Predicting Bicycle Usage for Preventive Maintenance

Executive Summary

Capital Bikeshare exists in a bicycle transportation sharing marketplace in Washington D.C. which competes with firms such as Mobike, LimeBike alongside electric scooter companies such as Jump and Lyft. The company provides users the ability to utilize any of its bikes across Washington D.C. for recreational or transportation services. Capital Bikeshare wants to ensure its customer base always has enough bicycles to meet demand, alongside keeping regular maintenance of the bikes to deliver a strong product. Currently Capital Bikeshare allocates its bicycles in an un-strategic way, and are unsure of the demand for bikes on a given day. Additionally, their maintenance services on their bikes are reactive instead of proactive.

The goal of this analysis is to provide a prediction tool for Capital Bikeshare to estimate how many bikes will be in demand on a given day due to forecasted weather conditions. From the prediction, Capital Bikeshare can allocate the correct number of bikes to a certain part of the city, and bring in bikes that may need maintenance on less busy days.

The data model for this prediction task will include the outcome variable, number of bikes used per day, given multiple input variables of weather, day of week and time of year. This data has been collected historically for bicycle demand, given weather conditions and time of year, by Capital Bikeshare.

The major challenge seen in this data, is that there is ‘2 seasons’ of data, one for riding usage in the warmer season, and one for the colder season usually seen in the winter. With this the data is clustered into 2 groups, with the major distinction being the seasonality, and predictive analysis is applied to each cluster and the original dataset. Linear regression, neural nets and random forest algorithms are applied to all the datasets and it is found that neural nets produce the best results and smallest prediction errors in RMSE. The prediction algorithm and input data should be used alongside weather forecasts for the next day or next week; anything beyond this results in more volatile weather predictions which may not produce as an accurate result.

Problem description

Business goal:

Capital Bikeshare wants to ensure customer satisfaction by increasing availability and reliability of its bikes. One of the biggest challenges faced by Capital Bikeshare is knowing when to take bikes off of their platform for proactive maintenance. Regular maintenance will ensure no catastrophic failures occur to a customer during a ride, which would decrease customer belief in the product. Additionally, having a bike well maintained will increase the longevity of the bikes and decrease capital costs of having to perform major fixes or replace bikes entirely. Having an estimate of usage on a daily or weekly basis will allow the maintenance and operations team to allocate the correct number of bikes that will be in demand and proactively maintain bikes.

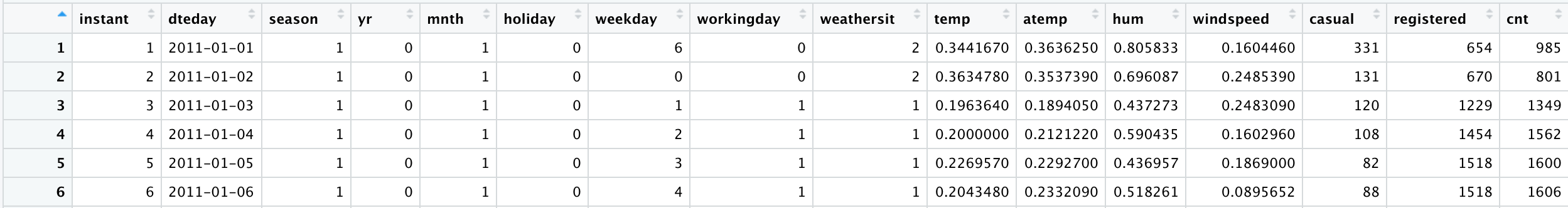
Analytics goal:

The operations analytics objective is to predict the number of bikes demanded on a given day or week subject to a certain season and weather conditions.

Data description

The data is a snapshot of daily demand (count of bikes used) in Washington D.C. based off of certain input variables such as weather, time of year, day of week & holiday.

*Raw Data:*



Data preparation details

There were several steps we had to take in preparing our data before running our prediction models. We first had to look into missing values, and correlation matrices to reduce our input variables. We had to relabel the seasonality column as most of the data was incorrectly label to the wrong season label.

*Missing Value Treatment:*

There existed no NAs in the dataset. Additionally, we wanted to check for any outliers that could influence the outcome variables, we used a box plot chart as shown in figure 1.1 to determine the inter quartile range and observed that there were no outliers.

