Predicting Hourly Bike Rentals in the Seoul Bike Sharing System

여의도 지역 안전모 시범 운영
기간 : 2018.9.1.(토)~2018.9.30.(일)

Acey Vogelstein, Christopher Regan, Max Cohen
Stat 471: Modern Data Mining
Professor Eugene Katsevich

May 2021

Table of Contents

Executive Summary

The goal of this study is to predict the requisite number of bikes in a public urban bike sharing system to ensure a stable supply. To do so, we look at data on rentals in the Seoul bike sharing system in the years 2017 and 2018 from the UCI machine learning repository (SeoulBikeData.csv). This data gives the number of bikes rented per hour as it relates to weather conditions and time of the year. We use the clean dataset, **clean_bike_data.csv**, to predict the continuous response variable, the number of bikes rented in a given hour.

We considered four models: a simple linear regression, an elastic net regression, a decision tree, and a random forest. Ultimately, we concluded that the random forest has the best predictive performance, as it has the lowest test MSE, 56232.8. This was significantly better than the other three models, and we believe this is due to the fact that tree-based models automatically incorporate non-linearities and interactions between features.

Our final model includes 12 features, detailed later in the report.

- Hour of the day
- Temperature
- Humidity
- Wind Speed
- Visibility
- Dew Point Temperature
- Solar Radiation
- Rainfall
- Snowfall
- Season
- Whether the day is a holiday or not
- Year (2017 or 2018)

The three most important features for predicting the rented bike count are hour of the day, temperature, and humidity. Based on this, we conclude that the Seoul city government could better maintain a stable supply of bikes for rental by offering more rental bikes during the warmer months. Furthermore, we think this study is informative on whether or not a bike sharing system would be successful in another city. However, this may be stretching the external validity, so further research in this area could make use of (1) bike sharing data from other cities and (2) demographic information of the bike users.

Introduction

In recent years, rental bikes have developed into a key means of transportation in city life. For the common city dweller, biking can be an enjoyable and effective way to get around. However, it can become inconvenient to the public when supply does not meet demand for rental bikes. With too few available bikes, citizens often are forced to wait and waste time. For convenient mobility, it is important for cities to determine the optimal supply of bikes at any given time. Unfortunately, this can be difficult to determine due to a number of variables, including factors related to weather, holidays, and time of the day.

Having identified this issue, we would like to predict the requisite number of rental bikes at each hour of the day in order to achieve a stable supply. With data for the Seoul Bike Sharing System spanning 2017-2018, we strive to optimize this prediction for a given observation. Fortunately, the dataset offers a number of seemingly useful features and nearly 9,000 observations that can facilitate our predictive performance.

With this information at our disposal, we can address relevant questions to the bike sharing problem. For instance, how does the bike count vary by season? What can we expect demand to look like at 4pm on a humid summer day? If we can answer these questions and more, it could lead to significant implications “down the road” for both suppliers and renters in the rental bike industry.

Data Cleaning and Exploratory Analysis

Data Extraction and Cleaning

The dataset used for this project was taken from the UCI machine learning repository, linked here. It contains the count of public bikes rented at each hour in the Seoul Bike Sharing System with the corresponding weather data and holiday information. The raw data contains 8760 observations across the years 2017 and 2018.

The data was already fairly clean -- each observation corresponded to a single row, and there were no null values anywhere in the dataset. However, there were a number of rows for which the bike sharing system was not functioning, which automatically resulted in a bike count of zero. Since this project focuses on predicting bike demand based on the features, these rows did not seem relevant as bikes were not even offered for consumption. Thus, we dropped all rows where the sharing system was not functioning, leaving the clean dataset with 8465 observations. Furthermore, we extracted the year from the date column in the raw data to include as an extra feature. Aside from this, the only other cleaning done was to rename the columns in the tibble to more convenient names.

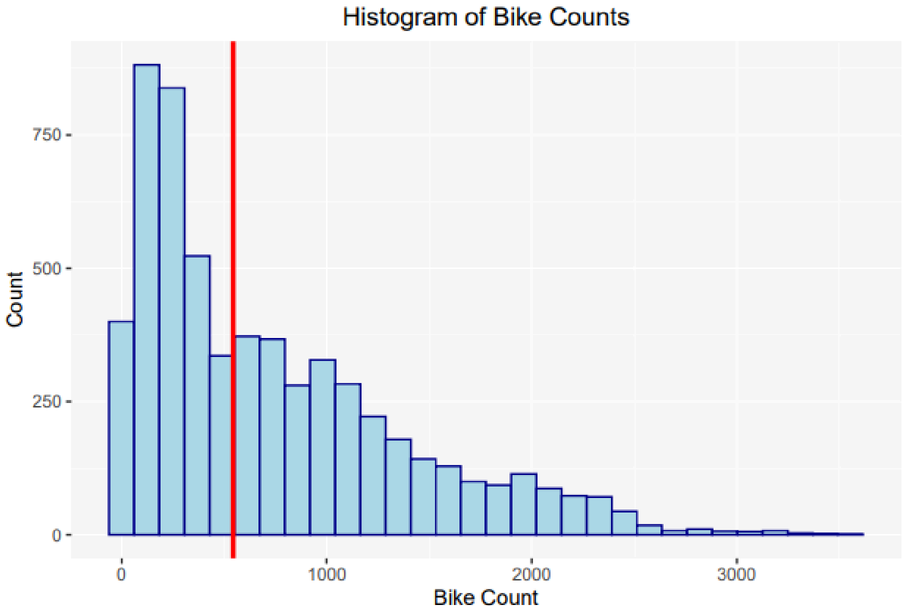
Data Summary

The response variable in this dataset is **bike_count**, which is the number of bikes rented at each hour (since each observation represents one hour of one day). Included below is a table of the 12 features after data cleaning.

