Diego Fernandez
Df356
GitHub repo: github.com/DFL01/ORIE_4741_Project

Traffic Flow Prediction for NYC's Broadway Avenue

NOTE: Given the lack of available data for construction or traffic interruption in NYC, we will pivot this project's goal from multivariate data analysis to a single data prediction project.

Introduction

Throughout the report we will perform an analysis of traffic flow in New York City's Broadway area, with a focus on predicting daily traffic volumes.

The goal is to leverage these predictions to enhance urban planning and reduce congestion.

We will implement several techniques learnt in class, evaluate their performance, and discuss their potential implementation in practical settings.

Background Information

Traffic congestion in urban areas such as NYC is a significant challenges in our modern society that affects everything from commute times to air quality. With a population of over 8 million people and over 1 million commuters daily, NYC traffic management is one of the central issues that the local government has been trying to address for decades.

Throughout the report we will focus on one of the busiest commercial and entertainment avenues in the city: Broadway, a major artery in the city's transport network. This street was not only due to its high traffic volume and importance but also due to the high volume of data available

Data Set

The dataset used in this report was obtained from New York City's open data portal, which includes traffic volume counts across various streets. The dataset contains data for most streets, bridges and tunnels in the city and provides hourly traffic volume data recorded over several months.

The data attributes as shown in image 1 below include:

- **Segment ID:** Unique identifier for the road segment
- **Roadway Name:** Name of the road

- **From/To:** Start and end of segment
- **Date and Time:** Timestamps for traffic counts
- **Hourly Volumes:** Traffic counts for each hour of the day

	ID	SegmentID	Roadway Name	From	To	Direction	Date	12:00-1:00 AM	1:00-2:00AM	2:00-3:00AM	...	2:00-3:00PM	3:00-4:00PM	4:00-5:00PM	5:00-6:00PM	6:00-7:00PM	7:00-8:00PM	8:00-9:00PM	9:00-10:00PM	10:00-11:00PM	11:00-12:00AM
0	1	15540	BEACH STREET	UNION PLACE	VAN DUZER STREET	NB	2012-01-09	20.0	10.0	11.0	...	104.0	105.0	147.0	120.0	91.0	83.0	74.0	49.0	42.0	42.0
1	2	15540	BEACH STREET	UNION PLACE	VAN DUZER STREET	NB	2012-01-10	21.0	16.0	8.0	...	102.0	98.0	133.0	131.0	95.0	73.0	70.0	63.0	42.0	35.0
2	3	15540	BEACH STREET	UNION PLACE	VAN DUZER STREET	NB	2012-01-11	27.0	14.0	6.0	...	115.0	115.0	130.0	143.0	106.0	89.0	68.0	64.0	56.0	43.0
3	4	15540	BEACH STREET	UNION PLACE	VAN DUZER STREET	NB	2012-01-12	22.0	7.0	7.0	...	71.0	127.0	122.0	144.0	122.0	76.0	64.0	58.0	64.0	43.0
4	5	15540	BEACH STREET	UNION PLACE	VAN DUZER STREET	NB	2012-01-13	31.0	17.0	7.0	...	113.0	126.0	133.0	135.0	102.0	106.0	58.0	58.0	55.0	54.0

Image 1

Methodology

1. *Data Processing*:

   We first start by cleaning and filtering the data.

   I. Converting date fields to datetime objects.

   ```
   traffic_data['Date'] = pd.to_datetime(traffic_data['Date'],
   errors='coerce')
   ```

   II. This includes filtering for Broadway data

   ```
   broadway_data = traffic_data[traffic_data['Roadway
   Name'].str.contains("Broadway")]
   ```

2. *Feature Engineering*:

   I. We then move onto feature engineering where we start off by extracting the day of the week, whether it's a weekend, and month from each row's date to use as predictive features.

   ```
   broadway_data.loc[:, 'day_of_week'] =
   broadway_data['Date'].dt.dayofweek
   broadway_data.loc[:, 'is_weekend'] =
   broadway_data['day_of_week'].apply(lambda x: 1 if x >= 5 else
   0)
   broadway_data.loc[:, 'month'] = broadway_data['Date'].dt.month
   ```

   II. Traffic data is recorded on an hourly basis.
   However, in order to be able to generate more accurate predictions of traffic patterns across Broadway, the hourly data is aggregated into total daily traffic volume.

