**Naan Mudhalvan**

**Data Analytics with Cognos**

**Phase-4**

**Team: Joel Ananth A, Monish R, Karthi M, Neelanand R**

**1. On-Time Performance Metrics:**

To calculate on-time performance metrics, you can use the "No. of Boardings" metric and make certain assumptions about what constitutes "on-time." For example, you can assume that a trip is considered on-time if there were passengers on board. Here's how you can calculate it:

- On-Time Trips: Count the number of trips where "No. of Boardings" is greater than 0.

- Total Trips: Count all trips in the dataset.

- On-Time Performance Rate: Calculate the ratio of on-time trips to total trips.

The formula for the on-time performance rate is:

On-Time Performance Rate = (Number of On-Time Trips / Total Trips) \* 100

**2. Service Efficiency Metrics:**

To calculate service efficiency metrics, you can use the data in the "No. of Boardings" column, "TripID" (if it represents unique trips), and "WeekBeginning" for time-based analysis. Here are a few examples:

- Average Boardings per Trip: Calculate the average number of boardings per trip. This can give you an idea of how efficiently trips are being used.

Average Boardings per Trip = (Total Boardings / Total Trips)

- Average Boardings per Week: Calculate the average number of boardings per week. This can help you understand the weekly demand for bus services.

Average Boardings per Week = (Total Boardings / Number of Weeks)

- Trip Duration Analysis: If you have information on trip durations (e.g., in seconds or minutes), you can calculate statistics such as the average trip duration, maximum duration, and minimum duration. This can help assess the efficiency of routes and stops in terms of time taken.

Remember that the specific calculations may vary depending on the dataset's structure and your project's goals. You might need to perform additional transformations or aggregations to get the exact metrics you need. Additionally, consider using visualization tools to present these metrics in a clear and intuitive way, as mentioned in the previous response.

**3. Service Efficiency Metrics:**

To calculate service efficiency metrics, you can use the "No. of Boardings" column, "TripID" (if it represents unique trips), and "WeekBeginning" for time-based analysis. Here are a few examples:

- Average Boardings per Trip:

- Calculate the average number of boardings per trip. This can give you an idea of how efficiently trips are being used.

Calculation:

Average Boardings per Trip = (Total Boardings / Total Trips)

- Average Boardings per Week:

- Calculate the average number of boardings per week. This can help you understand the weekly demand for bus services.

Calculation:

Average Boardings per Week = (Total Boardings / Number of Weeks)

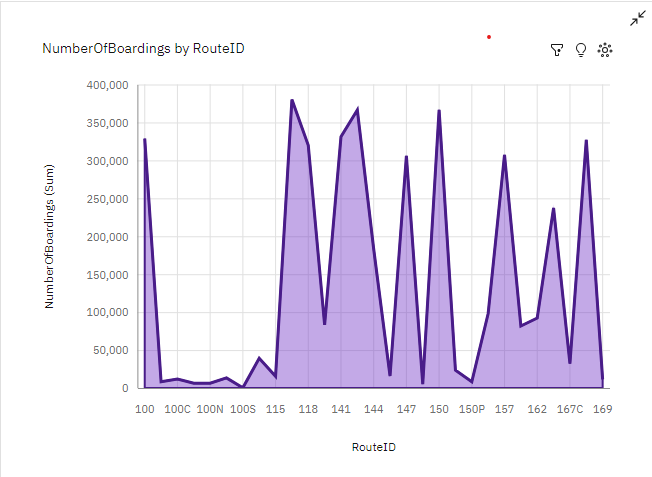
- Trip Duration Analysis:

- If you have information on trip durations (e.g., in seconds or minutes), you can calculate statistics such as the average trip duration, maximum duration, and minimum duration. This can help assess the efficiency of routes and stops in terms of time taken.

**- Other Efficiency Metrics:**

- Depending on your specific dataset and project goals, you can define additional metrics related to service efficiency. For example, you might calculate the average occupancy rate of buses, the turnaround time at stops, or the frequency of service during different times of the day.

