Names: Seth Graber and Henry Katis
Instructor: Professor Philip Pare
GitHub Usernames: SG295 and Katis009
Purdue Usernames: graber29 and hkatis

Mini Project Write Up: Path 1, Bike Traffic

Introduction:

The main premise of the project is to analyze a data set and answer various statistical questions revolving around said data set. For this project particularly, the dataset is of bike traffic across different bridges in New York City, as well as the corresponding date, time, and weather data for each days' individual traffic. Using the various topics taught throughout the semester, the group aimed to answer three questions about the dataset. First, if only given three sensors for the four bridges, which bridges should receive sensors to get the best prediction of overall traffic? Second, using the next day's weather forecast (low/high temperature and precipitation) can the total number of bicyclists for that day be predicted? Lastly, based on the number of bicyclists on the bridges, can the day of the week be predicted? This group's responses can be seen below, with the analysis mainly revolving around regression and classifier models.

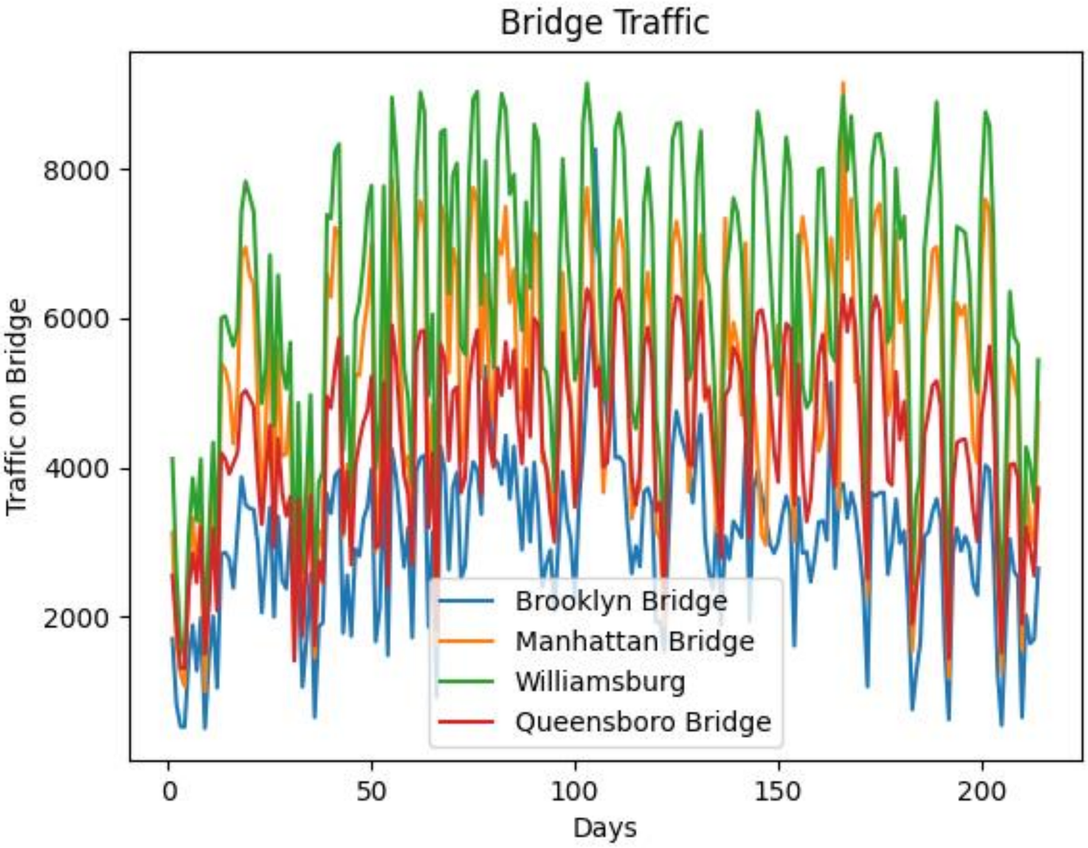


Figure 1. Bicycle Traffic Across Each of the Four Bridges

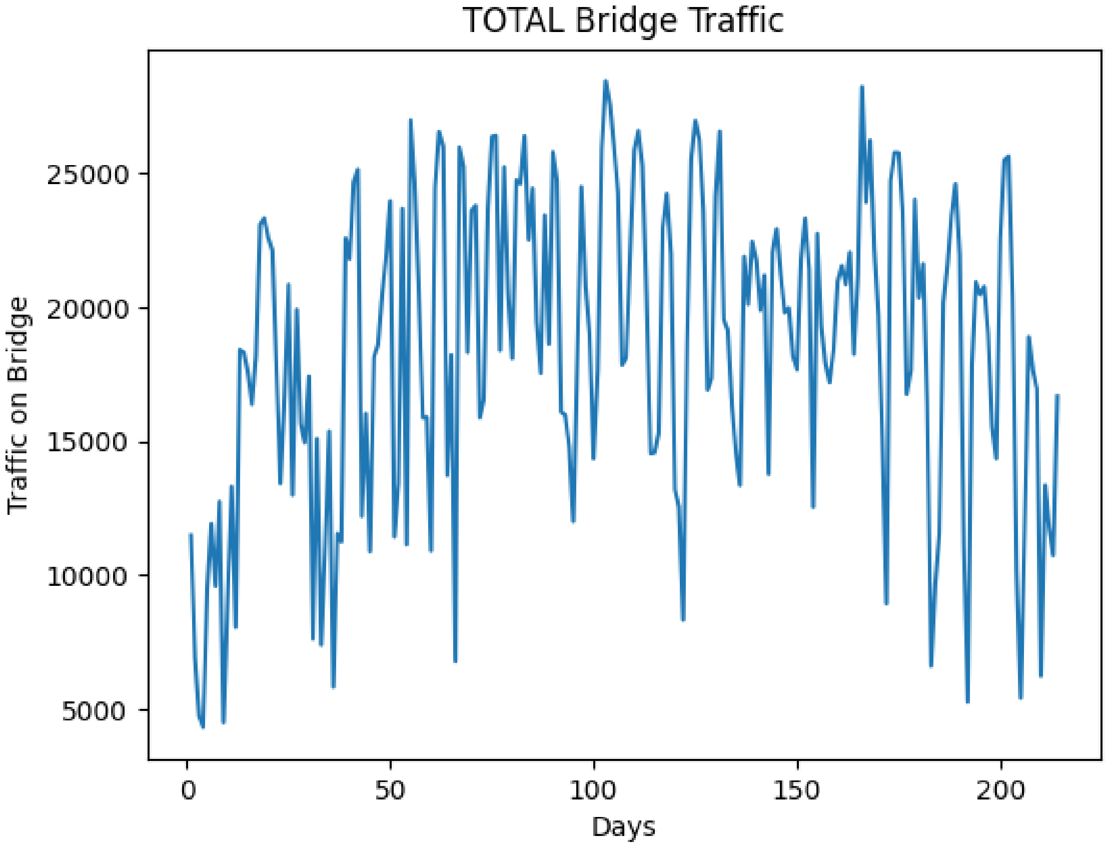


Figure 2. Total Cyclist Traffic Across All Four Bridges

Methods:

Question 1: The objective of this question is to determine which bridge should be left without a sensor to best predict the total traffic across all four bridges. Four multiple linear regression models were created to predict the total traffic using data from three bridges (leaving a different bridge out for each model). The model with the highest R-squared value and the lowest mean squared error tells us which three bridges the sensors should be installed on to get the best prediction of traffic across all four bridges.

