Public Transportation Analysis
Phase-3

Data Preprocessing :

Data preprocessing in public transportation analysis is a crucial step that involves cleaning, transforming, and organizing raw transportation data to make it suitable for analysis. Public transportation systems generate vast amounts of data from various sources, such as ticketing systems, GPS trackers, sensors, and schedules. Preprocessing this data is necessary to extract meaningful insights, improve data quality, and ensure that it's ready for analytical and modeling tasks. Here are the key aspects of data preprocessing in public transportation analysis:

1. Data Collection:
   - Data collection involves gathering information from various sources, such as fare collection systems, vehicle sensors, passenger counts, and scheduling systems. This raw data can be in different formats and structures.

2. Data Cleaning:
   - Data cleaning is the process of identifying and correcting errors, inconsistencies, and missing values in the dataset. This can include dealing with duplicated records, removing outliers, and addressing data entry errors.

3. Data Integration:
   - Public transportation data often comes from different sources and in various formats. Data integration involves merging, aligning, and transforming data so that it can be analyzed as a cohesive dataset.

4. Data Transformation:
   - Transformation tasks may include converting data into a standardized format, resampling temporal data, and aggregating data to different time intervals (e.g., hourly or daily) to align with analysis requirements.

5. Geospatial Data Processing:
   - Public transportation analysis often involves geospatial data, including GPS coordinates, routes, and geographical boundaries. Preprocessing may involve geocoding, spatial indexing, and the calculation of distances or travel times between locations.

The code used :

```
#!/usr/bin/env python
# coding: utf-8

# In[2]:


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


# In[3]:


#Loading the dataset
data =
pd.read_csv("C:\\Users\\Maha\\Downloads\\Dataset\\PublicTransportDataset.CSV", low_memory=False)
```

In[4]:

#Displaying the first 20 rows
data.head(20)

In[5]:

# Dropping records which have duplicate values
data.drop_duplicates(inplace=True)

In[6]:

# Filling missing values with mean
data.fillna(data.mean(), inplace=True)

In[7]:

# Printing the first few rows
print(data.head())

In[8]:

# Generating descriptive statistics of the dataset
print(data.describe())

In[9]:

Generating concise summary of the dataset

```
print(data.info())


# In[11]:


# Shape of the dataset
print(data.shape)


# In[12]:


# Displaying first few rows after preprocessing
data.head()
```

In [2]:
```
#Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
#Loading the dataset
data = pd.read_csv("C:\\Users\\AbiramiSV\\Downloads\\Dataset\\PublicT
```

In [4]:
```
#Displaying the first 20 rows
data.head(20)
```

Out[4]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
0	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
1	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1
2	23632	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
3	23633	100	12266	Zone A Arndale Interchange	2013-06-30 00:00:00	2
4	23633	100	14147	178 Cross Rd	2013-06-30 00:00:00	1
5	23634	100	13907	9A Marion Rd	2013-06-30 00:00:00	1
6	23634	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
7	23634	100	13335	9A Holbrooks Rd	2013-06-30 00:00:00	1
8	23634	100	13875	9 Marion Rd	2013-06-30 00:00:00	1
9	23634	100	13045	206 Holbrooks Rd	2013-06-30 00:00:00	1
10	23635	100	13335	9A Holbrooks Rd	2013-06-30 00:00:00	1
11	23635	100	13383	8A Marion Rd	2013-06-30 00:00:00	1
12	23635	100	13586	8D Marion Rd	2013-06-30 00:00:00	2
13	23635	100	12726	23 Findon Rd	2013-06-30 00:00:00	1
14	23635	100	13813	8K Marion Rd	2013-06-30 00:00:00	1
15	23635	100	14062	20 Cross Rd	2013-06-30 00:00:00	1
16	23636	100	12780	22A Crittenden Rd	2013-06-30 00:00:00	1
17	23636	100	13383	8A Marion Rd	2013-06-30 00:00:00	1
18	23636	100	14154	180 Cross Rd	2013-06-30 00:00:00	2
19	23636	100	13524	8C Marion Rd	2013-06-30 00:00:00	3

In [5]:
```
# Dropping records which have duplicate values
data.drop_duplicates(inplace=True)
```

In [6]:
```
# Filling missing values with mean
data.fillna(data.mean(), inplace=True)
```

In [7]:
```
# Printing the first few rows
print(data.head())
```

```
   TripID RouteID  StopID                    StopName        WeekBeg
inning  \
0   23631     100   14156                181 Cross Rd  2013-06-30 0
0:00:00
1   23631     100   14144                177 Cross Rd  2013-06-30 0
0:00:00
2   23632     100   14132                175 Cross Rd  2013-06-30 0
0:00:00
3   23633     100   12266  Zone A Arndale Interchange  2013-06-30 0
0:00:00
4   23633     100   14147                178 Cross Rd  2013-06-30 0
0:00:00

   NumberOfBoardings
0                  1
1                  1
2                  1
3                  2
4                  1
```

In [8]:
```
# Generating descriptive statistics of the dataset
print(data.describe())
```

```
             TripID        StopID  NumberOfBoardings
count  1.085723e+07  1.085723e+07       1.085723e+07
mean   2.952100e+04  1.366132e+04       4.743737e+00
std    1.960938e+04  1.971760e+03       9.382286e+00
min    7.900000e+01  1.000100e+04       1.000000e+00
25%    1.191700e+04  1.231100e+04       1.000000e+00
50%    2.747900e+04  1.334600e+04       2.000000e+00
75%    4.885800e+04  1.491600e+04       4.000000e+00
max    6.553500e+04  1.871500e+04       9.770000e+02
```

In [9]:
```
# Generating concise summary of the dataset
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10857234 entries, 0 to 10857233
Data columns (total 6 columns):
 #   Column             Dtype
---  ------             -----
 0   TripID             int64
 1   RouteID            object
 2   StopID             int64
 3   StopName           object
 4   WeekBeginning      object
 5   NumberOfBoardings  int64
dtypes: int64(3), object(3)
memory usage: 579.8+ MB
None
```

In [11]:
```
# Shape of the dataset
print(data.shape)
```

```
(10857234, 6)
```

In [12]:
```
# Displaying first few rows after preprocessing
data.head()
```

Out[12]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
0	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
1	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1
2	23632	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
3	23633	100	12266	Zone A Arndale Interchange	2013-06-30 00:00:00	2
4	23633	100	14147	178 Cross Rd	2013-06-30 00:00:00	1

In [ ]: