Pan Jialin 119020040
Wang Muying 119010301
Dai Hui 119010051
Xu Jinyu 119010357

Sharing bike demand prediction

2021/12/27

Abstract

This project aims at solving the problem of imbalance between the demand and supply for sharing bikes in Seoul. Our goal is to predict the hourly demand for sharing bikes in Seoul using regression methods and get inferences about the most influential factors. We first tried to draw inferences by fitting linear regression models, but it turns out that the linear assumptions are violated even after some transformations on the linear model, so we turn to nonlinear methods. We tried three nonlinear models (regression tree, random forest, and gradient boosting) and estimated their test MSE using 10-fold cross-validation. Among them, the random forest model gives us the best estimated test error, hence it is chosen as our final model. The test error of the prediction based on the random forest is 8719.738. The random forest illustrates that the most influential factors on demand of sharing bikes are the hour in a day, temperature, and humidity. The limitations of this study lie in the neglect of time-varying effect on data splitting process, failure to include interaction terms, limitation of hardware ability, and failure to address post-pandemic issues since the data is from 2017 to 2018.

1. Introduction

Sharing bikes have become a big part of the daily life of modern citizens ever since their introduction. It provides great mobility for people to commute around the city, allowing users to rent and return the bikes in different places. To maintain the sharing bike system, sharing bikes companies need to make regular replacements of sharing bikes from one place to another. The decision of bike placement is largely based on the demand for sharing bikes in different districts. While undersupply brings inconvenience to citizens and overcrowds other transportations, oversupply of sharing bikes can clog the payment and cause unwanted troubles. Hence, the balance of the demand and supply of sharing bikes becomes an essential issue to be taken care of by sharing bikes companies.

We collect two-year data of weather conditions, date information, and hourly amounts of bikes rented in one of the biggest cities in the world, Seoul. We believe that the weather conditions, such as rain, temperature, humidity, wind, etc. can largely affect people's choice of transportation and their willingness to commute by bike. Furthermore, the date information, such as weekdays, holidays and so on determines people's needs to commute, hence affecting the demand for rental bikes. With a clear explanatory relationship between the variables and a goal of sharing bikes demand prediction, we believe that building models by regression and machine learning algorithms are hopeful in solving our problem, and thus will be our research methods. Using the above-mentioned dataset and methods, we build linear regression models and tree models, with the goal of investigating the explanatory power of different variables on the prediction of sharing bikes demand, and

eventually select a model that gives the most satisfying prediction results.

This paper is structured as followed. Section 2 describes how the original data is dealt with and gives a rough illustration of variables through data visualization. In section 3 and 4 we provide details on the linear models and nonlinear models that we build. Section 5 provides the results and discussion. Section 6 is our conclusion.

2. Data Description

The data is collected from the UCI (https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand) website. The data set contains the count of public bikes rented at each hour in Seoul Bike hiring system with the corresponding weather data and holidays information from January 2017 to November 2018.

There is no missing data in the original data set.

There are in total 8760 observations, 1 response, and 22 features.

2.1 Attributes Information

Table 1. Attributes information

Variable Name	Type	Description
Response		
count	continuous	Count of bikes rented each hour
Predictors		
hour	categorical	Hour of the day
temp	continuous	Temperature, in Celsius
hum	continuous	Humidity, in %
wind	continuous	Windspeed, in m/s
visb	continuous	Visibility, in 10m
dew	continuous	Dew point temperature, in Celsius
solar	continuous	Solar radiation, in MJ/m^2
rain	continuous	Rainfall, in mm
snow	continuous	Snowfall, in cm
holiday	dummy	Holiday/non-holiday
temp	continuous	Temperature, in Celsius
season	categorical	Winter, Spring, Summer Autumn
spring, summer, autumn	dummies	One-hot encoding result from season, with winter as the base variable
day	categorical	Day in a week
monday, tuesday, wednesday, thursday, friday, saturday	dummies	One-hot encoding result from day, with sunday as the base variable
weekend	dummy	Weekend or non-weekend,
funcday	dummy	Whether the system is functioning

2.2 Data Cleaning

（1）Encoding

We transform the text data from the original data set *SouelBikeData* to numerical data by encoding.

For the *season* variables, the encoding is as follows: winter 0, autumn 1, spring 2, summer 3. Then we create 3 dummy variables *autumn, spring* and *summer* by one-hot encoding, taking *winter* as the base variable.

We further extract *day* information and do the following encoding: Sunday 0, Monday 1, Tuesday 2, Wednesday 3, Thursday 4, Friday 5, Saturday 6. We also create an indicator variable *weekend*, with 1 as *weekend* and 0 as *weekdays*. We further create 6 dummy variables *monday, tuesday, wednesday, thursday, friday,* and *saturday* by one-hot encoding, taking Sunady as the base variable.

