MACHINE LEARNING PROJECT *TBD*

A PREPRINT

Hyun Suk Ryoo (Max)
hr2ee@virginia.edu

Johnny Wong
jw6qs@virginia.edu

Sujin Park
sjp7yf@virginia.edu

April 27, 2019

1 Abstract

The number of people using electric bikes instead of riding the bus or car has been increasing due to the fact that they are eco-friendly and efficient. For this reason, more companies are starting the electric bike rental business and trying to put out more electric bike to accomodate for the increase the number of users. In order to maximize the profit of the electric bike rental companies, our group wanted to predict the busiest and least busy hours of people riding bikes in order for the companies to determine when to put out more bikes and when to take them back to be recharged. Additionally, we wanted to use the prediction model for other bike users for them to see the hours with the lowest bike traffic so that they can enjoy their bike rides in a more efficient manner.

In order to determine the busiest and least busy hours, we took the data for number of bike users in a given hour in the NOVA/DC area and classified them into three classes based on the count percentile: 0 being the least busy, 1 being the middle, and 2 being the busiest. 0 would be the time when there is a minimal amount of traffic. 1 would be regular with normal amount of traffic. 2 would be the time when there is heavy traffic and many electric bike users in the area. Using this data, we tested several different models to determine the best performing model for our data set. In order to compare the performance, we calculated the accuracy, precision, recall, and F-1 values for Softmax, Random Forest, Decision Tree, and KNN models. After analyzing the performance of the models and fine-tuning them to improve their performance, we create ensembles with different combinations of the models to determine the best ensemble by looking at each of the accuracy, precision, recall, and F-1 values of each class. After determining the ensemble with the best performance, we compared the performance of hard voting and soft voting as well. With our prediction model, both the users and the electric bike rental companies can benefit and maximize the efficiency of bike usage in Virginia.

2 Introduction

Using the data set that contains the number of bike users during a give hour of the day, we wanted to predict the busiest and least busy hours of the day. With this prediction, we can collaborate with electric bike rental companies that are becoming a big trend and apply the information to determine the hours to put out more bikes as well as the hours to take them back to recharge them. During the hours with the greatest number of bike users, there will be more people who are likely to use electric bikes in order to get to places or just for leisure. Similarly, during the hours with the least number of bike users, there will be less people wanting to use electric bikes, which means that bike companies can take most of them back to be recharged. Additionally, from the prediction, other regular bike users will be able to know whether to ride the bike or not based on the conditions such as weather, temperature, humidity, and many more factors. Regular bike users will most likely want to avoid riding bikes during the busiest hours since it might take them longer to get to places due to traffic, especially if they are in a hurry. However, during the less busy hours, it might be faster for them to ride bikes than to take the bus or drive for short distances since they would get slowed down by the car traffic. The data will be classified into 3 types: 0 for not busy, 1 for average, and 2 for busy. Therefore, the electric bike rental companies will put out the most number of bikes during the hours that are classified as 2, less bikes during the hours that are classified as 1, and very few number of bikes during the hours that are classified as 0. For the users, they should ride the bike during the hours that are classified as 0, ride the bike during the hours that are classified as 1 if they have time, and not ride the bike during the hours that are classified as 2. This will result in maximizing the efficiency of people riding bikes as well as the profit for the electric bike rental companies. From the user's perspective, it can decrease bike traffic, and with more electric bikes available, more people will choose to ride electric bikes over regular scooters, cars, or buses, decreasing air pollution as well. Furthermore, this can be applied to locations other than Virginia, though Charlottesville's recent boom of electric scooters was our inspiration.

3 Method

At first, our main focus was training and testing the Softmax model in order to see if the Softmax model is a good prediction model for our data set. Before training and testing the Softmax model, we first checked the data to see if data cleaning was necessary by checking for missing values and to see if scaling was necessary. Then we determined the mean, standard deviation, 25 percentile, and 75 percentile of the bike rider count in order to classify them into three classes: 0 within the 25 percentile, 1 being the middle (between 25 and 75th percentiles), and 2 above the 75 percentile. This classification was necessary for the users to determine whether they should ride the bike and for the electric bike rental companies to determine whether they should put out more bikes or not. After classifying the data, we split the data into training and testing sets in order to prepare for training and testing the Softmax model. In order to improve the performance from the original results that we got from the Softmax model, we fine-tuned the model using grid search by trying different values for the hyper-parameter C to find the optimal value that will improve the model. Since we were not restricted by time for our experiment, we did a grid search for C values between 0 and 10 with 0.1 intervals.

