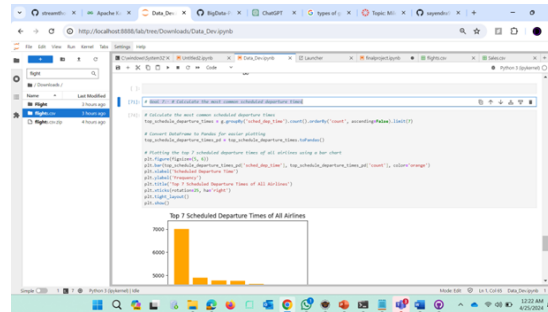
- Source Code:

Ensure accurate airline performance data.

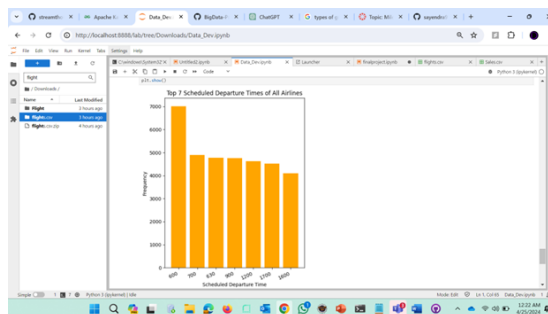


- **Results:** Finding top 5 trending airlines in an efficient manner.

Goal 6: Calculate the Most Common Scheduled Departure Times



- **Source Code:** Ensure accurate and complete departure time data.



- **Results:**

Conclusion

Through a rigorous analysis of airline fare data, several key insights have emerged, offering valuable guidance for strategic decision-making within the aviation sector. By calculating the average fare for each airline and identifying the cheapest fares and airlines, both travelers and industry stakeholders gain invaluable in-

sights into pricing dynamics and competitive positioning. Moreover, the examination of minimum fares by route allows airlines to fine-tune pricing strategies, optimizing revenue generation and meeting consumer demand effectively. Tracking trends in ticket purchases on both daily and monthly scales reveals patterns in travel demand, facilitating resource allocation, scheduling optimization, and targeted marketing initiatives. Additionally, recognizing the top trending airlines and understanding common scheduled departure times empower airlines to enhance service offerings, expand routes, and streamline operations to meet customer needs and maintain competitiveness in the market.

Citations

4.1 Kaggle Dataset: <https://www.kaggle.com/datasets/mahoora00135/flights>

4.2 GitHub Repository: https://github.com/sayendra99/Big_data_Project.git