# Contents

Section 1: About the project and Contributors ..... 1
Section 2: Dataset we work on ..... 1
Section 3 & 4: Analysis and results ..... 1
Q1 Analysis ..... 1
Q1 Results ..... 3
Q2 Analysis ..... 5
Q2 Results ..... 6
Q3 Analysis ..... 7
Q3 Results ..... 8

## Section 1: About the project and Contributors

We work on path 1: Bike traffic

Authors:

Jimmy Jiang jiang679@purdue.edu

Ziyue He he532@purdue.edu

## Section 2: Dataset we work on

Our work is based on the dataset “NYC_Bicycle_Counts_2016_Corrected.csv’ and adjust the dataset slightly to accommodate our analysis on data in three questions below.

## Section 3 & 4: Analysis and results

### Q1 Analysis

For question 1, the dataset we are working with are: Day, High Temp, Low Temp, Precipitation, Brooklyn Bridge, Manhattan Bridge, Williamsburg Bridge, Queensboro Bridge.

The question is asking for choosing the three bridges to install sensors on. It means that in

order to obtain the overall traffic data, traffic on one bridge will need to be predicted by the data we have. There's no difference between predicting traffic on one bridge and predicting the overall traffic. We can obtain one another by simply doing addition or subtraction.

We think that multiple factors can affect the traffic. The first factor is the most obvious one, which is traffic on other bridges. If the traffic increase on one bridge, it is likely that traffic on other bridges may decrease. Of course, this is not always true. If there's a traffic jam on bridge, people may tend to choose its alternative. It doesn't matter too much for what assumption we're making right here as the relationship will be revealed by our code. Another factor that may affect the traffic on one bridge is the weather. When it's raining, it will be less likely for people to ride a bike. Moreover, temperature may also affect people's behavior. When it's too hot or too cold outside, people may also not ride a bike. The day in the week may also affect the bike traffic. It may be more likely for people to bike during weekend instead of weekdays. The time in the year may also affect people's behavior. It is possible that people may ride a bike more often in spring instead of winter. However, the data provided didn't record the data for one whole year. Thereby, the time in the year will not be in our consideration.

Therefore, the model we build should consider all these factors. In order to build up the model, we will need to preprocess the data. First, there is a comma for numbers having more than three digits, which may cause some problems on python. We choose to eliminate the comma on Excel by choosing another format for it. After that, we still have to transform day in a week to number. We define Monday to be 1, Tuesday to be 2 and so on. We noticed that precipitation is having one data in this form "0.47 (S)". We choose to ignore the (S), which probably stands for snow. We did this by using regular expression when parsing, so even there are lots of data having the similar form, we are confident that python will help us make the format correct. We also noticed the "T" in precipitation. According to the information online, "T" stands for a very small amount that is very hard to measure with existing unit. Therefore, we replace "T" with zero.

We plan to build up four linear regression models, each treating traffic on one bridge as target variable and all other data as explanatory variables. We will compare the MSE calculated from the four models and choose the model that has the lowest MSE. After finding the model that has lowest MSE, we will choose not to install the sensor on the bridge that is treated as target

variable in this case. This is because we are confident that the missing information on one bridge can be accurately predicted by processing the sensors data with the model we have. We make the assumption that the sensors being installed will collect the same data as shown in the csv file provided, which means data like temperature, precipitation will also be provided. It is true that we can only use the traffic information without considering other factors like weather to build up this model. However, we think that the model will not be as accurate as the one considering multiple factors. Indeed, if the sensors being installed on the bridge will not collect the data like weather information, the less accurate model will be our only option. For building up the model, I choose to use the similar approach as in hw5 question 2. We will do cross validation by splitting the data we have. After that, we'll do normalization and test the MSE with different lambda value. After all, we'll be able to get the MSE value for each model.

## Q1 Results

After running the code Q1.py, I got plot of MSE versus lambda for each model, which is shown below from figure 1.0 to 1.3.