(2) Variable exclusion

funcday is an indicator of whether the rental bike system is functioning. Since it makes no sense to include predictors data on days when the rental bike is out of service, we make the following adjustment to our data set:
i. we deleted the rows with *funcday* equal to zero.
ii. we deleted the entire column of *funcday.*

2.3 Data Visualization

We visualize our response *count* using the box plot and histogram. We can see that the medium of the number of bikes rented at each hour is approximately 500, with the middle 50% of the data values lying in between 250 and 1000. The spread for the daily counts larger than the 75% quartile is wide, with an upper limit set at around 2400. It appears that there is a considerable number of outliers lying above the upper limits. However, we do not deal with these outliers since the maximum value is about 3500, which is a reasonable amount for the number of bikes rented hourly in the busy city Seoul.

From the histogram, we can see that the distribution of *count* is largely skewed, which implies that the residuals may have a skewed distribution, and that may violate the assumptions of the linear regression model. The problem is partially addressed by taking log-transformation on the response, resulting in a more normal-like distribution.

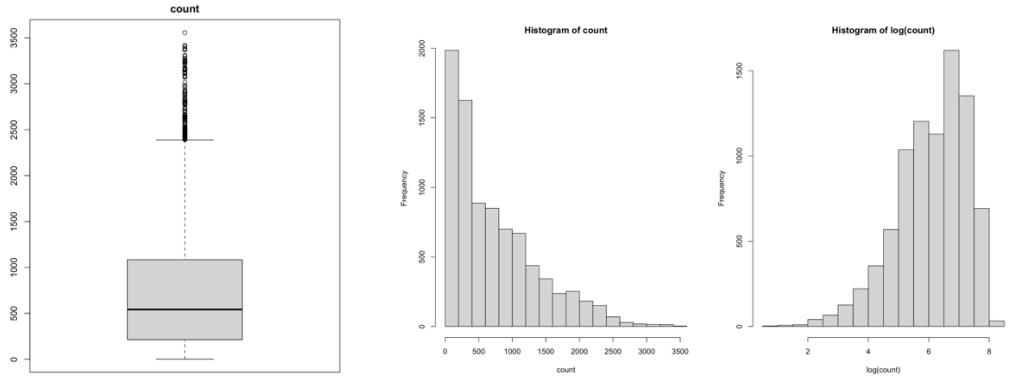


Figure 1. Box plot and histograms of response, count

Secondly, we check the relationship between the response and the categorical variables.

From the boxplot created for categorical variables *season*, we get the following patterns for the demand for rental bikes in Seoul:

1. The demand is lowest in *winter*, highest in *summer*. The demand for sharing bikes is similar in the two seasons *spring* and *autumn*.

2. The demand for sharing bikes is slightly higher on weekdays than on weekends. Moreover, the hourly count of bike rented on weekdays have significantly more outliers than on weekends.

3. The demand for sharing bikes is lower on holiday than on non-holiday.

The difference in demand for sharing bikes in season could be a result of different weather conditions, such as temperature, humidity, snow, rain, wind, and etc. These factors may have strong correlation with the *season* factor, to which we shall pay more attention in our later analysis.

That the demand for sharing bikes on weekdays is higher than on weekend implies that sharing bikes have become a daily commuter tool for citizens in Seoul. It makes sense that citizens would choose sharing bikes for daily commuting since sharing bikes are cheap, convenient, and eco-friendly. However, as the holiday boxplot implies, people might prefer other ways of traveling during the holiday.

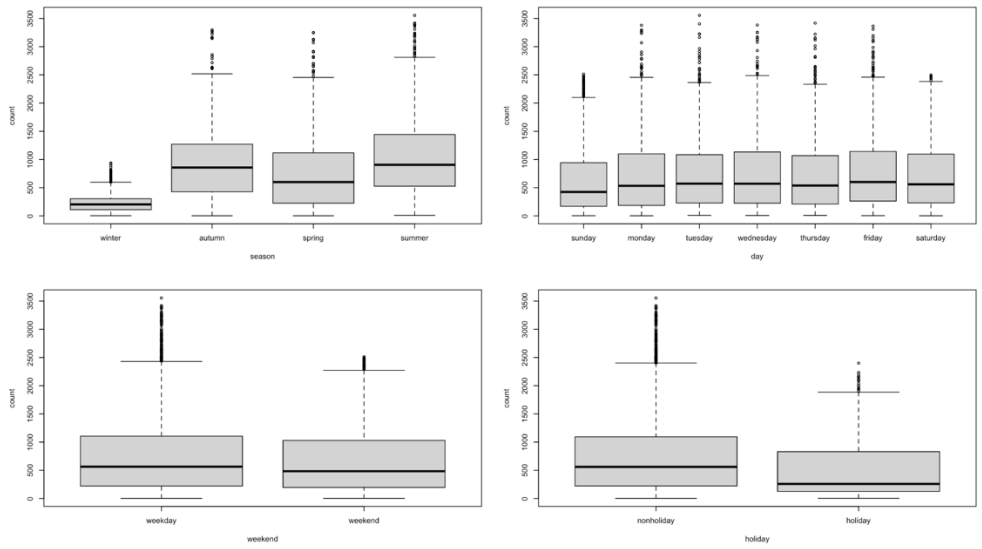


Figure 2. Box plots of response *count* and categorical variables

The scatter plots between the response and other variables show that the relationship between the response and these predictors is highly nonlinear; thus a linear model might not produce satisfying performance.

