An Empirical Investigation into Predicting BIXI Montréal Bikeshare Hourly Volumes

Millian Gehrer
Nicholas Johnson

May 9th, 2019

1 Introduction

1.1 Motivation

In response to concerns over traffic congestion, pollution, and climate change, many cities are subsidizing bikeshare systems to encourage a greener, healthier mode of transportation. As more consumers adopt and rely on such systems as a primary means of transit, it is crucial for bikeshare operators to maintain and restock bike stations in real time. To this end, accurate trip volume forecasting is crucial to the successful operation of any bikesharing system. Using Montréal's public bikesharing system BIXI as a case study, we set out to answer the research question:

For a given hour on any day of the year, given the precipitation (rain/snow) forecast, how well can we predict the number of BIXI bikes on the road in Montréal?

1.2 Data Gathering

To answer our research question, we set out to find a dataset which combined the exact start and end dates and times of BIXI trips in Montréal with the weather conditions that day. Though BIXI's public data offering documents all users' trips in a given year (including start and end times and locations), it does not include relevant weather data. Thus, we complemented BIXI's 2018 dataset [1] with data on temperature by hour, and total rainfall and total snow by day, from the Canadian government's historical climate database [2].

2 Data Pre-Processing

In the BIXI dataset, each row corresponds to the metrics for a single user's trip. We first converted the start and end time stamps into Hour, Weekday, and Month. Next, we constructed a column for the Trip Volume at any given minute, and iterated through the trips, incrementing the volume count by one if a given minute was between the start and end times (both inclusive) of a trip. In the weather data, we replaced any non-number entries with 0, and then constructed dictionaries in which the key was a date (along with time for Temperature) in 2018, and the corresponding value was the respective weather metric of that dictionary. We then matched each trip to its weather conditions by start date (and time if applicable). Dropping all other columns, we were left with our six predictors - Temperature, Total Rain (mm), Total Snow (cm), Hour, Weekday, and Month - as well as our response variable, the Trip Volume. We refer to this as our original dataset.

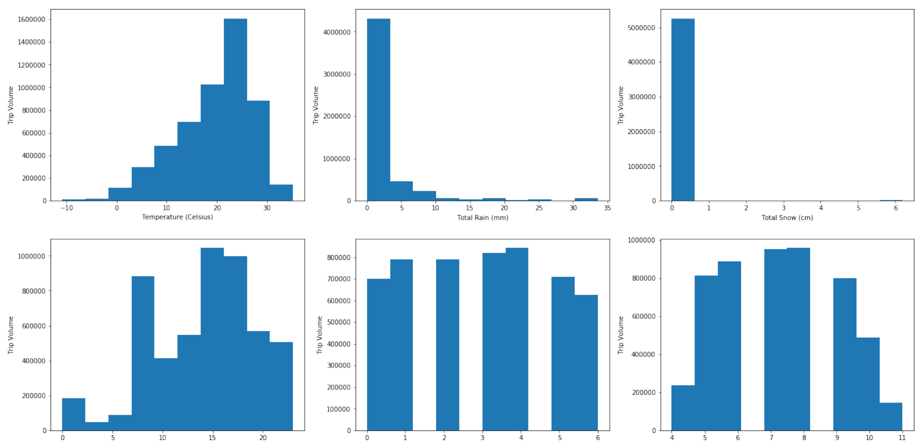


Figure 1: Trip volume plotted against each predictor in the original dataset.