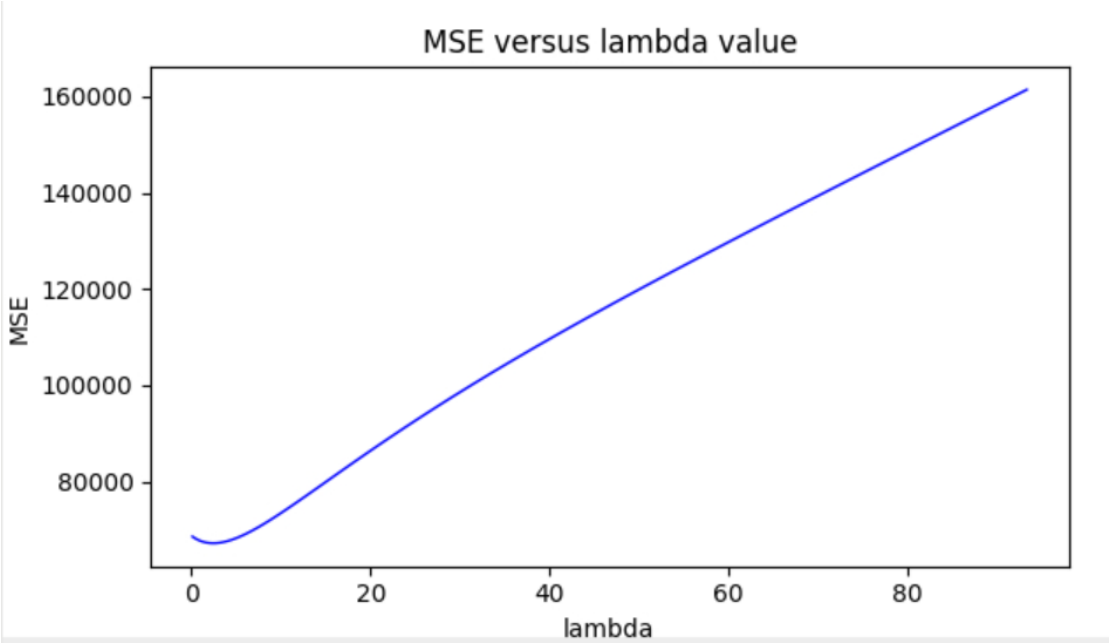



Figure 1.0: MSE versus lambda value plot for treating Queensboro Bridge as target variable

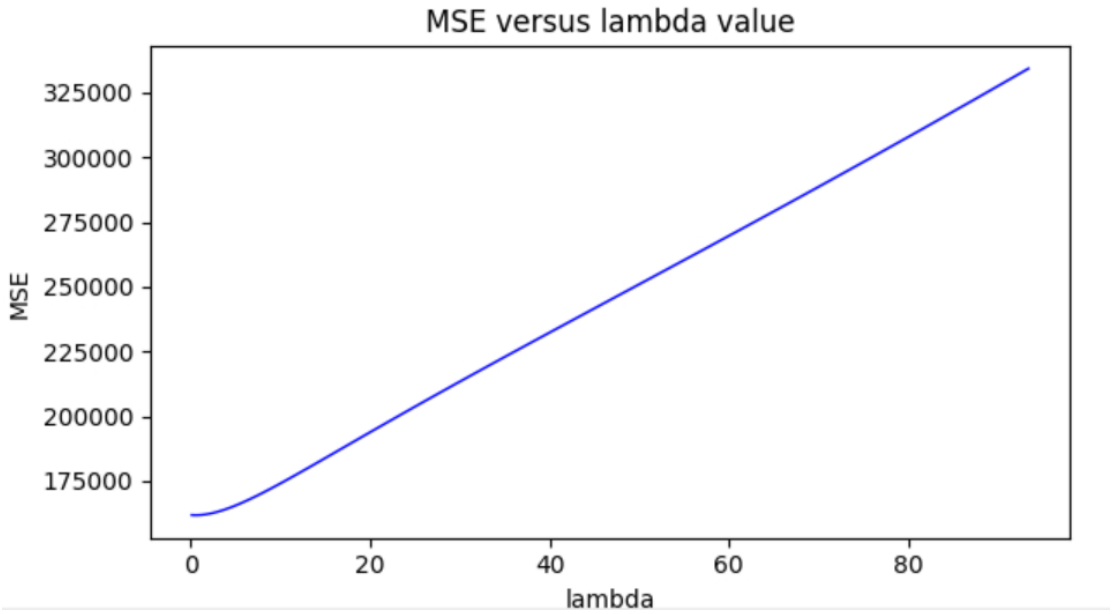



Figure 1.1: MSE versus lambda value plot for treating Williamsburg Bridge as target variable

