

# PHASE-3

## Development Part 1

<b>DATE</b>	<b>25-10-2023</b>
<b>TEAM ID</b>	<b>9277</b>
<b>PROJECT NAME</b>	<b>8301-PUBLIC TRANSPORTATION EFFICIENCY ANALYSIS</b>
<b>TEAM NAME</b>	<b>PROJ_207140_TEAM_1</b>

# ANALYTICS OBJECTIVES

## Data Preprocessing:

1. Data Collection
2. Data Inspection
3. Data Cleaning
4. Data Transformation
5. Data Splitting
6. Data Normalization
7. Data Validation
8. Data Visualization

## 1. Loading Data:

- Use pandas `.read_csv()` to load data from a CSV file.
- Use pandas `.read_excel()` for Excel files.

```
In [40]: import pandas as pd  
df=pd.read_csv("C:\transportexcel.csv")
```

## 2. Exploring Data:

- Use `df.head()` to view the first few rows of the dataset.

In [41]: `df.head()`

Out[41]:

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

- Use `df.info()` to get information about data types and missing values.

```
In [42]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1074 entries, 0 to 1073
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   TripID                1074 non-null   int64
1   RouteID               1074 non-null   int64
2   StopID                1074 non-null   int64
3   StopName              1074 non-null   object
4   WeekBeginning          1074 non-null   datetime64[ns]
5   NumberOfBoardings     1074 non-null   int64
dtypes: datetime64[ns](1), int64(4), object(1)
memory usage: 50.5+ KB
```

- Use df.describe() for summary statistics python.

In [6]: df.describe()

Out[6]:

	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

### 3. Handling Missing Values:

- Use `df.isnull()` to identify missing values.
- Use `df.dropna()` to handle missing values.

In [44]: `df.isnull()`

Out[44]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
1069	False	False	False	False	False	False
1070	False	False	False	False	False	False
1071	False	False	False	False	False	False
1072	False	False	False	False	False	False
1073	False	False	False	False	False	False

1074 rows × 6 columns

## 4. Data Cleaning:

- Remove duplicates rows with `df.drop_duplicates()`.
- Rename columns using `df.rename()` if necessary.
- Convert datatypes with `df.astype()`.

In [48]: `df.drop_duplicates()`

Out[48]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
0	23631	100	14156	181 Cross Rd	2013-06-30	1
1	23631	100	14144	177 Cross Rd	2013-06-30	1
2	23632	100	14132	175 Cross Rd	2013-06-30	1
3	23633	100	12266	Zone A Arndale Interchange	2013-06-30	2
4	23633	100	14147	178 Cross Rd	2013-06-30	1
...	...	...	...	...	...	...
1069	44705	100	14124	174 Cross Rd	2013-06-30	3
1070	44705	100	14147	178 Cross Rd	2013-06-30	2
1071	44705	100	12216	224 Woodville Rd	2013-06-30	1
1072	44705	100	14024	10 Marion Rd	2013-06-30	2
1073	44705	100	14112	171 Cross Rd	2013-06-30	1

