Ekam Bhullar

ECE 20875 – 002

Prof. Qiang Qiu

PUID: 00371-70892

Mini Project Path 2

GitHub : https://github.com/ECEDataScience/project-s25-ebhulla

**Description of dataset:**

The data provided for this path is the “nyc\_bicycle\_count\_2016.csv” file. In this comma-separated file, we have the information on cycle traffic across several bridges in New York City. The data is divided into columns as follows:

* Date: In the form of (day-month) format, for eg, 1-Apr, denoting 1st of April, we do not see any year as it is collected over the year 2016 as described in the title.
* Day: This represents the day of the week, such as Monday, Tuesday and Wednesday, etc. It is in correspondence with the date data.
* High Temp: This is the highest temperature recorded on that day or date, and it is given in Fahrenheit.
* Low Temp: This is the lowest temperature recorded on that particular day, and it is also given in Fahrenheit.
* Precipitation: The amount of precipitation that occurred that day in inches.
* Brooklyn, Queensboro, Manhattan, Williamsburg bridges: Number of cyclists recorded on each of these bridges on a particular day.
* Total: The total number of cyclists across all four bridges on that day.

**Method of Analysis:**

**Question 1:**

**You want to install sensors on the bridges to estimate overall traffic across all the bridges. But you only have enough budget to install sensors on three of the four bridges. Which bridges should you install the sensors on to get the best prediction of overall traffic?**

For this question, I used the linear regression model to estimate the overall traffic across all the bridges. I declared variable X as the number of cyclists on a particular bridge and Y as the total number of cyclists across all four bridges. I calculated the R^2 values for each linear regression model I plotted, for e.g., Number of Cyclists on Queensboro Bridge vs Total Number of cyclists. The fit quality was determined using the R^2 value, also known as the coefficient of determination. The closer the values were to one, the better the fit of the model. The top three bridges were ranked based on this value. Finally, we use one-tailed hypothesis testing to check if the average R^2 value across all bridges was greater than 0.25 or not.

Justification:

I chose a simple linear regression model because it is easy to understand and interpret without any need to normalize or smooth the data. It can help predict the value of a variable based on the value of another variable (IBM, n.p). I rank them by R^2 value because it helps in sorting and picking out the strongest predictors. The idea behind the null hypothesis is that we assume that each bridge will have an equal amount of traffic, which is an ideal condition. In simple terms, the traffic is 100% and divided among four bridges, so each bridge shall have 25% of all the traffic, which roughly converts to a population mean (mu) of 0.25. I take the average of all R^2 values of the bridges and compose a sample mean. Since the null hypothesis requires a Z-score for a dataset with more than 30 values, and the standard deviation is also known. Hence, we calculate the Z-score and thereafter the p-value. Alpha, or the significance level, is chosen as 0.05, which means we are accepting a 5% error, and this level has become standard in industry and research fields. Also, it's widely accepted because it provides a good tradeoff between error risks, making it a reliable and conventional choice. (Tenny & Abdelgawad, 2023). Therefore, comparing the sample means to a hypothesised population means using a Z-test provides statistical validity to our findings.

Expectation:

We expect certain bridges to have a higher R^2 value than others. It would be good news if we could reject the null hypothesis. Our null hypothesis is H0: The Sample mean of R^2 values is less than or equal to 0.25, directly hinting that a number of cyclists on bridges on a particular day are weak predictors of the total count of bridges, which shows local authorities cannot decide where to place the sensors. If we can reject it, this will show that we have enough evidence that the number of cyclists on a bridge is a good predictor of where to place the sensors by the authorities. In other words, we have stronger, more meaningful relationships between the bridge cyclist counts and the total cyclist count.

Conclusion:

R^2 value for Brooklyn Bridge: 0.7646

R^2 value for Manhattan Bridge: 0.8751

R^2 value for Queensboro Bridge: 0.9277

R^2 value for Williamsburg Bridge: 0.9508

Top 3 bridges based on R^2 value:

1. Williamsburg Bridge (R^2: 0.9508)

2. Queensboro Bridge (R^2: 0.9277)

3. Manhattan Bridge (R^2: 0.8751)

Hypothesis Test:

H₀: Sample mean ≤ 0.25

H₁: Sample mean > 0.25

Reject the null hypothesis (p-value: 0.0000)

These are the outcomes of the code, which very clearly represent that we were able to reject the null hypothesis and display the top three bridges where sensors can be placed.

A collage of graphs and charts

AI-generated content may be incorrect.

**Question 2: The city administration is cracking down on helmet laws and wants to deploy police officers on days with high traffic to hand out citations. Can they use the next day's weather forecast (low/high temperature and precipitation) to predict the total number of bicyclists that day?**

For this question, we consider the weather features such as high temperature, low temperature, and precipitation to make predictions for the number of cyclists. Our X for this question is the features of the weather, and Y is the total number of cyclists on all bridges on that day. We split the data into test and train and then use linear regression to see if the R^2 value is close to 1, and also compare the Mean Squared Error to give a robust prediction on whether weather is a predictor of the total number of bicyclists that day or not. We also generate a scatter plot with a line of best fit to help us visually analyse our predictions.

