ECE 20875

Final Project -New York Bridges Bicycle counts.

Introduction:

Contributors: Archis Raykar, Siddharth Mitra.

Purdue Username: araykar, mitra30

GitHub Username: araykar0412, siddmitra10

Project Path: 1

Section 1: Dataset:

The dataset provided to us for this group project(path 1) is a CSV file containing information on bike traffic across 4 bridges(Brooklyn, Manhattan, Williamsburg, Queensboro) in New York City. In this Data frame, we are provided with data from 1st of April to 31st October (7 months) containing the day, the High temperature and Low temperature at that day (in deg F), Precipitation that happened on that day(ranging from 0 inches to 1.65 inches), the number of bike users on each of the bridges on that day and lastly, the total number of bikers present on that same day.

Section 1: Analysis:

Q.1 Analysis:

For the first question, they are asking us to install sensors on the bridges to estimate overall traffic across all the bridges. We are supposed to install 3 sensors and thus need to predict which bridges to install so that we can predict the overall traffic data optimally.

As we had to predict which 3 bridges to choose out of 4. So we tried fitting a model with our feature variables as x1,x2,x3,x4 each denoting a bridge's data set, and then trying to find the predicted value of the total number of our bikers on a given day. Firstly, we will choose 3 bridges out of the 4 available where we will install our sensors and create a model for each of the 4 combinations present using the 3 bridges' dataset as our feature variables for the model. We will then fit a linear regression for the 4 models. We will cross-validate our predicted data with the observed data obtained from the total number of bikers dataset. We will use the train_test_split() inbuilt function with a 90% split for our training data and also to analyze and debug our problem, we are setting a random state for the purpose consistency. We will then calculate the MSE for each of the models and then choose the model which has the least MSE out of the models present as that particular model has the least error in its predicted value when compared to the observed value thus predicting the data of the total number of bikers optimally. Hence, installing our sensors onto those 3 bridges.

Q.2 Analysis:

For this problem, we have to check whether we can use the next day's weather data to predict the number of bicyclists that day for the city administration to deploy police officers to hand out citations. As we have 3 datasets High temp, Low temp, and Precipitation. As we have multiple variables we have tried to fit a model using the 3 variables' data as our training set and give us a predicted value for the number of bicyclists for the next day. The data provided to us cannot be sorted and is in a non-linear form thus we are going to use a multivariable regression(explain the relationship between one continuous dependent variable and multiple independent variables) and do not have our data overfitted we are using regularization.

Our target variable is going to be the next day's total number of bikers' data set.

Typically break our data set into an 80% and 20% (using test_train_split inbuilt function.) split.

As we need every data to be of the same unit we need to normalize the data(X_train, Y_train, and Y_test) by subtracting each dataset by its mean and then dividing it by its standard deviation and also unnormalizing the Y_predicted data by undoing the arithmetic calculations done for normalizing the data using the mean and std of Y_train dataset. We then use this normalized data to calculate the optimal value of the regularization parameter. To implement our regularization technique we are going to fit a ridge regression on our model and another reason for using this regression is that we did not want any correlation between our independent variables to minimize the error in our predicted model. Also, just like our previous question ridge expression helps us identify the more influential feature variables thus helping us increase our efficiency of prediction.

Once our model is fit, we will choose the coefficients of the model with the least MSE to minimize our error as a part of our regularization method.

For checking the accuracy of our model that we have tried to fit we will check our R2 score for the same and we consider 0.85 to be a threshold fit level of our R2 score for the model to accurately predict the number of bicyclists the next day. If it is below 0.85 we will consider our model to not predict our output correctly and vice versa.

