**PUBLIC TRANSPORTATION EFFICIENCY ANALYSIS**

**TEAM MEMBER**

**732121104302:NIDHARSHANAA M**

**PHASE-2 DOCUMENT SUBMISSION**

**PROJECT: PUBLIC TRANSPORTATION EFFICIENCY**

****

**INTRODUCTION:**

Public transportation plays a pivotal role in urban and rural areas, offering an alternative to private vehicles and addressing the challenges of congestion, pollution, and limited mobility options. The efficiency of public transportation systems is crucial for their success in serving the needs of communities and promoting sustainable urban development. This introduction will explore the concept of public transportation efficiency and the analysis methods employed to assess and enhance its effectiveness.

Efficient public transportation is a vital component of sustainable urban development. It reduces traffic congestion, lowers greenhouse gas emissions, and promotes equitable access to opportunities for all members of the community. As cities continue to grow, the need for efficient public transportation systems becomes increasingly pressing, not only to ease the burden on infrastructure but also to address pressing issues like climate change.

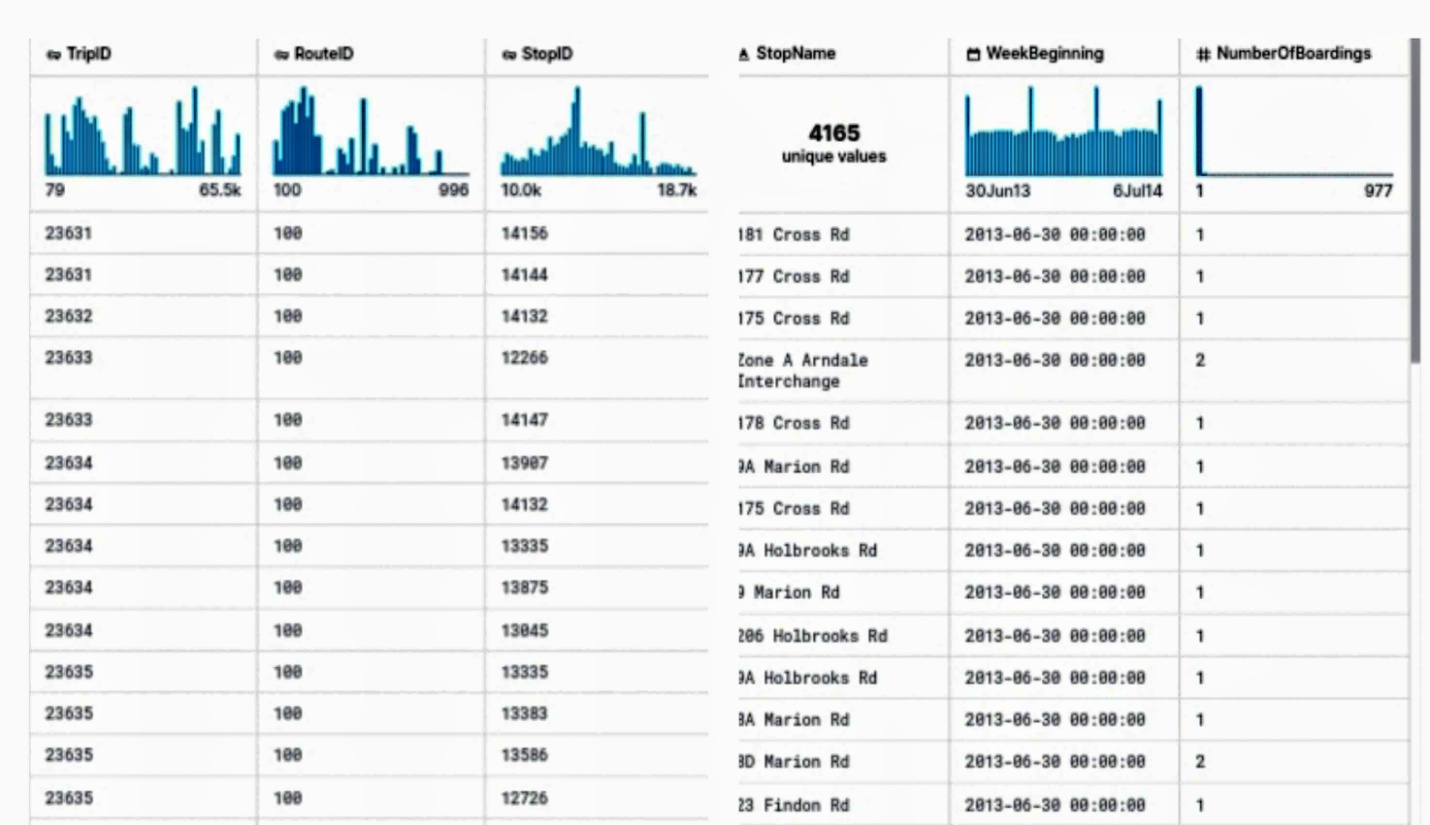
**CONTENT FOR PHASE-2 PROJECT:**

Explain in detail the complete steps that will be taken by you to put your design that you thought of in previous phase into transformation.

**DATA SOURCE:**

**Dataset Link:** **https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV**

We have explored how people are travelling from different stops in Adelaide Metropolitan area and managing the buses on each route according to the no of passenger commuting through the buses.



**Public Transportation Efficiency:**

Public transportation efficiency refers to the ability of a transit system to provide reliable, convenient, and cost-effective services that meet the transportation needs of a community. Efficient public transportation systems should accomplish the following objectives:

**Accessibility:** Provide access to key destinations such as employment centers, educational institutions, healthcare facilities, and recreational areas.

**Reliability:** Offer reliable and consistent services that minimize waiting times and delays for passengers.

**Affordability:** Maintain reasonable fares that are accessible to a broad range of residents, ensuring that public transportation remains an attractive option.

**Safety and Comfort:** Ensure the safety and comfort of passengers, addressing concerns related to security, cleanliness, and passenger experience.

**Environmental Sustainability:** Minimize the environmental impact of transit operations, including reducing emissions and resource consumption

**PROGRAM:**

matplotlib inline

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import datetime

import os

from math import sqrt

import warnings

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

warnings.filterwarnings('ignore')

data = pd.read\_csv('../input/unisys/ptsboardingsummary/20140711.CSV')

data.shape

data.head(10)

**Output :** ( 10857234,6)



out\_geo = pd.read\_csv('../input/outgeo/output\_geo.csv')

out\_geo.shape

out\_geo.head()

**Output:**

(4165, 10)



**EXTERNAL FEATURES:**

from math import sin, cos, sqrt, atan2, radians

def calc\_dist(lat1,lon1):

R = 6373.0

dlon = radians(138.604801) - radians(lon1)

dlat = radians(-34.921247) - radians(lat1)

a = sin (dlat / 2)\*2 + cos(radians(lat1)) \*cos(radians(-34.921247)) \* sin(dlon / 2)\*2

c = 2 \* atan2(sqrt(a), sqrt(1 - a))

return R \* c

out\_geo['dist\_from\_centre']=out\_geo[['latitude','longitude']].apply(lambda x: calc\_dist(\*x), axis=1)

out\_geo.head()

**Output:**



out\_geo['type'].fillna('street\_address',inplace=True)

out\_geo['type'] = out\_geo['type'].apply(lambda x: str(x).split(',')[-1])

out\_geo['type'].unique()

**Output:**

array(['street\_address', 'transit\_station', 'premise', 'political',

'school', 'route', 'intersection', 'point\_of\_interest',

'subpremise', 'real\_estate\_agency', 'university', 'travel\_agency',

'restaurant', 'supermarket', 'store', 'post\_office'], dtype=object)

**Input:**

data['WeekBeginning'] = pd.to\_datetime(data['WeekBeginning']).dt.date

data['WeekBeginning'][1]

**Output:**

datetime.date(2013, 6, 30)