*Test for Multicollinearity:*

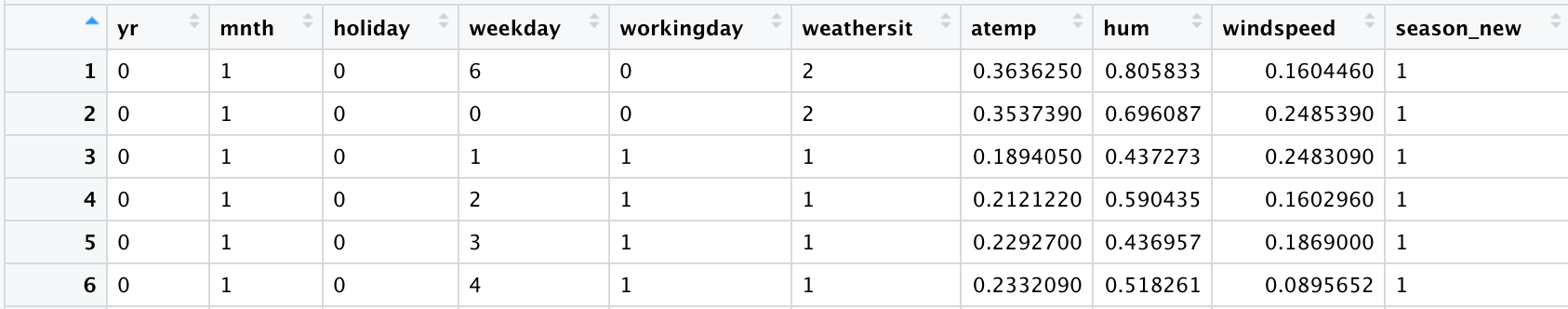
The easiest way to look for multicollinearity is by looking at a correlation matrix. Here is how the correlation matrix looked after including all relevant variables of the dataset.

Looking at figure 1.2a, we can see that there is strong multi-collinearity between variables

1. “temp” and “atemp”: Since “temp” indicates normalized average temperature on that particular day and “atemp” indicates the normalized feeling temperature, it was expected to have high correlation. We will be removing variable “temp” from our dataset. Since the correlation between these two variables was more than 90%, we were indifferent between selecting any one of these variables.
2. “registered”, “cnt” and “casual”: Sum of column “registered” and “casual” gives us “cnt” total count column. We removed “registered” and “casual” from our dataset as we only wanted to focus on the total count of bikes rented in a day instead of focusing only on the registered or non-registered users.

Figure 1.2b displays the correlation matrix after removing the variables which have high multi-collinearity.

*Cleaned Data:*



Datamining solution:

*Clustering Analysis:*

The given dataset had indicators of consumer demand dependent on time of year, in order to get better predictions, we decided to segment our dataset and create two prediction models.

For the segmentation analysis we decided to use K-Means clustering analysis. We generated k-means up to 15 values of K. Figure 2.1 shows overall average within-cluster distance for different choice of k. We can see that the adding more clusters beyond 2 brings less improvement to cluster homogeneity. We decide to go ahead with k=2.

Next we looked at the cluster centeroids from figure 3.1. The clusters are formed based on season and month. Both clusters have roughly 50% of the dataset. The clusters can be described as follows:

Cluster 1: This cluster has summer, autum and winter seasons and months are from July till December.

Cluster 2: This cluster has winter, spring and summer seasons and months are from January till June.

*Prediction Models:*

The prediction models that were determined to be used to predict customer bike demand were Linear Regression, Neural Nets and Random Forests. Given the clustering method created distinct groups based off of month, we ran these models on the two unique clusters and the original dataset.

Evaluation of these models looked into two ideas, will clustering help create better prediction models, and additionally which method should we use? These methods were utilized to create a prediction for each individual cluster and for the entire dataset. To numerically evaluate these models, we looked at Root Mean Square Error as the most important metric, therefore a confidence interval could be given to the operations team to determine roughly how much bike demand is expected.

The multivariable regression model ran on both the clusters separately as well as the combined dataset. Figures 4.1, 4.2, and 4.3 depict the decrease in RMSE for the validation set with the clustering. Furthermore, the significant variables observed for this model were yr, atemp, and month.

The Random Forest algorithm was run iteratively changing the intervals of number of trees from 300-1300, in addition to using an interval for mtry from 1:20 at each split. The RMSEs range was quite minimal based off these changing input parameters, thus most models gave reasonable RMSE train.

The Neural Net algorithms were run with two hidden layers and 1:10 neurons in each layer. The two layers with X number of neurons with the lowest RMSE were determined as the optimal model for each cluster and the entire dataset. Overfit was an additional concern with Neural nets, thus the minimum test RMSE was the focus. Figure 5.1, 5.2, 5.3 show the optimal models for the Entire Dataset, Cluster 1 and Cluster 2 models respectively.