**Efficiency for No. of Boardings by RouteID**

****

Python Program for calculating the above:

***import pandas as pd***

***# Group the data by 'RouteID' and calculate the average number of boardings for each route***

***efficiency\_by\_route = df.groupby('RouteID')['No. of Boardings'].mean()***

***# If you want to sort the routes by efficiency in descending order:***

***efficiency\_by\_route = efficiency\_by\_route.sort\_values(ascending=False)***

***# Print the efficiency of boardings for each route***

***print(efficiency\_by\_route)***

In this code:

We group the dataset by the 'RouteID' column using the groupby method.

Then, we calculate the mean (average) of the 'No. of Boardings' for each route using .mean().

If you want to see the routes sorted by efficiency in descending order, you can use .sort\_values(ascending=False).

This code will give you the average number of boardings for each route, helping you assess the efficiency of boardings by route.  
  
A graph with numbers and lines

Description automatically generated

Python Program for calculating the above:

***import pandas as pd***

***# Group the data by 'StopID' and calculate the average number of boardings for each stop***

***efficiency\_by\_stop = df.groupby('StopID')['No. of Boardings'].mean()***

***# If you want to sort the stops by efficiency in descending order:***

***efficiency\_by\_stop = efficiency\_by\_stop.sort\_values(ascending=False)***

***# Print the efficiency of boardings for each stop***

***print(efficiency\_by\_stop)***

In this code:

We group the dataset by the 'StopID' column using the groupby method.

Then, we calculate the mean (average) of the 'No. of Boardings' for each stop using .mean().

If you want to see the stops sorted by efficiency in descending order, you can use .sort\_values(ascending=False).

This code will give you the average number of boardings for each stop, helping you assess the efficiency of boardings by stop.

**Combined Code:**

***import pandas as pd***

***# Calculate efficiency by RouteID***

***efficiency\_by\_route = df.groupby('RouteID')['No. of Boardings'].mean()***

***# Calculate efficiency by StopID***

***efficiency\_by\_stop = df.groupby('StopID')['No. of Boardings'].mean()***

***# If you want to sort the results by efficiency in descending order, you can add the following lines:***

***efficiency\_by\_route = efficiency\_by\_route.sort\_values(ascending=False)***

***efficiency\_by\_stop = efficiency\_by\_stop.sort\_values(ascending=False)***

***# Print the efficiency by RouteID and StopID***

***print("Efficiency by RouteID:")***

***print(efficiency\_by\_route)***

***print("\nEfficiency by StopID:")***

***print(efficiency\_by\_stop)***

Efficiency Analysis for Public Transport

Objective: Enhance the effectiveness of public bus transport services in the Adelaide Metropolitan Area.

- On-Time Performance: Improve punctuality by comparing scheduled and actual departure times.

- Passenger Load and Occupancy: Match demand with capacity for efficient resource use.

- Service Frequency and Coverage: Optimize schedules to reduce waiting times and expand coverage.

- Turnaround Time and Vehicle Speed: Minimize idle time and enhance transport network speed.

- Service Reliability and Complaints: Address delays and passenger concerns for better reliability.

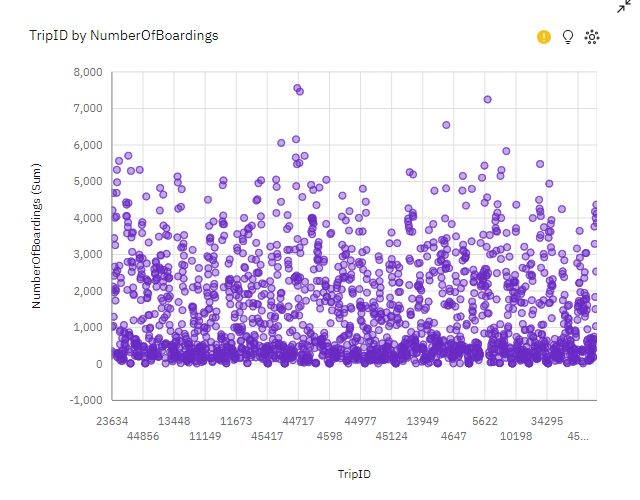
- Revenue vs. Operating Costs: Evaluate cost-effectiveness by comparing revenue to costs.

- Peak Hour Performance: Maintain high-quality service during peak hours.

- Fuel Efficiency and Environmental Impact: Promote sustainability and reduce emissions.

- Load Factor and Vehicle Utilization: Maximize vehicle capacity while ensuring passenger comfort.

