Analyzing Citi Bike through Machine Learning

Sri Manda, Yusuf Senturk, and Max Wilde

Introduction

The goal of this project was to analyze Citi Bike data using Logistic Regression and Decision tree in hopes of better understanding the users. By better understanding the needs and uses for customers versus subscribers, the Citi Bike business could be better tailored to suit the needs of these individuals.

The Task

Citi Bike is a prominent bike sharing system in New York City

The privately owned operation has over 1 million riders in an average month and over 2 million riders during the Spring and Summer

Users are able to subscribe to the service for $169 a month or pay for a variety of single ride, daily, or 3 day pass options

Using data that is uploaded from Citi Bike from March 2019, we wanted to see if we could use machine learning to identify subscribers versus customers

The Data

Citi Bike uploads a monthly CSV to https://www.citibikenyc.com/system-data that includes various information on individual rides taken over the course of the month

This data includes: Trip Duration, Start Time, Stop Time, Start Station, Stop Station, Station ID, Station Lat/Long, Bike ID, User Type, Gender, and Year of Birth

In order to work with this data we converted the trip duration to minutes and changed gender to a binary numeric value

```
print(len(df))
```

```
2092573
```

```
df.head()
```

tripduration	starttime	stoptime	start station id	start station name	start station latitude	start station longitude	end station id	end station name	end station latitude	end station longitude	bikeid	usertype	birth year	gender
527	2019-10-01 00:00:05.6180	2019-10-01 00:08:52.9430	3746	6 Ave & Broome St	40.724308	-74.004730	223	W 13 St & 7 Ave	40.737815	-73.999947	41750	Subscriber	1993	1
174	2019-10-01 00:00:15.8750	2019-10-01 00:03:10.1680	3301	Columbus Ave & W 95 St	40.791956	-73.968087	3283	W 89 St & Columbus Ave	40.788221	-73.970416	18264	Subscriber	1992	1
759	2019-10-01 00:00:19.8240	2019-10-01 00:12:59.7070	161	LaGuardia Pl & W 3 St	40.729170	-73.998102	174	E 25 St & 1 Ave	40.738177	-73.977387	25525	Subscriber	1995	1
615	2019-10-01 00:00:21.0680	2019-10-01 00:10:36.6790	254	W 11 St & 6 Ave	40.735324	-73.998004	477	W 41 St & 8 Ave	40.756405	-73.990026	30186	Subscriber	1992	1
761	2019-10-01 00:00:26.3800	2019-10-01 00:13:08.3130	161	LaGuardia Pl & W 3 St	40.729170	-73.998102	174	E 25 St & 1 Ave	40.738177	-73.977387	25597	Subscriber	1992	1

Parameters

Number of rows : 2092573

Number of Columns : 14

Target Variable (Y) is UserType : Trying to predict if UserType is Subscriber or Customer

The simplest model tried is logistic regression with an accuracy score of 0.78
The best model tried is decision tree with an accuracy score of 0.88

ML Type - Classification, Two classes and Imbalanced

Logistic Regression

- Elimination of Columns
- Dummy Encoding
- Correlation Matrix
- Confusion Matrix
- Conclusion for Logistic Regression

```
data
```

	Trip_Duration	Birth_Year	Gender	Trip_Duration_Mins	Subscriber
0	527	1993	1	9	1
1	174	1992	1	3	1
2	759	1995	1	13	1
3	615	1992	1	10	1
4	761	1992	1	13	1
...	...	...	...	...	...
2092568	729	1995	1	12	1
2092569	645	1969	0	11	0
2092570	257	1985	1	4	1
2092571	466	1989	0	8	1
2092572	81	1990	1	1	1

2092573 rows × 5 columns

```
data.dtypes
```

```
Trip_Duration         int64
Birth_Year            int32
Gender                int64
Trip_Duration_Mins    int32
Subscriber            uint8
dtype: object
```

Correlations

```
data.corr()
```

	Trip_Duration	Birth_Year	Gender	Trip_Duration_Mins	Subscriber
Trip_Duration	1.000000	-0.006633	-0.015041	0.999997	-0.046817
Birth_Year	-0.006633	1.000000	0.171064	-0.006636	0.024576
Gender	-0.015041	0.171064	1.000000	-0.015041	0.283830
Trip_Duration_Mins	0.999997	-0.006636	-0.015041	1.000000	-0.046817
Subscriber	-0.046817	0.024576	0.283830	-0.046817	1.000000

- Through an unbalanced logistic regression, we were able to find a 88% match, while for a balanced regression the match decreased to 78%.