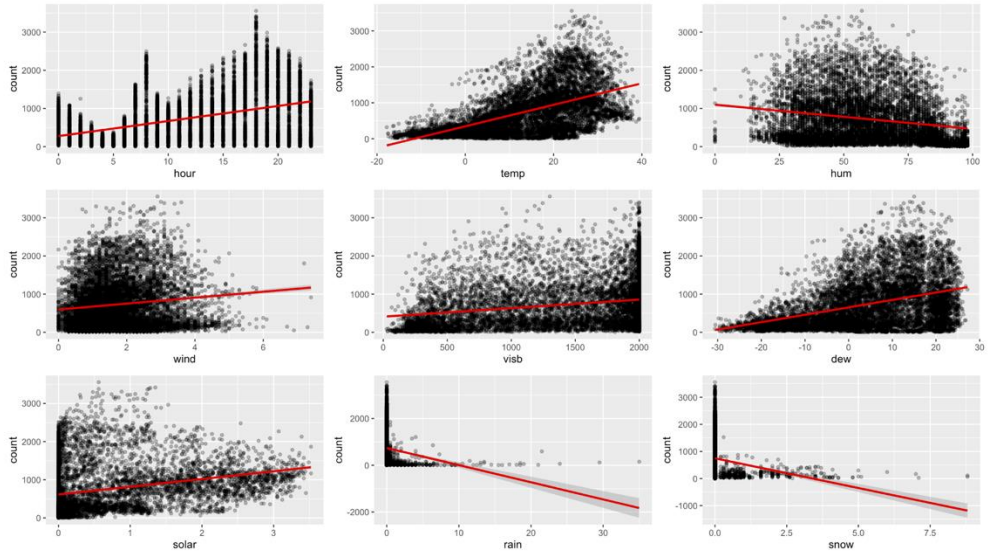


Figure 3. Scatter plots of response and other variables

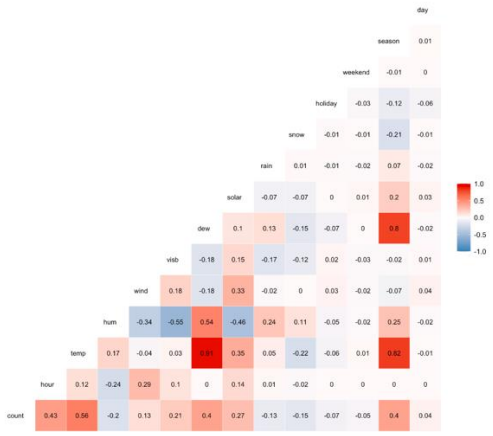


Figure 4. Correlation heat map between variables

From the correlation heat map, we can see that the demand for sharing bikes has relatively strong positive correlation with the *hour* in a day, *temp*, *dew*, and *season*, all with correlations larger than 0.4. The demand for sharing bikes appears to have a negative relation with *humidity*, *rain*, *snow*, and whether the day is a holiday or weekend, even though the correlations are weak, with absolute values smaller than 0.25.

Moreover, we can see that temperature has a strong correlation with dew point temperature and season. The correlation between season and dew point temperature is also strong. The problem of strong correlation between explanatory variables will be addressed later through feature selections.

2.4 Data Splitting

We randomly split the original data set and use 70% as our training data set and 30% as our test set. Since the splitting is done randomly, we do not take time-variation issue into account. Instead, we consider that data at different time of the year are randomly distributed to the two sets. The original data set contains data from 2017 to 2018, a time range of two years. Since no major event has significant effect on the way people commute during the two years' time, we neglect the changes within the two years and believe that splitting the data altogether will not affect our analysis.

3. Model Construction: Linear Models

3.1 Linear Regression Model

We fit a full model on the training data set using *count* as response and all other variables as predictors, except for *weekend, season* and *day*, since their effects have been included in the one-hot encoding dummies.