There was no simple Naïve Forecast for this prediction model. Thus, the method with the lowest Test RMSE was considered as the top method. Additionally, the changing RMSE between the entire dataset and clusters was considered to determine whether cluster analysis is useful in prediction.

|  |  |  |
| --- | --- | --- |
| **Non-Clustered** | |  |
| ***Methods*** | ***RMSE Train*** | ***RMSE Test*** |
| Linear Regression | 766.9 | 910.3 |
| Random Forest (ntree = 500) | 766.6 | 704.2 |
| Neural Nets (1 Layer: 3) | 619.6 | 778.6 |
| **Cluster 1** | |  |
| ***Methods*** | ***RMSE Train*** | ***RMSE Test*** |
| Linear Regression | 694.7 | 653.7 |
| Random Forest (ntree = 500) | 836.9 | 855.8 |
| Neural Net (2 Layers: 3,3) | 464.9 | 597.9 |
| Cluster 2 | |  |
| *Methods* | ***RMSE Train*** | ***RMSE Test*** |
| Linear Regression | 936.6 | 843.8 |
| Random Forest (ntree = 500) | 656.7 | 612.1 |
| Neural Nets (2 Layers: 3,8) | 542.5 | 748.9 |

**Table 1: Performance across all models considered with and without clustering.**

Conclusions

The clustered-based approach shown above, indicates that neural nets with clustering results in the lowest RMSE values for the prediction models (average of 672 Test RMSE across 2 clusters). Considering RMSE gives a confidence interval of the predicted value, then the operations team has the ability to estimate with certain levels of confidence the +/- for tomorrows or certain points in the week's predictions.

Clustering and partitioning the data into two separate groups generally resulted in a lower Test RMSE. Additionally, using Neural Nets results in the most predictive method, compared to regression, and slightly better than Random Forests. We generally see more overfit data in Neural Nets, so there is caution from users of the model that overfit could be an issue, especially on unseen data. Considering we want the smallest range between our prediction and confidence interval, we look for the lowest RMSE in the test set to determine best model, across clustered and non-clustered data, Neural Nets are the best.

If the operations team only wants to work with a non-clustered dataset, then the Random Forest model is recommended.

Limitations

The major limitation of this prediction model is its high dependence on the weather input variables as an indicator of bike demand in Washington D.C. Since most weather predictions are forecasts, there is uncertainty and variance of what will occur in the future, thus the error in the weather forecast model will leak into the error of the model just presented.

Operational Recommendations

Recommendations for the operations & maintenance department as follows:

1. Classify each new prediction as whether it would belong to cluster 1 or cluster 2.

2. Use Neural Nets when creating a prediction of demand for each day.

3. The prediction model should only be utilized in next day or next week circumstances, otherwise there is more variance in input weather variables which strongly impact the demand for bikes.

4. Schedule maintenance periods based on lower predicted daily or weekly demand.

5. Manage inventory and determine how many bikes need to be dispatched each day from inventory.

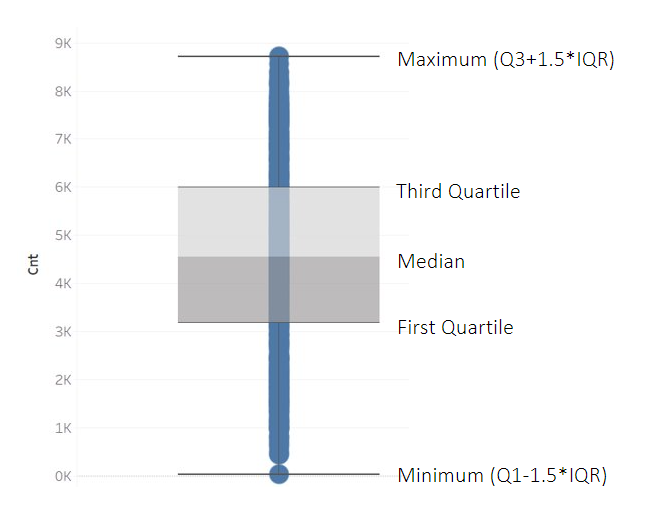
6. Take bikes off of streets to reduce depreciation of physical asset due to weathering, given lower demand predictions.