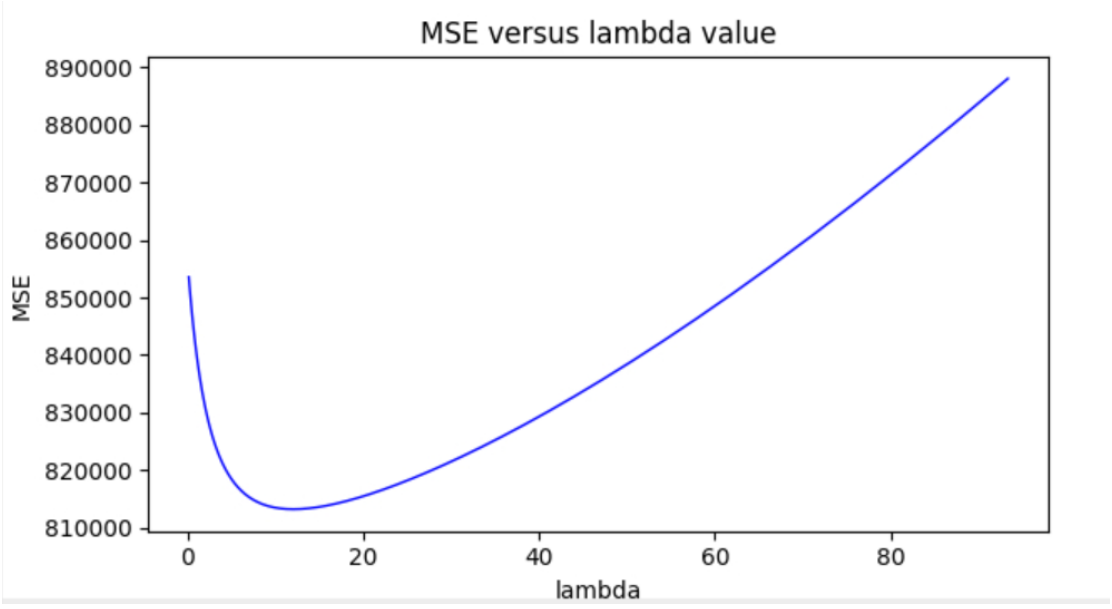



Figure 1.2: MSE versus lambda value plot for treating Manhattan Bridge as target variable

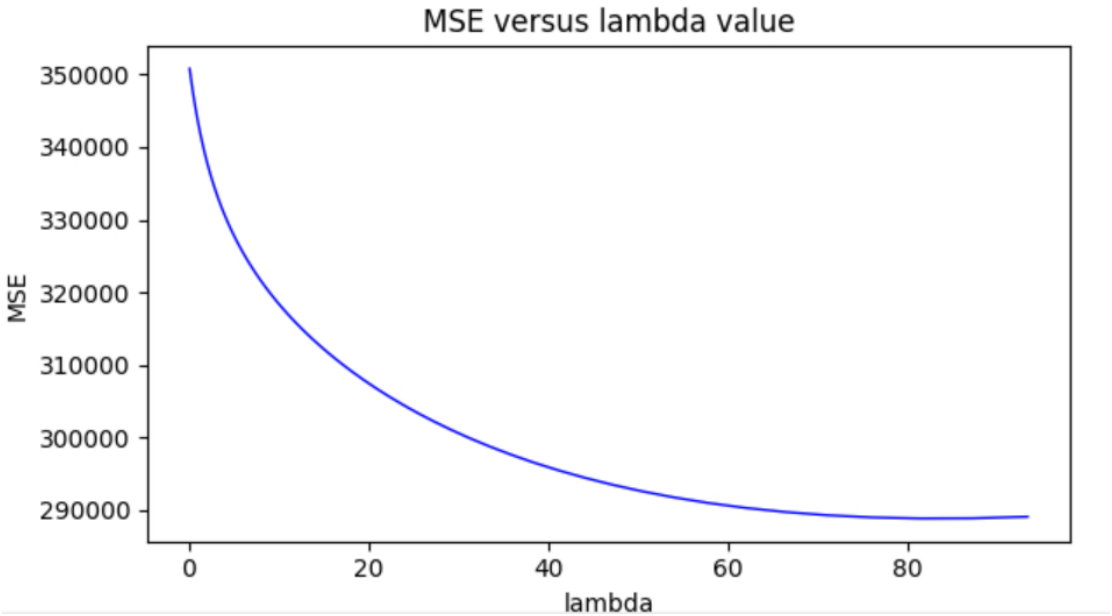



Figure 1.3: MSE versus lambda value plot for treating Brooklyn Bridge as target variable

I got the result as below in figure 1.4.

```
The MSE for model treating Queensboro Bridge as target variable is 67198.98837224378. The best lambda tested for it is 2.48908
4291235577
The MSE for model treating Williamsburg Bridge as target variable is 161841.55066608064. The best lambda tested for it is 0.55
28079211966477
The MSE for model treating Manhattan Bridge as target variable is 813232.3664305279. The best lambda tested for it is 12.00073
7073062885
The MSE for model treating Brooklyn Bridge as target variable is 288835.30549984117. The best lambda tested for it is 81.45000
00870929
```

Figure 1.4: Python code result

We can easily tell that the model treating Queensboro Bridge data as target variable will get the lowest MSE among the four models, which is 67198.988. Therefore, I choose not to install sensor on Queensboro Bridge, which means that we'll install sensors on Brooklyn Bridge, Manhattan Bridge, and Williamsburg Bridge.