$$count = \beta_0 + \beta_1 * hour + \beta_2 * temp + \cdots + \beta_{19} * wednesday$$

```
Call:
lm(formula = count ~ . - weekend - season - day, data = d.trn)

Residuals:
    Min      1Q  Median      3Q     Max
-1225.8  -275.4   -57.9   211.0  2256.9

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.258e+02  1.085e+02   4.846 1.29e-06 ***
hour         2.877e+01  8.908e-01  32.295  < 2e-16 ***
temp         1.726e+01  4.256e+00   4.055 5.08e-05 ***
hum         -1.120e+01  1.195e+00  -9.378  < 2e-16 ***
wind         1.624e+01  6.185e+00   2.626  0.00866 **
visb         5.350e-04  1.197e-02   0.045  0.96434
dew          1.027e+01  4.452e+00   2.306  0.02112 *
solar       -8.607e+01  9.238e+00  -9.317  < 2e-16 ***
rain        -6.197e+01  5.421e+00 -11.431  < 2e-16 ***
snow         3.414e+01  1.298e+01   2.630  0.00856 **
holiday     -1.452e+02  2.640e+01  -5.499 3.97e-08 ***
autumn       3.801e+02  2.347e+01  16.192  < 2e-16 ***
spring       2.219e+02  2.223e+01   9.981  < 2e-16 ***
summer       2.130e+02  3.373e+01   6.315 2.90e-10 ***
friday       1.409e+02  2.065e+01   6.824 9.71e-12 ***
monday       9.789e+01  2.079e+01   4.709 2.55e-06 ***
saturday     5.874e+01  2.084e+01   2.818  0.00485 **
thursday     1.138e+02  2.108e+01   5.400 6.91e-08 ***
tuesday      1.116e+02  2.101e+01   5.313 1.12e-07 ***
wednesday    1.272e+02  2.083e+01   6.106 1.09e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 430.1 on 5907 degrees of freedom
Multiple R-squared:  0.5494,    Adjusted R-squared:  0.548
F-statistic: 379.1 on 19 and 5907 DF,  p-value: < 2.2e-16
```

Figure 5. Simple linear regression result

The regression result shows that all predictors are of 99% significant in explaining the response, except for *visb*, the coefficient of whom is not significantly different from 0. The adjusted R-squared of this model is 0.548, with a F-statistics of 379.1, which is very significant and demonstrates that the model can capture more than half of the variability of the demand for sharing bikes.

The coefficients of the predictors illustrate that the variables humidity, solar radiation, and holidays have negative effects on the demand for sharing bikes. Since all coefficients of the season dummies *spring, summer, autumn* are positive, we also conclude that the average demand for sharing bikes is the lowest in winter. Similarly, it appears that Sunday, which is the base variable for days in a week, is also related to the lowest demand for sharing bikes.

Through examining the VIFs of the variables, we find that the variables *hour, visb, temp,* and *spring* may suffer from multicollinearity, with their VIF all larger than 5. We will try to address this problem later on by using Lasso Regression and Ridge Regression.

```
```{r}
d.std = data.frame(scale(d.trn))
lm.fit.std = lm(count~.-weekend-season-day, data = d.std)
summary(lm.fit.std)
```
```

```
Call:
lm(formula = count ~ . - weekend - season - day, data = d.std)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9162 -0.4305 -0.0905  0.3298  3.5279

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.754e-15  8.733e-03   0.000  1.00000
hour         3.106e-01  9.618e-03  32.295  < 2e-16 ***
temp         3.276e-01  8.079e-02   4.055 5.08e-05 ***
hum         -3.581e-01  3.818e-02  -9.378  < 2e-16 ***
wind         2.632e-02  1.002e-02   2.626  0.00866 **
visb         5.084e-04  1.137e-02   0.045  0.96434
dew          2.132e-01  9.242e-02   2.306  0.02112 *
solar       -1.157e-01  1.242e-02  -9.317  < 2e-16 ***
rain        -1.040e-01  9.094e-03 -11.431  < 2e-16 ***
snow         2.430e-02  9.241e-03   2.630  0.00856 **
holiday     -4.872e-02  8.860e-03  -5.499 3.97e-08 ***
autumn       2.500e-01  1.544e-02  16.192  < 2e-16 ***
spring       1.509e-01  1.511e-02   9.981  < 2e-16 ***
summer       1.463e-01  2.316e-02   6.315 2.90e-10 ***
friday       7.815e-02  1.145e-02   6.824 9.71e-12 ***
monday       5.366e-02  1.140e-02   4.709 2.55e-06 ***
saturday     3.207e-02  1.138e-02   2.818  0.00485 **
thursday     6.109e-02  1.131e-02   5.400 6.91e-08 ***
tuesday      6.045e-02  1.138e-02   5.313 1.12e-07 ***
wednesday    6.976e-02  1.143e-02   6.106 1.09e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

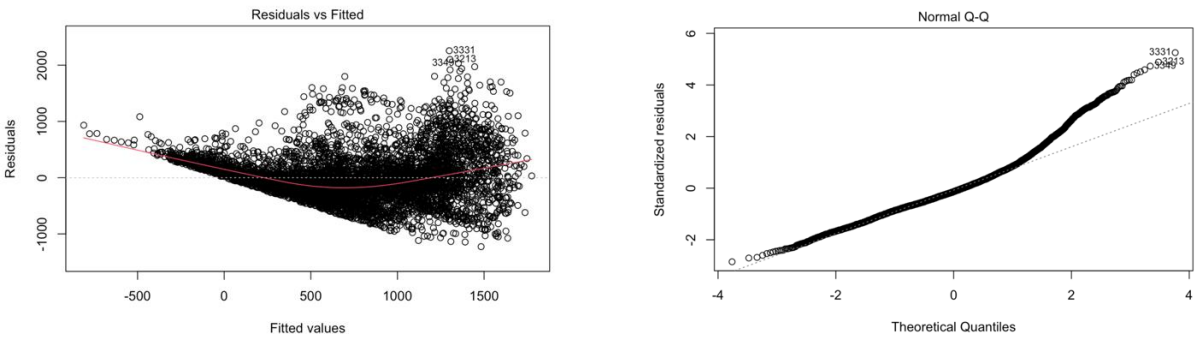
Residual standard error: 0.6723 on 5907 degrees of freedom
Multiple R-squared:  0.5494,    Adjusted R-squared:  0.548
F-statistic: 379.1 on 19 and 5907 DF,  p-value: < 2.2e-16
```