7. Update the algorithm with new datasets as there is a trend year over year in increasing usage.

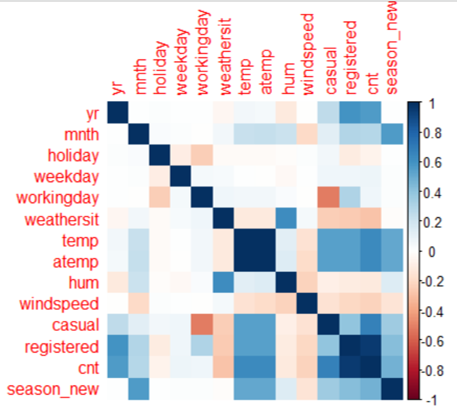
References:

Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.

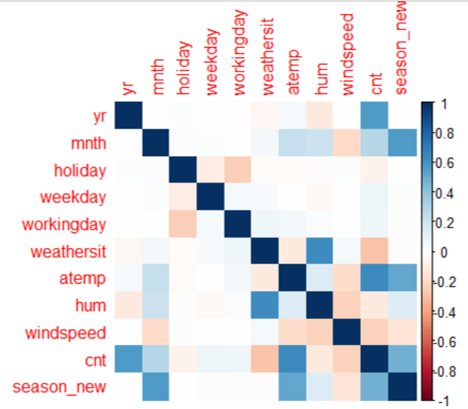
Appendix



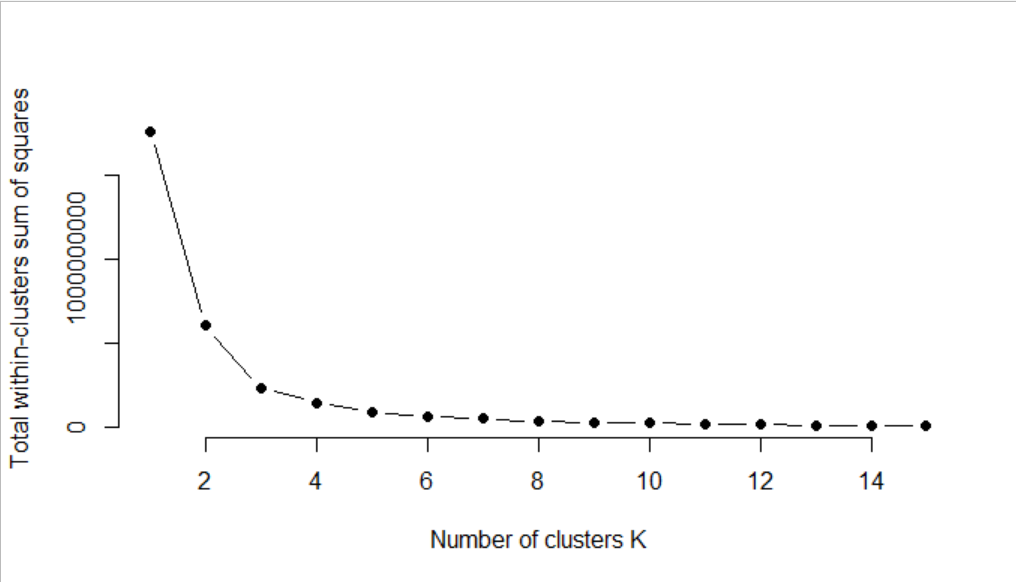
**Figure 1.1**



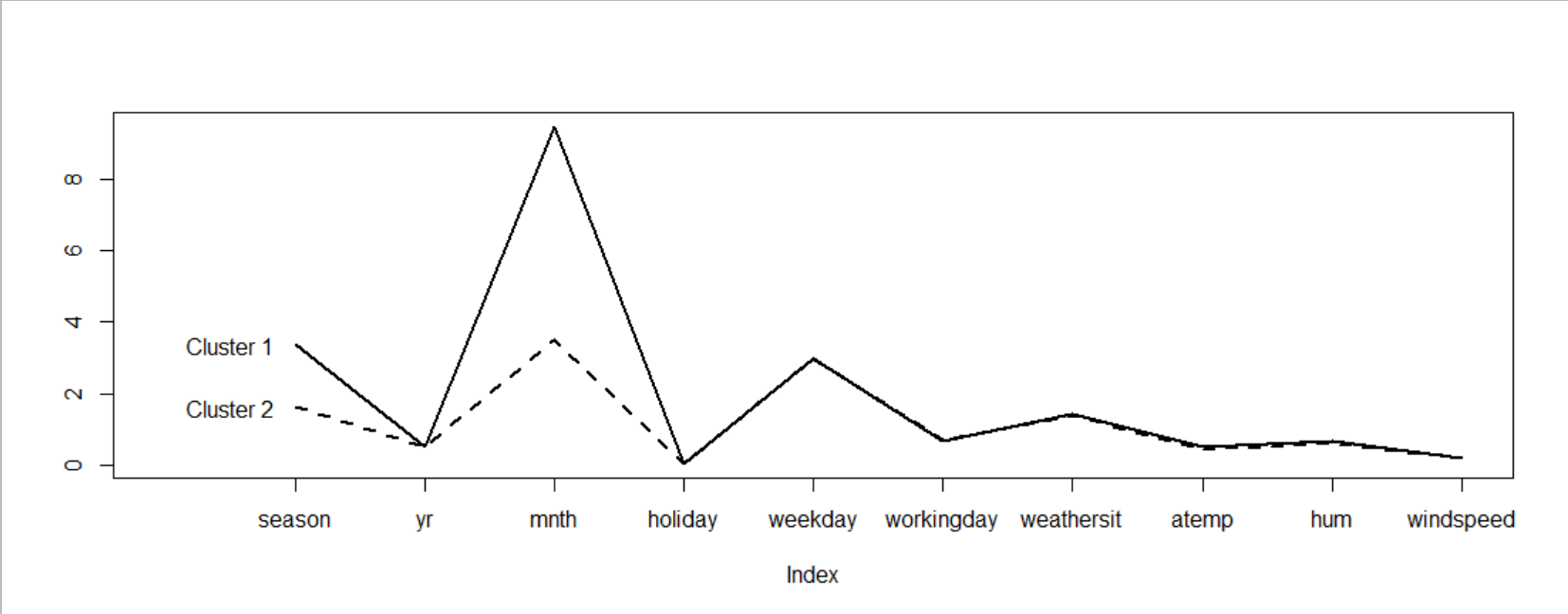
**Figure 1.2a**



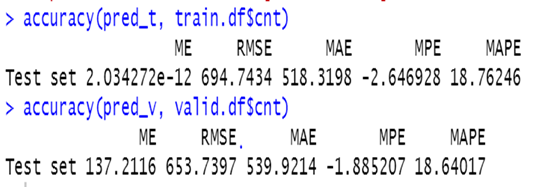
**Figure 1.2b**



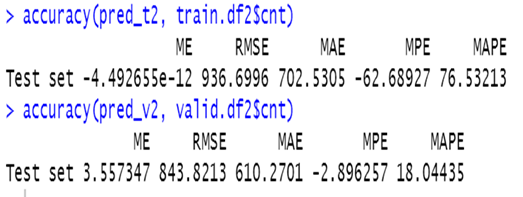
**Figure 2.1: Determining Number Clusters**



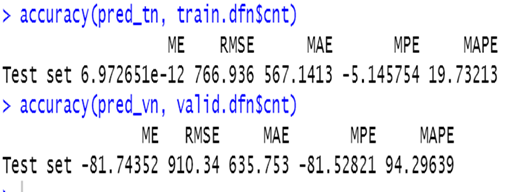