Question 2: There are many different types of models that can be used to predict the number of cyclists based on other factors, and, in order to determine which type of model fits best, a tool called “lazy predict” was used to easily fit and evaluate all the models from scikit-learn using temperature and weather data (Lewinson). The model with the highest R-squared value and lowest mean squared error is the model that best predicts the number of cyclists. These values will tell us how effective this particular model is at predicting the number of cyclists.

Question 3: This problem requires finding correlation between numerical values and categorical values, so a classification model should be used. The “lazy predict” tool used in question 2 was used here as well to test various classification models. These models were then compared and the model with the highest accuracy value and lowest mean squared error is the one that fits best. Once the best model was identified, it was evaluated to see how accurately it can predict the day of the week.

Results:

Question 1: The multiple linear regression models created follow the format of the equation shown in Figure 3. Once the models were fitted to the data, they were compared to the data and evaluated. It was found that the model that was the best at predicting the total traffic was the model that had no sensor on the Queensboro bridge with an R-squared value of 0.9957 (Fig. 4). The model that was the worst at predicting the total traffic was the model that had no sensor on the Manhattan Bridge with an R-squared value of 0.947. To get the best prediction of overall traffic, sensors should be put on the Brooklyn Bridge, the Manhattan Bridge, and the Williamsburg Bridge. The complete model can be seen below along with a graph of its predictions on a scatter plot on random days (Figure 5).