Figure 6. Standardized linear regression result

We fit the linear regression model again but standardize all variables using the Z-score method. The regression results are largely the same, but it shows that the predictors temperature, humidity, and hour in a day have the largest three effects on the response.

Recall that in order for the statistical inference of the model to hold true, the following assumptions need to be satisfied: 1. (L) the true relationship between the expected value of the response should be a linear function of the predictors; 2. (I) the errors should be independent; 3. (N) the errors at each value of the predictors should be normally distributed; 4. (E) the errors should have equal variance at each level of the predictors. We further plot the residual plot, Q-Q plot, Scale-Location plot, and Residual vs. Leverage plot to see if the LINE assumptions are satisfied.

There is a sharp cut in the left bottom part of the residual plot, which is due to that all values of the response *count* are positive. We can tell from the residual plot that the relationship between the predictors is largely nonlinear since the residual does not bounce randomly around the zero line. The Q-Q plot is heavy-tailed, indicating that the residuals do not have a normal distribution. The scale-location plot shows that the residuals do not have equal variance, and this model suffers from heteroscedasticity. The residual vs. leverage plot illustrates that no residual is outside of the Cook’s distance, so there is no influential point (no outliers and no leverage points). Since the linear assumptions are hardly satisfied, we need to make some adjustments to our model.



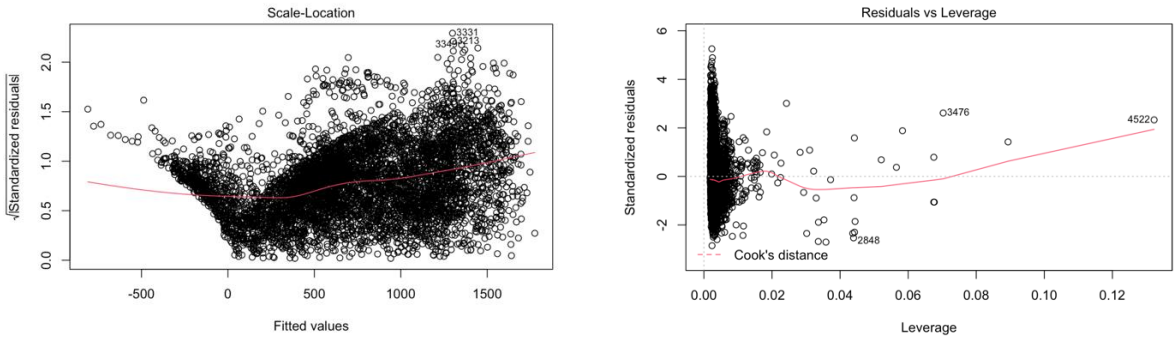


Figure 7. Residual plot, Q-Q plot, Scale-Location plot, and Residual vs Leverage plot

3.2 Improvements for Linear Regression Model

The simplest linear model suffers from the non-linearity, heteroscedasticity, and non-normality problems. We try to solve these problems using two methods:

(1) Log-transformation for the response *count*;

(2) Weighted Least Square. For each method, we use feature selection (best subset selection, Ridge & Lasso) to find the best model.

3.2-1. Log-transformation Model

We fit a linear regression model using *log(count)* as response and all other 19 variables as predictors. To further improve this model, we perform subset selection, Ridge, and Lasso regression.

Table 2. Results from subset selection

	Feature Selection Method	Best Model	Estimated Test MSE
Log Transformation	Best Subset Selection	7-variable model	**210289**
	Ridge		211203
	Lasso		212512

By comparing the estimated test MSE generated by cross-validation, the best model is a 7-variable model using best subset selection method.

Now we take a closer look at the best subset selection for the log transformation model.
Firstly, we fit the model.

```
d.trn$lcount = log(d.trn$count)
n = ncol(d.trn)-1-4
regfit.full = regsubsets(lcount~.-weekend-season-day-count, data=d.trn, nvmax=n)
```

Secondly, we choose among models using cross-validation and one standard error rule.