**Figure 3.1: Differentiating Factors of Clusters**



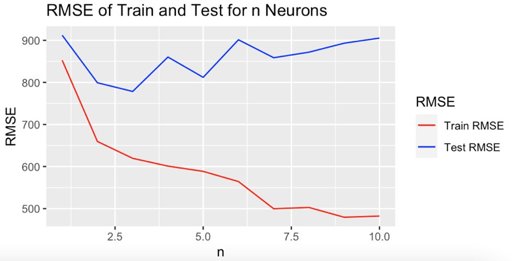
**Figure 4.1: Multivariable Regression on Cluster 1**



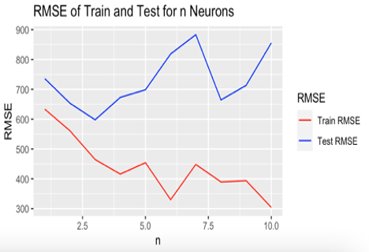
**Figure 4.2: Multivariable Regression on Cluster 2**



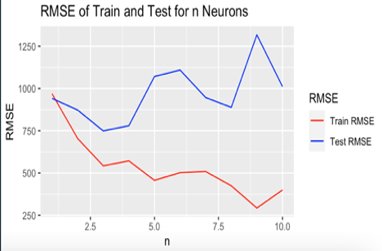
**Figure 4.3: Multivariable Regression on Combined Dataset**

****

**Figure 5.1: Determining Number of Neurons in Entire Dataset**

****

**Figure 5.2: Determining Number of Neurons in Cluster 1**

****

**Figure 5.3: Determining Number of Neurons in Cluster 2**