# PUBLIC TRANSPORTATION EFFIENCY ANALYSIS

| | |
|---|---|
| TEAM LEADER / REGISTER NO | **SHUSHIL SP/410121104045** |
| DEPT | **COMPUTER SCIENCE** |
| DOMAIN | **DATA ANALYTICS** |
| COLLEGE CODE | **4101** |
| NM I'd | **au410121104045** |

**Analysis Objectives**:

1. **Improving On-Time Performance**: Ensure that public transportation services operate on time and identify areas with consistently poor punctuality.

2. **Enhancing Passenger Satisfaction**: Understand passenger feedback to address issues and improve the overall quality of service.

3. **Optimizing Service Efficiency**: Increase the efficiency of public transportation services by monitoring key operational metrics.

**Design Thinking Process**:

Design thinking is an iterative process that helps solve complex problems and create innovative solutions. In the context of public transportation efficiency analysis, you can follow these steps:

1. **Empathize**:

- Understand the needs and pain points of both passengers and transportation operators.

- Identify key metrics for efficiency, on-time performance, and passenger satisfaction.

2. **Define**:

- Clearly define the project's objectives and scope.

- Determine the specific metrics and KPIs that will be used to measure efficiency, on-time performance, and passenger satisfaction.

3. **Ideate**:

- Brainstorm potential solutions and data sources.

- Consider innovative ways to visualize and analyze the data.

4. **Prototype**:

- Create mock-ups or wireframes of the IBM Cognos dashboards and reports you plan to build.

- Use these prototypes to gather feedback from stakeholders.

5. **Test and Iterate**:

- Develop a minimum viable product (MVP) of the dashboards and reports.

- Gather feedback from users and stakeholders, and make iterative improvements.

**Development Phases**:

The development of the project can be broken down into several phases:

1. **Data Collection Process**:

- Collect data from various sources, which may include:

  - GPS tracking systems on vehicles for on-time performance data.

  - Passenger surveys or feedback forms.

  - Operational data, including maintenance records and schedules.

- Implement data extraction, transformation, and loading (ETL) processes to consolidate and clean the data.

2. **Data Storage and Management**:

- Store the cleaned and transformed data in a suitable data repository, such as a relational database or a data warehouse.

- Ensure data is regularly updated and maintained.

3. **Data Visualization Using IBM Cognos**:

- Create a data model in IBM Cognos Framework Manager to support reporting and dashboard creation.

- Design and develop reports and dashboards using IBM Cognos Report Studio and Dashboard.

- Visualize on-time performance with line charts or heat maps, passenger feedback using word clouds and tables, and service efficiency with bar charts or KPIs.

- Implement interactivity, such as filtering and drill-through options, for enhanced user experience.

4. **Code Integration**:

- If necessary, integrate custom code or scripts to enhance data processing or create advanced visualizations.

- Ensure that the code is well-documented and maintainable.

5. **Testing and Optimization**:

- Thoroughly test the dashboards, reports, and data processes for accuracy, performance, and usability.

- Optimize the performance of data queries and dashboard rendering.

6. **Deployment and User Training**:

- Deploy the dashboards and reports to the appropriate platform for users to access.

- Provide training to end users and administrators on how to utilize the tools effectively.

7. **Feedback and Continuous Improvement**:

- Continuously collect feedback from users and stakeholders to identify areas for improvement and make iterative enhancements to the project.

## CODE :

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset (replace 'your_dataset.csv' with your actual dataset file)
data = pd.read_csv('your_dataset.csv')

# Data Preprocessing
data['Date'] = pd.to_datetime(data['Date'])
data['ScheduledArrival'] = pd.to_datetime(data['ScheduledArrival'])
data['ActualArrival'] = pd.to_datetime(data['ActualArrival'])

# Calculate delay
data['Delay'] = (data['ActualArrival'] - data['ScheduledArrival']).dt.total_seconds()
/ 60

# Calculate key performance indicators
avg_delay = data['Delay'].mean()
on_time_percentage = (data['Delay'] <= 0).mean()
average_passenger_load = data['PassengerCount'].mean()
```

```
# Data Visualization
plt.figure(figsize=(12, 6))
sns.histplot(data['Delay'], bins=20, kde=True)
plt.title('Distribution of Delays')
plt.xlabel('Delay (minutes)')
plt.ylabel('Frequency')
plt.show()

# Print analysis results
print(f'Average Delay: {avg_delay:.2f} minutes')
print(f'On-Time Percentage: {on_time_percentage:.2%}')
print(f'Average Passenger Load: {average_passenger_load:.2f} passengers')

# Route optimization analysis
# You can add code here to optimize transportation routes, e.g., using network analysis libraries like NetworkX.

# Advanced visualization
# Create visualizations that provide insights into route efficiency, passenger distribution, etc.

# Machine learning models
# Implement ML models to predict delays or optimize schedules based on historical data.
```

# Export results

# Save the analysis results and visualizations to files or databases for reporting.

# Error handling, data cleaning, and other project-specific tasks should be included as well.

SAMPLE OUTPUT:

Average Delay: 3.50 minutes

On-Time Percentage: 75.00%

Average Passenger Load: 50.25 passengers