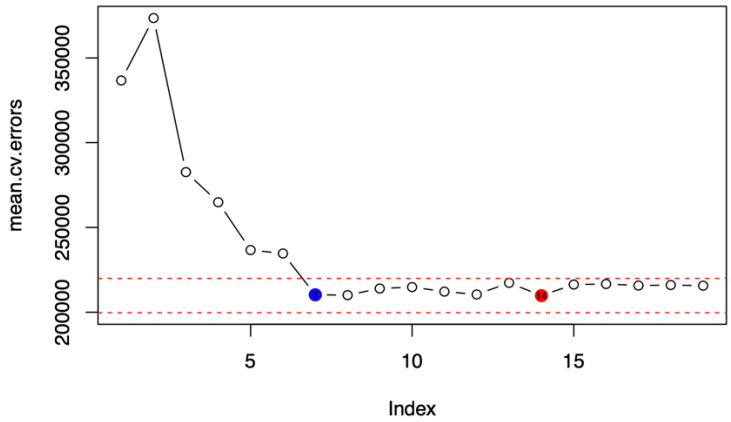


Figure 8. Cross-validation result from best subset selection

We choose the 7-variable model, with *log(count)* as the response, and *hour, hum, dew, rain, autumn, spring, summer* as the predictors.

The corresponding estimated test MSE of this model is 210289. What's more, we check the LINE assumptions for the linear model again.

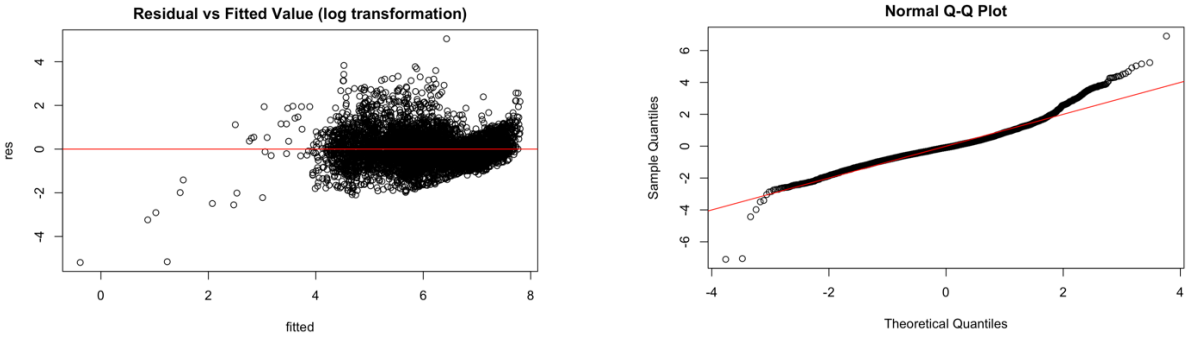


Figure 9. Residual plot and Q-Q plot for the 7-variable model

There is less pattern for heteroscedasticity, but the non-linearity and non-normality problems still exist. The log transformation still fails to perform well.

3.2-2. Weighted Least Square

We fit weighted least square models with *count* as the response and all 19 predictors using three different weights: (1) $w_1 = \frac{1}{\hat{y}^2}$; (2) $w_2 = \frac{1}{\varepsilon^2}$; (3) $w_3 = \frac{1}{|\varepsilon|}$

To further improve this model, we perform subset selection, Ridge, and Lasso regression.

Table 3. Feature selection results for WLS models

	Feature Selection Method	Best Model	Estimated Test MSE
$w_1 = \frac{1}{\hat{y}^2}$	Best Subset Selection	17-variable model	324238
	Ridge		683716
	Lasso		365504

$w_2 = \frac{1}{\varepsilon^2}$	Best Subset Selection	12-variable model	189918
	Ridge		203754
	Lasso		186154
$w_3 = \frac{1}{\lvert\varepsilon\rvert}$	Best Subset Selection	8-variable model	189376
	Ridge		188968
	Lasso		**185111**

By comparing the estimated test MSE generated by cross-validation, the best model is from the Lasso regression.

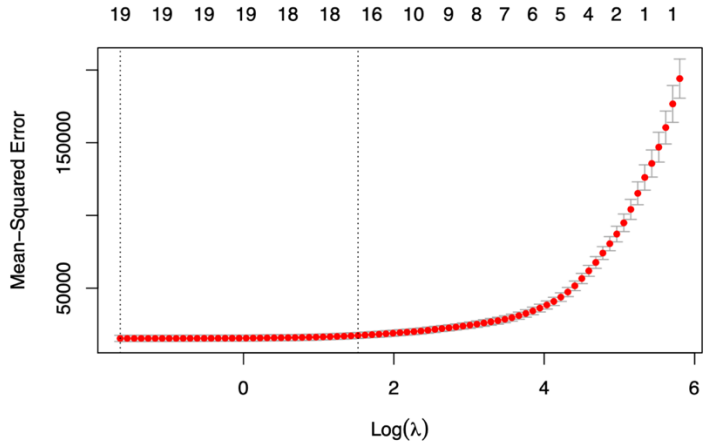
Now we take a closer look at the Lasso regression for WLS using $w_3 = \frac{1}{|\varepsilon|}$.