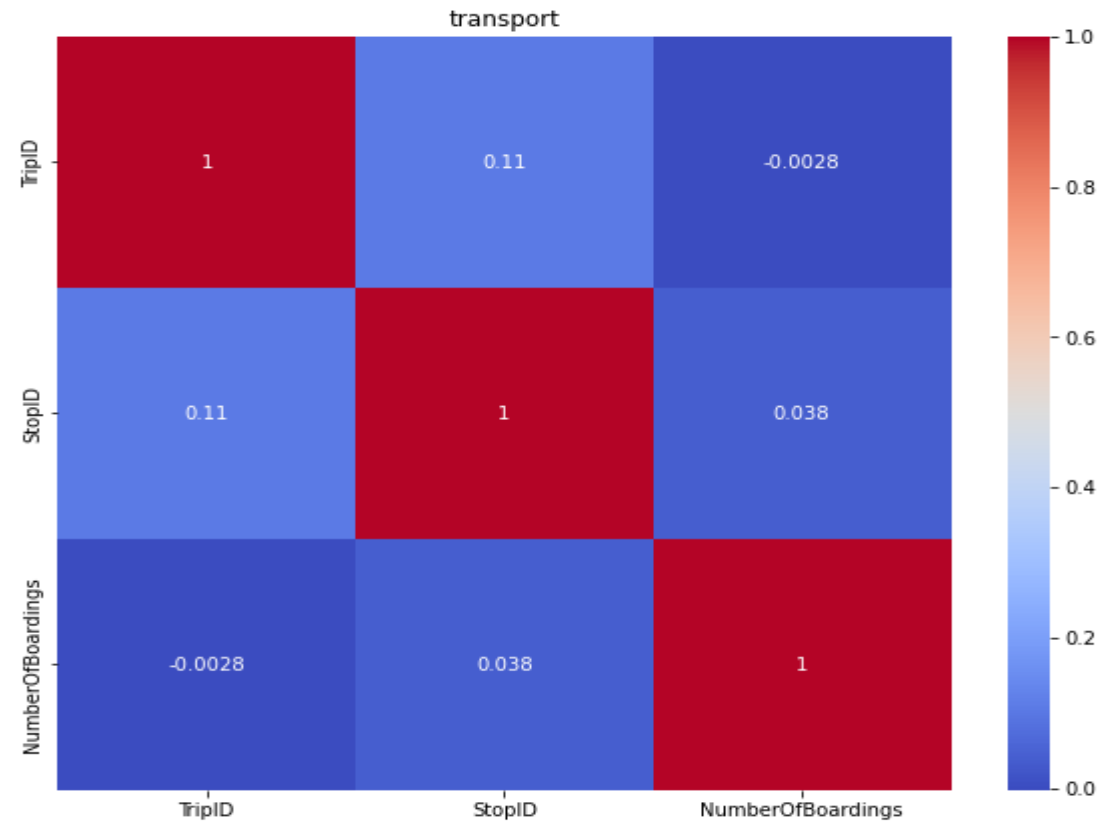
1074 rows × 6 columns



## 5. Handling Outliers:

- Detect and deal with outliers using statistical methods or visualization.
- You can use techniques like z-scores or IQR(Interquartile Range)

```
In [54]: import seaborn as sns
import matplotlib.pyplot as plt
corr_matrix = df.corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix,annot=True,cmap="coolwarm")
plt.title("transport")
plt.show()
```



In [ ]:

## 6. Saving Data:

- Save the preprocessed data back to a file if needed

```
df.to_csv('Downloads/preprocessed_transportexcel.csv',index=False)
```

IBM COGNOS ANALYTICS

# IN COGNOS-DATA LOADING

The screenshot shows the IBM Cognos Analytics interface. On the left, the 'Data module' pane lists several modules, with 'transportexcel.xlsx' highlighted. Below it, a list of fields is visible, including '# Row Id', '# TripID', '# RouteID', '# StopID', 'abc StopName', and '🕒 WeekBeginning'. The main area displays a table with columns: Row Id, TripID, RouteID, StopID, StopName, and WeekBeginning. The table contains 9 rows of data.

Row Id	TripID	RouteID	StopID	StopName	WeekBeginning
1	23631	100	14156	181 Cross Rd	6/
2	23631	100	14144	177 Cross Rd	6/
3	23632	100	14132	175 Cross Rd	6/
4	23633	100	12266	Zone A Arndale Interchange	6/
5	23633	100	14147	178 Cross Rd	6/
6	23634	100	13907	9A Marion Rd	6/
7	23634	100	14132	175 Cross Rd	6/
8	23634	100	13335	9A Holbrooks Rd	6/
9	23634	100	13875	9 Marion Rd	6/