Q.3 Analysis:

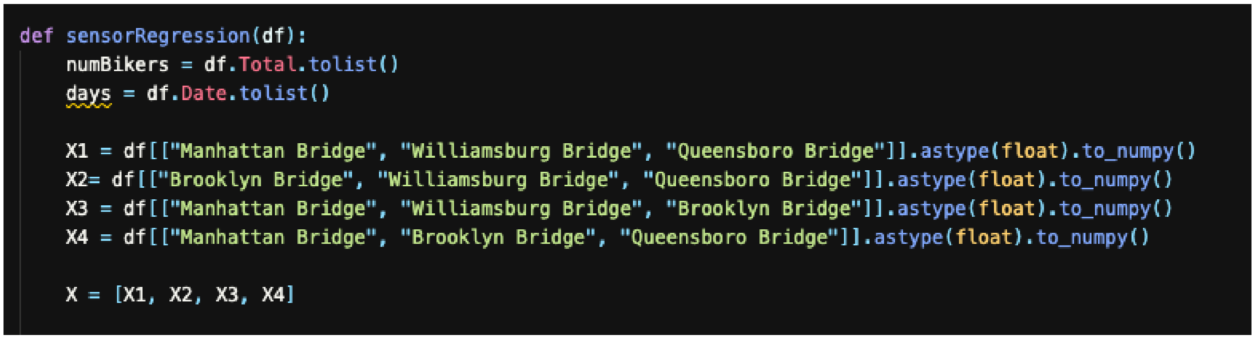
Question 3 asks us to predict whether it is raining or not, based on the number of bicyclists on that given day. This question has more of a binary answer as in Yes it is raining or No it is not raining. Thus for this kind of answer, we thought that the best way to tackle this problem was setting up a logistic regression as it is mainly used to predict binary answers like Yes or No (0 or 1). We used the total number of bikers dataset as our feature variable to set this regression and our target dataset was the precipitation dataset. We created a new array of our dataset in a binary format where if there is a 0 or a T denoting Trace amounts of precipitation we would consider that as our boolean False value as no precipitation occurred and if there was any numerical data present (ignoring the S in this condition) we would consider that day to have precipitation thus, boolean True.

We set our logistic regression with a split of 80% train and 20% test with random states ranging from 1-100 to calculate the F-score for each state and then averaged our F-1 score for the entire set of random states. F-1 score ranges from 0-1(1 being perfectly accurate) and as it is a binary problem the minimum F-1 score we needed was 0.5 as there is one out of 2 options for the model to predict it correctly. If the averaged F-1 score is below 0.5 we will conclude that our logistic regression model is not a great fit for our data and cannot accurately predict if it is raining or not on that particular day.

Section 2: Results:

Q.1 Results:

We created 4 combinations of bridges each containing 3 of the bridges as shown in the screenshot below:

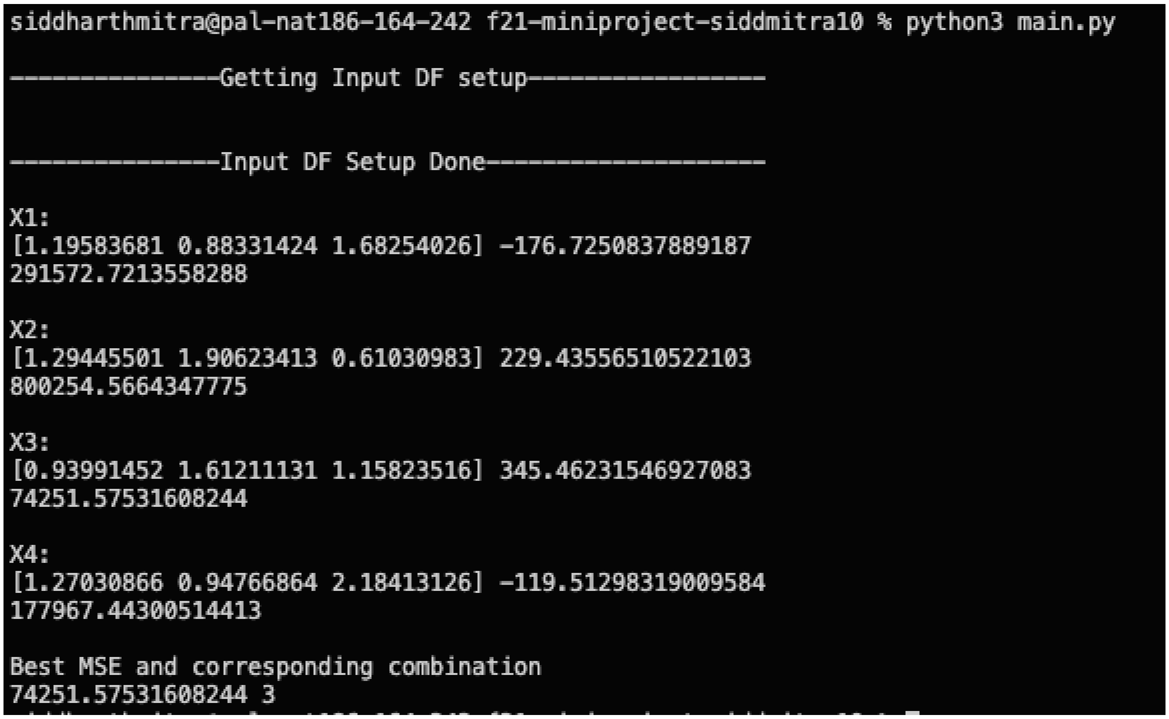


```
def sensorRegression(df):
    numBikers = df.Total.tolist()
    days = df.Date.tolist()

    X1 = df[["Manhattan Bridge", "Williamsburg Bridge", "Queensboro Bridge"]].astype(float).to_numpy()
    X2= df[["Brooklyn Bridge", "Williamsburg Bridge", "Queensboro Bridge"]].astype(float).to_numpy()
    X3 = df[["Manhattan Bridge", "Williamsburg Bridge", "Brooklyn Bridge"]].astype(float).to_numpy()
    X4 = df[["Manhattan Bridge", "Brooklyn Bridge", "Queensboro Bridge"]].astype(float).to_numpy()

    X = [X1, X2, X3, X4]
```

Using these 4 combinations we set up a linear regression model for each of the models with the combination of bridges in each model as their respective feature variables. After fitting these models, we calculated the MSE for each model using an inbuilt function from sklearns.metrics library as shown below. We then outputted the coefficient matrixes of each of the models with their respective intercepts as shown below with the MSE for each of the models just below their respective coefficients matrixes as shown below:



```
siddharthmitra@pal-nat186-164-242 f21-miniproject-siddmitra10 % python3 main.py

---------------Getting Input DF setup-----------------


---------------Input DF Setup Done--------------------

X1:
[1.19583681 0.88331424 1.68254026] -176.7250837889187
291572.7213558288

X2:
[1.29445501 1.90623413 0.61030983] 229.43556510522103
800254.5664347775

X3:
[0.93991452 1.61211131 1.15823516] 345.46231546927083
74251.57531608244

X4:
[1.27030866 0.94766864 2.18413126] -119.51298319009584
177967.44300514413

Best MSE and corresponding combination
74251.57531608244 3
```

