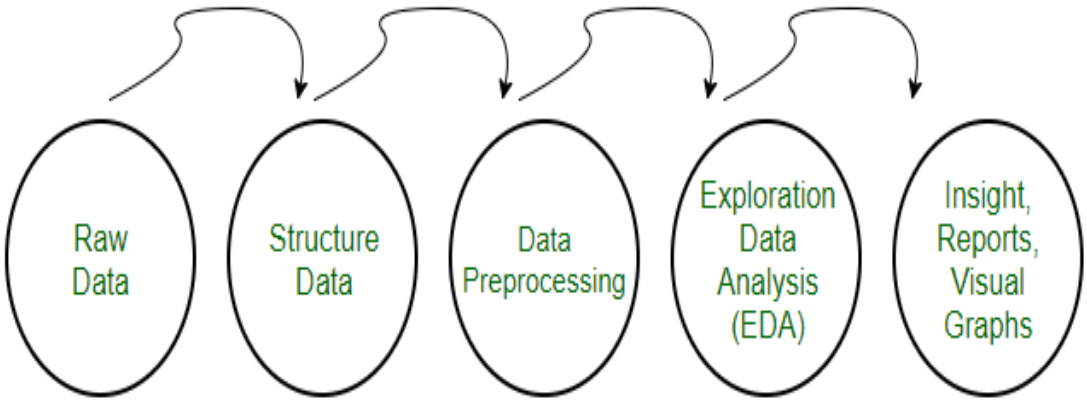
PUBLIC TRANSPORT EFFICENCY ANALYSIS

PHASE-3

DATA PREPROCESS:

- ✓ Data preprocessing is an essential step in preparing your data for analysis. Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.
- ✓ Data preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.



Data Cleaning:

- ✓ Handle missing values: Decide whether to remove rows with missing data or impute missing values using methods like mean, median, or predictive modeling.

Remove duplicates:

- ✓ Identify and remove duplicate rows if they exist in your dataset.

Data Transformation:

- ✓ Encode categorical variables: Convert categorical data into numerical format using techniques like one-hot encoding or label encoding.

Scale numerical features:

- ✓ Normalize or standardize numerical data to ensure consistent scales.

Program:

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

```
%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')
```

In [2]:

```
data = pd.read_csv('../input/unisys/ptsboardingsummary/20140711.CSV')
data.shape
data.head
```

```
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])
```

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

Out[27]:

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

Out[27]:

```
Text(0.5,1,'WeekBeginning')
```

Out[27]:

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

Out[27]:

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

Out[27]:

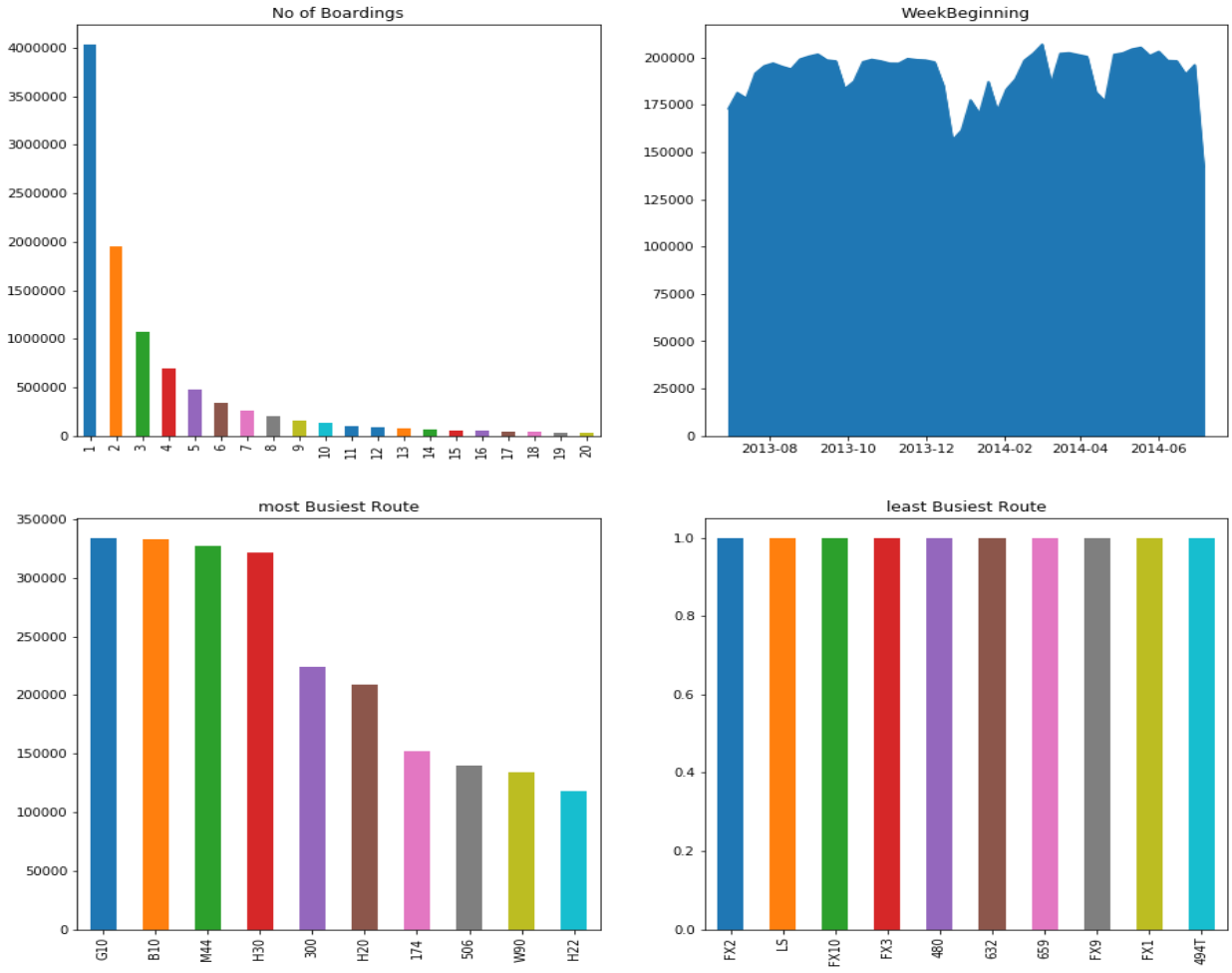
<matplotlib.axes._subplots.AxesSubplot at 0x7ff709a48e10>