However, we were not satisfied with the performance of the Softmax model, which will be further discussed in the results section, so we tested other models such as KNN, Random Forrest, and Decision Tree. However, in order to create an even better performing models, we fine-tuned the Random Forrest and Decision Tree models as well using various hyper-parameters for each model. For the KNN model, we thought it will be best to keep these values since we didn't find any improvement to the model. For Decision Tree we experimented with the hyper-parameter of max-depth. The values that we specifically looked at for this fine-tuning procedure were values of 1 to 64 incremented by 1. For our Random Forest model, the hyper-parameters that were looked at were n estimator and max features. The n estimator was a list of numbers ranging from 53 and 58 with increments of one. Originally the values had a much higher range, but upon trial and error we decided to have the range of 53 to 58 as our hyper-parameter values. Using a similar process, we decided to use the values 8, 10, 14, and 16 to use a grid search for max features. In addition to these hyper-tuned models, we created different ensembles with various combinations of the models to compare and determine which combination works the best with our data set. Additionally, we decided to further analyze the ensemble voting classifier by comparing the performance of hard voting and soft voting in regards to accuracy, precision, recall, and f1 score.

4 Experiments

After doing data cleaning and feature scaling to get our data ready for the experiments using Imputer, Column Transformer, and Pipeline, we classified the data into three classes of 0, 1, and 2. The three classes were defined by determining the 25 percentile and 75 percentile of the counts of bike users in the data with the counts less than or equal to the 25 percentile being classified as 0, the counts in between the 25 and 75 percentiles being classified as 1, and the counts greater than or equal to being classified as 2. We first began the prediction process by training and testing the Softmax model, calculating for the accuracy, precision, recall, and F-1 score to determine its performance. However, since the performance of the initial Softmax model was not great, we also used grid search to fine-tune the hyper-parameter C in order to find the optimal parameter for this model. In addition, since we wanted to find the model that performs the best with our data set, after testing the Softmax model, we used Decision Tree, Random Forest, and KNN, calculating for accuracy, precision, recall, and F-1 score for each of those models as well. When analyzing the performance of the three models, the Random Forrest, Decision Tree, and KNN, these models showed better performance compared to the Softmax model even before being fine-tuned. Comparing the performance of each of the four models, random forest seemed to perform the best with our problem, followed by Decision Tree, KNN, and Softmax.

After looking at the initial performance of each model and the performance after fine tuning the models, we created ensembles with various combinations of the four models that we created in order to determine the best combination of models that would give the best performance result. The first ensemble was created using Random Forest, Decision Tree, and Softmax, the second ensemble was created using Random Forest, Decision Tree, KNN, and Softmax, and the third ensemble was created using Random Forest, Decision Tree, and KNN. Similar to analyzing each of the models previously, we calculated for the accuracy, precision, recall, and F-1 score to compare the performance of each of the ensembles. After analyzing the performance of each of the ensembles, we determined that the third ensemble that consisted of Random Forest, Decision Tree, and KNN had the best performance out of the three ensembles since it displayed the best accuracy value. When we looked at the hard voting and soft voting performance of the third ensemble, hard voting showed higher accuracy score.

5 Results

During the classification of our data into 3 classes of 0, 1, and 2, the mean was 189.46, standard deviation 181.38, 25 percentile 40.0, and 75 percentile 281.0. When splitting the data, the size of the training set was 13903 and the size of the testing set was 3476. Based on the original data set's values of bike counts, a new column of classifiers were added, which was used as our response variable for all our models. Before fine-tuning, the Softmax model had accuracy value of 0.6683, precision value of [0.7190 0.6478 0.6508], recall value of [0.7815 0.7312 0.4334], and F-1 score value of [0.7489 0.6870 0.5203]. Using a grid search as described previously on the experiment section with a Cross validation of 5, we found that the optimal C value was 0.2. After fine-tuning with the best hyper-parameter value C of 0.2, the Softmax model had accuracy value of 0.6683, precision value of [0.7195 0.6473 0.6513], recall value of [0.7837 0.7318 0.4300], and F-1 score value of [0.7503 0.6870 0.5180]. Therefore, after fine-tuning, there was no increase in accuracy of the Softmax model. However, the changes in precision was [0.0006 -0.0005 0.0004], recall was [0.0023 0.0006 -0.0034], and F-1 score was [1.3352e-03 -1.6924e-05 -2.3102e-03].

In terms of fine-tuning Decision Tree, the accuracy for before fine-tuning was 0.8277, precision was [0.9050 0.8138 0.7818], recall was [0.8495 0.8387 0.7844], and F-1 score was [0.8763 0.8261 0.7831]. This was when the model was fit with default variables DecisionTreeClassifier. After fine-tuning the Decision Tree model with depth as 20.0, the accuracy was 0.8334, precision was [0.9091 0.8195 0.7885], recall was [0.8584 0.8447 0.7867], F-1 score [0.8830 0.8319 0.7876]. Therefore, the accuracy increased by 0.0057, precision increased by [0.0041 0.0057 0.0067], recall increased by [0.0089 0.0059 0.0023], and F-1 score increased by [0.0067 0.0058 0.0045]. In all fields of measurement there were slight increases in the model's performance and proof that fine tuning did improve the model.

