Contents

Problem Statement and Goal

Problem Statement

- Inconvenience to find and stop the bikes

  If customers are often inconvenienced, they will stop using the service because they go to the station to check out their bikes for their morning commute, but there are no bikes there.

- Increase conversion rate

  There are still many one-time customers. If divvy can carry out a strategic market campaign on the potential subscribers, it will be able to increase its conversion rate.

Goal

1. Predict how many bikes will be checked out from a particular station at a given time. If the number of bikes checked out can be known ahead of time, Divvy will know when and where to reload bikes so customers can continue to use the site.

2. Predict for the potential subscribers that Divvy may better target for a market campaign.

Data Overview

Data Source: City of Chicago Data portal

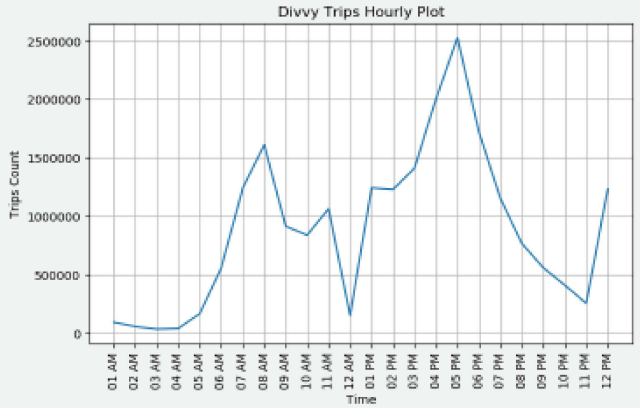
Data Introduction:

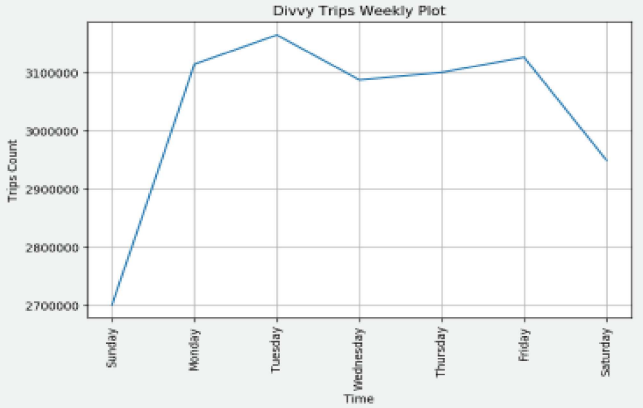
Divvy: includes the origin, destination, and timestamps for each trip. Trips using a subscriber pass will include some basic demographic data (gender and age) that is associated with the account.

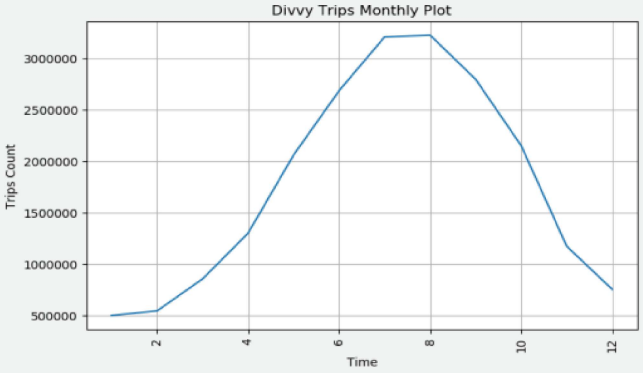
Weather: includes temperature, wind, rain, solar radiation, pressure, sensor info measured from sensors at beaches along Michigan Lake.