This sum represents the total count of vehicles in a day.

```
hourly_columns = [col for col in broadway_data.columns if 'AM'
in col or 'PM' in col]
broadway_data.loc[:, 'daily_traffic_volume'] =
broadway_data[hourly_columns].sum(axis=1)
```

III. We then select the features that are relevant for our regressions. Given that we are trying to predict the volume of traffic in Broadway, we will select the **day of the week**, **whether it's a weekend** and the **month** as key predictors for the daily traffic volume on this NYC avenue.

```
features = broadway_data[['day_of_week', 'is_weekend',
'month']]
target = broadway_data['daily_traffic_volume']
```

3. *Data Visualization*

Before we start implementing our models, we will first visualize Broadway data's daily traffic volume.

a. A summary of the stats for daily traffic volume

```
count      423.000000
mean     10619.848700
std       5311.714444
min       2130.000000
25%       7140.000000
50%       9791.000000
75%      12414.000000
max      28498.000000
```

b. A histogram to view the distribution of daily traffic volume (image 2).

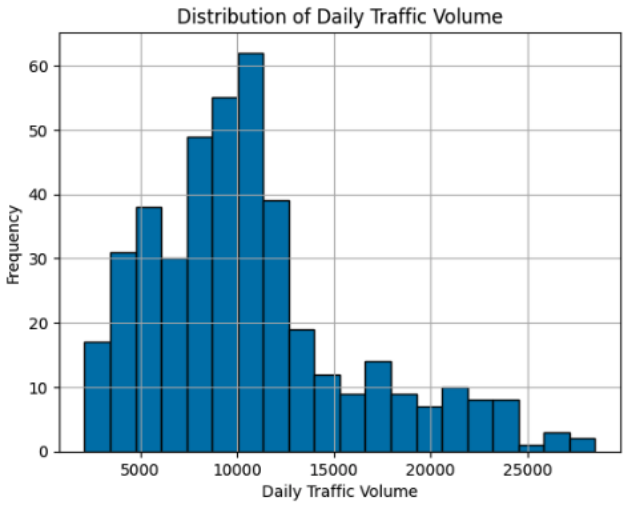

Image 2

c. A box plot of daily traffic volume to help visualize outliers and the spread of the data (image 3).

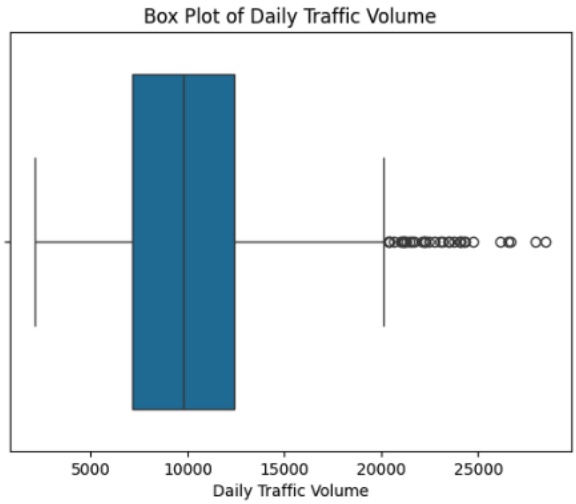

Image 3

4. *Data Split*

We then proceed to shuffle and split our data into training and testing datasets. To do so we use sklearn.model_selection's train_test_split() function and shuffle our data.

Given that our filtered data contains 42756 rows we will shuffle our data in batches of 21 rows to achieve as much consistency in our shuffling (given that 42756 is divisible by 21).

We will use a 20/80 split: 20% of the data will be used to test and 80% will be used to train.

```
X_train, X_test, y_train, y_test = train_test_split(features,
target, train_size= 0.8, test_size=0.2, random_state=21)
```

Given the relatively small size of our data set (42756 rows), the validation data set has not been taken into consideration. Instead, if necessary the method of cross validation will be used.

5. *Model Implementation*

We will employ two different regression models to predict total traffic volume in NYC and compare how these two models perform.

I. Linear Regression

To predict the number of vehicles on Broadway daily, we will start with the simplest regression model: linear regression.

This model has a low computational overhead and can give a quantitative measure of the number of cars that will be passing through this NYC avenue.

The linear regression model will be implemented using sklearn.linear_model's library.

```
reg_model = LinearRegression()
reg_model.fit(X_train, y_train)
y_pred_reg = reg_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred_reg)
r2 = r2_score(y_test, y_pred_reg)
```

II. Logistic Regression

To obtain a regression that is easier to understand, we will implement logistic regression to categorize traffic into three different classes: low, medium and high traffic volumes.