The parameters Random Forest had before fine-tuning were 100 n estimators and max depth of 10. This declaration of the Random Forest model had an accuracy of 0.8075, precision of [0.9256 0.7445 0.8701], recall of [0.8462 0.9209 0.5519], and F-1 score of [0.8841 0.8233 0.6754]. After fine-tuning the Random Forest model with max features as 14 and n estimators as 56, the accuracy was 0.8536, precision was [0.9305 0.8264 0.8337], recall was [0.8807 0.8860 0.7641], and the F-1 score was [0.9049 0.8552 0.7974]. Therefore, from before fine-tuning to after fine-tuning, the accuracy changed by 0.0460, precision changed by [0.0049 0.0819 -0.0364], recall changed by [0.0346 -0.0348 0.2122], and F-1 score changed by [0.0208 0.0318 0.1220]. There was an increase in all measurements, accuracy, precision, recall, and F-1 score, which showed that our Random Forest model increased upon fine tuning the parameters.

The last classification model that was used for our study was the KNN model. For the KNN model, the accuracy was 0.7491, precision was [0.7689 0.7589 0.7090], recall was [0.8161 0.7401 0.6986], and F-1 score was [0.7918 0.7494 0.7038].

As there is only benefits, in each of our models we decided to fine-tune them with Grid Search in order to find the best combination of hyper-parameters. This allows each model to provide us a better score, even if only slightly in some cases.

From these models we created voting classifiers. For the first voting ensemble model, the models that were used were Random Forest, Decision Tree, and Softmax. The accuracy value was for this voting classifier was 0.8455, precision was [0.9090 0.8191 0.8367], recall was [0.8796 0.8825 0.7404], and F-1 score was [0.8941 0.8496 0.7856]. We also decided to test a different voting ensemble model consisting of different classification models. The second ensemble classifier that we used had the models of Random Forest, Decision Tree, KNN, and Softmax models, the accuracy value was 0.8170, precision was [0.8423 0.7837 0.8776], recall was [0.9231 0.8665 0.6151], and F-1 score was [0.8809 0.8230 0.7233]. The third and final model that we used for ensemble voting classifier consisted of Random Forest, Decision Tree, and KNN models had accuracy of 0.8579, precision of [0.9131 0.8440 0.8296], recall of [0.8907 0.8724 0.7968], and F-1 score of [0.9018 0.8580 0.8129]. From the three voting ensembles that we constructed,the measurements were higher in all fields, accuracy, precision, recall, and F-1 score, in our last model, which was the reason why we chose this to be our classifier. Lastly, from the third ensemble, we decided to see whether hard voting or soft voting had a significant impact. Hard voting had an accuracy of 0.8579, precision of [0.9130 0.8432 0.8314], recall of [0.8896 0.8736 0.7957], and F-1 score of [0.9012 0.8581 0.8131], and soft voting had accuracy of 0.8564, precision of [0.9256 0.8371 0.8281], recall of [0.8740 0.8771 0.7991], and F-1 score of [0.8991 0.8566 0.8133].

URL to our Google Collaboratory:
https://colab.research.google.com/drive/1fTgs6AM7ACXNNSXVGEUFq26kislo9z2e

6 Conclusion

After comparing several type of models and ensembles, the hard voting ensemble consisting of the Random Forest, Decision Tree, and KNN models performed the best with the highest accuracy value compared to other ensembles that either included the SVM model or did not include the KNN model. From this performance analysis and comparison, we can see that even though soft voting tends to perform better in many cases, there are exceptions as seen with our data set and the higher accuracy value from hard voting compared to soft voting. Since the accuracy value from hard voting ensemble consisting of the Random Forest, Decision Tree, and KNN models was about 85.79%, this seems accurate enough to help electric bike companies as well as regular bike users in Virginia in determining the busiest and least busy hours to maximize profit as well as avoid bike traffic. Even though this prediction will not be used to predict something that can greatly harm one's health and therefore there will not be a huge risk in making wrong predictions, we would still like to find ways to improve the accuracy of our prediction model in the future. For future experiments, we can develop our own Neural Network model in order to solve our problem of predicting the busiest and least busy ours and also try using transfer learning of a model that has already been established in order to achieve more accurate model that provides better results, helping the electric bike companies and regular bike users in Virginia, making Virginia more eco-friendly.

7 References

https://brage.bibsys.no/xmlui/bitstream/handle/11250/2563560/Master$_2$017$_Kanestrom.pdf?sequence = 1isAllowed = y$

https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

https://www.researchgate.net/publication/326542794$_Use_of_Deep_Learning_to_Predict_Daily_Usage_of_Bike_Sharing_Systems$

https://stats.stackexchange.com/questions/349540/hard-voting-soft-voting-in-ensemble-based-methods

https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a

8 Member Contribution

We all worked on the proposal, checkpoint, video, and final report together, having meetings to talk about what we are going to do and checking over each others work to make edits and additions. We also all worked to perform the necessary experiments, checking with the TAs as well.