## Q2 Analysis

In question 2, we are asked whether we can use weather forecast to predict number of bicyclists. The weather forecast includes information like high temperature, low temperature, precipitation. We need to employ a method which takes in several variables and returns a prediction based on all of them. Thus, we use multivariable linear regression to analyze the data. Admittedly, there will

certainly be other aspects of weather condition having effects on the number of riders on the bridges. But since we are only given data on high temperature, low temperature, and precipitation, we will only consider these three as independent variables.

We expect to see a good model explicitly showing linear relationship between three independent variables mentioned above and riders on that day. The most ideal scenario is the model has large R-squared value and small MSE. If the model we get does not satisfy our expectation, we would improve the algorithm by normalizing/scaling raw data first and continue with regression.

## Q2 Results

The linear regression on original data:

$$y = 390.92 * HighTemp - 162.32 * LowTemp - 7951.49 * Precipitation + 178.2$$

We can use this model to predict the number of bicyclists in a certain day.

The regression results table is shown below. For this model, R-squared is 0.499, which is not very high.

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Total | R-squared: | 0.499 |
| Model: | OLS | Adj. R-squared: | 0.492 |
| Method: | Least Squares | F-statistic: | 69.85 |
| Date: | Sun, 02 Aug 2020 | Prob (F-statistic): | 2.28e-31 |
| Time: | 19:00:06 | Log-Likelihood: | -2079.9 |
| No. Observations: | 214 | AIC: | 4168. |
| Df Residuals: | 210 | BIC: | 4181. |
| Df Model: | 3 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 178.2009 | 1706.345 | 0.104 | 0.917 | -3185.560 | 3541.962 |
| High Temp (°F) | 390.9183 | 57.206 | 6.834 | 0.000 | 278.147 | 503.690 |
| Low Temp (°F) | -162.3201 | 61.461 | -2.641 | 0.009 | -283.480 | -41.160 |
| Precipitation | -7951.4864 | 1099.722 | -7.230 | 0.000 | -1.01e+04 | -5783.576 |

| | | | |
|---|---|---|---|
| Omnibus: | 10.936 | Durbin-Watson: | 1.101 |
| Prob(Omnibus): | 0.004 | Jarque-Bera (JB): | 6.251 |
| Skew: | -0.239 | Prob(JB): | 0.0439 |
| Kurtosis: | 2.313 | Cond. No. | 610. |

Next, we try to normalize the data before doing regression to improve the precision.

The model we get is

$$y = 4892 * NormalHighTemp - 1889 * NoralLowTemp - 2062 * NormalPrecipitation + 18544$$

