Predicting Citi Bike Trip Duration Using Data Mining Algorithms

Katie Cao

Abstract

Purpose:

- Explore new revenue opportunities for bike share systems through data mining techniques.
- As these systems grow, sponsorship may not be able to scale at the same pace.

Goal:

- Predict Citi Bike trip duration based on geographic and demographic factors of trip-level data.
- Use this knowledge to implement dynamic pricing for peak use periods.

Methods:

- Downloaded Citi Bike system data and monthly operation reports.
- Applied data mining techniques including Decision Tree analysis, Logistic Regression, and Artificial Neural Networks.

Introduction

750+ active stations, across Manhattan, Brooklyn, Queens, and Jersey City
Largest bike share system in North America - 6th largest in the world.

- 8K-9K active bikes on the fleet on any given day
- September 2018:
    - 146K active annual members
    - 121K casual passes purchased (single trip, 1 day, and 3 day)
    - 62K rides taken, with each bike averaging 7 rides per day
- Revenue for 2017 reached $47M

Introduction

The current pricing model

Membership	Included	add'l 15 min
Subscriber	45 min	$2.50
Customer	30 min	$4.00

9.9 mins **average trip duration** during Nov 2017 - Oct 2018

not charging dynamically

The flat rent rate should be around the same level as the average trip duration.

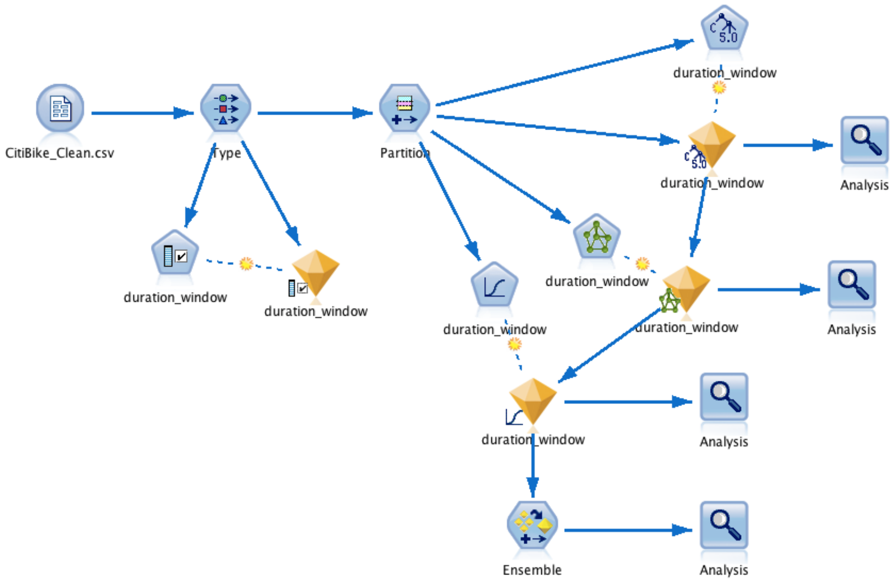
Predict the ride duration → **Charge a higher, dynamic rate** → **Increase revenue**

Learning Model

Gaining an understanding of within which time range the trip is going to be.