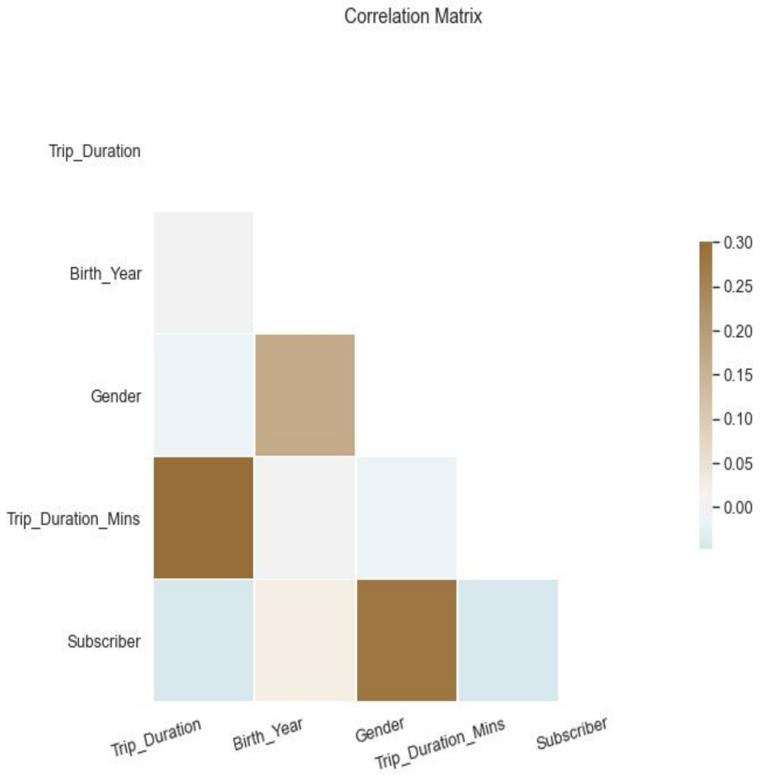
```
training_score = lg_model.score(X_train,y_train)
testing_score = lg_model.score(X_test,y_test)

print(f"Training Score: {training_score}")
print(f"Testing Score: {testing_score}")
```

```
Training Score: 0.8848294507110548
Testing Score: 0.8848959368739774
```

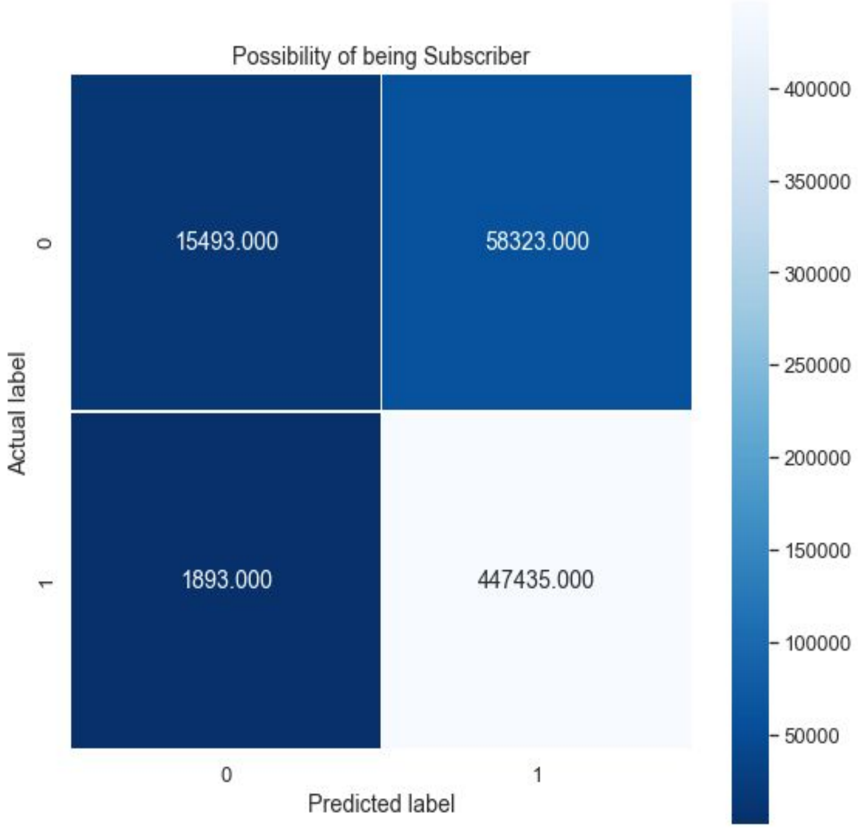
Prediction with Balanced Logistic Regression

```
Testing Score: 0.7779445047635068
```

	Feature	Coefficients
0	Trip_Duration	-0.000413
1	Birth_Year	0.000231
2	Gender	1.654900
3	Trip_Duration_Mins	0.002268



Result of Logistic Regression



Conclusion

Analysis of predictions:

- True Negatives = 15.493
- False Positive = 58.323
- True Positives = 447.435
- False Negatives = 1.893

True Negatives + True Positives = Model Performance

From the above analysis we found that the our model performance is 88.49 %.