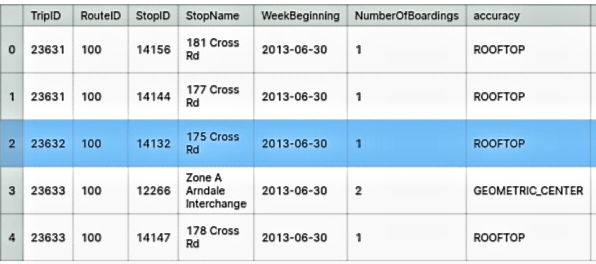
**DATA AGGREGATION:**

data= pd.merge(data,out\_geo,how='left',left\_on = 'StopName',right\_on = 'input\_string')

data.head(5)

data.shape

**Output:**



(10857234, 17)

**Input:**

col=['TripID','RouteID','StopID','StopName','WeekBeginning','NumberOfBoardings', 'latitude', 'longitude','postcode','type','dist\_from\_centre']

data = data[col]

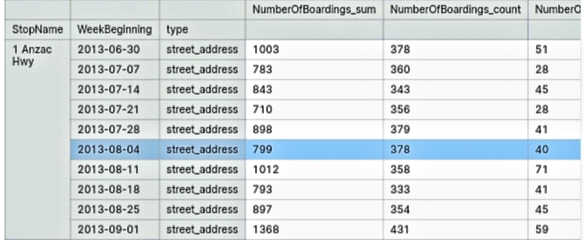
grouped = data.groupby(['StopName','WeekBeginning','type'])

grouped.columns = ["\_".join(x) for x in grouped.columns.ravel()]

grouped.head(10)

grouped.columns

**Output:**



Index(['NumberOfBoardings\_sum', 'NumberOfBoardings\_count','NumberOfBoardings\_max'],dtype='object')

**Input:**

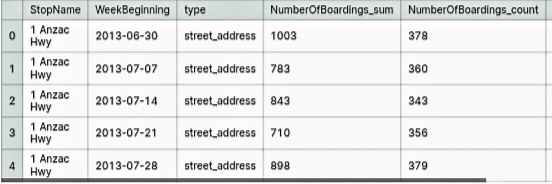
st\_week\_grp = pd.DataFrame(grouped).reset\_index()

st\_week\_grp.shape

st\_week\_grp.head()

**Output:**

(207864, 6)

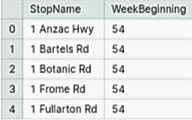


**Input:**

st\_week\_grp1 = pd.DataFrame(st\_week\_grp.groupby('StopName')["WeekBeginning"].count()).reset\_index()

st\_week\_grp1.head()

**Output:**

****

**Input:**

aa = list(st\_week\_grp1[st\_week\_grp1['WeekBeginning'] == 54]['StopName'])

aa[1:10]

**Output:**

['1 Bartels Rd',

'1 Botanic Rd',

'1 Frome Rd',

'1 Fullarton Rd',

'1 George St',

'1 Glen Osmond Rd',

'1 Goodwood Rd',

'1 Henley Beach Rd',

'1 Kensington Rd']

**Input:**

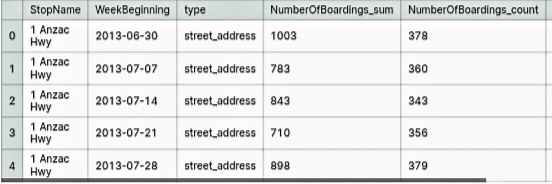
bb = st\_week\_grp[st\_week\_grp['StopName'].isin(aa)]

bb.head()

bb.shape

type(bb)

**Output:**



(175446, 6)

pandas.core.frame.DataFrame

**Input:**

new\_data = data[data['StopName'].isin(aa)]

new\_data.shape

print("data without stopage removing: ", data.shape)

print("data, after removing stoppage not having the data of whole 54 weeks: ", new\_data.shape)

**Output:**

(10567931, 11)

data without stopage removing: (10857234, 11)

data, after removing stoppage not having the data of whole 54 weeks: (10567931, 11)

**Input:**

new\_data.head(2)

filtered\_data = new\_data[new\_data['dist\_from\_centre'] <= 100]

filtered\_data.shape

**Output:**



(10341468, 11)