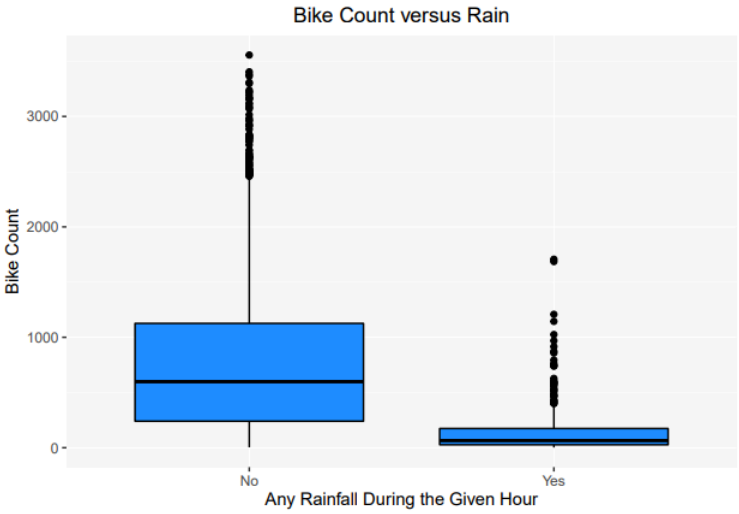
Variable	Type	Description
hour	categorical	Hour of the day (integer 0-23)
temp	continuous	Temperature (℃)
humidity	continuous	Humidity (%)
wind_speed	continuous	Wind speed (m/s)
visibility	continuous	Visibility (10m)
dew_point	continuous	Dew point temperature (℃)
solar_rad	continuous	Solar radiation (MJ/m^2)
rainfall	continuous	Rainfall (mm)
snowfall	continuous	Snowfall (cm)
seasons	categorical	Winter, Spring, Summer, or Autumn
holiday	categorical	Holiday/No Holiday
year	categorical	Whether the observation comes from 2017 or 2018

Data Exploration

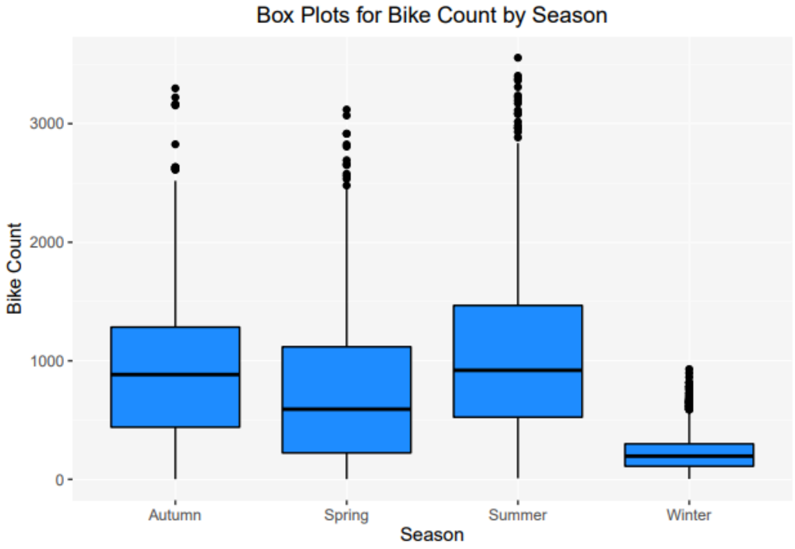
First, we divide the data into a training set and a testing set. We use 70% of the data for training and hold out 30% for testing. To get a sense for the data, we conduct some exploratory analysis using the training set.



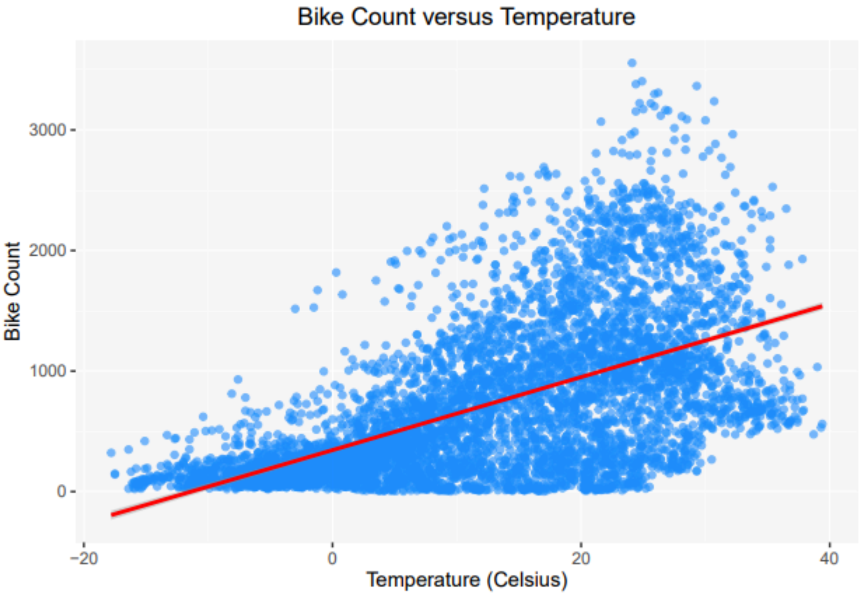
First, we look at the distribution of our response, the count of bikes rented at each hour. Looking at the histogram to the right, one can see that the distribution is heavily skewed, with a long right tail. The median bike count was 544, which is marked on the plot by the vertical red line. Next, we look at the relationship between the response and some of the features. First, we considered looking at a scatter plot between the amount of rainfall in a given hour versus the count of rented bikes, as we expect that rain would lead to fewer bikes being rented. However, this led to a non-linear relationship that was difficult to interpret. Thus, we instead looked at box plots for the rented bike count in 2 cases: (1) when



