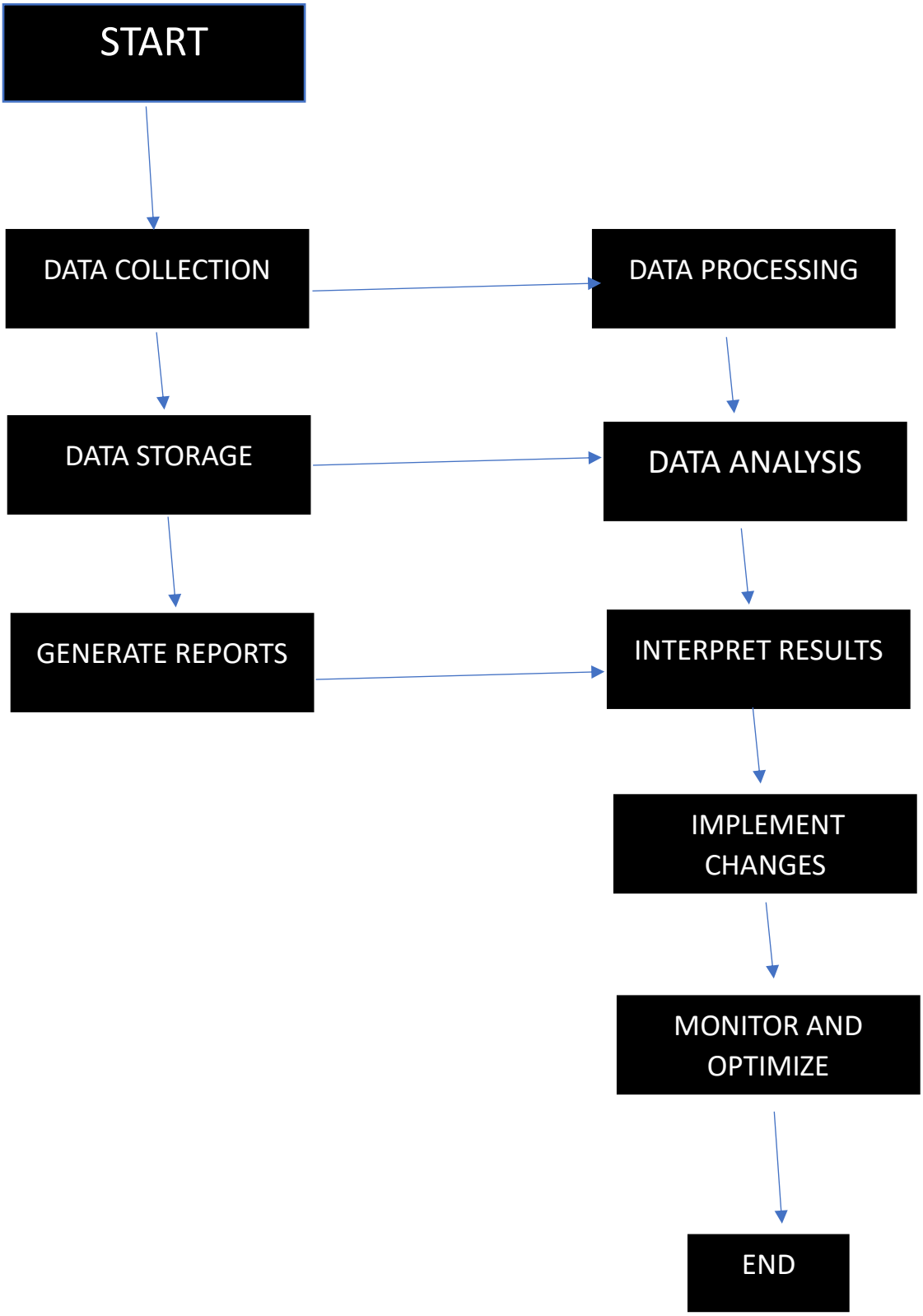
Public Transportation Efficiency Analysis

ARCHITECTURE OF OUR PROJECT



Public Transportation Efficiency

Define the Scope and Objectives:

Clearly define the objectives of your analysis. Are you trying to improve service reliability, reduce operational costs, or enhance the overall passenger experience? Knowing the scope and goals of your analysis will guide your model-building efforts.

Gather Data:

Collect data from various sources, including transit agencies, government records, and surveys. Relevant data may include:

- Ridership data
- Financial data
- External factors

Ensure that the data is accurate, up-to-date, and relevant to your analysis.

Data Preprocessing:

Clean, preprocess, and format the data for analysis. This may involve handling missing values, outliers, and

data normalization. You may also need to merge data from different sources.