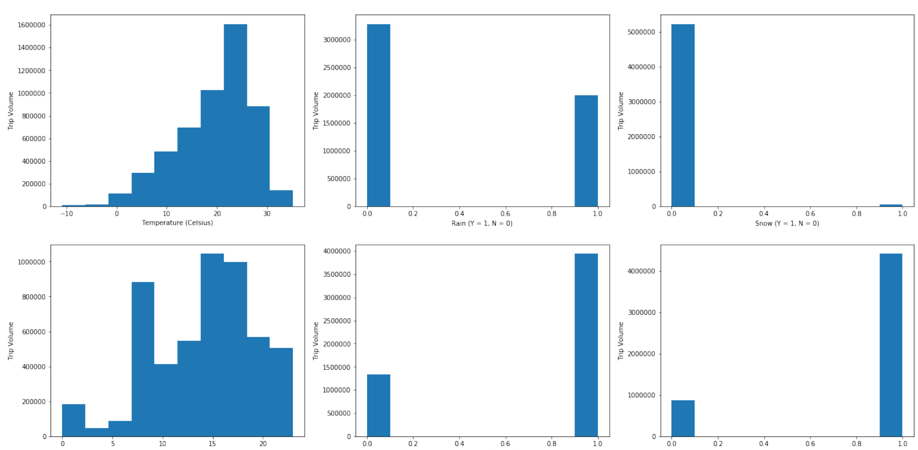


Figure 2: Trip volume plotted against each predictor in the engineered dataset.

To improve interpretability, we transformed four of our six predictors into binary indicators. We let the rain and snow predictors take value 1 if the respective form of precipitation was non-zero on a given day, and 0 otherwise. We let the day of the week take value 1 if the day was a weekday, and 0 if the day was over a weekend. Finally, we converted the month into a seasonality indicator, where we defined May-September as warm (1), and all other months as cold (0). We refer to this as our engineered dataset.

We used the same 70%-30% train-test splits to divide both datasets for fitting regression models.

3 Regression Techniques

We fit four types of regressors to our dataset. Table 1 summarizes the results, which we explain in detail in the following subsections. Note 'RMSE' denotes the root mean square error, and 'Percent' denotes the mean absolute percent error. Though the RMSE is often lower, we care most about the mean absolute percent error, as it best captures the accuracy of our predictions.

Errors by Regression Technique				
Technique	RMSE (O)	Percent (O)	RMSE (E)	Percent (E)
LLS	0.1747	0.4256	0.1899	0.4086
Decision Tree	0.0316	0.0668	0.0453	0.0911
Random Forests	0.0316	0.0669	0.0453	0.0912
MLP	0.0001	0.0949	-	-

Table 1: Note (O) = original dataset, (E) = engineered dataset

3.1 Linear Least Squares Regression

Having completed preliminary data visualization and preprocessing, we first investigated how well a simple linear least squares (LLS) regression model would perform using the original features and thereafter the engineered features. Although the histograms suggest that few predictors exhibit clear linear relationships with the response variable, it is good practice to begin an analysis by using simple methods whenever possible. LLS achieved a root mean square error of 0.1747 and mean absolute percent deviation of 0.4256 with the original predictors. With the engineered predictors, LLS achieved a root mean square error of 0.1899 and mean absolute percent deviation of 0.4086. Note that LLS using the engineered predictors achieved a lower average absolute percent error which suggests that our feature engineering is useful for addressing our problem using this regression technique. However, 41% error in predicting trip volume is far too poor to have any practical usefulness in decision making.

3.2 Decision Trees

Due to the relatively poor performance of LLS and because the predictor histograms do not suggest that a linear model would do particularly well, we next turned to decision trees which are non linear. Utilizing decision trees is also an attractive option by virtue of their often inherent interpretability. A decision tree was fitted using the original predictors and thereafter using the engineered predictors. Let n^* denote the minimum number of samples required to be in a leaf node. We performed 5-fold cross validation to select the hyperparameter $n^* \in \{10, 50, 100, 250, 500, 750, 1000, 1250, 1500, 1750, 2000\}$. In each case, the optimal hyperparameter setting was found to be $n^* = 10$. The decision tree achieved a root mean square error of 0.0316 and mean absolute percent deviation of 0.0668 with the original predictors. With the engineered predictors, the decision tree achieved a root mean square error of 0.0453 and mean absolute percent deviation of 0.0911. The decision tree performed significantly better than LLS using either set of predictors. Figure 3 and Figure 4 indicate that Temperature and Hour are the two most important features for this task. Note that the decision tree performed worse in terms of root mean square error and mean absolute percent error using the engineered features compared to the original features. We revisit this observation in a later section.