# IN COGNOS-DATA RELATIONSHIP

IBM Cognos Analytics

\* New exploration

5

Search

Share

Help

Notifications

User

Save

Share

Undo

Redo

+

Cards

3

4

NumberOfBoard... by StopName

Data relationships

Relationship diagram

10% 100%

## Explore data relationships

transportexcel.xlsx

Reset to original

Search NumberOfBoardings

Edit diagram

StopName

NumberOfBoardings

StopID

TripID

NumberOfBoardings by StopName colored by TripID

Add +

NumberOfBoardings by StopID colored by StopName

Add +

The screenshot displays the IBM Cognos Analytics interface. At the top, there's a navigation bar with the IBM Cognos Analytics logo, a 'New exploration' button, and a search bar. Below this is a toolbar with icons for save, share, undo, redo, and a plus sign. The main area is divided into several sections. On the left, there's a 'Cards' section with two cards: '3' and '4'. Card '3' shows a bar chart with multiple colored bars. Card '4' shows a bar chart titled 'NumberOfBoard... by StopName'. Below these cards is a 'Data relationships' section with a diagram showing a central node connected to four other nodes, and three empty boxes for additional relationships. The central part of the interface is titled 'Explore data relationships' and shows a relationship diagram for 'transportexcel.xlsx'. The diagram has a central node 'NumberOfBoardings' connected to three other nodes: 'StopName', 'StopID', and 'TripID'. There are search bars and buttons like 'Reset to original' and 'Edit diagram'. At the bottom of this section is a 'Relationship diagram' legend and a zoom slider from 10% to 100%. On the right side, there are two more cards. The top card is titled 'NumberOfBoardings by StopName colored by TripID' and shows a bar chart. The bottom card is titled 'NumberOfBoardings by StopID colored by StopName' and also shows a bar chart. Both cards have an 'Add +' button.

# IN COGNOS -TRANSPORT DASHBOARD

The screenshot displays the IBM Cognos Analytics interface for a transport dashboard. The top navigation bar includes the IBM Cognos Analytics logo, a '+ New dashboard' button, and a series of icons for notifications (5), search, chat, help, and user profile. Below the navigation bar is a toolbar with icons for edit, save, share, undo, redo, and zoom. The left sidebar shows the 'Selected sources' section with a search bar and a list of data sources: 'transportexcel.xlsx' (expanded) showing fields like TripID, RouteID, StopID, StopName, WeekBeginning, and NumberOfBoardings. The main workspace features a tabbed interface with 'Tab 1', 'Tab 2' (selected), and 'Tab 3'. Above the tabs is a filter area with 'All tabs' and 'This tab' options, and a prompt to 'Drag and drop data here to filter this tab.' The dashboard content includes three widgets: a table for 'WeekBeginning' showing a date '6/30/2013, 12:00 AM', a large number card for 'NumberOfBoardings' displaying '3883', and a list for 'StopName' showing addresses like '10 Holbrooks Rd', '10 Marion Rd', '10A Marion Rd', '11 Marion Rd', and '11 Portrush Rd'.

IBM Cognos Analytics

\* New dashboard

5

Edit

Selected sources /

transportexcel.xlsx

Search

Navigation paths

transportexcel.xlsx

- # TripID
- # RouteID
- # StopID
- abc StopName
- 🕒 WeekBeginning
- 📊 NumberOfBoardings

All tabs

This tab

Drag and drop data here to filter this tab.