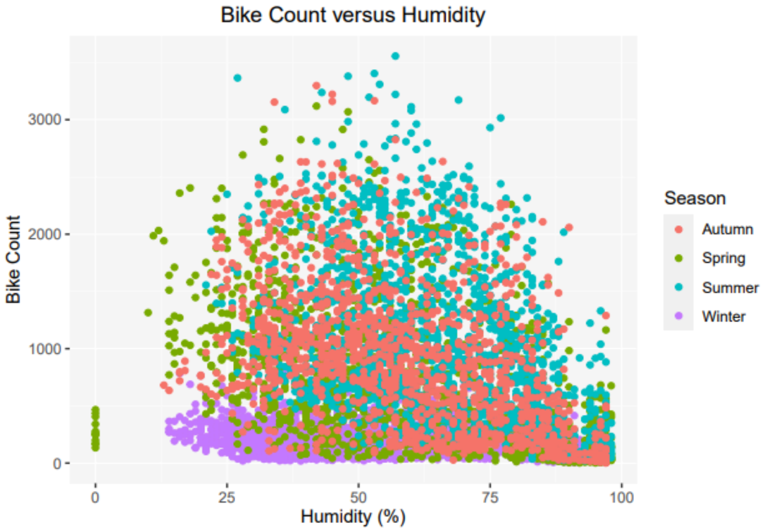
there is no rainfall in the given hour and (2) when there is any rainfall in the given hour. These boxplots are included above.



We can see that the rented bike count is far lower when there is any amount of rain in a given hour. In fact, the “box” portion of the plot for when there is rain lies entirely under the bottom of the “box” portion for when there is no rain. Thus, based on this, we expect rainfall will be an important predictor of bike count. Next, we looked at box plots of the bike count grouped by seasons, included to the left. Based on this, we conclude that, on average, the rented bike count is highest in the summer (median 920 bikes) and lowest in the winter (median 196 bikes). This makes some intuitive sense, as weather conditions are generally better in the summer compared to the winter. This prompted us to investigate the relationship between the rented bike count and the other weather conditions. Thus, we first make a scatter plot of bike count versus temperature, which is included to the right. There appears to be an upward trend that is not completely linear, as the plot exhibits a “fanning out” shape. Nonetheless, we fit a simple linear regression line to the scatter plot, as shown by the red line. This slope of the line is clearly positive, so despite the non-linear “fanning out” behavior, linear regression still captures the upward trend.

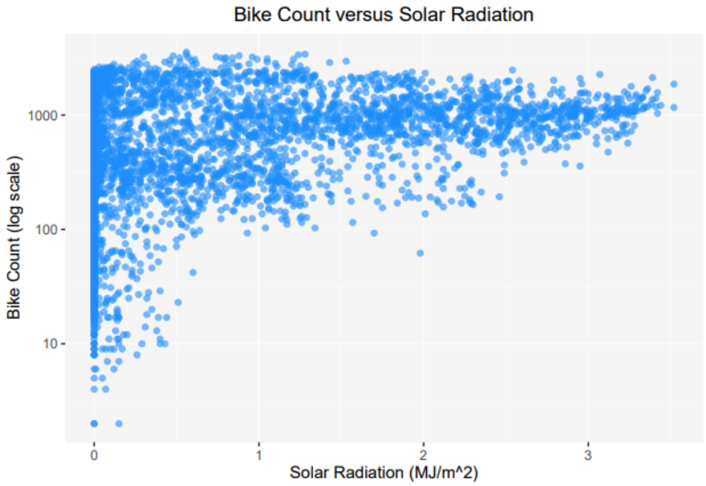


Next, we consider another important weather feature -- humidity. The plot below is a scatter plot of bike count versus humidity, colored by the season. The overall relationship here is less clear than in the temperature plot as for all humidities on the x-axis, there are a similar number of bike count points from 0 to 2000. However, above a bike count of ~2000, a trend is discernible -- humidities in the middle of the distribution generally correspond to a higher bike count. Furthermore, because we colored by season, we can see how the different ranges in humidity correspond to different seasons (e.g. winter tends to be less humid).

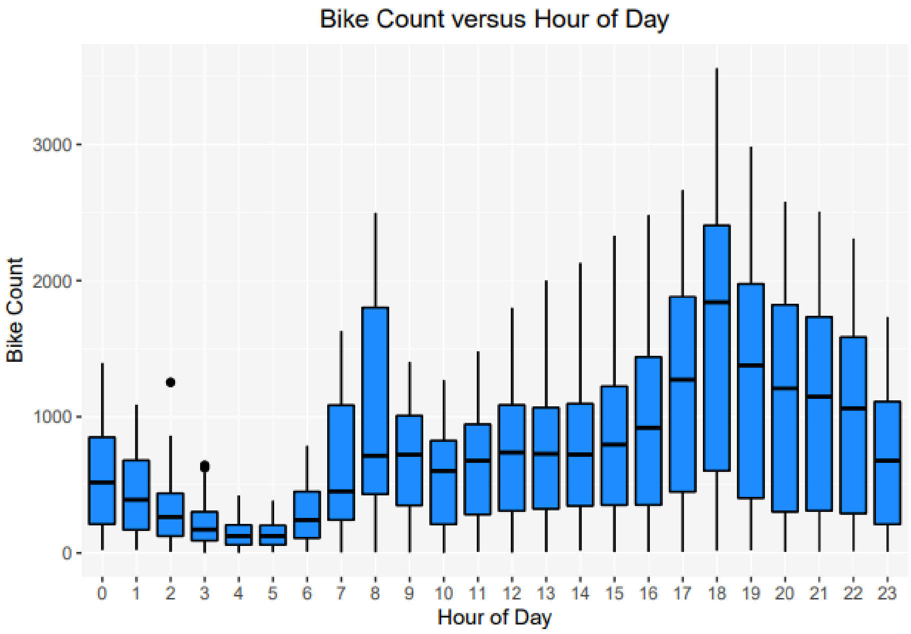


Next, we look at bike count versus the amount of solar radiation, plotted below. To most clearly visualize the relationship, the y-axis has been put on log-scale. While the trend is still not linear, almost all the points with a bike count of below 100 correspond to a solar radiation of less than 1 MJ/m^2. This suggests that fewer bikes are rented on days that are less sunny.