Firstly, we fit the model.

```
x = model.matrix(count~.-weekend-season-day-lcount, d.trn)[,-1] # delete the intercept
y = d.trn$count
grid =10^seq(10,-2, length =100) # 1e-2 = 0.01
```

```
lasso.mod = glmnet(x, y, alpha = 1, lambda = grid, weights=w3)
```

Secondly, we use cross-validation to find the best λ, which is 0.1939782.




```
bestlam =cv.out$lambda.min
bestlam

## [1] 0.1939782
```

Figure 10. CV result for selection of λ

Finally, using the best λ chosen by cross-validation, we get the WLS model from Lasso regression. Since the λ is small, no coefficients shrink exactly to 0.

```
out = glmnet(x,y,alpha = 1, lambda = grid, weights=w3)
(lasso.coef = predict(out, type = "coefficients", s = bestlam)[,])

##   (Intercept)          hour          temp           hum          wind
##  4.951319e+02  2.749082e+01  1.823708e+01 -1.060968e+01  1.391003e+01
##          visb           dew         solar          rain          snow
##  8.972557e-03  9.556209e+00 -7.810602e+01 -6.156472e+01  3.634721e+01
##       holiday        autumn        spring        summer        friday
## -1.422731e+02  3.603905e+02  2.064326e+02  1.850792e+02  1.343998e+02
##        monday      saturday      thursday       tuesday     wednesday
##  9.221958e+01  5.989167e+01  1.044012e+02  1.032958e+02  1.109103e+02
```

The corresponding estimated test MSE is 185111. What's more, we check the LINE assumptions for the

linear model again.

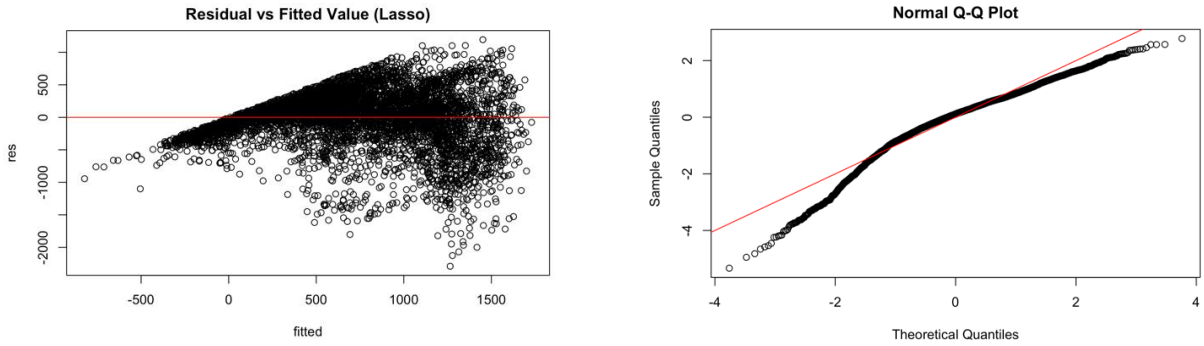


Figure 11. Residual plots and QQ plot for WLS model

The problems of heteroscedasticity, non-linearity, and non-normality still exist. The WLS model still fails to perform well.

Since the linear models all fail to satisfy the LINE assumptions, we turn our efforts to the nonlinear models.

4. Model Construction： Nonlinear Models

4.1 Regression Tree

To fit a regression tree model, we firstly build an unpruned tree using the training data:

```
# Grow an unpruned tree
tree.bike = tree(count~., data=training)
```

Then we plot the tree below:

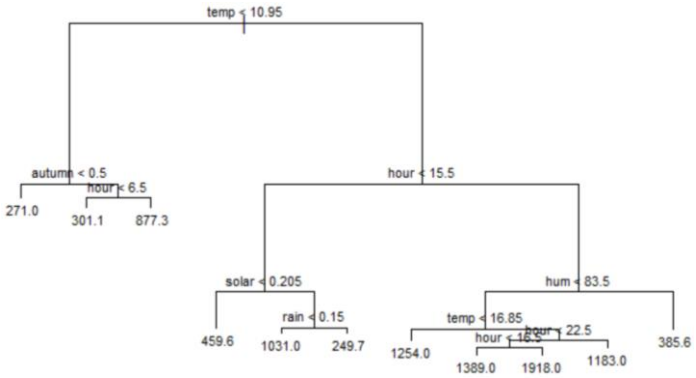


Figure 12. Regression Tree, Unpruned

From the above tree, we find that only 6 variables are used to construct the tree.