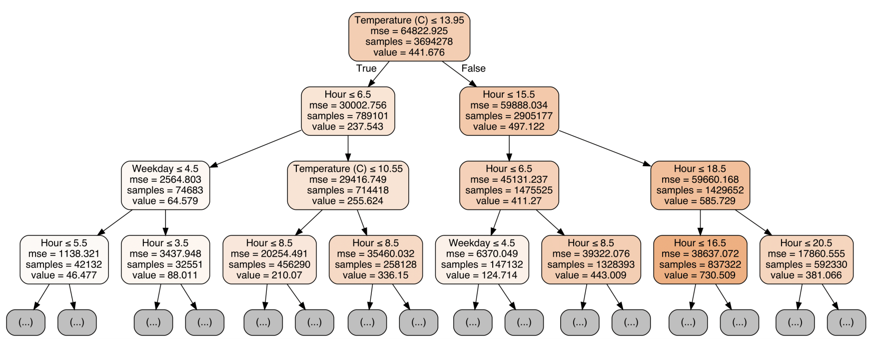


Figure 3: First three layers of decision tree fitted to original dataset.

3.3 Random Forests

The performance of the decision tree on this task was very promising. This prompted us to explore the performance of random forests which often lead to more robust (less unstable) regression functions with correspondingly better performance. Let m^* denote the number of estimators used in a random forest. We fitted random forests using the original predictors and the engineered predictors for values $m^* \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and $n^* = 10$. Using the original predictors, a random forest with 60 estimators produced the lowest root mean square error (0.0316) and the lowest mean absolute percent error

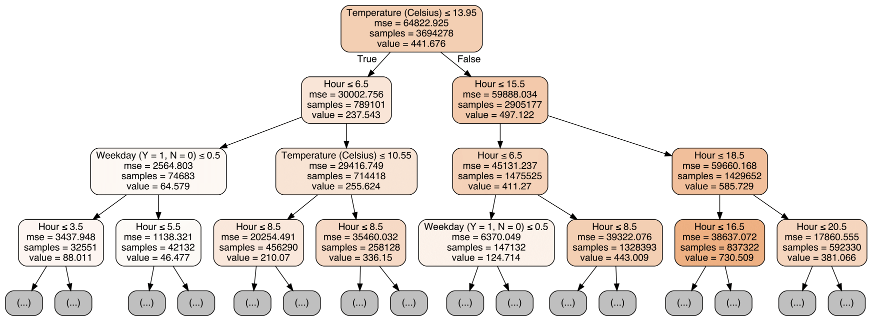


Figure 4: First three layers of decision tree fitted to engineered dataset.

(0.0669). Using the engineered predictors, a random forest with 50 estimators produced the lowest root mean square error (0.0453) and a random forest with 40 estimators produced the lowest mean absolute percent error (0.0912). However, as illustrated in Figure 5, these evaluation metrics are fairly stable across random forests of different sizes (evaluation metrics only differed in the fifth decimal place). Furthermore, the performance of the random forests was effectively the same as the performance of a single decision tree. This might occur because the large amount of training data ($\approx 3.5 \cdot 10^6$) results in a single decision tree being relatively stable. Note that similarly to the single decision tree performance, the random forest performance is better when using the original predictors than when using the engineered predictors.