Classification + **Ensemble**

↑ **Higher Accuracy**

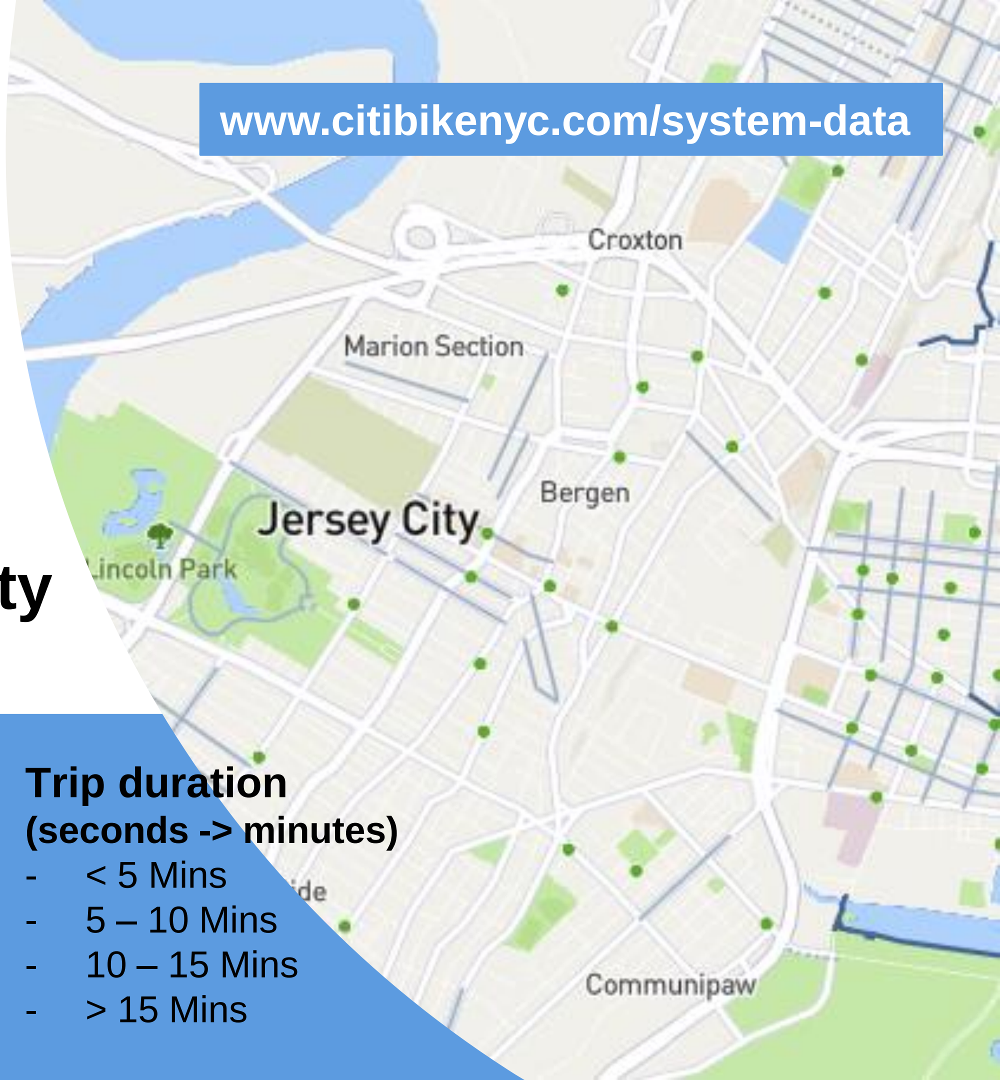


Compare several classification models

Recommend a classifier

Data Description
Processing Data

www.citibikenyc.com/system-data



12 months

Nov 2017 - Oct 2018

top 10 stations in Jersey City

based on Start frequency

Start time ->
- Morning
- Afternoon
- Evening
- Night

Birth year -> Age
- Youth
- Young adult
- Adult
- Senior

Trip duration (seconds -> minutes)
- < 5 Mins
- 5 – 10 Mins
- 10 – 15 Mins
- > 15 Mins

Data Description - *Variables*

165,401 records in total with 60% training data, 30% testing and 10% validation to reduce overfitting problem

Field	Measurement	Values	Missing	Check	Role
field1	Continuous	[0,42010]		None	None
tripduration	Continuous	[61.0,1193290...		None	None
starttime	Continuous	[2017-11-01 0...		None	None
stoptime	Continuous	[2017-11-01 0...		None	None
start station id	Nominal	3183.0,3186.0...		None	None
start station name	Nominal	"Exchange Pla...		None	Input
start station latitu...	Continuous	[40.71241882...		None	None
start station longi...	Continuous	[-74.06378388...		None	None
end station id	Continuous	[212.0,3694.0]		None	None
end station name	Nominal	"12 Ave & W 4...		None	None
end station latitude	Continuous	[40.69263996...		None	None
end station longit...	Continuous	[-74.09693659...		None	None
bikeid	Continuous	[14793.0,3500...		None	None
usertype	Flag	Subscriber/Cu...		None	Input
birth year	Continuous	[1887.0,2002.0]		None	None
gender	Nominal	0,1,2		None	Input
age	Continuous	[16.0,131.0]		None	None
agegroup	Nominal	adult,senior,"y...		None	Input
time	Nominal	afternoon,even...		None	Input
tripdurationinmins	Continuous	[1.016666666...		None	None
duration_window	Nominal	"10 - 15 mins",...		None	Target