Justification:

Since we needed to predict the usefulness of weather data in this question, we split the data into training and testing sets. Training sets help to teach the data as it learns certain patterns from the data. The testing set then evaluates the model and shows us how well the model can predict new and unknown data. This has two main benefits, first that it checks if the model generalises the data well or not, and secondly, it avoids overfitting, additionally helping us to not just memorise the patterns but see if they exist or not. We specify the test size to be 0.2, which means 20% of the data will be used for testing and 80 % for training. If we search the Internet for the best train-test ratio, the first answer to pop up will be 80:20 (Tokuç, 2021). Setting a random state equal to 42 means we are controlling the randomness of how the split is made. This is primarily done when you have to re-run your code several times, for eg, if a grader wants to run your code, you want the results to match exactly when you ran it individually.

We generate a scatter plot to compare what the model predicted vs what real-world values are. The X-axis has the actual number of cyclists, and the Y-axis has the predicted total number of cyclists. We additionally plot a dotted line known as the line of perfect fits. If the model is perfect, then the dots shall lie exactly on the line; if the model is just fine, then the dots should be near the line; otherwise, the model is bad at prediction. This helps us judge the model without worrying about numbers like R^2 and mean squared error.

Expectation:

We expect weather conditions like temperature and precipitation to have a noticeable impact on bike traffic on bridges. Weather generally affects the movement of lot of things, hence we expect it to influence bike traffic as well. However, since the dataset is too large, we might end up with slight variations and discrepancies.

Conclusion:

The results are as follows:

Mean Squared Error for weather data: 17498379.3172

R^2 value for weather data: 0.5750

A graph with blue dots and red line

AI-generated content may be incorrect.

Since we get a big MSE, which itself is not a bad thing, since Total Cyclist is itself a very big number. R^2 tells us about the variability between the variables, and our R^2 is equal to 0.5750, which means our model explains 57.5% variation in the number of cyclists based on weather conditions. In the real world, anything around 0.5 and 0.7 is considered decent in real-world modelling. Our model predicts traffic questionably well however, it still fails to explain 42.5% of the variation in data. Some suggestions are to use complex models like Random Forest Regression, which can capture nonlinear relationships. Make use of cross-validation (K-Fold) to get a more reliable performance.

Thus, city administration cannot deploy police officers solely based on the weather data; they need to consider more features before making a decision.

Question 3: Can **you analyse and visualise the data to identify any patterns or trends associated with specific days of the week? (Hint: One way is that you can average the values over all weekdays and then see if there are some weekly patterns.) Can you use this data to predict what *\*day\** (Monday to Sunday) it is today based on the number of bicyclists on the bridges?**

For this question, we specifically focus on the date and day data provided. We add the year 2016 to the data provided in the form of “1-Apr”. We also drop any Nan values and group by data based on the days of the week. After doing some preprocessing on the target data, we finally make use of the Random Forest Classifier to predict the day of the week based on the number of cyclists on all bridges on that particular day. Again, we split the data into a testing and training set. We also analyse the accuracy of the model by using the accuracy score. We plot a bar chart to visualise the number of cyclists for each day of the week.

Justification:

We add 2016 to the date format because it is necessary for performing certain date-related operations, like extracting the day of the week. By adding the year, we can make use of Python’s built-in datetime function, which otherwise would have been rendered useless as the date data is not in the correct format. After adding the year, we were able to extract a particular day of the week for our analysis. The coerce argument in the pd.to\_datetime () function is to handle unparseable or invalid date values, this argument will automatically convert those values into Nat (Not A Time) instead of throwing an error every time it iterates over the data. Later on in the code, we use the dropna () built-in function in Python which removes the rows with invalid datetime values. By doing this, we are not including any invalid values in our calculations and predictions. For each group, we calculate the average number of cyclists. Reindex () makes sure that days are ordered correctly, even if they are random in the data. All these arguments in a groupby function help us to aggregate the data, which is useful in identifying any trends and also simplifies it based on our custom parameters.

A bar chart is an excellent way of comparing categorical data, like day of week, with discrete values like the average number of cyclists. The X-axis represents the day of the week, and the Y-axis represents the average number of cyclists. Data might be too complex therefore, a bar chart condenses it into a single average value, which is very easy to interpret without getting lost in the numbers.

The main purpose behind choosing the Random Forest Classifier model is to make a good prediction in case the relationship between the chosen variables is nonlinear. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting (Scikit-Learn, 2018). Its combination and prediction of many trees make it quite robust. Moreover, it is resilient to missing values in data and works well with categorical values. To summarize, in our case for the specific task of predicting the day of week, Random Forest Classifier is indeed a reliable choice of model.

Expectations:

We do anticipate seeing some pattern in the day of the week and the number of cyclists on the bridges. However, we could also notice some deviations since we are only considering a handful of features while making a prediction.

If the prediction accuracy based upon our choice of model is between 80% to 90%, we can confidently say that our model can tell which day of the week it is based on upon number of bikes on the bridge.

The results are as follows:

Accuracy of Random Forest model for predicting day of the week: 0.2326

A graph of blue bars with black text

AI-generated content may be incorrect.

Conclusion:

Given the low accuracy of our model, about 23.26%, it is very clear that is currently not reliable to predict the day of the week. A better choice of model is required. Some suggestions are to add more relevant features, as the number of cyclists alone is not enough. Use cross-validation, like K-fold, to get a more useful evaluation. We need to eliminate any unimportant features and evaluate variety of metrics like precision, recall and F1- score.

References:

Scikit-Learn. (2018). *sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 documentation*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Tenny, S., & Abdelgawad, I. (2023, November 23). *Statistical significance*. National Library of Medicine; StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK459346/

Tokuç, A. A. (2021, January 14). *Splitting a Dataset into Train and Test Sets | Baeldung on Computer Science*. Www.baeldung.com. https://www.baeldung.com/cs/train-test-datasets-ratio