Finally, we explore one non-weather characteristic -- the hour of the day, since the observations of the data are hourly. Based on this plot, one can see that the number of bikes rented clearly varies with the time of day. Specifically, the number of rentals peaks at 6pm, around the end of the work day. Furthermore, there appears to be a smaller local maximum around 8am or 9am, which is the beginning of the workday.



Therefore, based on our exploratory analysis, we expect that the weather features along with the hour of the day will be able to predict the hourly rented bike count. The next section focuses on building a predictive model to do this.



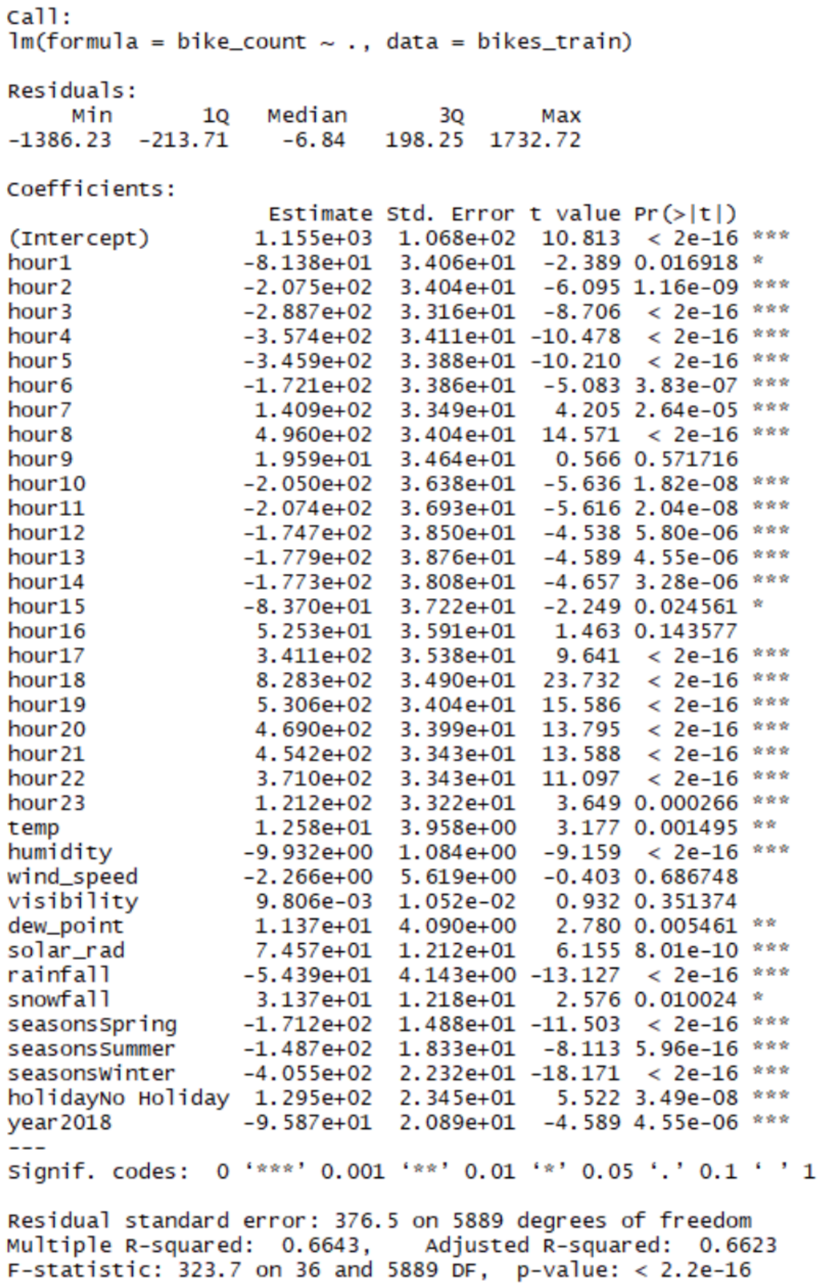
Model Building, Interpretation, and Evaluation

Objective

Using the features in the table from the “Data Summary” subsection, we want to build a predictive model that predicts the rented bike count in a given hour. We fit a series of models (linear regression, elastic net regression, a decision tree, and a random forest) to the training data, and then, we evaluate the success of each model on the test data. Finally, we compare the predictive performance of each model. The following sections describe the details of each model, and the final section addresses the predictive performance.

Linear Regression

At first, we build a simple linear regression model on the training data and evaluate it on the test set. For this predictive model, we initially decide which features we should code as factor variables. Because they can be classified as categorical and not continuous, we choose “hour,” “seasons,” “holiday,” and “year.” Using these categorical variables in addition to the continuous variables, we then run a linear model fit. This optimal OLS fit to the training set is our linear regression predictive model. The summary of our linear model is illustrated to the left. In the summary, we can see that the Multiple R-Squared is 0.6643, which means that variance in our chosen regressors accounts for 66.43% of the variance in rented bike count in the training data.