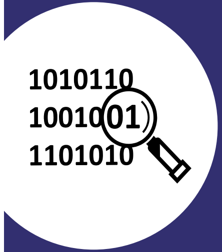
Methodology – *Data Source and Data Pre-processing*

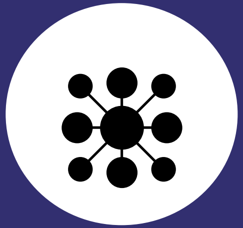
- Data extracted from Citi Bike website directly
- Utilize Pandas and NumPy in Python to conduct feature engineering
    - Filter the **top 10 stations** to start and filter records with missing data
    - Derive feature **“age”** using **“birthyear”** in the original dataset
    - Rescale trip duration from seconds to minutes and catergorize into four duration ranges

Methodology – *Variable Selection and Model Building*

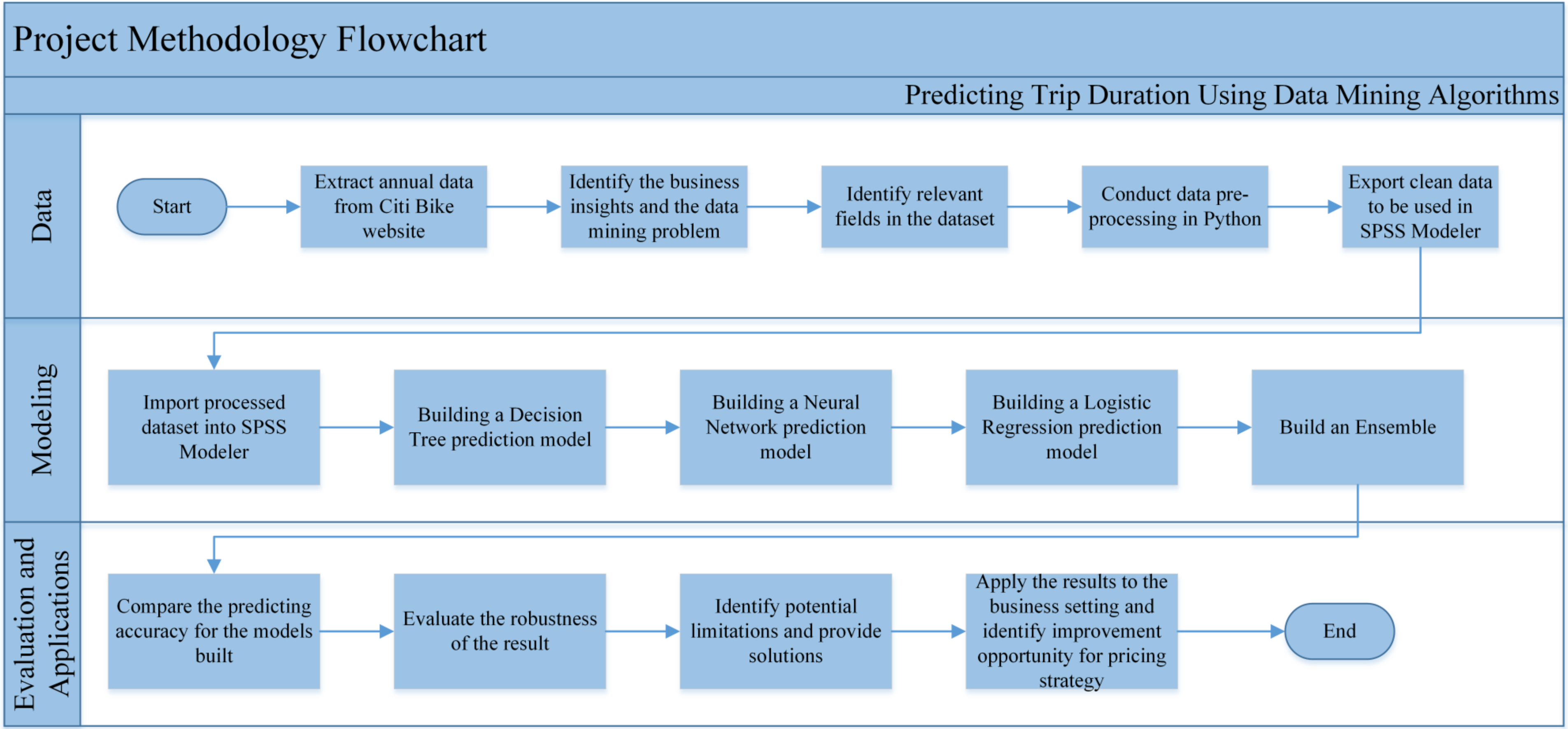
- **Utilize SPSS Feature Selection node to identify important features**
    - However we didn't take out "usertype" as suggested by SPSS Modeler because doing so reduces the accuracy of the model
- **Construct the following models to predict the duration window of the trip:**