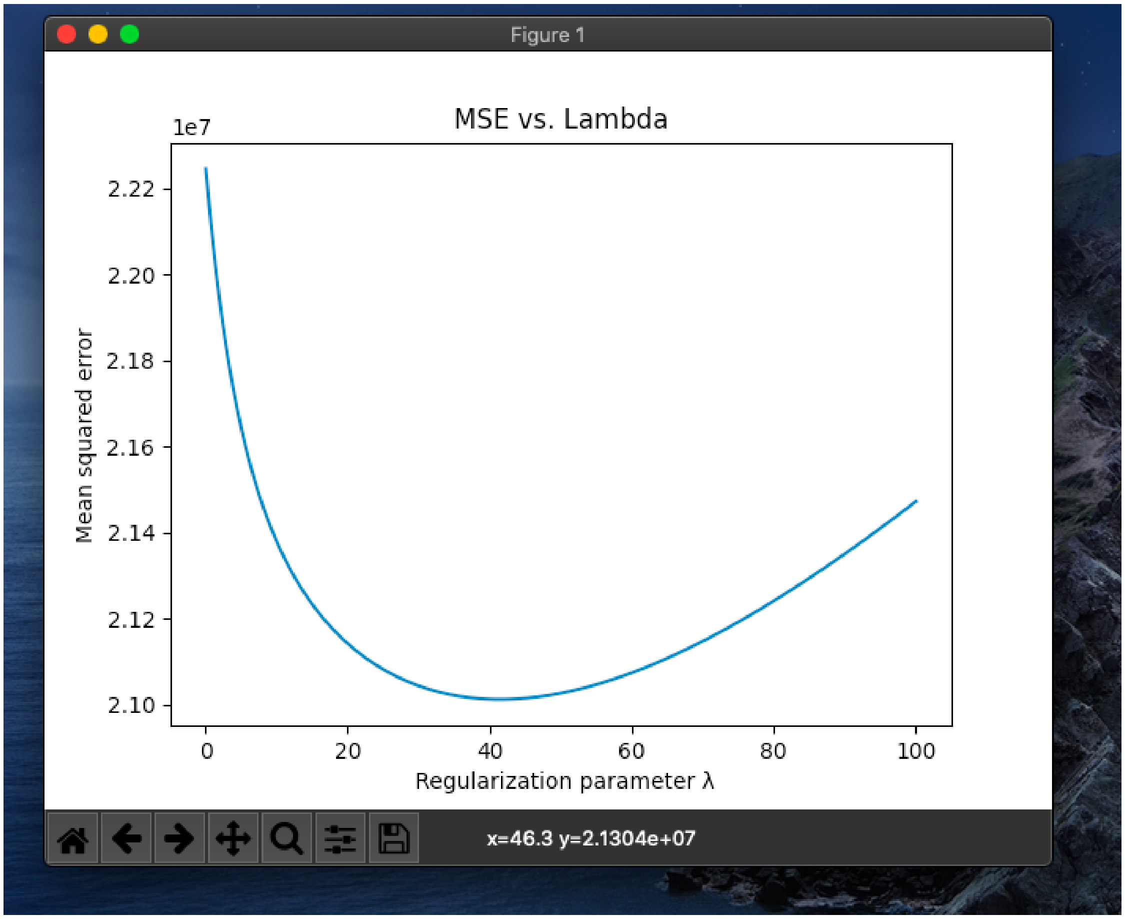
We then choose that model which has the least MSE as it shows the least error between the predicted and observed values. As seen from the screenshot given above, X3 i.e the Manhattan Bridge, Williamsburg Bridge, and Brooklyn Bridge combination model seems to predict the required data optimally. Thus, by this reasoning we think that the above three bridges stated(M, W, B) should be the bridges chosen to install the 3 sensors out of the 4 New York Bridges given.

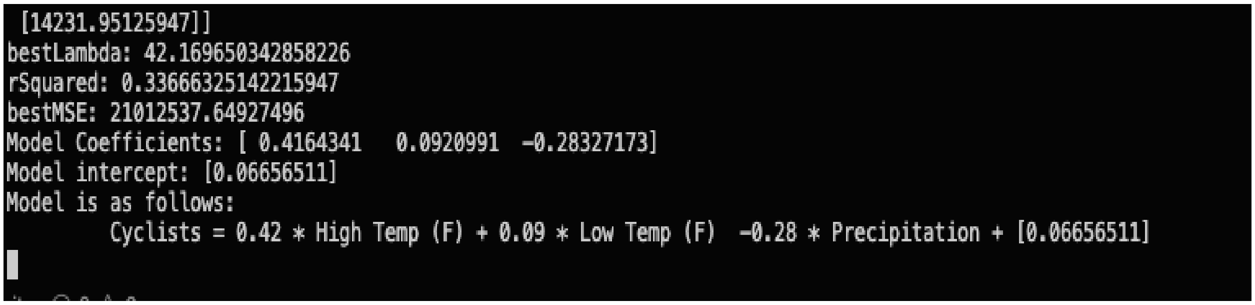
Q.2 Results:

For our second problem, we had to check if can use the next day's weather forecast data (High temp, Low temp, precipitation) to predict the number of bicyclists present on that day. For this, we fitted a multivariable linear regression model with all the 3 weather datasets as our feature variables. As we had multiple variables we did not want our data to be overfitted thus, we used the regularization technique to mitigate overfitting by penalizing non-zero model coefficients.

The expression that we used to perform this technique was: min(model error) + λ (coeff weights).

We obtained a relation between MSE and λ as shown below in the diagram.





```
 [14231.95125947]]
bestLambda: 42.169650342858226
rSquared: 0.33666325142215947
bestMSE: 21012537.64927496
Model Coefficients: [ 0.4164341   0.0920991  -0.28327173]
Model intercept: [0.06656511]
Model is as follows:
        Cyclists = 0.42 * High Temp (F) + 0.09 * Low Temp (F)  -0.28 * Precipitation + [0.06656511]
```

The best λ that we obtained after fitting this model was : 42.1696503

The R^2 score obtained from our model was: 0.336663251.

The MSE obtained for the best λ was: 21012537.6493

Also, we have formed the normalized equation for the predicted y values as shown above in the picture using the normalized feature variable datasets. To interpret this result, you will have to un-normalize the variable 'Cyclists' using the mean and standard deviation of Y.

As mentioned above the R^2 score that we obtained from our model was 0.33666 which tells us that there is a slight correlation between the weather forecast and the number of bicyclists on that day. We had already declared above in our analysis for question 2 that we at least need an R^2 score of 0.85 to conclude that the model we have set up is a good fit for these prediction values. Thus, using these aforementioned reasons, we would like to conclude that our multivariable linear regression model is not a good fit for predicting the number of bicyclists on that given day.

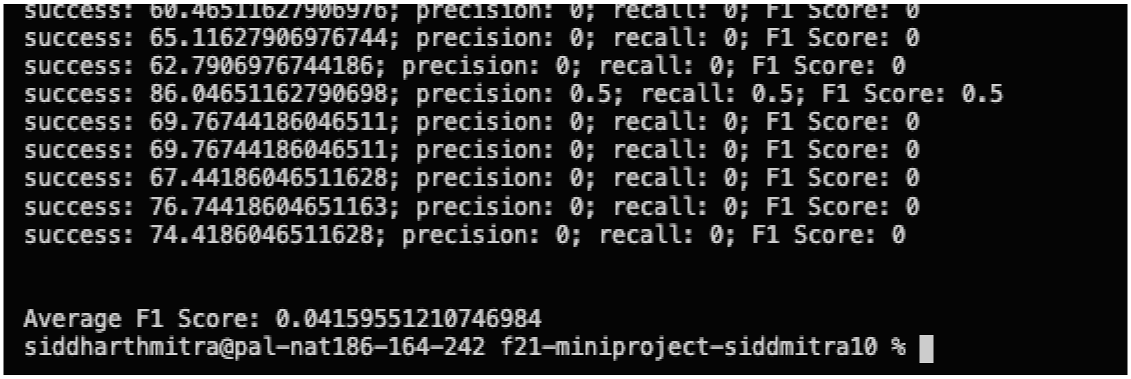
We realized after analyzing our results and data that, rather than fitting a linear model, if we had approached this problem with fitting a non-linear regression model like the SVM it would be a better fit as the data seems to be more fitting for a non-linear model.

<u>Q.3 Results:</u>

For the third problem, we have to predict whether it is going to rain or not depending on the number of bicyclists present on the bridge that day. It seemed to us that the answer that they needed for the question is a binary answer: either Yes it is raining or No it is not raining. Thus, as we needed a binary answer we fitted a logistic regression for this problem.