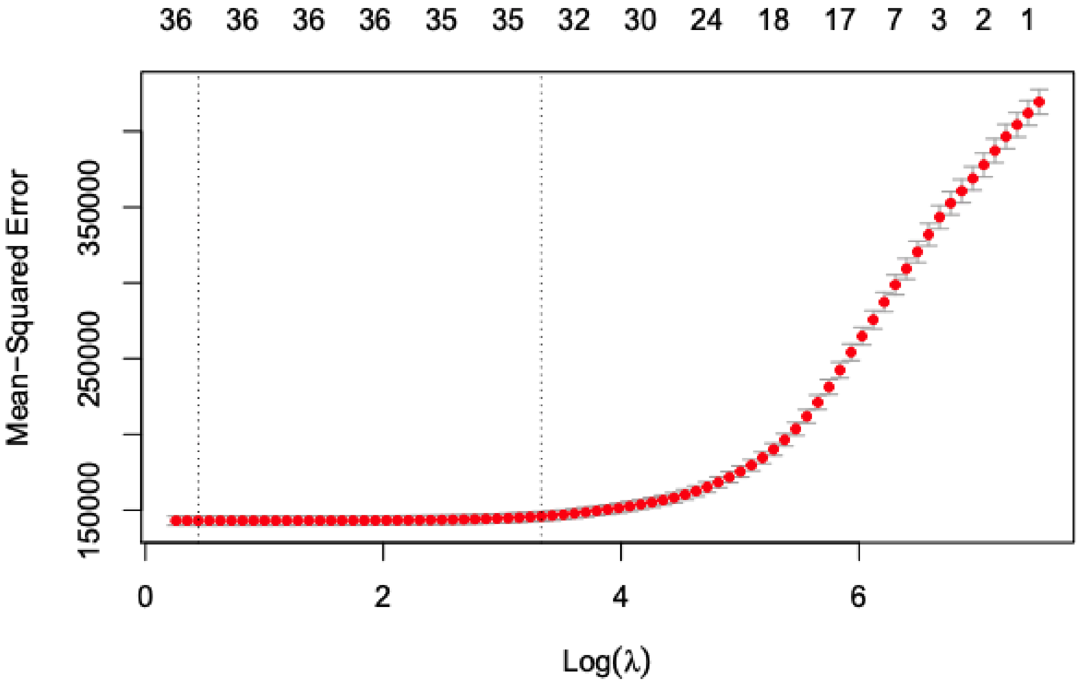


```
Call:
lm(formula = bike_count ~ ., data = bikes_train)

Residuals:
     Min       1Q   Median       3Q      Max 
-1386.23  -213.71    -6.84   198.25  1732.72 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1.155e+03  1.068e+02  10.813  < 2e-16 ***
hour1            -8.138e+01  3.406e+01  -2.389 0.016918 *  
hour2            -2.075e+02  3.404e+01  -6.095 1.16e-09 ***
hour3            -2.887e+02  3.316e+01  -8.706  < 2e-16 ***
hour4            -3.574e+02  3.411e+01 -10.478  < 2e-16 ***
hour5            -3.459e+02  3.388e+01 -10.210  < 2e-16 ***
hour6            -1.721e+02  3.386e+01  -5.083 3.83e-07 ***
hour7             1.409e+02  3.349e+01   4.205 2.64e-05 ***
hour8             4.960e+02  3.404e+01  14.571  < 2e-16 ***
hour9             1.959e+01  3.464e+01   0.566 0.571716    
hour10           -2.050e+02  3.638e+01  -5.636 1.82e-08 ***
hour11           -2.074e+02  3.693e+01  -5.616 2.04e-08 ***
hour12           -1.747e+02  3.850e+01  -4.538 5.80e-06 ***
hour13           -1.779e+02  3.876e+01  -4.589 4.55e-06 ***
hour14           -1.773e+02  3.808e+01  -4.657 3.28e-06 ***
hour15           -8.370e+01  3.722e+01  -2.249 0.024561 *  
hour16            5.253e+01  3.591e+01   1.463 0.143577    
hour17            3.411e+02  3.538e+01   9.641  < 2e-16 ***
hour18            8.283e+02  3.490e+01  23.732  < 2e-16 ***
hour19            5.306e+02  3.404e+01  15.586  < 2e-16 ***
hour20            4.690e+02  3.399e+01  13.795  < 2e-16 ***
hour21            4.542e+02  3.343e+01  13.588  < 2e-16 ***
hour22            3.710e+02  3.343e+01  11.097  < 2e-16 ***
hour23            1.212e+02  3.322e+01   3.649 0.000266 ***
temp              1.258e+01  3.958e+00   3.177 0.001495 ** 
humidity         -9.932e+00  1.084e+00  -9.159  < 2e-16 ***
wind_speed       -2.266e+00  5.619e+00  -0.403 0.686748    
visibility        9.806e-03  1.052e-02   0.932 0.351374    
dew_point         1.137e+01  4.090e+00   2.780 0.005461 ** 
solar_rad         7.457e+01  1.212e+01   6.155 8.01e-10 ***
rainfall         -5.439e+01  4.143e+00 -13.127  < 2e-16 ***
snowfall          3.137e+01  1.218e+01   2.576 0.010024 *  
seasonsSpring    -1.712e+02  1.488e+01 -11.503  < 2e-16 ***
seasonsSummer    -1.487e+02  1.833e+01  -8.113 5.96e-16 ***
seasonsWinter    -4.055e+02  2.232e+01 -18.171  < 2e-16 ***
holidayNo Holiday  1.295e+02  2.345e+01   5.522 3.49e-08 ***
year2018         -9.587e+01  2.089e+01  -4.589 4.55e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 376.5 on 5889 degrees of freedom
Multiple R-squared:  0.6643,	Adjusted R-squared:  0.6623 
F-statistic: 323.7 on 36 and 5889 DF,  p-value: < 2.2e-16
```

