**Enhancing Public Transportation through Data Analysis**

**PHASE 3**

**DEVELOPMENT(Part1)**

* **INTRODUCTION :**

The project aims to assess the service efficiency, on-time performance, and passenger feedback of public transportation systems. It begins by defining the objectives of the analysis and collecting transportation data from a provided source, which is a dataset available on Kaggle. The collected data is then processed and cleaned to ensure its quality and accuracy. The next step involves utilizing IBM Cognos, a business intelligence, and data visualization tool, to create insightful visualizations. These visualizations will help explore and analyse the dataset, ultimately supporting transportation improvement initiatives and enhancing the overall public transportation experience.

* **Install Libraries:**

You may need to install some Python libraries. Use the following commands if you have not already:

pip install pandas

* **NOTE:**
* Additionally, make sure the dataset file is accessible or downloaded to your local environment, and provide the appropriate path in the read csv function. Additionally, make sure the dataset file is accessible or downloaded to your local environment, and provide the appropriate path in the read csv function.
* **ACCESS THE DATASET:**

import pandas as pd

# Load the dataset

df = pd.read\_csv('https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV')

# Perform data analysis and preprocessing as required

# For example, you can check the first few rows of the dataset

print(df.head())

# You can also perform other operations like statistical analysis

# For example, calculate the average on-time performance

avg\_on\_time = df['On-time Performance'].mean()

print('Average On-time Performance:', avg\_on\_time)

# Further analyze the dataset, perform route optimization, and passenger feedback analysis

# Generate insights and recommendations based on the analysis

# Visualize the data and insights using matplotlib or other visualization libraries

# Save the generated insights and reports or present them as needed

* **PREPROCESS THE DATASET:**

import pandas as pd

# Load the dataset

df = pd.read\_csv('https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV')

# Explore the dataset

print(df.head()) # Check the first few rows of the dataset

# Perform data preprocessing

# Clean the data, handle missing values, and ensure data quality and accuracy

# Drop irrelevant columns

df = df.drop(['Column1', 'Column2'], axis=1)

# Handle missing values, if any

df = df.dropna()

# Convert data types if necessary

df['Date'] = pd.to\_datetime(df['Date'])

# Check the cleaned dataset

print(df.head())

# Further preprocessing steps as per your analysis objectives

# Save the cleaned dataset

df.to\_csv('cleaned\_dataset.csv', index=False)

* **PYTHON PROGRAM:**

import pandas as pd

import matplotlib.pyplot as plt

# Load transportation data (example: CSV file)

transport\_data = pd.read\_csv('transport\_data.csv')

# Assuming 'ScheduledTime' and 'ActualTime' columns in the dataset

# Convert time columns to datetime objects for analysis

transport\_data['ScheduledTime'] = pd.to\_datetime(transport\_data['ScheduledTime'])

transport\_data['ActualTime'] = pd.to\_datetime(transport\_data['ActualTime'])

# Calculate the time difference between scheduled and actual times

transport\_data['TimeDifference'] = (transport\_data['ActualTime'] - transport\_data['ScheduledTime']).dt.total\_seconds()

# Calculate the percentage of on-time arrivals

on\_time\_threshold = 300 # Assuming 5 minutes (300 seconds) delay is considered on-time

on\_time\_percentage = (transport\_data['TimeDifference'] <= on\_time\_threshold).mean() \* 100

# Visualize on-time performance

plt.figure(figsize=(8, 6))

plt.hist(transport\_data['TimeDifference'], bins=30, color='skyblue', edgecolor='black')

plt.axvline(x=on\_time\_threshold, color='red', linestyle='--', label=f'On-Time Threshold ({on\_time\_threshold} sec)')

plt.xlabel('Time Difference (seconds)')

plt.ylabel('Frequency')

plt.title('Distribution of Arrival Time Differences')

plt.legend()

plt.show()

# Print on-time performance percentage

print(f'Percentage of on-time arrivals: {on\_time\_percentage:.2f}%')

plt.xlabel('Time Difference (seconds)')

plt.ylabel('Frequency')

plt.title('Distribution of Arrival Time Differences')

plt.legend()

plt.show()

# Print on-time performance percentage

print(f'Percentage of on-time arrivals: {on\_time\_percentage:.2f}%')

* **DATA VISUALIZATIUON:**

The provided code loads the transportation data from a CSV file and performs analysis on the arrival time differences. It assumes the presence of 'ScheduledTime' and 'ActualTime' columns in the dataset, which are then converted to datetime objects for analysis. The code calculates the time difference between the scheduled and actual times of arrival and determines the percentage of on-time arrivals based on a threshold of 300 seconds (5 minutes). It visualizes the distribution of arrival time differences using a histogram and adds a vertical line to represent the on-time threshold. Finally, it prints the percentage of on-time arrivals as an output. This code provides insights into the efficiency and punctuality of the transportation service.

* **CONCLUSION:**

In conclusion, the initial phase of the project has successfully established a strong foundation for further analysis and reporting. By defining the problem statement and adopting a design thinking approach, we have set clear objectives and strategies for data gathering and visualization. This structured approach ensures that our project will generate actionable insights to improve the public transportation experience. With a well-planned data collection process and the utilization of visualization techniques, we are well-prepared to move forward to the subsequent phases of the project. By following this organized and systematic approach, we aim to provide valuable recommendations and solutions to enhance the efficiency, on-time performance, and overall satisfaction of public transportation systems.