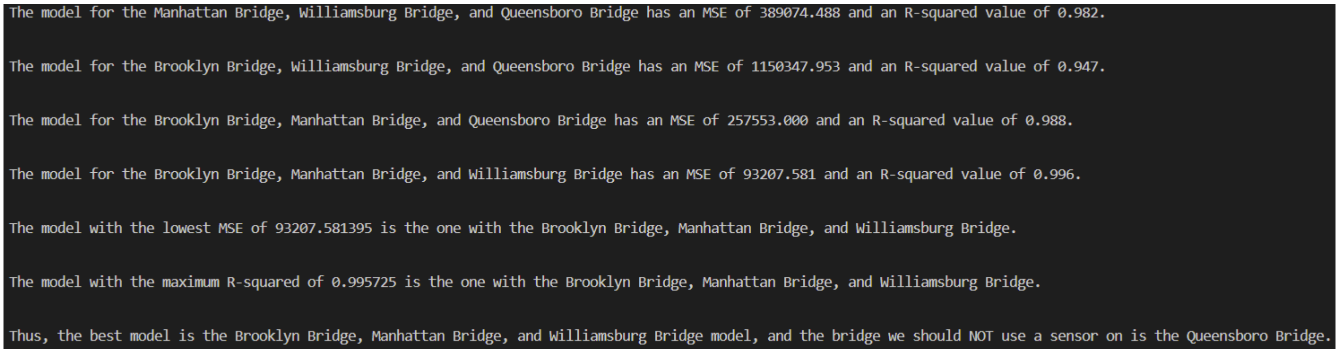
$$Y = 1.139 * X_1 + 0.947 * X_2 + 1.609 * X_3 + 382.746$$

$$X_1 = Brooklyn\ Traffic\ ;\ X_2 = Manhattan\ Traffic;\ X_3 = Williamsburg\ Traffic$$

$$\boxed{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ ... + \beta_i X_i}$$

Y : Dependent variable
β_0 : Intercept
β_i : Slope for X_i
X = Independent variable

Figure 3. Format of a Multiple Linear Regression Model



```
The model for the Manhattan Bridge, Williamsburg Bridge, and Queensboro Bridge has an MSE of 389074.488 and an R-squared value of 0.982.

The model for the Brooklyn Bridge, Williamsburg Bridge, and Queensboro Bridge has an MSE of 1150347.953 and an R-squared value of 0.947.

The model for the Brooklyn Bridge, Manhattan Bridge, and Queensboro Bridge has an MSE of 257553.000 and an R-squared value of 0.988.

The model for the Brooklyn Bridge, Manhattan Bridge, and Williamsburg Bridge has an MSE of 93207.581 and an R-squared value of 0.996.

The model with the lowest MSE of 93207.581395 is the one with the Brooklyn Bridge, Manhattan Bridge, and Williamsburg Bridge.

The model with the maximum R-squared of 0.995725 is the one with the Brooklyn Bridge, Manhattan Bridge, and Williamsburg Bridge.

Thus, the best model is the Brooklyn Bridge, Manhattan Bridge, and Williamsburg Bridge model, and the bridge we should NOT use a sensor on is the Queensboro Bridge.
```