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Total   R-squared:                       0.499
Model:                            OLS   Adj. R-squared:                  0.492
Method:                 Least Squares   F-statistic:                     69.85
Date:                Sun, 02 Aug 2020   Prob (F-statistic):           2.28e-31
Time:                        19:00:06   Log-Likelihood:                -2079.9
No. Observations:                 214   AIC:                             4168.
Df Residuals:                     210   BIC:                             4181.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.854e+04    277.733     66.771      0.000     1.8e+04    1.91e+04
x1          4892.7615    715.995      6.834      0.000    3481.303    6304.220
x2         -1889.9358    715.607     -2.641      0.009   -3300.630    -479.241
x3         -2062.2219    285.214     -7.230      0.000   -2624.470   -1499.973
==============================================================================
Omnibus:                       10.936   Durbin-Watson:                   1.101
Prob(Omnibus):                  0.004   Jarque-Bera (JB):                6.251
Skew:                          -0.239   Prob(JB):                       0.0439
Kurtosis:                       2.313   Cond. No.                         4.96
==============================================================================
```

Through calculation, we find R-squared value didn't increase after we normalize data. Referring to https://stats.stackexchange.com/questions/29781/when-conducting-multiple-regression-when-should-you-center-your-predictor-varia, we find out normalizing data doesn't help decrease MSE or increase R_squared value. Though our model is not close to perfection, it still provides us with a reasonable prediction of bicyclists in a certain day. We believe the error could be caused by other aspects of weather conditions (such as wind) which are not included in this dataset.

## Q3 Analysis

In problem 3, our goal is to predict whether it is raining based on the given data of number of bicyclists.

We choose logistic analysis to predict the possibly of raining on a certain day. Mathematically, logistic analysis takes in an independent variable and returns an indictor of either 0 or 1(0 means fail while 1 means success). This character of logistic analysis confirms with our goal-to get a 0 or 1 value based on input bicyclists. Linear regression method is also discussed in brainstorming process but eliminated finally. This is because outliers can have much larger influence on the precision of linear model than that on logistic model. The natural log term in logistic model helps buffer the influence of outliers.

Our approach to this problem is to split and whole dataset into two parts: one training set and

a test set. We train and logistic model with training set and use test set to determine whether the trained model is good enough or not. In practice, we randomly choose 92% of data as training set and test on other 8% (18) testers.

## Q3 Results

First, we process the data so that if precipitation is larger than 0, precipitation = 1

Then, we conduct logistic regression on the processed datasets.

The logistic model we get from the logistic regression is:

$$y = \frac{1}{1 + e^{(0.979 + 1.165x)}}$$

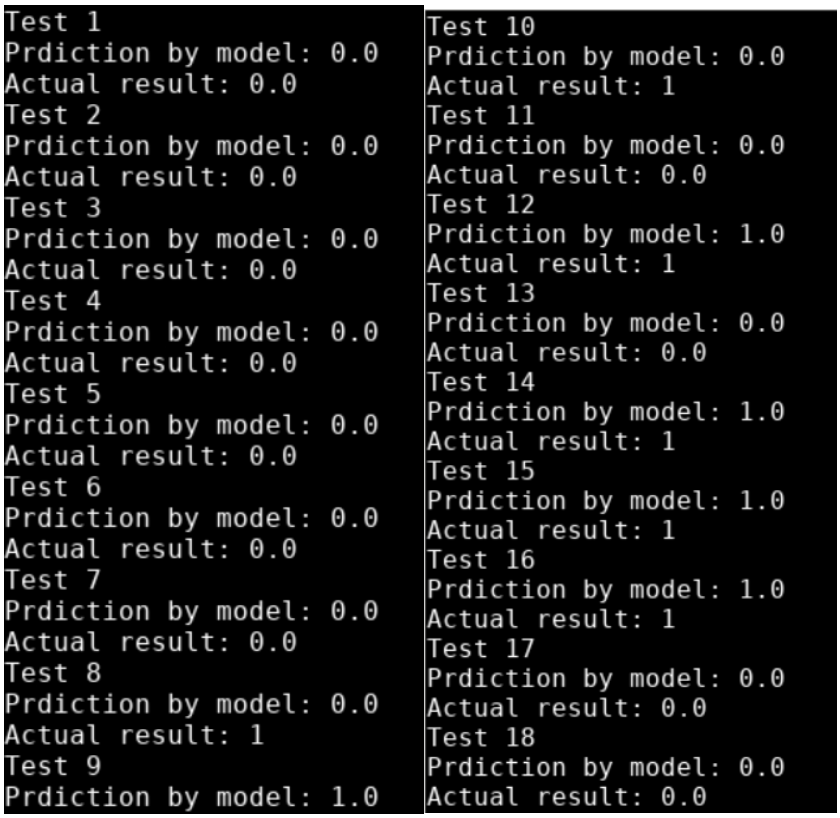
Y represents the possibility of raining and x represents total number of bikes on the bridge.

Then, we look at the predicted raining results of 18 test sets given by our model and compare them with the actual raining result of that day.

18 test sets which are randomly chosen in the train_test_split process:

| | Total |
|---|---|
| 21 | 17837 |
| 54 | 26978 |
| 84 | 24440 |
| 102 | 28437 |
| 26 | 19914 |
| 202 | 20403 |
| 208 | 16948 |
| 28 | 14954 |
| 6 | 9596 |
| 161 | 20850 |
| 188 | 24594 |
| 25 | 13005 |
| 74 | 26368 |
| 142 | 13771 |
| 42 | 12202 |
| 37 | 11254 |
| 122 | 18780 |
| 198 | 14356 |

Comparison between the predicted result and actual result:





In 18 test cases, our logistic model successfully predicts the outcome of 16 of them. The accuracy rate, which is also the score returned by calling LogisticRegression.score(), is 0.8888. This score is high and indicates our model is doing a great job on predicting whether it is raining based on the number of bicyclists on the bridges.