Classification Model : Decision Tree

1. Prediction with Decision Tree (Train, Test, Split)
2. Prediction with Balanced Decision Tree
3. Accuracy Improvements by Pruning
4. Decision Tree Visualization

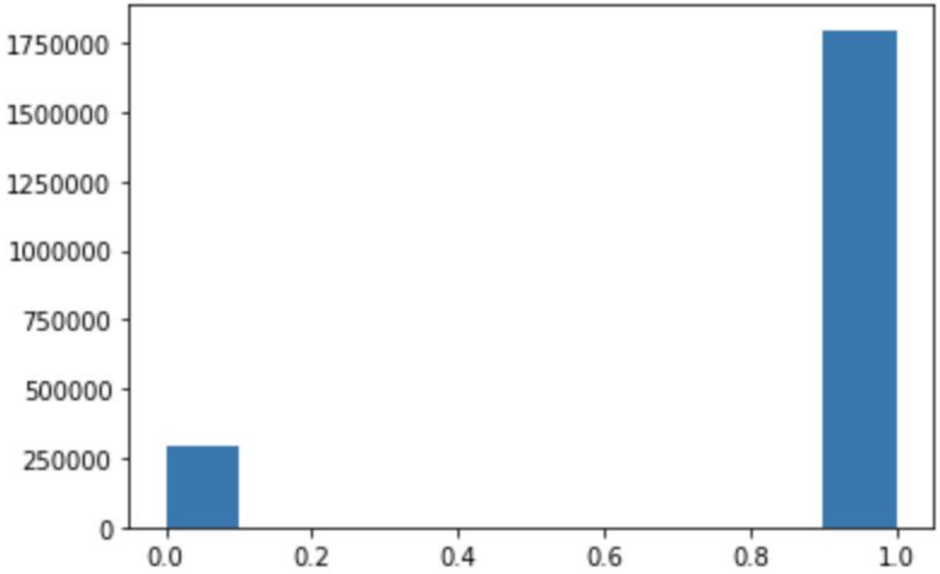
Feature Engineering

- Null Values, units conversion (trip time etc)
- One Hot Encoding Gender, since it's a categorical variable .get_dummies()
- One Hot Encoding User Type, same as above
- Imbalance for user-type, skewed towards subscriber

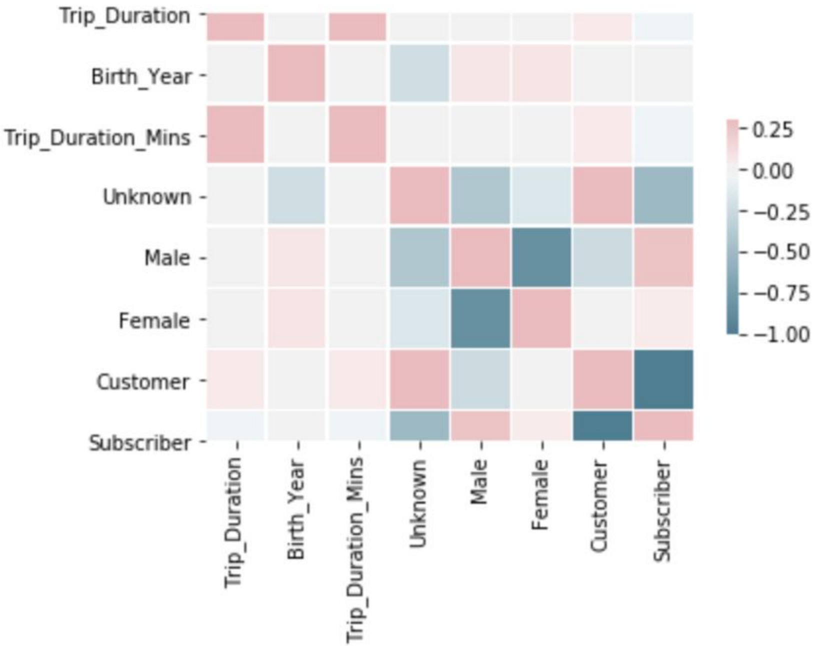
Trip_Duration	Start_Time	Stop_Time	Start_Station_Name	End_Station_Name	Bike_Id	User_Type	Birth_Year	Gender	Trip_Duration_mins
527	2019-10-01 00:00:05.6180	2019-10-01 00:08:52.9430	6 Ave & Broome St	W 13 St & 7 Ave	41750	Subscriber	1993	1	9.0
174	2019-10-01 00:00:15.8750	2019-10-01 00:03:10.1680	Columbus Ave & W 95 St	W 89 St & Columbus Ave	18264	Subscriber	1992	1	3.0
759	2019-10-01 00:00:19.8240	2019-10-01 00:12:59.7070	LaGuardia Pl & W 3 St	E 25 St & 1 Ave	25525	Subscriber	1995	1	13.0
615	2019-10-01 00:00:21.0680	2019-10-01 00:10:36.6790	W 11 St & 6 Ave	W 41 St & 8 Ave	30186	Subscriber	1992	1	10.0
761	2019-10-01 00:00:26.3800	2019-10-01 00:13:08.3130	LaGuardia Pl & W 3 St	E 25 St & 1 Ave	25597	Subscriber	1992	1	13.0