**Input:**

data = filtered\_data.copy()

data.shape

**Output:**

(10341468, 11)

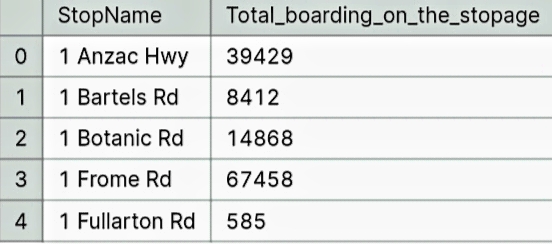
**Input:**

stopageName\_with\_boarding=bb.groupby(['StopName']).agg({'NumberOfBoardings\_sum': ['sum']}

stopageName\_with\_boarding.columns=["StopName","Total\_boarding\_on\_the\_stopage"]

stopageName\_with\_boarding.head()

**Output:**



**DATA EXPLORATION:**

data.nunique()

**Output:**

TripID 39211

RouteID 616

StopID 5838

StopName 3127

WeekBeginning 54

NumberOfBoardings 359

latitude 2393

longitude 2379

postcode 138

type 8

dist\_from\_centre 2397

dtype: int64

**DATA VISUALIZATION:**

fig,axrr=plt.subplots(2,2,figsize=(15,15))

ax=axrr[0][0]

ax.set\_title("No of Boardings")

data['NumberOfBoardings'].value\_counts().sort\_index().head(20).plot.bar(ax=axrr[0][0])

ax=axrr[0][1]

ax.set\_title("WeekBeginning")

data['WeekBeginning'].value\_counts().plot.area(ax=axrr[0][1])

ax=axrr[1][0]

ax.set\_title("most Busiest Route")

data['RouteID'].value\_counts().head(10).plot.bar(ax=axrr[1][0])

ax=axrr[1][1]

ax.set\_title("least Busiest Route")

data['RouteID'].value\_counts().tail(10).plot.bar(ax=axrr[1][1])

**Output:**

Text(0.5,1,'No of Boardings')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff880af0940>

Text(0.5,1,'WeekBeginning')

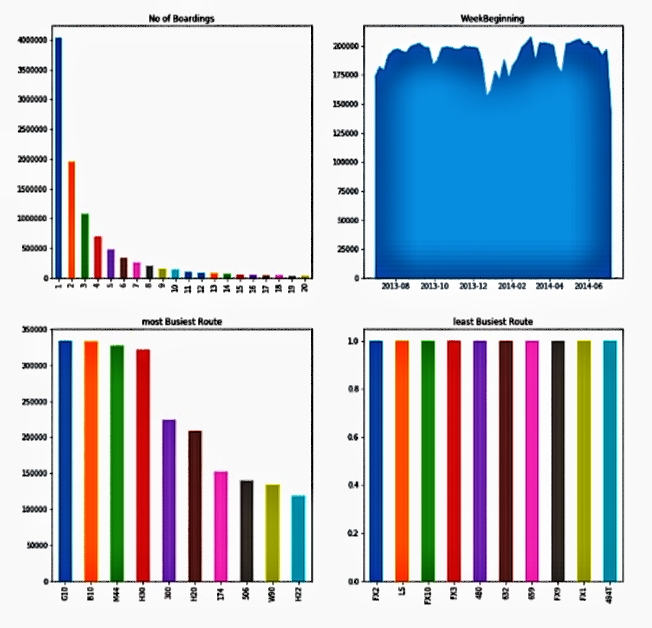
<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff709a6bb38>

Text(0.5,1,'most Busiest Route')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff709a48e10>

Text(0.5,1,'least Busiest Route')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff736bbafd0>

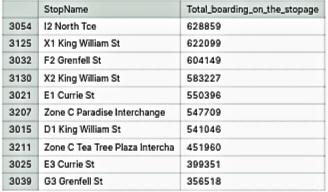


**Input:**

stopageName\_with\_boarding = stopageName\_with\_boarding.sort\_values('Total\_boarding\_on\_the\_stopage', ascending = False)

stopageName\_with\_boarding.head(10)