Figure 4. Print Statements from Code Showing Which Model Fits Best

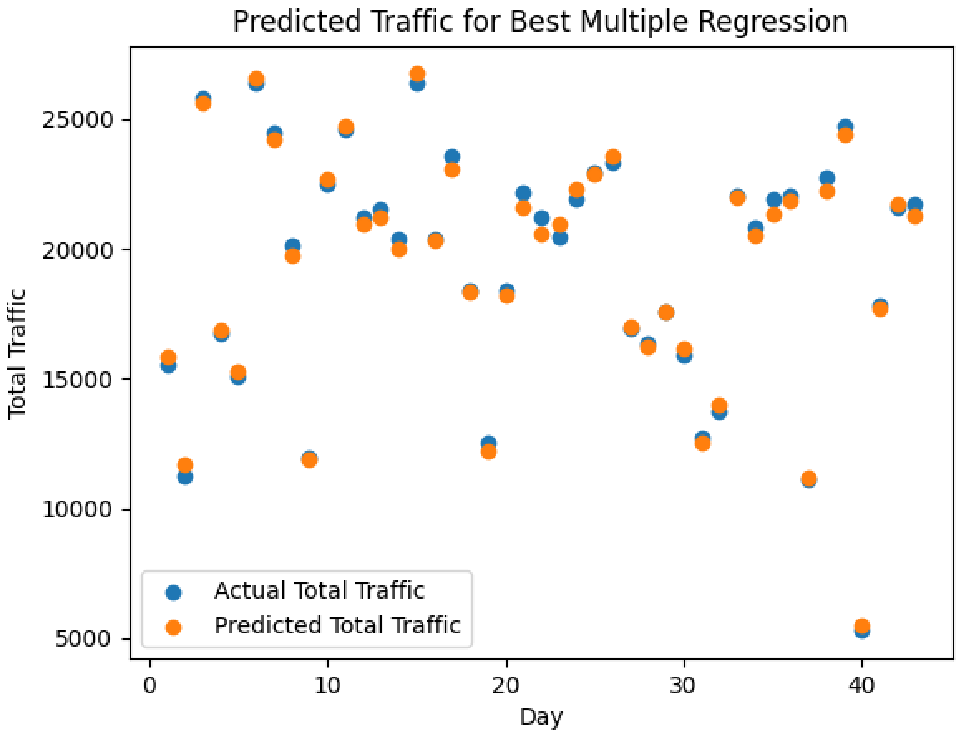
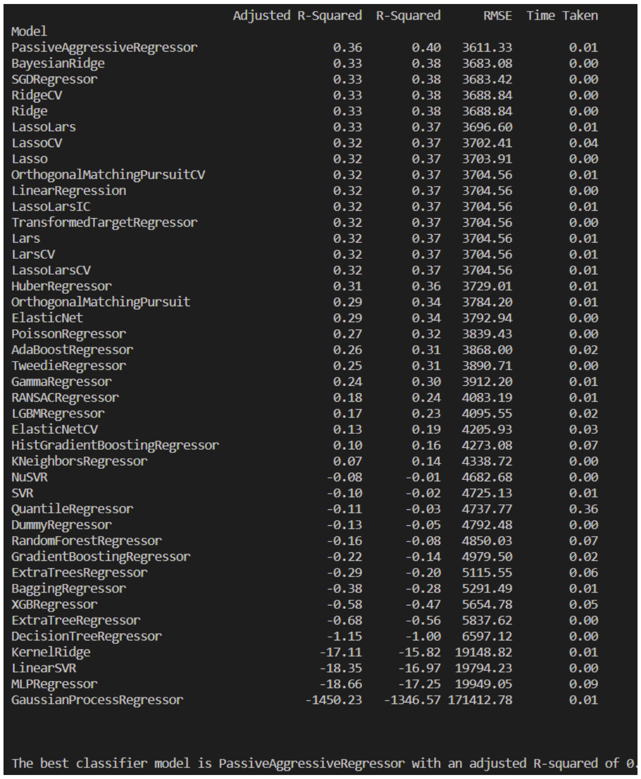


Figure 5. Predictions vs. Actual Total Traffic for Chosen Model

Question 2: Figure 6 shows the results of fitting and evaluating many different model types. The best model tested was a passive aggression regression model. Figure 7 shows the model's prediction, the true value, and the percent error for the test data. The test data was just 42 days because the rest of the days were used as training data to fit the model. While the passive aggressive regression model was the best fit model we tested, its R-squared value was still too low to confidently provide predictions for the number of cyclists. This could indicate that the number of cyclists is more heavily dependent on some other factor besides weather. The next day's weather forecast cannot be used to confidently and accurately predict the total number of cyclists that day. The Passive Aggressive Regressor model we produced using "sklearn" can be seen below along with a graph of the predictions on random days (Figure 8).