Decision Tree	**Neural Network**
Logistic Regression	**Ensemble**





Methodology – *Overall Flow of the Project*



Results – Decision Tree

- Predicting **61.07%** of the testing data correctly

```
usertype = Subscriber [ Mode: <5 mins ]
    start station name in [ "Exchange Place" "Grove St PATH" "Hamilton Park" "Harborside" "Marin Light Rail" "Morris Canal" "Newark Ave" ] [ Mode: <5 mins ] ⇨ <5 mins
    start station name in [ "Newport PATH" ] [ Mode: 5 - 10 mins ]
        time in [ "afternoon" ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
        time in [ "evening" ] [ Mode: <5 mins ]
            agegroup = adult [ Mode: 5 - 10 mins ]
                gender in [ 0 1 ] [ Mode: <5 mins ] ⇨ <5 mins
                gender in [ 2 ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
            agegroup = senior [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
            agegroup = young adult [ Mode: <5 mins ] ⇨ <5 mins
            agegroup = youth [ Mode: <5 mins ] ⇨ <5 mins
        time in [ "morning" "night" ] [ Mode: <5 mins ] ⇨ <5 mins
    start station name in [ "Newport Pkwy" ] [ Mode: 5 - 10 mins ]
        agegroup in [ "adult" ] [ Mode: <5 mins ]
            gender in [ 0 2 ] [ Mode: <5 mins ] ⇨ <5 mins
            gender in [ 1 ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
        agegroup in [ "senior" "young adult" ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
        agegroup in [ "youth" ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
    start station name in [ "Sip Ave" ] [ Mode: 5 - 10 mins ]
        time = afternoon [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
        time = evening [ Mode: <5 mins ]
            gender = 0 [ Mode: <5 mins ] ⇨ <5 mins
            gender = 1 [ Mode: <5 mins ]
                agegroup in [ "adult" "young adult" "youth" ] [ Mode: <5 mins ] ⇨ <5 mins
                agegroup in [ "senior" ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
            gender = 2 [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
        time = morning [ Mode: 5 - 10 mins ]
            gender in [ 0 2 ] [ Mode: <5 mins ] ⇨ <5 mins
            gender in [ 1 ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
        time = night [ Mode: <5 mins ]
            agegroup = adult [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
            agegroup = senior [ Mode: 10 - 15 mins ] ⇨ 10 - 15 mins
            agegroup = young adult [ Mode: <5 mins ]
                gender in [ 0 1 ] [ Mode: <5 mins ] ⇨ <5 mins
                gender in [ 2 ] [ Mode: 5 - 10 mins ] ⇨ 5 - 10 mins
            agegroup = youth [ Mode: <5 mins ] ⇨ <5 mins
usertype = Customer [ Mode: >15 mins ] ⇨ >15 mins
```

Results for output field duration_window

Comparing $C-duration_window with duration_window

'Partition'	1_Training		2_Testing		3_Validation	
Correct	60,889	61.39%	30,181	61.07%	10,265	61.1%
Wrong	38,292	38.61%	19,240	38.93%	6,534	38.9%
Total	99,181		49,421		16,799	

Coincidence Matrix for $C-duration_window (rows show actuals)

'Partition' = 1_Training	10 - 15 mins	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	30	1,831	4,329	860
5 - 10 mins	10	6,495	18,531	1,096
<5 mins	2	4,572	51,541	585
>15 mins	1	1,624	4,851	2,823