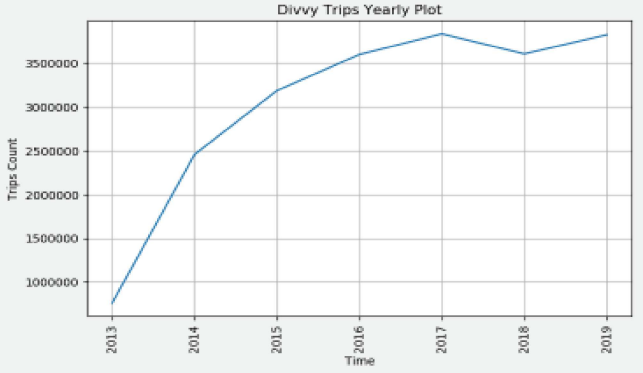
Dataset	Divvy Bike Trips Data	Weather Data
Size	5.13 GB	21.2 MB
Rows	20 M	141 K
Columns	18	18

EDA———Time series Plot

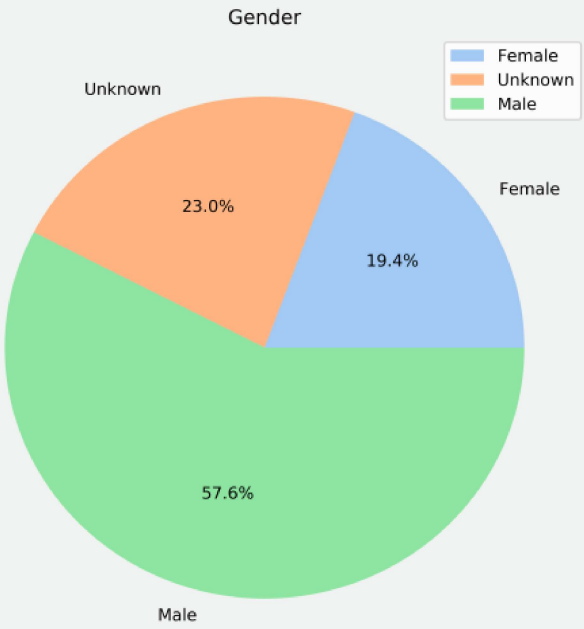


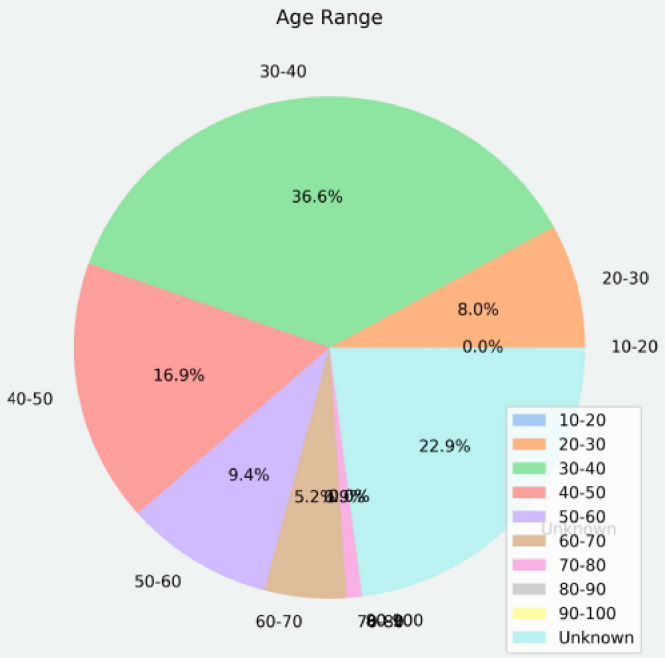


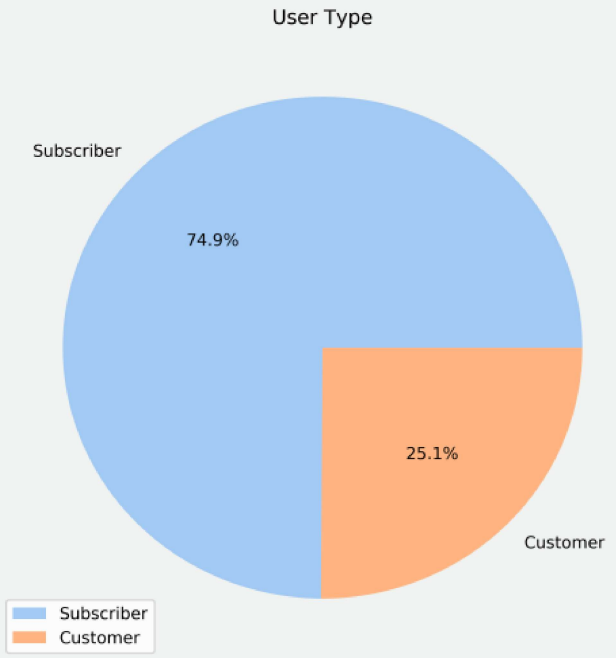




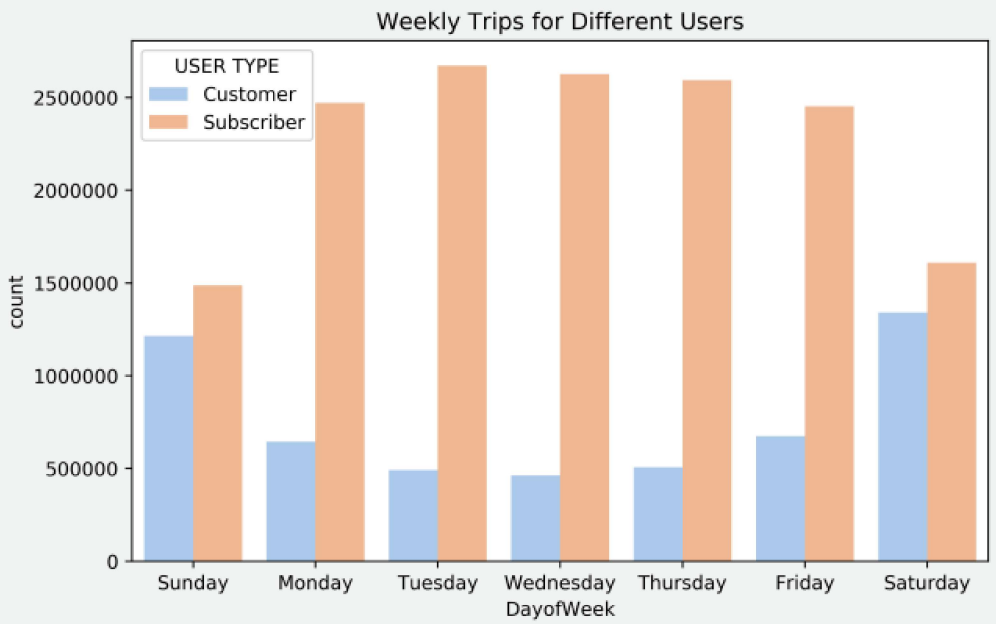
EDA———User Demographics

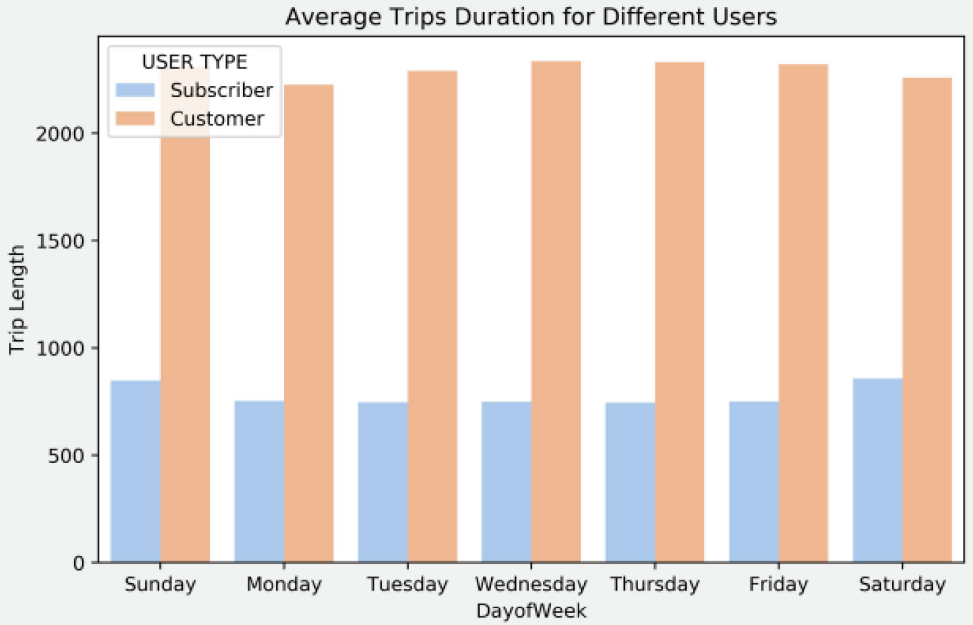




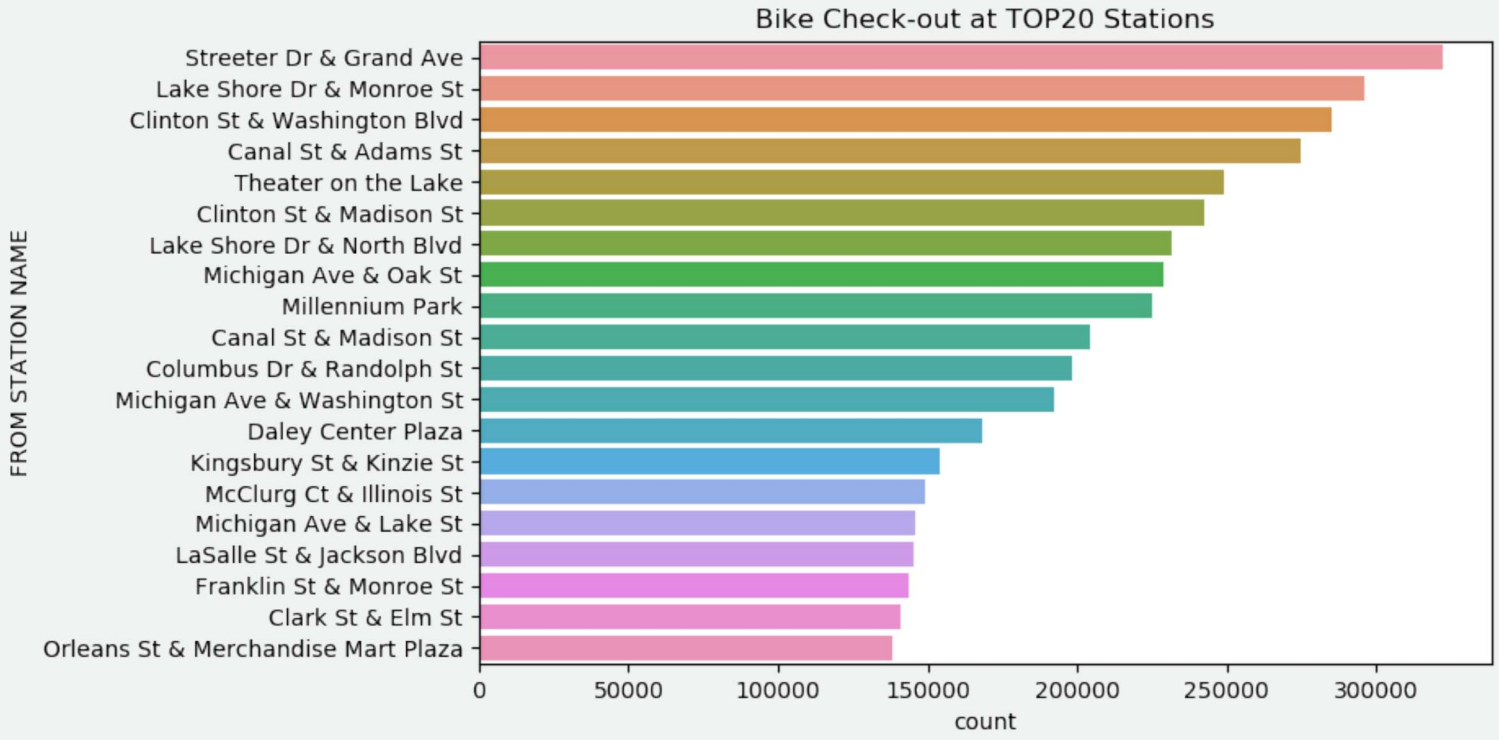


EDA———User Type Comparison





EDA———Popular Stations



Feature Engineering

- Drop Unnecessary Columns
  - Location features from **Divvy Dataset**
  - Measurement Device features from **Weather Dataset**
- Convert to appropriate Data types
  - Time column from String to Timestamp