$$Y = -222.901X_1 + 400.609X_2 - 46.509X_3 - 20.073$$

$$X_1 = Low\ Temp;\ X_2 = High\ Temp;\ X_3 = Precipitation$$

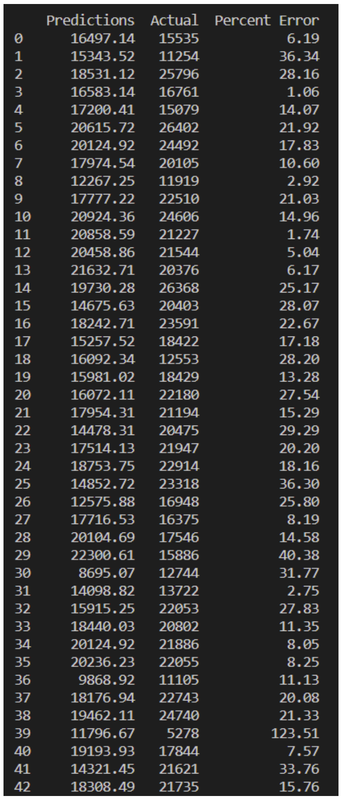


```
                               Adjusted R-Squared  R-Squared      RMSE  Time Taken
Model
PassiveAggressiveRegressor                   0.36       0.40   3611.33        0.01
BayesianRidge                                0.33       0.38   3683.08        0.00
SGDRegressor                                 0.33       0.38   3683.42        0.00
RidgeCV                                      0.33       0.38   3688.84        0.00
Ridge                                        0.33       0.38   3688.84        0.00
LassoLars                                    0.33       0.37   3696.60        0.01
LassoCV                                      0.32       0.37   3702.41        0.04
Lasso                                        0.32       0.37   3703.91        0.00
OrthogonalMatchingPursuitCV                  0.32       0.37   3704.56        0.01
LinearRegression                             0.32       0.37   3704.56        0.00
LassoLarsIC                                  0.32       0.37   3704.56        0.01
TransformedTargetRegressor                   0.32       0.37   3704.56        0.00
Lars                                         0.32       0.37   3704.56        0.01
LarsCV                                       0.32       0.37   3704.56        0.01
LassoLarsCV                                  0.32       0.37   3704.56        0.01
HuberRegressor                               0.31       0.36   3729.01        0.01
OrthogonalMatchingPursuit                    0.29       0.34   3784.20        0.01
ElasticNet                                   0.29       0.34   3792.94        0.00
PoissonRegressor                             0.27       0.32   3839.43        0.00
AdaBoostRegressor                            0.26       0.31   3868.00        0.02
TweedieRegressor                             0.25       0.31   3890.71        0.00
GammaRegressor                               0.24       0.30   3912.20        0.01
RANSACRegressor                              0.18       0.24   4083.19        0.01
LGBMRegressor                                0.17       0.23   4095.55        0.02
ElasticNetCV                                 0.13       0.19   4205.93        0.03
HistGradientBoostingRegressor                0.10       0.16   4273.08        0.07
KNeighborsRegressor                          0.07       0.14   4338.72        0.00
NuSVR                                       -0.08      -0.01   4682.68        0.00
SVR                                         -0.10      -0.02   4725.13        0.01
QuantileRegressor                           -0.11      -0.03   4737.77        0.36
DummyRegressor                              -0.13      -0.05   4792.48        0.00
RandomForestRegressor                       -0.16      -0.08   4850.03        0.07
GradientBoostingRegressor                   -0.22      -0.14   4979.50        0.02
ExtraTreesRegressor                         -0.29      -0.20   5115.55        0.06
BaggingRegressor                            -0.38      -0.28   5291.49        0.01
XGBRegressor                                -0.58      -0.47   5654.78        0.05
ExtraTreeRegressor                          -0.68      -0.56   5837.62        0.00
DecisionTreeRegressor                       -1.15      -1.00   6597.12        0.00
KernelRidge                                -17.11     -15.82  19148.82        0.01
LinearSVR                                  -18.35     -16.97  19794.23        0.00
MLPRegressor                               -18.66     -17.25  19949.05        0.09
GaussianProcessRegressor                 -1450.23   -1346.57 171412.78        0.01



The best classifier model is PassiveAggressiveRegressor with an adjusted R-squared of 0.
```