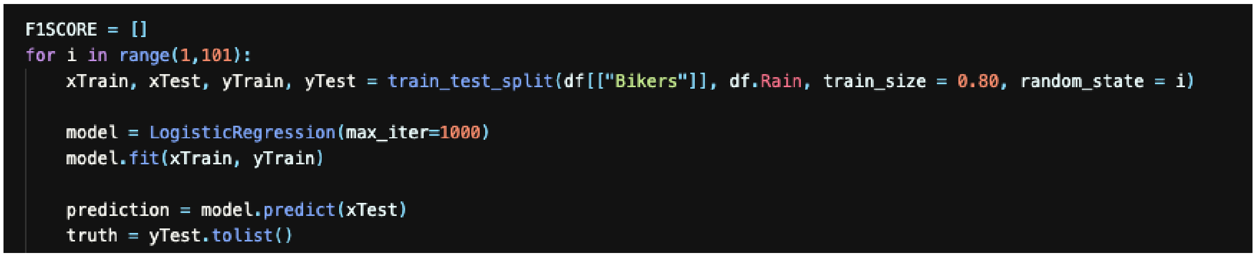
We used the number of bicyclists data present in the total number of bicyclists data-set in the CSV file to train our model and used the precipitation data as our testing data-set. We converted this precipitation data-set into a binary array resulting in boolean FALSE if the precipitation value was either 0 or T (denoting Trace) and boolean TRUE if it was any other numerical except the above 2 mentioned. For the purpose of this, we remove "(S)" in all applicable data points

We realized after analyzing our results and data that, rather than fitting a linear model, if we had approached this problem with fitting a non-linear regression model like the SVM it would be a better fit as the data seems to be more fitting for a non-linear model.



```
success: 60.46511627906976; precision: 0; recall: 0; F1 Score: 0
success: 65.11627906976744; precision: 0; recall: 0; F1 Score: 0
success: 62.7906976744186; precision: 0; recall: 0; F1 Score: 0
success: 86.04651162790698; precision: 0.5; recall: 0.5; F1 Score: 0.5
success: 69.76744186046511; precision: 0; recall: 0; F1 Score: 0
success: 69.76744186046511; precision: 0; recall: 0; F1 Score: 0
success: 67.44186046511628; precision: 0; recall: 0; F1 Score: 0
success: 76.74418604651163; precision: 0; recall: 0; F1 Score: 0
success: 74.4186046511628; precision: 0; recall: 0; F1 Score: 0


Average F1 Score: 0.041595512107469884
siddharthmitra@pal-nat186-164-242 f21-miniproject-siddmitra10 %
```



```
F1SCORE = []
for i in range(1,101):
    xTrain, xTest, yTrain, yTest = train_test_split(df[["Bikers"]], df.Rain, train_size = 0.80, random_state = i)

    model = LogisticRegression(max_iter=1000)
    model.fit(xTrain, yTrain)

    prediction = model.predict(xTest)
    truth = yTest.tolist()
```

The success prediction percentage can be shown above in the following screenshot. On an average our success percentage came out to be very high but, it is not the only factor determining if our model is a perfect fit or not. The reason behind this is that more often in the data there is 0 or T precipitation resulting in FALSE value and as our dataset contains just 214 datapoints it is a very low number for our model to predict the output, thus our model is more biased towards FALSE boolean value. To get a fair understanding of how good is our model we calculated and printed out the precision, recall and F1 score values as shown above.

As you can see in the diagram given above we calculated the precision, recall and F1 score for a range of 100 random states to get an understanding of how our data is being trained on an average and then calculated an average F1 score for the same which came out to be 0.0415955. An F1 score ranges from 0-1 and is a perfect way of checking the reliability and quality of our logistic model that we have fit on our training variables and it should atleast be 0.5 as our ans can be one out of 2 options. Thus as our average F1 score is way below we can conclude that the logistic regression model that we have tried to predict if it is going to rain or not is not a good fit for the problem at hand.