'Partition' = 2_Testing	10 - 15 mins	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	12	886	2,218	376
5 - 10 mins	9	3,156	9,273	528
<5 mins	1	2,309	25,647	272
>15 mins	2	814	2,552	1,366

'Partition' = 3_Validation	10 - 15 mins	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	6	314	726	153
5 - 10 mins	1	1,087	3,246	159
<5 mins	0	754	8,709	103
>15 mins	0	246	832	463

Results – Neural Network

- Network generated for classification

Classification for duration_window

Overall Percent Correct = 60.7%

Observed	Predicted			
	10 – 15 mins	5 – 10 mins	<5 mins	>15 mins
10 – 15 mins	0.0%	21.5%	66.7%	11.8%
5 – 10 mins	0.0%	20.6%	75.3%	4.1%
<5 mins	0.0%	7.2%	91.8%	1.0%
>15 mins	0.0%	13.9%	56.1%	29.9%

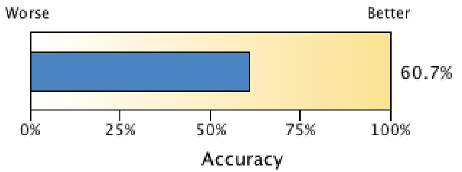
Row Percent
100.00
80.00
60.00
40.00
20.00
0.00

Results – Neural Network

- Predicting **60.34%** of data correctly for testing data

Model Summary

Target	duration_window
Model	Multilayer Perceptron
Stopping Rule Used	Minimum relative change in error achieved
Hidden Layer 1 Neurons	6



Results for output field duration_window

Comparing $N-duration_window with duration_window

'Partition'	1_Training		2_Testing		3_Validation	
Correct	60,212	60.71%	29,821	60.34%	10,203	60.74%
Wrong	38,969	39.29%	19,600	39.66%	6,596	39.26%
Total	99,181		49,421		16,799	

Coincidence Matrix for $N-duration_window (rows show actuals)

'Partition' = 1_Training	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	1,514	4,705	831
5 - 10 mins	5,378	19,683	1,071
<5 mins	4,072	52,050	578
>15 mins	1,294	5,221	2,784

'Partition' = 2_Testing	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	721	2,402	369
5 - 10 mins	2,573	9,877	516
<5 mins	2,068	25,897	264
>15 mins	641	2,742	1,351

'Partition' = 3_Validation	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	244	803	152
5 - 10 mins	913	3,423	157
<5 mins	638	8,829	99
>15 mins	199	881	461

Results – Logistic Regression

- Predicting **60.07%** of data correctly for testing data

Results for output field duration_window

Comparing $L-duration_window with duration_window

'Partition'	1_Training		2_Testing		3_Validation	
Correct	59,866	60.36%	29,685	60.07%	10,158	60.47%
Wrong	39,315	39.64%	19,736	39.93%	6,641	39.53%
Total	99,181		49,421		16,799	

Coincidence Matrix for $L-duration_window (rows show actuals)

'Partition' = 1_Training	10 - 15 mins	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	1	1,546	4,690	813
5 - 10 mins	2	5,415	19,672	1,043
<5 mins	0	4,437	51,696	567
>15 mins	8	1,354	5,183	2,754

'Partition' = 2_Testing	10 - 15 mins	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	0	744	2,385	363
5 - 10 mins	1	2,593	9,865	507
<5 mins	0	2,208	25,758	263
>15 mins	1	713	2,686	1,334

'Partition' = 3_Validation	10 - 15 mins	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	0	255	792	152
5 - 10 mins	1	931	3,411	150
<5 mins	0	696	8,770	100
>15 mins	1	208	875	457

Results – Ensemble

- Predicting **60.43%** of data correctly for testing data

- Use confidence weighted voting method

Results for output field duration_window

Comparing $XS-duration_window with duration_window

'Partition'	1_Training		2_Testing		3_Validation	
Correct	60,274	60.77%	29,865	60.43%	10,217	60.82%
Wrong	38,907	39.23%	19,556	39.57%	6,582	39.18%
Total	99,181		49,421		16,799	