Looking back at our data visualizations in section 3, we will take the thresholds to categorize traffic volumes to be at 25% and at 75%. Therefore we will make the following assumptions for the case of this analysis:

- Low traffic: 7,000 vehicles or fewer
- Medium traffic: 7,000 to 12,000 vehicles
- High traffic: 12,000 vehicles or more

```
low_threshold = target.quantile(0.33)
high_threshold = target.quantile(0.66)
```

In conclusion we have determined the following:

Low traffic days (25th percentile and below):

25% percentile is approx. 7140 vehicles per day

So low traffic days have around 7,000 vehicles or fewer.

Medium traffic days (25th to 75th percentile):

50% percentile (median) is 10619.848700 vehicles per day

Medium traffic days range from about 7,000 to 12,000 vehicles.

High traffic days (75th percentile and above):

75% percentile is 12414 vehicles per day
High traffic days have around 12,000 vehicles or more.

While some busy days can exceed 25,000 vehicles, these outliers are relatively rare, having most days fall between 5,000 to 20,000 vehicles, with the middle 50% of days seeing between 7,000 to 12,000 vehicles on Broadway.

We will therefore categorize the target data (daily traffic volume on Broadway) as follows:

```
y_train_categorized = pd.cut(y_train, bins=[0, low_threshold,
high_threshold, float('inf')], labels=['Low', 'Medium',
'High'])

y_test_categorized = pd.cut(y_test, bins=[0, low_threshold,
high_threshold, float('inf')], labels=['Low', 'Medium',
'High'])
```

Finally we implement the logistic regression model using sklearn.linear_model's library.

```
log_reg_model = LogisticRegression()
log_reg_model.fit(X_train, y_train_cat)
y_pred_class = log_reg_model.predict(X_test)
accuracy = accuracy_score(y_test_cat, y_pred_class)
class_report = classification_report(y_test_cat, y_pred_class)
```

Results

Linear Regression Model Metrics:
Mean Squared Error: 26653726.83142919
R-squared: 0.02097216803475599

Logistic Regression Model Metrics:
Accuracy: 0.4470588235294118

	precision	recall	f1-score	support
High	0.33	0.12	0.18	25
Low	0.00	0.00	0.00	20
Medium	0.46	0.88	0.60	40

```
              precision    recall  f1-score   support

    accuracy                           0.45        85
   macro avg       0.26      0.33      0.26        85
weighted avg       0.31      0.45      0.34        85
```

Discussion

Linear Regression Model

The linear regression model performed poorly at predicting daily traffic volumes with an R-squared value of approximately 0.021. This means that only 2% of all of the variability in daily traffic volume can be justified by the model.

Therefore, we can assume that the model is not predicting the underlying patterns correctly. This is also justified by the high mean squared error, which points to significant errors in the predictions of actual traffic volumes, and reinforces the poor performance of this model.

Logistic Regression Model

The logistic regression model, used to classify traffic volume into the three categories defined above (Low, Medium, High), managed to make predictions with a 44.71% accuracy.

Despite this performance being better than randomly guessing (33% accuracy) this accuracy level is still relatively low for any practical decision-making.

The model performs well in identifying 'medium' traffic days, performing with an 88% accuracy and an F1-score of 60%.

However the model is not accurate at predicting low or high traffic days. The zero precision and recall for low traffic days suggest that the model fails to correctly identify any days with low traffic.

Weapons of Math Destruction

It is important to take into account the ethical considerations to ensure that models like these do not become Weapons of Math Destruction (WMDs).

In the context of traffic volume prediction models like those discussed for the Broadway traffic analysis, inaccurate predictions like those produced by the Linear Regression Model can result in the loss of public funds.

The use of this inaccurate information could also have more serious effects if they are used in GPS and navigation devices of first responders, law enforcement and ambulances.

Conclusion

The results of both models show that there is room for improvement.

The linear regression model could be improved by reconsidering the implementation of feature selection or model complexity.
Despite the initial plan for this project to be a multivariate data analysis that cross references traffic volumes with construction of traffic interruption data, the unavailability of this latter data forced this project to pivot to a single variable data analysis. However, by incorporating additional predictors, such as weather conditions, special events, or real-time traffic data, the model's performance could be significantly improved.

The logistic regression model is relatively accurate in classifying days with medium traffic but is unreliable for days with high or low traffic.
Some improvements in the model could involve redefining the threshold values for classification, or integrating more contextual data that could influence traffic patterns as mentioned above for the linear regression model improvement.

In their current state, neither model is ready for deployment in a policy making environment for traffic management on Broadway.

Considering other modeling approaches or data sources might help improve results.

References

NYC Open Data, Traffic Volume Counts (https://data.cityofnewyork.us/Transportation/Traffic-Volume-Counts/btm5-ppia/about_data)