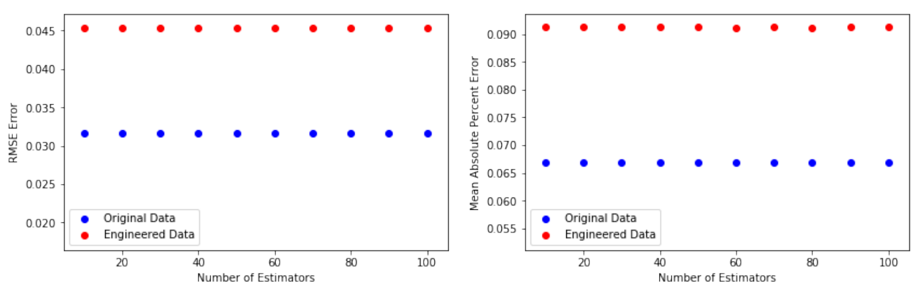
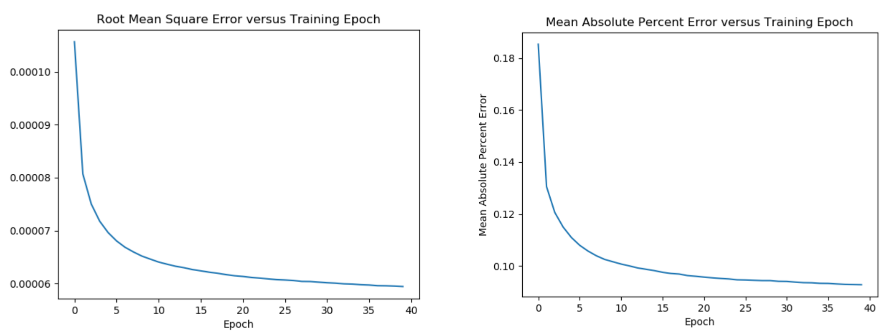


Figure 5: Errors from fitting random forest.

3.4 Multilayer Perceptron

Although the single Decision Tree and Random Forests performed well qualitatively, we thought we might be able to produce a model with even better performance. The fact that the Decision Tree and Random Forests performed better on the original data compared to the engineered data is somewhat surprising and suggests that our feature engineering is not effective in the sense that it does not make the underlying relationship between the predictors and the response variable easier to learn (although our feature engineering does make the predictors more interpretable to humans). Recognizing the ineffectiveness of our feature engineering on model performance, we turned to a Multilayer Percepton (MLP) Neural Network due to its end-to-end nature which avoids human feature engineering. As such, the MLP was trained only on the original dataset. Moreover, using an MLP for this problem is further justified because the dataset contains a large number of examples ($\approx 5 \cdot 10^6$ examples).

We used a 5 layer MLP (4 hidden layers and 1 output layer) where we set $n_h^{[1]} = 400$, $n_h^{[2]} = 300$, $n_h^{[3]} = 200$, $n_h^{[4]} = 100$, and $n_h^{[5]} = 1$. All neurons used the ReLU activation function. Training was done using the Adam optimizer for 40 epochs and the loss function was defined to be the mean squared error of the network on the training points. Note that due to the data intensive nature of Neural Networks, the train-test split was chosen to be 80%-20% for the MLP. After training, the MLP resulted in a root mean square error of 0.0001 and a mean absolute percent error of 0.0949. Although the MLP resulted in the lowest root mean square error of all tested regression techniques, the MLP's mean absolute percent error is greater than that achieved by random forests and the single decision tree.



(a) MLP Root Mean Square Error (b) MLP Mean Absolute Percent Deviation

Figure 6: Evaluation Metrics During MLP Training

4 Conclusion

In order to forecast the number of BIXI bikes on the road in Montréal for a given hour of any day of the year, given a precipitation forecast, we recommend using a random forest with $m^* = 60$ and $n^* = 10$. Using this model, the error of hourly bike volume predictions would be 6.7% on average - the predictions are useful enough to make practical business and logistics decisions. Linear Least Squares regression performed very poorly on this task and produced predictions with high error (41% error on average). Although a single decision tree achieved similar performance to the random forest, the random forest would likely generalize better to unseen data. Finally, using an MLP resulted in slightly less accurate predictions in terms of percentage error 9.5% compared to the random forest while being far more computationally expensive.

References

[1] "Trip History, Year 2018 - BIXI Open Data." https://bixi.com/en/open-data. Accessed: 2019-05-08.

[2] "Historical Climate Data - Government of Canada." http://climate.weather.gc.ca/historical$_{d}ata/search_{h}istoric_{d}ata_{e}.html.Accessed: 2019-05-08.$