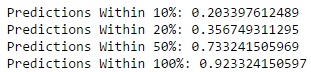
The topic for this project is to predict the amount of bikes being shared in a bikeshare system. I used a modified data set obtained from kaggle (<https://www.kaggle.com/c/bike-sharing-demand/>). The description of the variables included is shown below:



Since my main focus was on the total number of bikes being shared, I had removed the “Casual” and “Registered” fields from the table. I also decided to alter the datetime data as I felt the time of day was important but the day itself was not as important. I just can’t imagine someone saying “Oh, it’s January 5th, I shouldn’t ride a bike today” but I could imagine someone saying “It’s 1 in the morning, it’s probably not safe to ride right now.”

The method I decided to use was random forest classifier. With so much data, I was forced to limit my estimators to 128 which took about 30 seconds per model to run. As the data was estimating a number rather than giving a binary answer, I decided to change up what I felt were good metrics to measure. These new measurements were if the test data was within 10%, 20%, 50%, and 100% of the actual count using the percent error equation. The optimization process I used on this was to try removing one attribute at a time until I saw an improvement mainly on the “Within 10%” measurement. The reason behind this tedious method was that I felt all attributes included with that data were rather important and that turned out to be correct. The final result had me using all of the attributes.

The final results are shown below.



This shows that slightly over 20% of the predictions were within 10% of the actual result. Only slightly less than 8% of the predictions were greater than 100% of the actual result. This means that ~92% of the estimations were between 0 and 2x with x being the actual amount.

Another interesting view is the amount of estimations that were lower than the actual amount versus the amount of estimations that were higher than the actual amount.



Both of those values are relatively close to 50%. This shows that the model has no strong bias towards heavy over predictions or heavy under predictions.