Elastic Net Regression

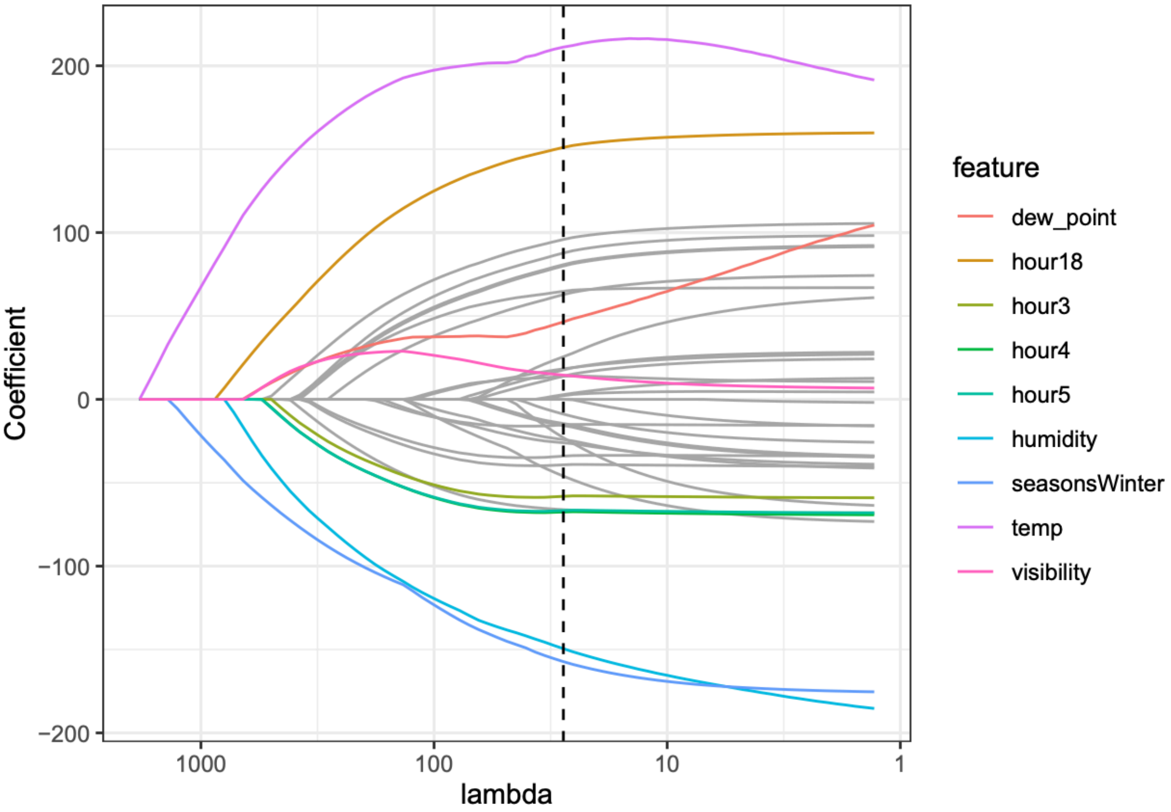
We next build a penalized regression model to generate a more parsimonious model. Because we believe that most, but not all, features in our dataset are conducive to strong predictive performance, we decide to use an elastic net regression model, which can be viewed as a hybrid model of Ridge and LASSO. Elastic net tends to both shrink and remove some predictor variables, which allows the model to become more interpretable. By choosing a value for the parameter alpha, we allow the model to induce some sparsity while not becoming as sparse as a true LASSO model. We decide that a reasonable value of alpha is 0.2. Our dataset does not have a large number of features; thus, we do not want to encourage too much sparsity. Therefore, we choose a relatively small value of alpha so that we emphasize shrinkage instead of selection.

To select the optimal value of our parameter λ, we start off by generating a ten-fold cross validation on our training set. To clarify, the optimal lambda minimizes the average cross-validation error. This analysis can be visualized in the plot below.



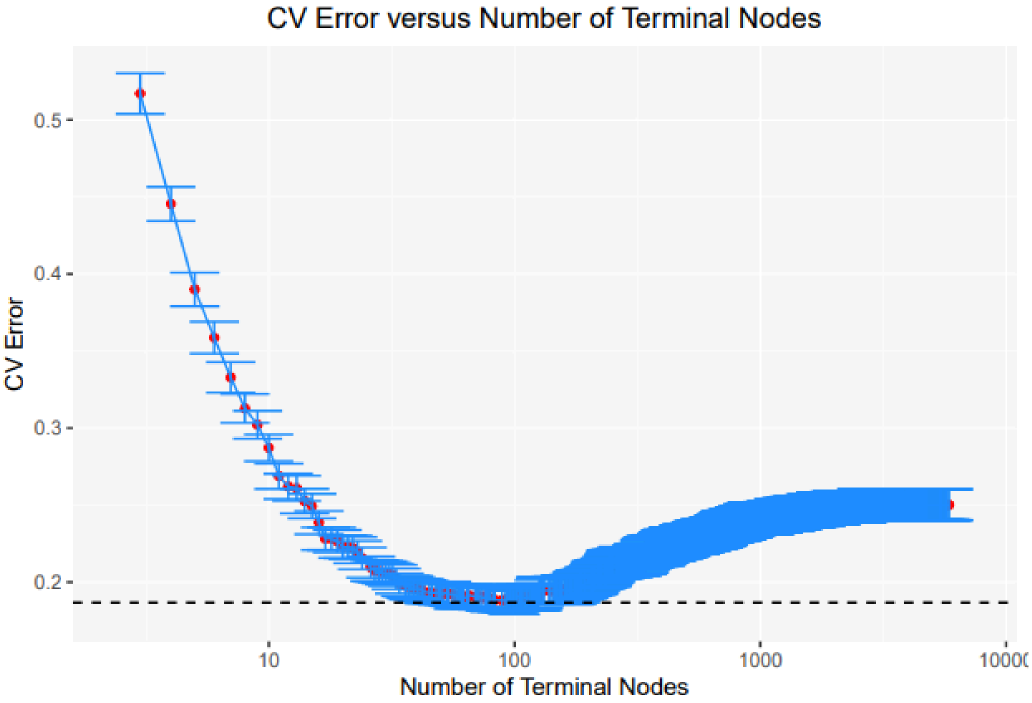
According to the plot, our desired λ value under the one standard error rule is about 27.9. Thus, we build our model using this penalty value and find that the optimal elastic net model reduces two of our variables' coefficients to zero (**hour15** and **wind_speed**). Because our data includes multiple categorical features, the numbers listed horizontally across the top of the plot do not necessarily accurately represent the number of included features at each value of λ.