  - Station ID from Int to String for one-hot encode
- Create new features
  - Year, month, day, day of week, hour from single time column
  - Age and age range from birth year column

- Drop duplicates in **Weather Dataset**
  - Data is measured from 3 stations, and some contains missing value
  - Ensure one record per day per hour using aggregation max()
  - Missing value reduced from 48149 to 180
- Handling Missing Values
  - Fill "Unknown" for missing gender
  - Fill 0 for missing age (age<10 or age>100 has "unknown" for age range)
  - Drop missing values in **Weather Dataset**
- Aggregate **Divvy Dataset**
  - Sum() bike check-out number at per station per day per hour level
- Merge Dataset
  - Join **divvy and weather dataset** on 'day' and 'hour'

- One-Hot Encoding
  - Regression: Station Name, Month, Day of Week, Hour
  - Classification: Station Name, Month, Day of Week, Hour, Age range, Gender
  - Consist of different **StringIndexer** and **OneHotEncoderEstimator** in a Pipeline

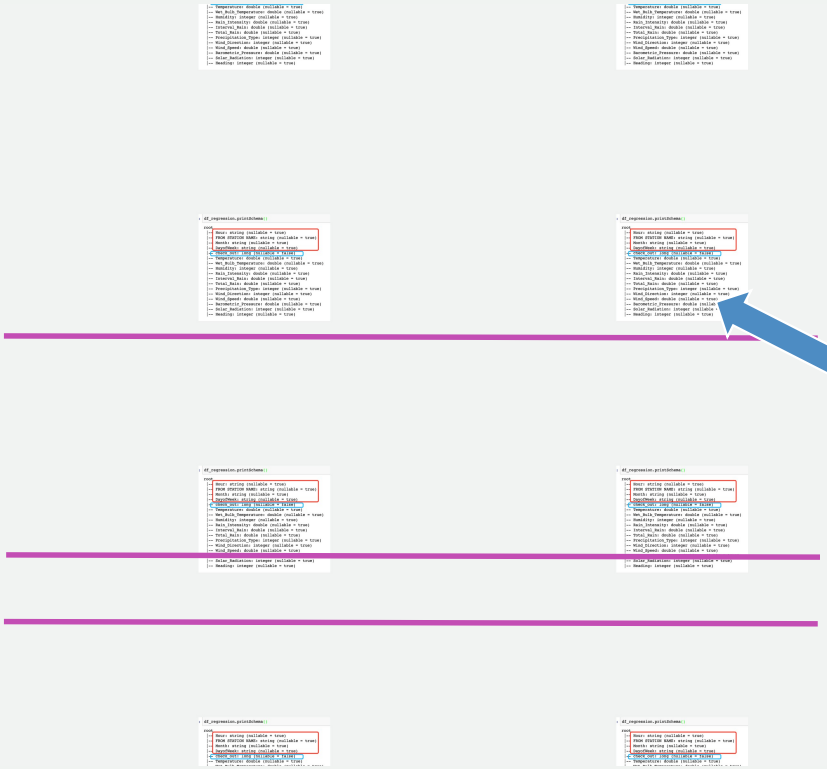
```
#convert relevant categorical into one hot encoded
indexer1 = StringIndexer(inputCol="FROM STATION NAME", outputCol="stationIdx").setHandleInvalid("skip")
indexer2 = StringIndexer(inputCol="Month", outputCol="monthIdx").setHandleInvalid("skip")
indexer3 = StringIndexer(inputCol="Hour", outputCol="hourIdx").setHandleInvalid("skip")
indexer4 = StringIndexer(inputCol="DayofWeek", outputCol="dayofweekIdx").setHandleInvalid("skip")
indexer5 = StringIndexer(inputCol="GENDER", outputCol="genderIdx").setHandleInvalid("skip")
indexer6 = StringIndexer(inputCol="Age_Range", outputCol="agerangeIdx").setHandleInvalid("skip")


#gather all indexers as inputs to the One Hot Encoder
inputs = [indexer1.getOutputCol(), indexer2.getOutputCol(), \
          indexer3.getOutputCol(), indexer4.getOutputCol(), \
          indexer5.getOutputCol(), indexer6.getOutputCol()]

#create the one hot encoder
encoder = OneHotEncoderEstimator(inputCols=inputs,  \
                                 outputCols=["stationVec", "monthVec", "hourVec", \
                                             "dayofweekVec","agerangeVec","genderVec"])

#run it through a pipeline
pipeline = Pipeline(stages=[indexer1,indexer2,indexer3,indexer4,indexer5,indexer6,encoder])
```