Figure 6. Table showing important metrics for each type of model once it has been fit to the data



```
    Predictions  Actual  Percent Error
0      16497.14   15535           6.19
1      15343.52   11254          36.34
2      18531.12   25796          28.16
3      16583.14   16761           1.06
4      17200.41   15079          14.07
5      20615.72   26402          21.92
6      20124.92   24492          17.83
7      17974.54   20105          10.60
8      12267.25   11919           2.92
9      17777.22   22510          21.03
10     20924.36   24606          14.96
11     20858.59   21227           1.74
12     20458.86   21544           5.04
13     21632.71   20376           6.17
14     19730.28   26368          25.17
15     14675.63   20403          28.07
16     18242.71   23591          22.67
17     15257.52   18422          17.18
18     16092.34   12553          28.20
19     15981.02   18429          13.28
20     16072.11   22180          27.54
21     17954.31   21194          15.29
22     14478.31   20475          29.29
23     17514.13   21947          20.20
24     18753.75   22914          18.16
25     14852.72   23318          36.30
26     12575.88   16948          25.80
27     17716.53   16375           8.19
28     20104.69   17546          14.58
29     22300.61   15886          40.38
30      8695.07   12744          31.77
31     14098.82   13722           2.75
32     15915.25   22053          27.83
33     18440.03   20802          11.35
34     20124.92   21886           8.05
35     20236.23   22055           8.25
36      9868.92   11105          11.13
37     18176.94   22743          20.08
38     19462.11   24740          21.33
39     11796.67    5278         123.51
40     19193.93   17844           7.57
41     14321.45   21621          33.76
42     18308.49   21735          15.76
```

Figure 7. Table of Trained Model, Test Data, and the Percent Error

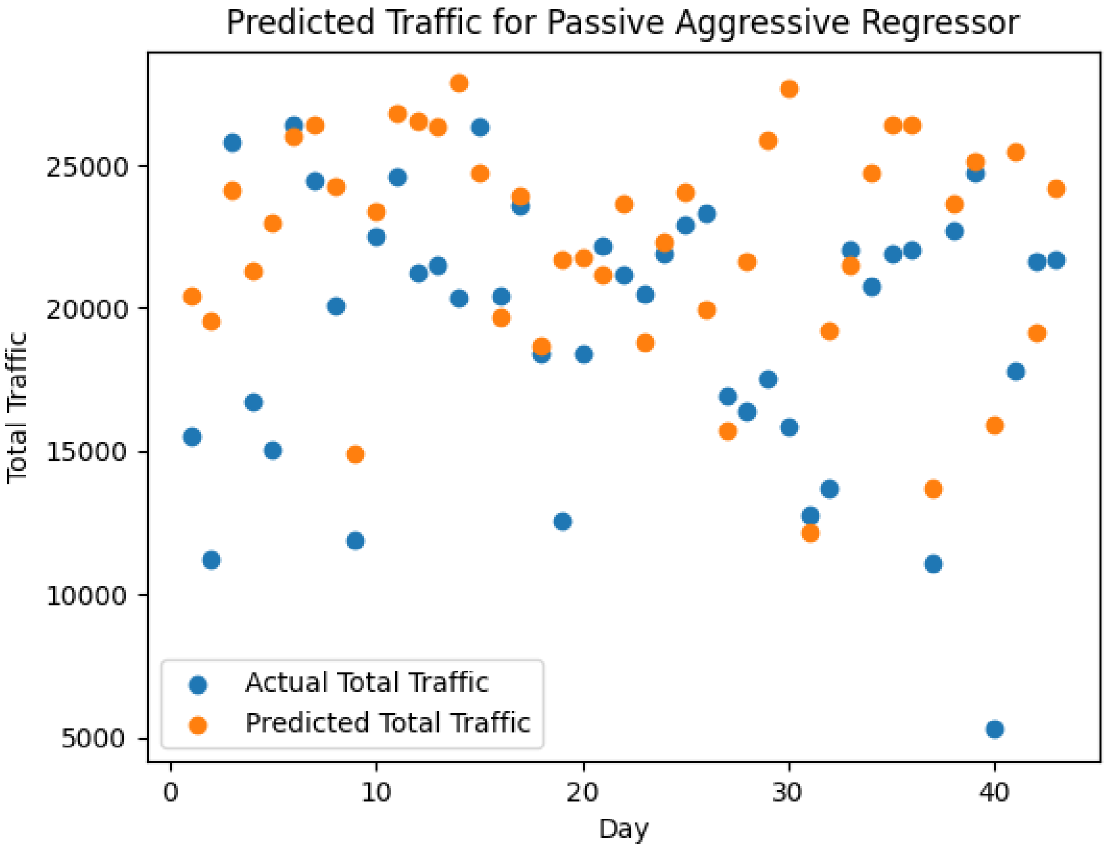
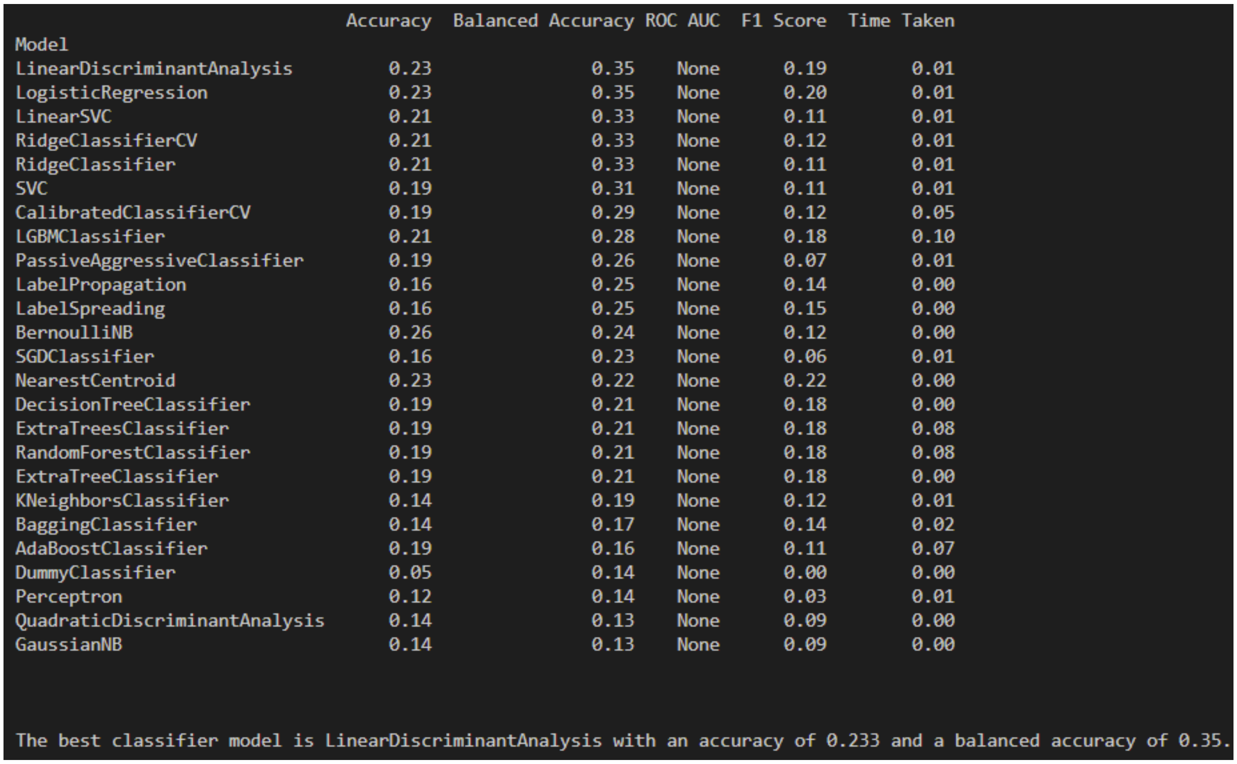