In order to interpret this model, we create a plot of the coefficients of each feature for each value of λ. The dotted vertical line represents our selected value of λ. According to our plot, features like "dew_point," "seasonsWinter," and "hour18" have among the largest coefficients in terms of magnitude.



Decision Tree

To get the best possible predictive performance for the tree, we want to find the optimal tree using cross validation with the training data. At a high level, the method for doing this is fitting the largest possible tree to the data, and cutting it down using the cost complexity pruning algorithm. The largest possible tree has 5851 splits (5852 terminal nodes). We fit 3036 different trees of varying sizes, and compute the cross validation error for each tree size. The CV plot is included above.



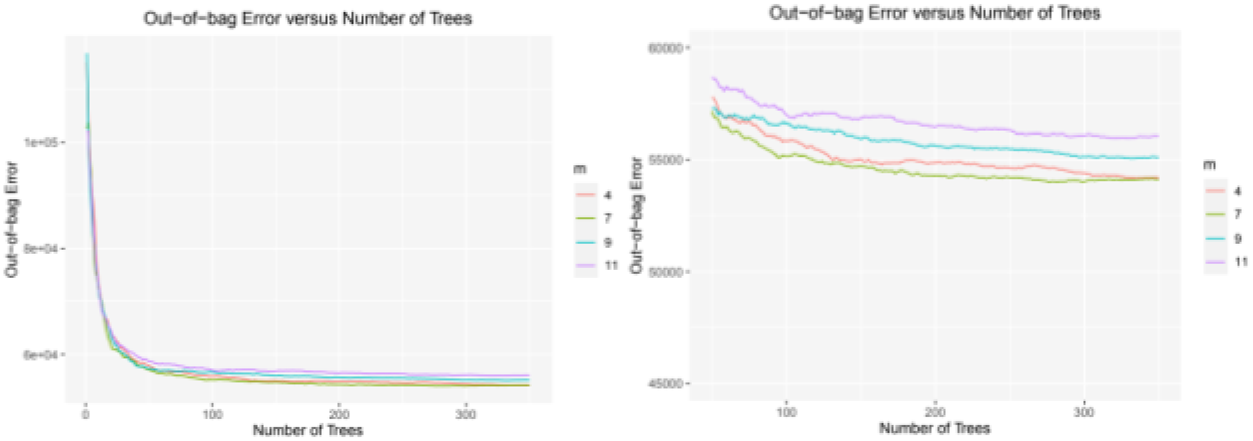
Using the one standard error rule, we conclude that the optimal tree has 44 splits (45 terminal nodes), and a complexity parameter of $\alpha = 0.00114$. Thus, we prune our large tree (the one with 5852 terminal nodes) down using this optimal value of the complexity parameter to get the optimal tree. This optimal tree fit to the training data is our final decision tree predictive model.

Random Forest

For our final predictive model we fit a random forest. Since this is a state of the art prediction method, we expect it will outperform the other three models if tuned correctly. The parameters of the random forest are *B* (the number of trees fit in the forest), m (the number of features considered at each split), and the criterion to stop growing the tree. As long as *B* is large enough (typically a value between 100 and 1000), the parameter *m* is the only one that needs to be tuned.

Since there are 12 features in our cleaned data, the default value for *m* is $12/3 = 4$, since this is a random forest for regression. To choose a value of *B*, we fit a tree with the default value of *m* and looked at a plot of the out of bag error as a function of the number of trees (*B*). We found that the out of bag error flattened out by ~300 trees, so we choose $B = 350$ to be conservative (plot not shown as it will be redundant with the out of bag error plot we show for tuning *m*).

With the value of *B* chosen, we move to tuning *m*. Since fitting trees to the dataset is computationally expensive, we choose a few different reasonable values of *m* and compare. Specifically, we choose $m = 4, 7, 9, 11$. For each of these values of *m*, we plot the out of bag error as a function of the number of trees to compare the performance. This plot is shown below.