Regression Model



Features

1. **Dummy Variables**: Hour/ FROM STATION NAME/ Month/ DayofWeek
2. **Numeric Variables**: Temperature/ Wind_Speed/ Precipitation_Type

Target Variable
check_out

Train/Test Split
80% + 20%

Regression Model Evaluation

	RMSE	MSE	MAE	R2
Linear Regression	3.580	12.818	1.874	0.277
Ridge Regression	3.582	12.831	1.864	0.276
Random Forest Regression	3.828	14.65	2.015	0.173

Conclusion:

1. Linear and Ridge regression perform better with lower RMSE, MSE and MAE, compared with random forest.
2. But Regression techniques did not perform very well as due to low R2 scores.
   - The Ridge regression explains 27.6% of the variance
   - Meaning we have 27.6% of the information that we need to make an accurate prediction on the number of bikes checked out of any station at any given hour.

Classification Model

```
root
 |-- TRIP DURATION: integer (nullable = true)
 |-- FROM STATION ID: string (nullable = true)
 |-- FROM STATION NAME: string (nullable = true)
 |-- TO STATION ID: string (nullable = true)
 |-- TO STATION NAME: string (nullable = true)
 |-- USER TYPE: string (nullable = true)
 |-- GENDER: string (nullable = false)
 |-- Month: string (nullable = true)
 |-- DayofWeek: string (nullable = true)
 |-- Hour: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Age_Range: string (nullable = true)
```

Features

FROM STATION NAME/
Month/ DayofWeek / Hour /
Gender / Age_Range

Target Variable

USER TYPE (subscriber or customer)

Train/Test Split

80% + 20%

Classification Model Evaluation

Conclusion:

1. Logistic Regression and Naïve perform pretty the same in all three metrics: Accuracy, F1 score, and AUC

2. Random forest performs a bit worse which could be the reason it's not well tuned. We had run it for hours!

3. For saving time cost and having better Interpretability, we decide to use logistic regression as our final classification model.

Limitations & Future Steps

Limitations

- Did not consider the external factors including environment changes, like disease (COVID 19), disaster (flood, sandstorm & hurricane).
- Not considering changes of users' behavior.
- Regression model needs to be improved by selecting or adding more features.

Future Steps

Data	End to end update of event tracking to complete data analysis process
Model	Add more weather-related and user-related features


Thank You !