Out[27]:

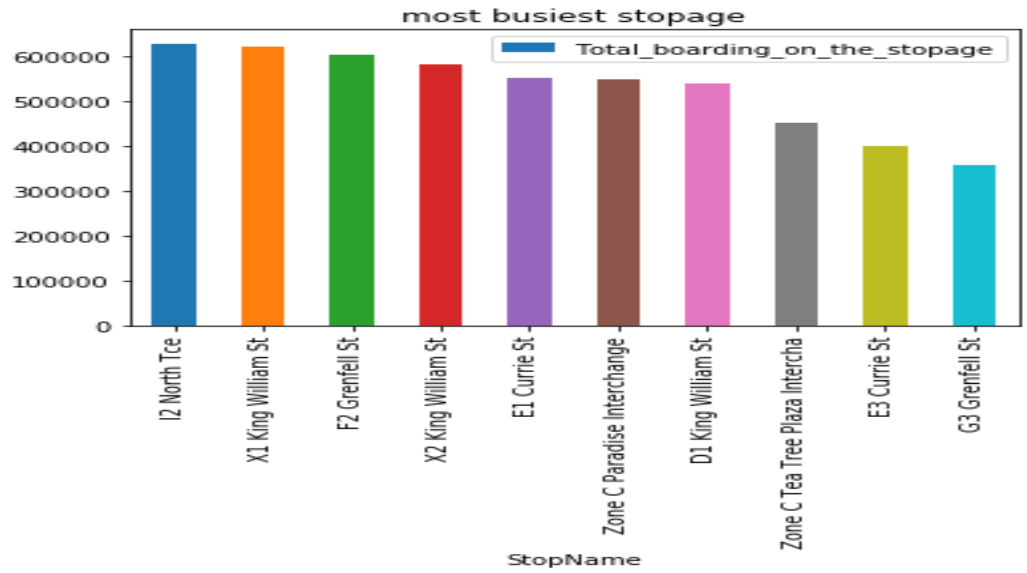
Text(0.5,1,'least Busiest Route')

Out[27]:

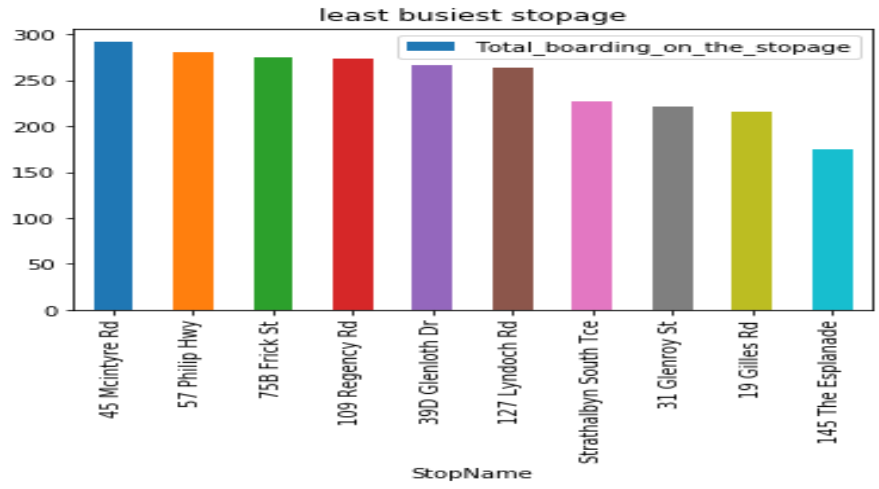
<matplotlib.axes._subplots.AxesSubplot at 0x7ff736bbafd0>



```
ax = stopageName_with_boarding.head(10).plot.bar(x='StopName', y='Total_boarding_on_the_stopage', rot=90)
ax.set_title("most busiest stopage")
```




```
ax = stopageName_with_boarding.tail(10).plot.bar(x='StopName', y='Total_boarding_on_the_stopage', rot=90)
ax.set_title("least busiest stopage")
```



```
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()
```

Out[33]:

	dist_from_centre	NumberOfBoardings
0	0.000018	1892435
1	0.131368	167535
2	0.309089	356518
3	0.314937	1484824
4	0.326005	120061

Out[33]:

```
Index(['dist_from_centre', 'NumberOfBoardings'], dtype='object')
```

Out[33]:

	dist_from_centre	NumberOfBoardings
2392	86.471064	18905
2393	94.826409	321
2394	99.625655	1101
2395	99.665190	4373
2396	99.748995	21216

In [34]:

```
        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 Willi
am 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)
```

In [35]:

```
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
```

In [36]:

```
print(outlier_)
```

```
0
```

In [37]:

```
y = len(distance_10)+len(distance_10_50)+len(distance_50_100)
```

In [38]:

```
print(total)
print("passangers, boarding the buses in the radious of 10Km from the city c
enter = ", (len(distance_10)/total)*100)

print("passanger, boarding the buses from the distance of 10Km to 50Km fr
om the city center = ", (len(distance_10_50)/total)*100)
```

```
print("passanger, boarding the buses from the distance of 50Km to 100 fro
m the city center = ", (len(distance_50_100)/total)*100)
```
10341468

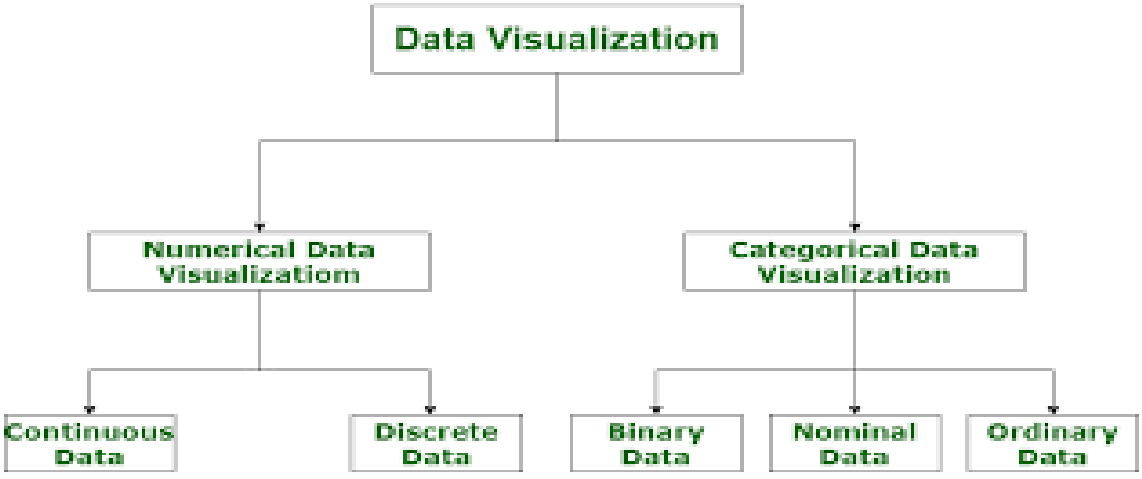
passangers, boarding the buses in the radious of 10Km from the city center
64.31275521038212