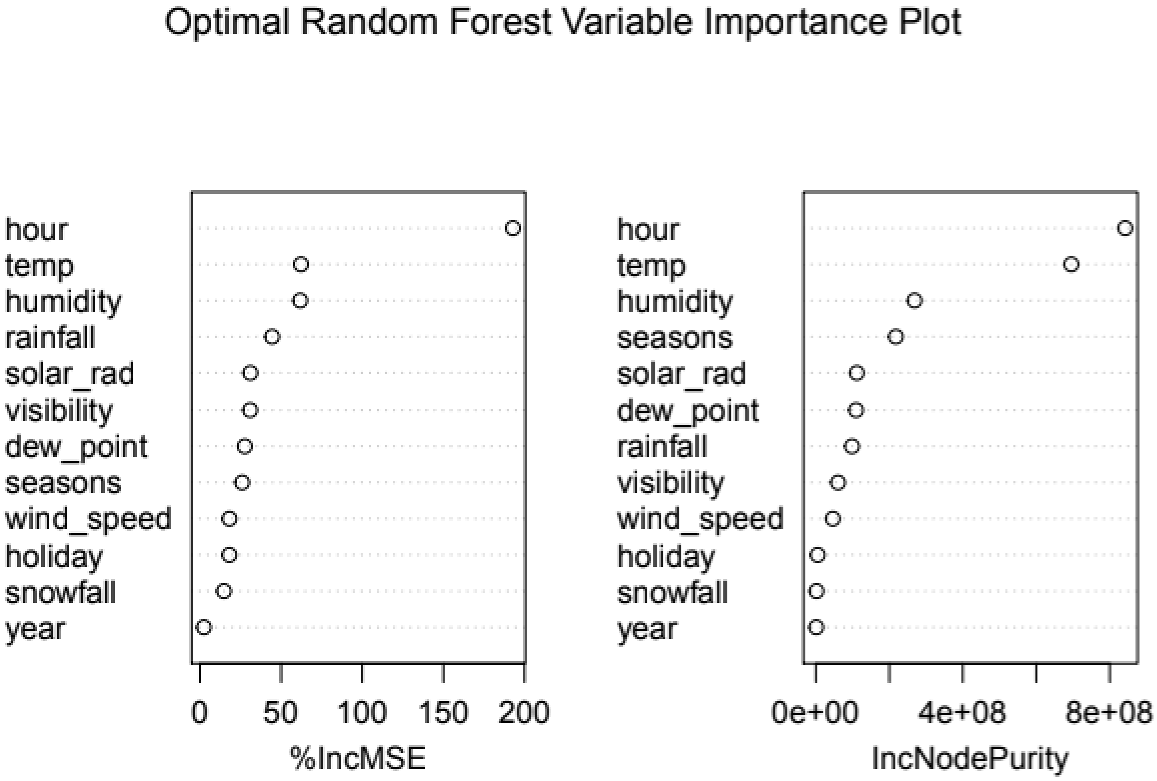


Out-of-bag Error versus Number of Trees

Out-of-bag Error versus Number of Trees

The left panel shows the entire plot, whereas the right panel zooms in to the end of the plot so we can get a better understanding of the OOB performance of the trees with the different values of m. First and foremost, the plot on the right panel verifies the $B = 350$ is a sufficiently large number of trees as the out of bag error clearly flattens out by then. In terms of tuning m, we see that by the right panel that at $B = 350$, the forests with $m = 4$ and $m = 7$ perform quite similarly (and better than the other two values of m). The forest with $m = 7$performs just slightly better at that point, and for lower values of *B*, it performs noticeably better than the forest with $m = 4$. For this reason, we choose $m = 7$ for our optimal random forest (though $m = 4$ would probably be an equally valid choice).

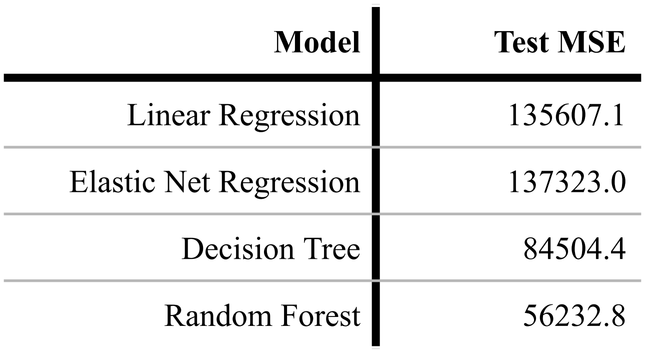
Thus, our final random forest predictive model is fit with $m = 7$ and $B = 350$. In order to interpret this model, we create a plot of variable importance below.



The left plot shows the importance using an OOB-based method, whereas the plot on the right shows the importance of each variable using a purity based metric. For both metrics, the three most important variables (in order from most to least important) are hour of the day, temperature, and humidity.

Model Evaluation and Comparison

For each model described above, we calculated the test error as the MSE of the model predictions on the test data. The table below shows the test errors for each model.



Model	Test MSE
Linear Regression	135607.1
Elastic Net Regression	137323.0
Decision Tree	84504.4
Random Forest	56232.8

First, we notice that simple linear regression and elastic net regression perform very similarly to one another, suggesting that the penalty term in elastic net does not improve predictive performance in this case. The decision tree performs significantly better than the two linear models, and the random forest outperforms everything else, with a dramatically lower test error.

We attribute the poor performance of the linear models to the fact that they don't incorporate interactions between the features, as we simply used first order, non-interaction terms in both regressions. In hindsight, including interactions between every combination of variables could have improved the predictive performance of these models, as interactions between weather characteristics could be important (e.g. low humidity and cold is quite unpleasant, whereas low humidity and high temperature could be considered desirable weather). In contrast, the two tree-based models automatically incorporate interactions between features.

It also makes sense that of the two tree-based models, the random forest outperforms the decision tree. The random forest aggregates predictions from a large number of trees, randomly selecting a subset of features to consider at each split. As a state of the art prediction method, it should outperform the single decision tree. Since the test MSE for the random forest is 56232.8, it has an RMSE of 237.13. Given the spread we see in the distribution of the bike counts from the "Data Exploration" section, this is fairly accurate.

Thus, the random forest appears to be the best predictive model in this case. Our results suggest that a successful predictive model for this data likely needs to account for non-linearities and interactions between the features.

Conclusions

The goal of this project is to predict the number of bikes rented in a given hour in the Seoul bike sharing system using information on weather conditions and time of day. After testing a series of models, we found that a random forest has the best predictive performance. This random forest model is able to predict the hourly rented bike count with a test MSE of 56232.8 (RMSE 237.13).

Some of the most important features for predicting the rented bike count are hour of the day, temperature, humidity, rainfall, solar radiation, and season. Particularly, the fact that both tree-based models significantly outperformed the linear models suggests that the best predictive model needs to account for non-linearities and interactions between features.

While a random forest is somewhat difficult to interpret, we can infer the relationship between rented bike count and each feature based on the earlier fitted models and the results from the exploratory data analysis. For example, higher temperature generally leads to predicting a higher bike count.

This could be informative to the Seoul city government on how to maintain a stable supply of bikes. For example, it could help them conclude that they need to be offering more bikes for rent during the warmer, sunnier, and less rainy months of the year. Furthermore, they could use this information to implement price discrimination, charging more during times of the year with better weather conditions.

While it might be extending the external validity of this study too far, it could also be used to determine whether a bike sharing system would be successful in another city, considering the overall climate of the new city in terms of how it relates to the important features from this study. That said, to make this conclusion, it would be better to replicate this analysis on either (1) a dataset that includes bike sharing data from multiple cities in different climates or (2) information on the demographics of bike users.