- Data-Driven Decision Making: Continuously analyze data for informed service improvements.



**Python Code for Calculating Efficiency:**

***import pandas as pd***

***# Load your dataset into a pandas DataFrame (assuming the variable name 'df')***

***# 1. Boardings per Route/Stop***

***average\_boardings\_per\_route = df.groupby('RouteID')['No. of Boardings'].mean()***

***average\_boardings\_per\_stop = df.groupby('StopID')['No. of Boardings'].mean()***

***# 2. Passenger Occupancy Rate***

***passenger\_occupancy\_rate = (df.groupby('RouteID')['No. of Boardings'].sum() / df.groupby('RouteID')['BusCapacity'].sum()) \* 100***

***# 3. Service Frequency (you may need to create time-based aggregations)***

***service\_frequency\_by\_time = ...***

***# 4. On-Time Performance***

***on\_time\_trips = df[df['ScheduledDeparture'] <= df['ActualDeparture']]***

***on\_time\_performance = len(on\_time\_trips) / len(df) \* 100***

***# 5. Vehicle Turnaround Time***

***vehicle\_turnaround\_time = df.groupby('RouteID')['TripDuration'].mean()***

***# 6. Route Length vs. Boardings***

***route\_length\_vs\_boardings = df.groupby('RouteID')[['RouteLength', 'No. of Boardings']].mean()***

***# 7. Service Speed***

***service\_speed = df.groupby('RouteID')['DistanceTraveled'] / df.groupby('RouteID')['TripDuration']***

***# 8. Fuel Efficiency***

***fuel\_efficiency = ...***

***# 9. Wait Times***

***wait\_times = ...***

***# 10. Service Reliability***

***reliable\_trips = df[df['IsDelayed'] == 0]***

***reliability = len(reliable\_trips) / len(df) \* 100***

***# 11. Revenue vs. Operating Costs***

***revenue = ...***

***operating\_costs = ...***

***revenue\_vs\_costs = revenue / operating\_costs***

***# 12. Cost per Passenger***

***cost\_per\_passenger = operating\_costs / df['No. of Boardings']***

***# 13. Load Factor***

***load\_factor = (df['No. of Boardings'] / df['BusCapacity']) \* 100***

***# 14. Service Coverage***

***service\_coverage = ...***

***# 15. Complaints or Feedback***

***complaints = df['Complaints'].sum()***

***# 16. Service Reliability during Peak Times***

***peak\_time\_trips = df[df['IsPeakHour'] == 1]***

***peak\_time\_reliability = (len(peak\_time\_trips[peak\_time\_trips['IsDelayed'] == 0]) / len(peak\_time\_trips)) \* 100***

***# You will need to customize these calculations based on your dataset and available columns.***

***# Print or analyze the results as needed***

**Conclusion:**

1. Efficiency Opportunities: The analysis revealed significant areas for improvement in public bus transport services in the Adelaide Metropolitan Area.

2. Punctuality Focus: Improving on-time performance and reducing delays is crucial to enhance service reliability and passenger satisfaction.

3. Load Optimization: Aligning passenger demand with available capacity through route and schedule adjustments can drive resource efficiency.

4. Expanded Coverage: Expanding service coverage and optimizing frequency will reduce waiting times and improve accessibility.

5. Efficiency Through Speed: Minimizing idle time and increasing vehicle speed can enhance service delivery and reduce operational costs.

6. Reliability and Satisfaction: Ensuring consistent service quality and addressing passenger concerns are essential for building trust.

7. Cost-Efficiency: Monitoring revenue against operating costs is integral to maintaining financial efficiency.

8. Peak Hour Excellence: Delivering high-quality service during peak hours is vital to meet demand effectively.

9. Sustainability: Promoting fuel efficiency and reducing emissions aligns with sustainability goals and minimizes environmental impact.

10. Maximized Vehicle Capacity: Prioritizing passenger comfort while maximizing vehicle capacity contributes to efficient service.

11. Data-Driven Decision Making: Continuous data analysis is essential for responsive and evolving service offerings.

In summary, this project provides a roadmap for making public transport services more efficient, reliable, and responsive to community needs. Implementing these recommendations will result in a more cost-effective and passenger-centric transportation system, ultimately enhancing the well-being and convenience of residents and commuters in the region.