Attributes

Class Imbalance



Feature Selection



Check Correlations.¶

- Subscriber and being Male seem to be correlated.
- Stronger relationship between Male/Female/Unknown. This is expected since being one gender requires not being the other two.
- Trip Durations and Trip duration minutes are correlated.
- From the features that correlate, we have to pick just one

P value & Coefficient

P Value is the probability that there is no relationship between the variables in X (features) and Y (what we're trying to predict)

Coefficients

```
     coef
-----------
   0.2712
  -0.0001
   0.0002
   0.1915
```

The P > t column here is the p_value for that feature which is all 0.00

The industry/scientific standard is for the value to be less than 0.05 - or - less than 5% chance of the feature having no relationship with Y. This is ML speak is called being 'statistically significant'

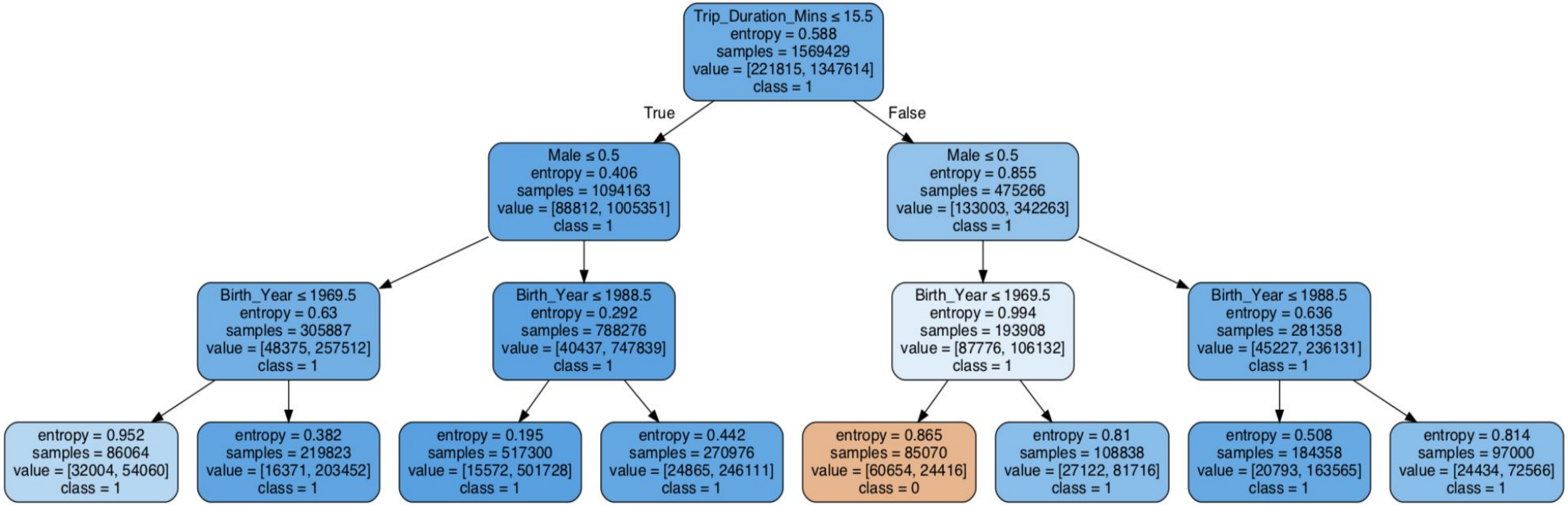
The Coef gives the strength of the relationship. The higher the coef the higher the relationship btw the feature and Y. Negative values translates to inverse relationship

In this case all features are significant / they have a relationship with Y

Selected Features

Trip_Duration_Mins	
Birth_Year	
Male	
Subscriber	Y

Decision Tree Visualization



Conclusion

1. Prediction with Decision Tree (Train, Test, Split)

   Accuracy: 0.9061615922193507
2. Prediction with Balanced Decision Tree

   Accuracy: 0.8109335097028734, F1 Score: 0.88
3. Accuracy Improvements by Pruning

   Accuracy: 0.882108941323995, F1 Score: 0.93

Decision Tree does much better in classifying with Unbalanced class