Feature Engineering:

Create relevant features from the raw data to help the model better understand the factors that affect public transportation efficiency. For example, you could create features like on-time performance, route density, or passenger load factors.

Select a Modeling Approach:

Choose the appropriate modeling technique based on your objectives. Common modeling approaches for public transportation efficiency analysis include:

- Regression analysis: For identifying relationships between variables
- Time series analysis: For forecasting ridership and service demand.
- Machine learning models: For complex, data-driven analyses, such as predicting customer satisfaction or optimizing routes.

Model Development:

Develop the chosen model using the preprocessed data. This may involve the use of machine learning libraries, statistical software, or custom code.

Model Validation:

Evaluate the model's performance using appropriate metrics and validation techniques. Cross-validation, mean absolute error, or R-squared values are commonly used metrics for assessing model accuracy.

Interpret the Results:

Understand the insights provided by the model. Determine which factors have the most significant impact on public transportation efficiency. Identify potential areas for improvement.

Scenario Analysis:

Use the model to perform scenario analysis. For instance, you can simulate the impact of changes in service frequency, route modifications, or fare adjustments on efficiency.

Implement Recommendations:

Translate the insights from your model into actionable recommendations for public transportation authorities. These recommendations may include operational changes, infrastructure improvements, or policy adjustments.

Monitor and Update:

Continuously monitor the system's performance, collect new data, and update the model to ensure that the analysis remains relevant and useful.

Reporting and Visualization:

Present your findings and recommendations in a clear and understandable manner through reports, dashboards, or visualizations. This helps stakeholders make informed decisions.

DATA CLEANING AND ANALYSIS

```
data.dropna(inplace=True)

data.drop_duplicates(inplace=True)

data.head()
```

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
1	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
2	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1
3	23632	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
4	23633	100	12266	Zone A Arndale Interchange	2013-06-30 00:00:00	2
5	23633	100	14147	178 Cross Rd	2013-06-30 00:00:00	1

```
[ ] data['WeekBeginning'] = pd.to_datetime(data['WeekBeginning'], format='%Y-%m-%d %H:%M:%S')
```

```
[ ] data['DayOfWeek'] = data['WeekBeginning'].dt.day_name()

    data.head()
```

+ Code + Text

T4 RAM Disk

```
[ ] data = pd.get_dummies(data, columns=[c for c in data.columns if c.strip() == "categorical_column"])
```

```
[ ] columns_to_drop = ['ColumnToDrop1', 'ColumnToDrop2']
    data = data.drop([c for c in columns_to_drop if c in data.columns], axis=1)
```

```
[ ] target_column = 'target_column'
    target_column = target_column.strip()

    data.head()
```

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	DayOfWeek
1	23631	100	14156	181 Cross Rd	2013-06-30	0.000000	Sunday
2	23631	100	14144	177 Cross Rd	2013-06-30	0.000000	Sunday
3	23632	100	14132	175 Cross Rd	2013-06-30	0.000000	Sunday
4	23633	100	12266	Zone A Arndale Interchange	2013-06-30	0.001025	Sunday
5	23633	100	14147	178 Cross Rd	2013-06-30	0.000000	Sunday

```
[ ] data.to_csv('preprocessed_dataset.csv', index=False)
```

```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import csv
import sys
```

```python
df = "20140711(1) .CSV"

try:
    with open(file_path, 'r', errors='replace') as csvfile:
        reader = csv.reader(csvfile)
        rows = []

        for row in reader:
            rows.append(row)
        data = pd.DataFrame(rows, columns=rows[0])
        data = data.iloc[1:]

except pd.errors.ParserError as e:
    sys.exit(f'Error parsing CSV: {e}')
```

```python
[ ] data.head()
```

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
1	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
2	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1
3	23632	100	14132	175 Cross Rd	2013-06-30 00:00:00	1

✓ 0s completed at 10:12 PM

```python
[ ] data.head()
```

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
1	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
2	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1
3	23632	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
4	23633	100	12266	Zone A Arndale Interchange	2013-06-30 00:00:00	2
5	23633	100	14147	178 Cross Rd	2013-06-30 00:00:00	1

```python
data.describe()
```

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
count	3585808	3585808	3585808	3585808	3585808	3585807
unique	12094	99	2629	1495	54	215
top	27478	178	16279	2 King William Rd	2014-03-02 00:00:00	1
freq	2733	181303	22679	27353	72849	1435663

```python
[ ] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3585808 entries, 1 to 3585808
Data columns (total 6 columns):
 #   Column            Dtype
```

✓ 0s completed at 10:12 PM