Tab 1

Tab 2

Tab 3

WeekBeginning

WeekBeginning
6/30/2013, 12:00 AM

NumberOfBoardings

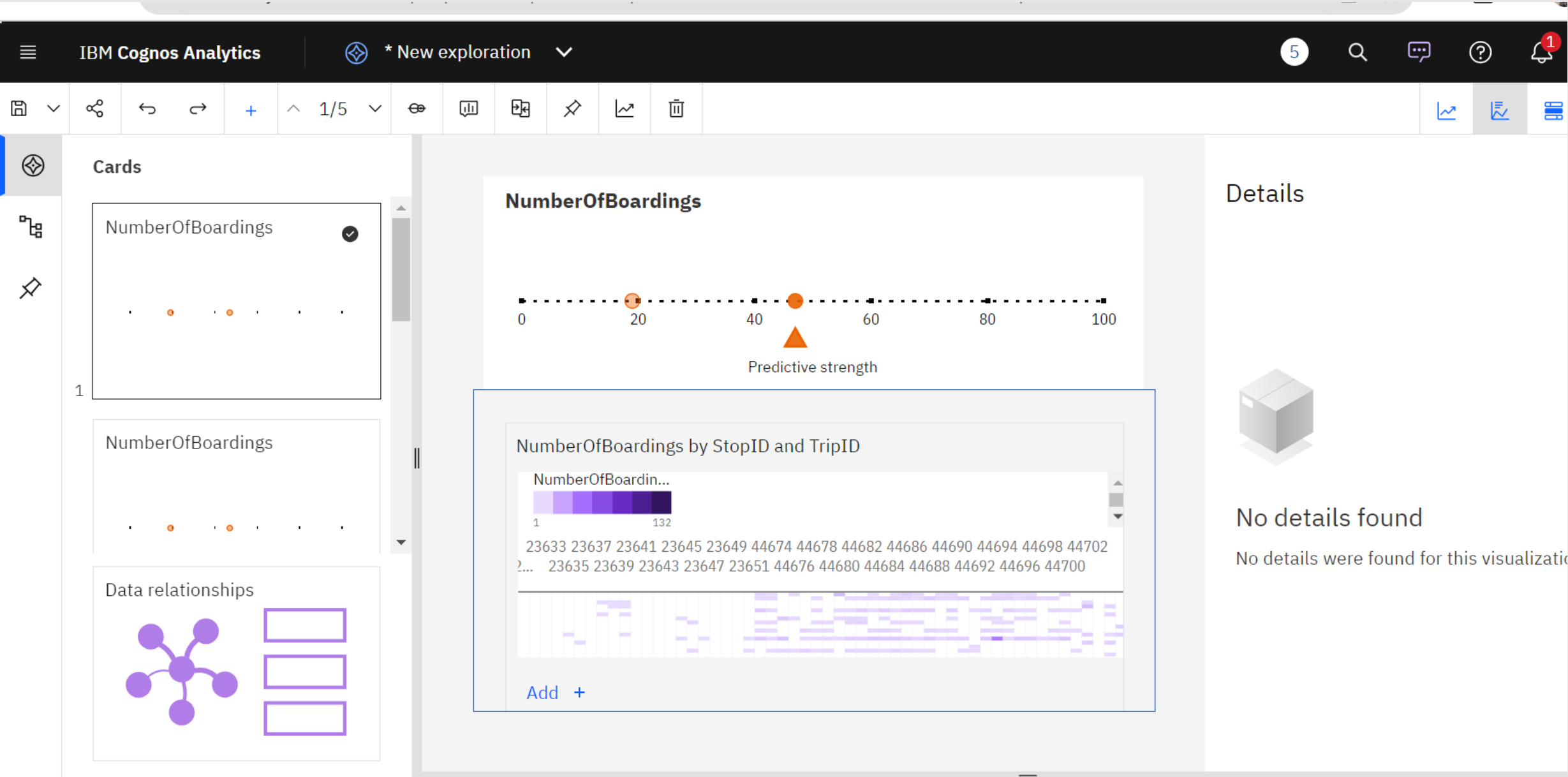
3883

NumberOfBoardings

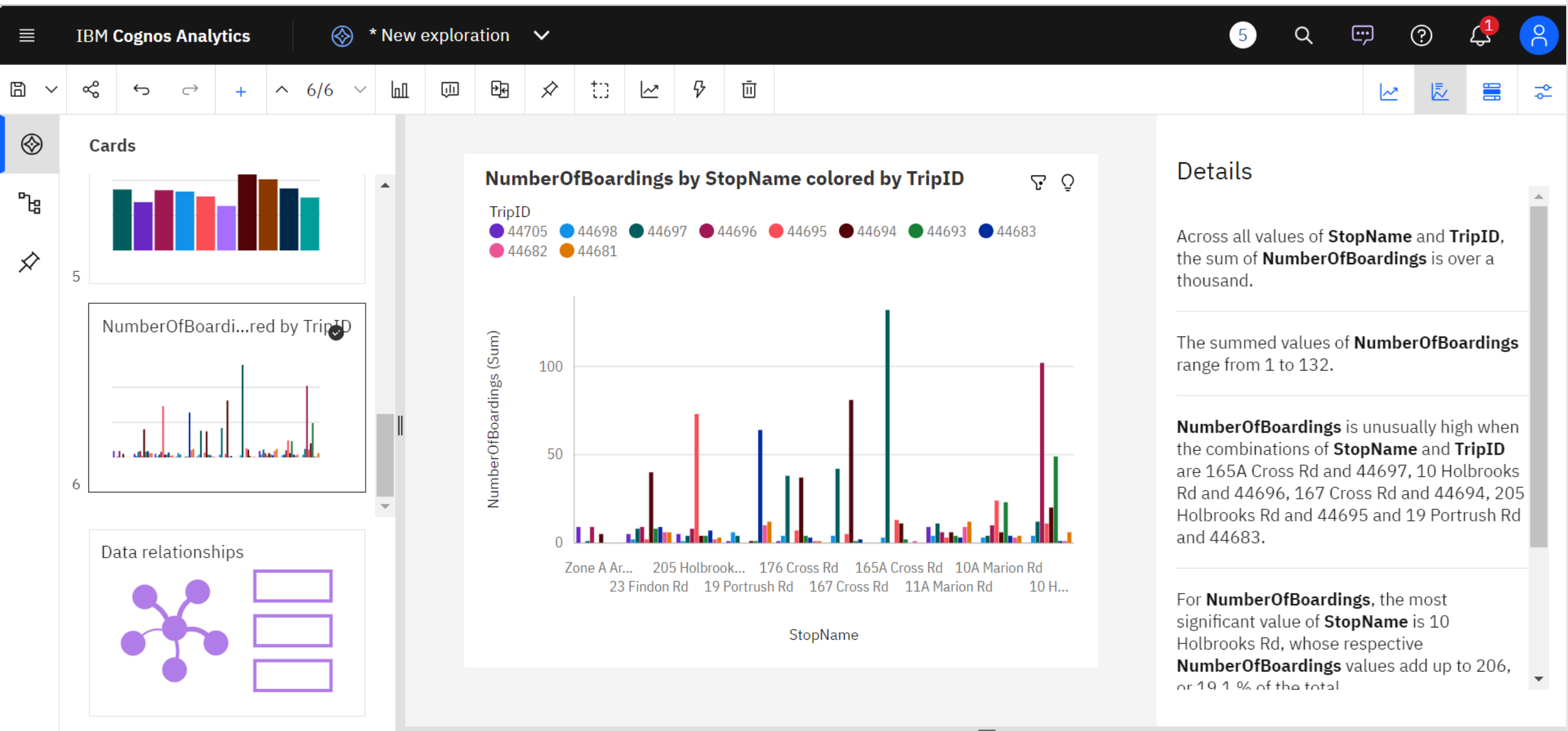
StopName

StopName
10 Holbrooks Rd
10 Marion Rd
10A Marion Rd
11 Marion Rd
11 Portrush Rd

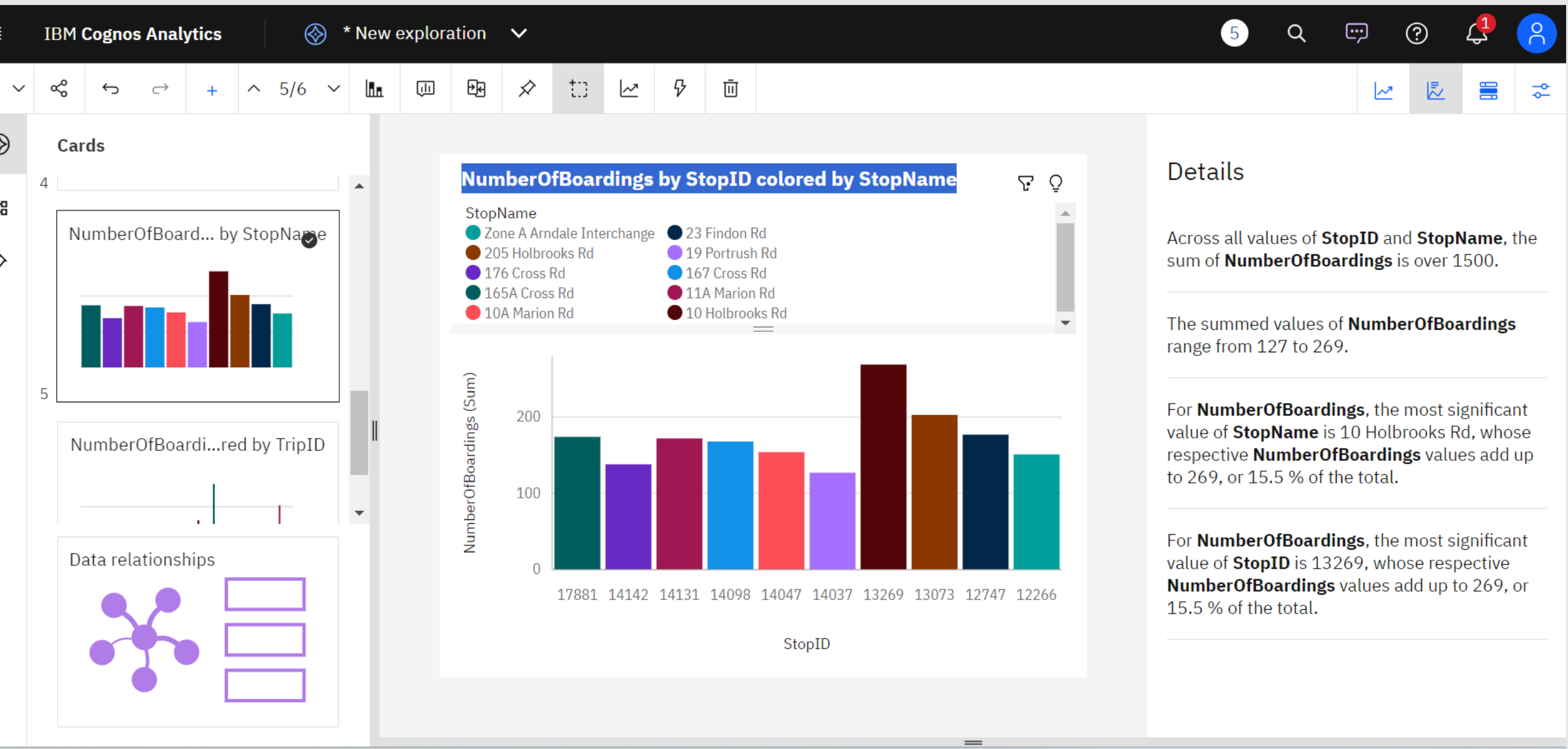
# IN COGNOS-NUMBER OF BOARDINGS



# IN-COGNOS NUMBEROFBOARDINGS BY STOPNAME COLORED BY TRIPID



# IN-COGNOS NUMBEROFBOARDINGS BY STOPID COLORED BY STOPNAME





## CONCULSION

Data preprocessing is a crucial step in analyzing public transportation efficiency. It involves data collection, data cleaning, data transformation, data aggregation. The main steps include data collection, data inspection, cleansing, data transformation, data splitting, normalization, data validation.

By effectively preprocessing public transportation data, we can derive meaningful insights to improve efficiency, optimize routes, and enhance overall transportation system.