Figure 8. Prediction vs. Actual Total Traffic for Chosen Model

Question 3: Figure 9 shows the results of fitting and evaluating several types of classification models. A linear discriminant analysis model had the best accuracy and balanced accuracy of the types of models tested. Accuracy is the percentage of days correctly predicted by the model and balanced accuracy is the average of the accuracy for each day of the week. Balanced accuracy is a useful metric for analyzing data when the data is imbalanced and one of the target classes appears much more frequently than the others. Figure 10 shows a confusion matrix for the linear discriminant analysis model created from the test data. The entries along the diagonal from top left to bottom right represent the correct predictions. The non-zero values outside the diagonal represent false positives. The calculated balanced accuracy of 35% is better than the calculated accuracy of 23%, but both are still far too low to say that this model is effective at guessing the day of the week based on the number of cyclists.



```
                               Accuracy  Balanced Accuracy ROC AUC  F1 Score  Time Taken
Model
LinearDiscriminantAnalysis         0.23               0.35    None      0.19        0.01
LogisticRegression                 0.23               0.35    None      0.20        0.01
LinearSVC                          0.21               0.33    None      0.11        0.01
RidgeClassifierCV                  0.21               0.33    None      0.12        0.01
RidgeClassifier                    0.21               0.33    None      0.11        0.01
SVC                                0.19               0.31    None      0.11        0.01
CalibratedClassifierCV             0.19               0.29    None      0.12        0.05
LGBMClassifier                     0.21               0.28    None      0.18        0.10
PassiveAggressiveClassifier        0.19               0.26    None      0.07        0.01
LabelPropagation                   0.16               0.25    None      0.14        0.00
LabelSpreading                     0.16               0.25    None      0.15        0.00
BernoulliNB                        0.26               0.24    None      0.12        0.00
SGDClassifier                      0.16               0.23    None      0.06        0.01
NearestCentroid                    0.23               0.22    None      0.22        0.00
DecisionTreeClassifier             0.19               0.21    None      0.18        0.00
ExtraTreesClassifier               0.19               0.21    None      0.18        0.08
RandomForestClassifier             0.19               0.21    None      0.18        0.08
ExtraTreeClassifier                0.19               0.21    None      0.18        0.00
KNeighborsClassifier               0.14               0.19    None      0.12        0.01
BaggingClassifier                  0.14               0.17    None      0.14        0.02
AdaBoostClassifier                 0.19               0.16    None      0.11        0.07
DummyClassifier                    0.05               0.14    None      0.00        0.00
Perceptron                         0.12               0.14    None      0.03        0.01
QuadraticDiscriminantAnalysis      0.14               0.13    None      0.09        0.00
GaussianNB                         0.14               0.13    None      0.09        0.00


The best classifier model is LinearDiscriminantAnalysis with an accuracy of 0.233 and a balanced accuracy of 0.35.
```