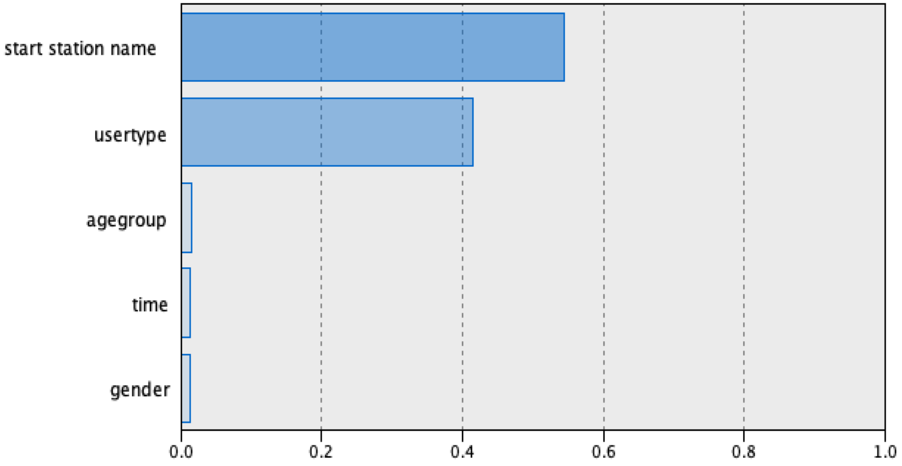
Coincidence Matrix for $XS-duration_window (rows show actuals)

'Partition' = 1_Training	5 - 10 mins	<5 mins	>15 mins
10 - 15 mins	1,534	4,671	845
5 - 10 mins	5,597	19,468	1,067
<5 mins	4,246	51,875	579
>15 mins	1,307	5,190	2,802
'Partition' = 2_Testing	**5 - 10 mins**	**<5 mins**	**>15 mins**
10 - 15 mins	729	2,392	371
5 - 10 mins	2,687	9,759	520
<5 mins	2,142	25,817	270
>15 mins	664	2,709	1,361
'Partition' = 3_Validation	**5 - 10 mins**	**<5 mins**	**>15 mins**
10 - 15 mins	248	799	152
5 - 10 mins	949	3,388	156
<5 mins	658	8,806	102
>15 mins	203	876	462

Results

Decision tree outperforms all the other models in terms of **predicting the testing data correctly (61.07%)**

Most Important Factor:
Start Station



Predictor Importance

<u>Rule 1</u>
User type: Subscriber
Start Station: “Newport PATH”
Time of day: evening
Age Group: adult
Gender: 2
mode: 5 - 10 min
(210; 0.476)

<u>Rule 2</u>
User type: Subscriber
Start Station: “Exchange Place”, “Grov St PATH”, “Hamilton Park”, “Harborside”, “Marin Light Rail”, “Morris Canal”, “Newark Ave”
mode: < 5 min
(69,558; 0.671)

<u>Rule 3</u>
User type: Customer

mode: > 15 min
(5,364; 0.526)

Results – Limitation and Solution

Limitation: The data is not evenly distributed for the four trip duration range – “<5 mins” may be given more weights during the learning process

Reasons

- Relatively shorter distance between stations in Jersey city
- Re-docking errors

Solution: conduct k-fold cross validation for training data, however, it may incur significant computational cost

```
In [29]: duration_class

Out[29]: duration_window
10 - 15 mins     11741
5 - 10 mins      43591
<5 mins          94495
>15 mins         15574
dtype: int64
```

duration_window		
10 - 15 mins	11741	7.10%
5 - 10 mins	43591	26.35%
<5 mins	94495	57.13%
>15 mins	15574	9.42%

Conclusion

Predicting **trip duration range** is beneficial for Citi Bike to improve dynamic pricing strategy

- Implement flat-rate pricing during peak periods, similar to Lyft.
  - ex: Evening commuters, charge $3 premium for subscribers, $5 for casual customers.
- Premium for certain stations (Rule 2)
  - Ex: $1.50 surcharge
  - Need to dig deeper into more trends for these stations specifically, such as time of day.
- Reduce the “free” period for subscribers to fall closer in-line with the average trip duration.

Decision Tree as the prediction model because of:

- The highest **prediction accuracy** (61.07%) in testing data
- **Readiness** of explaining the results and rules to the management of Citi Bike
- Large dataset allows post-pruning hence **reduces overfitting problem**

Future Directions

- Explore more approaches to slice and dice the dataset for more accurate predictions
- Expand the size of the dataset to include records from a larger time span and wider stations