**Output:**



**Input:**

stopageName\_with\_boarding.tail(10)

**Output:**



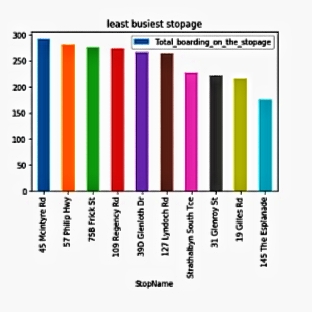
**Input:**

ax=stopageName\_with\_boarding.head(10).plot.bar(x='StopName', y='Total\_boarding\_on\_the\_stopage', rot=90)

ax.set\_title("most busiest stopage")

**Output:**

Text(0.5,1,'most busiest stopage')



**Input:**

data['WeekBeginning'].value\_counts().mean()

**Output:**

191508.66666666666

**Input:**

bb\_grp=data.groupby(['dist\_from\_centre']).agg({'NumberOfBoardings': ['sum']}).reset\_index()

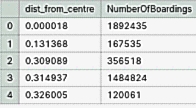
bb\_grp.columns = bb\_grp.columns.get\_level\_values(0)

bb\_grp.head()

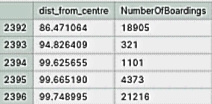
bb\_grp.columns

bb\_grp.tail()

**Output:**



Index(['dist\_from\_centre', 'NumberOfBoardings'], dtype='object')



**Input:**

import plotly.graph\_objs as go

from plotly.offline import iplot

trace0 = go.Scatter(

x = bb\_grp['dist\_from\_centre'],

y = bb\_grp['NumberOfBoardings'],mode = 'lines+markers',name = 'X2 King William St')

data1 = [trace0]

layout = dict(title = 'Distance Vs Number of boarding',

xaxis = dict(title = 'Distance from centre'),

yaxis = dict(title = 'Number of Boardings'))

fig = dict(data=data1, layout=layout)

iplot(fig)

x = data["dist\_from\_centre"]

distance\_10 = []

distance\_10\_50 = []

distance\_50\_100 = []

distance\_100\_more = []

total = 0

outlier = []

outlier\_ = 0

for i in x:

if(i<=10):

distance\_10.append(i)

total += 1

elif(i<=50):

distance\_10\_50.append(i)

total += 1

elif(i<=100):

distance\_50\_100.append(i)

total += 1

print(outlier\_)

0

y = len(distance\_10)+len(distance\_10\_50)+len(distance\_50\_100)

print(total)

print("passangers, boarding the buses in the radious of 10Km from the city center = ", (len(distance\_10)/total)\*100)

print("passanger, boarding the buses from the distance of 10Km to 50Km from the city center = ", (len(distance\_10\_50)/total)\*100)

print("passanger, boarding the buses from the distance of 50Km to 100 from the city center = ", (len(distance\_50\_100)/total)\*100)

print("passanger, boarding the buses from the distance of 100Km and more from the city center = ", (len(distance\_100\_more)/total)\*100)

10341468

passangers, boarding the buses in the radious of 10Km from the city center = 64.31275521038212

passanger, boarding the buses from the distance of 10Km to 50Km from the city center = 33.16731241638035

passanger, boarding the buses from the distance of 50Km to 100 from the city center = 2.5199323732375323

grouped\_route=data.groupby(['RouteID']).agg({'NumberOfBoardings':['sum', 'max']})

grouped\_route.columns=["\_".join(x) for x in grouped\_route.columns.ravel()]

**Output:**

'route\_data=grouped\_route[grouped\_route[\'RouteID\']=="G10"]\nroute\_data.head()'

**CONCLUSION:**

public transportation efficiency and analysis are crucial elements in the development and maintenance of effective transit systems that enhance the quality of life in urban and rural areas. By continuously evaluating and improving public transportation services, communities can enjoy the benefits of reduced traffic congestion, improved air quality, and enhanced accessibility while fostering sustainable and inclusive urban environments.