Figure 9. Table showing important metrics for each model once it has been fit to the data

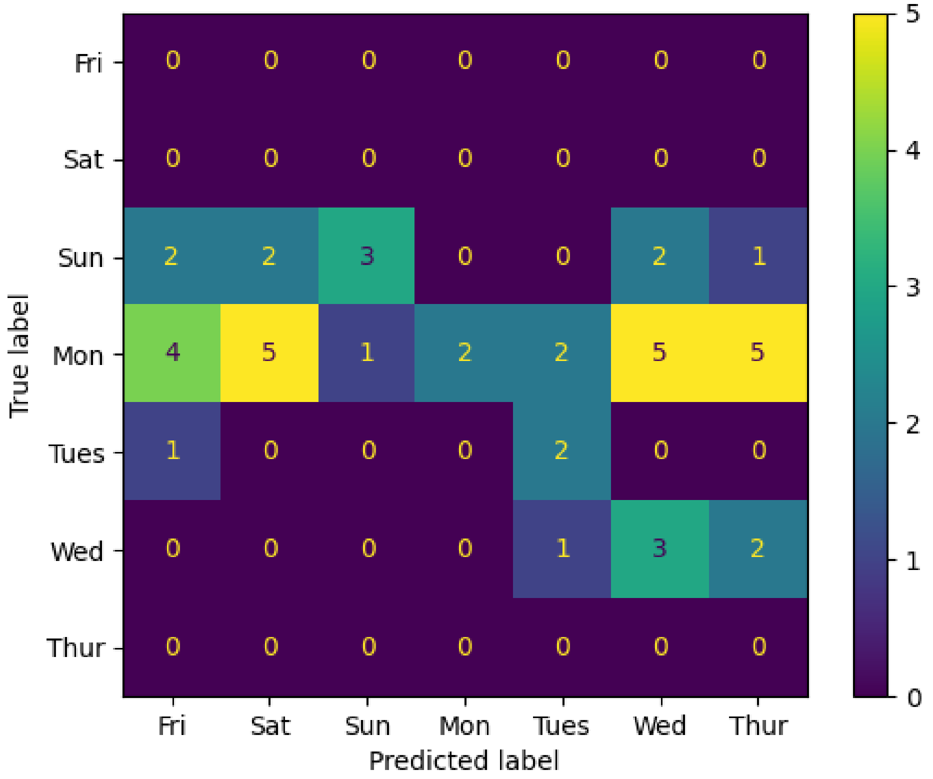


Figure 10. Confusion Matrix of Trained Model Predictions Vs Test Data

Sources:

1. "Example of Multiple Linear Regression in Python." *Data to Fish*, 30 Jul. 2022, datatofish.com/multiple-linear-regression-python/.
2. Lewinson, Eryk. "Lazy Predict: fit and evaluate all the models from scikit-learn with a single line of code." *Towards Data Science*, 8 Jan. 2021, towardsdatascience.com/lazy-predict-fit-and-evaluate-all-the-models-from-scikit-learn-with-a-single-line-of-code-7fe510c7281.