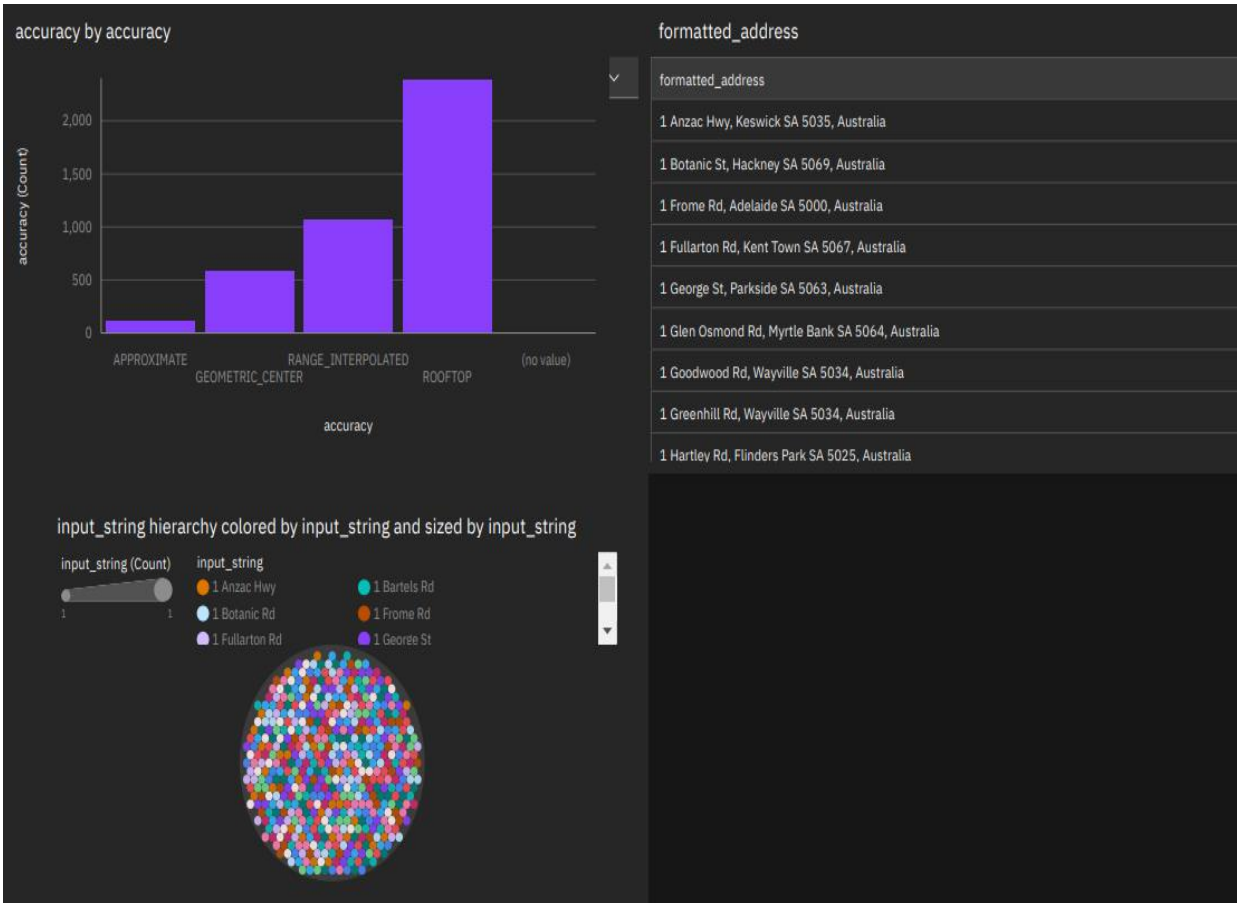
passanger, boarding the buses from the distance of 10Km to 50Km from the city center = 33.16731241638035

passanger, boarding the buses from the distance of 50Km to 100 from the city enter = 2.5199323732375327

Data Visualization:

- ✓ These tools are used to represent your data through charts, graphs, and maps t hat allow you to find patterns and trends in the data.
- ✓ datapine's already mentioned BI platform also offers a wealth of powerful onl ine data visualization tools with several benefits. Some of them include: deliv ering compelling data-driven presentations to share with your entire company
- ✓ the ability to see your data online with any device wherever you are, an intera ctive dashboard design feature that enables you to showcase your results in a n interactive and understandable way, and to perform online self-service repo rts that can be used simultaneously with several other people to enhance team productivity.





Conclusion:

- ✓ Data collection is an essential part of the research process, whether you're conducting scientific experiments, market research, or surveys. The methods and tools used for data collection will vary depending on the research type, the sample size required, and the resources available.