Among all of them, temperature(*temp*) is the dominant variable. When *temp* < 10.95℃, the demand for bikes is relatively smaller. When *temp* > 10.95℃, the demand quantity for bikes is relatively larger.

Then we use 10-fold cross-validation to choose the tree-complexity:

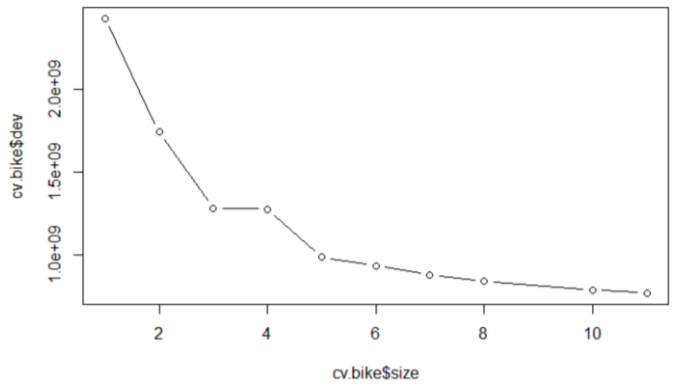


Figure 13. CV result for tree pruning

The graph above shows that the unpruned tree has the lowest deviation. Thus, we do not need to prune the tree and should use the whole tree in our model.

After applying CV on the training set, the estimated MSE of the regression tree is 205345.7

4.2 Random Forest

Firstly, we use cross-validation to select the tunning parameter: *mtry*, which is the number of predictors we considered for each split of the tree.

```
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   2    309.4504  0.8179604  219.6159
  12    177.9153  0.9239158  110.2631
  22    178.0139  0.9228347  108.4998
```

Figure 14. CV Result for random forest tuning parameter

We find that *mtry* = 12 gives the smallest RMSE, so we select *mtry* = 12 to build our random forest model.

Note that more trees in the random forest model will not result in overfitting. Therefore theoretically, the more trees there are in our model, the lower the error should be. This fact can also be verified by the graph below:

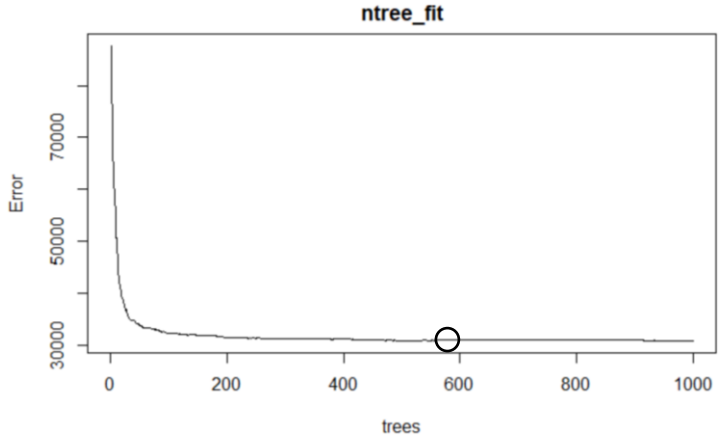


Figure 15. Estimated test error for different number of trees in random forest

Thus we choose a very large number B = 1000 in our random forest model.

We use *mtry* = 12 and *ntree* = 1000 to build our random forest model:

```
library(randomForest)
set.seed(42)
ntree_fit<-randomForest(count ~ ., data=training, mtry=12, ntree=1000, importance=TRUE)
```

We also want to use the varImpPlot() function to view and compare the importance of each variable:

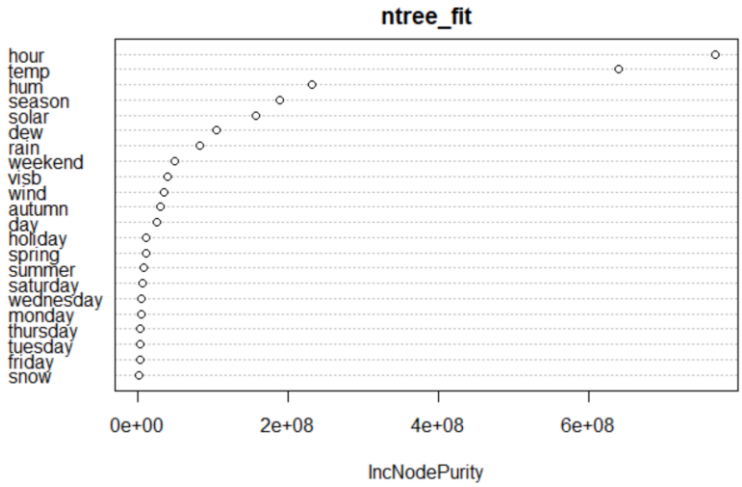


Figure 16. Comparison of model importance, random forest

The plot indicates that taking all the trees in the random forest into consideration, *hour* and *temp* are the top 2 most important variables.

After applying CV on the training set, the estimated MSE of random forest is 31653.85.

4.3 Gradient Boosting Machine

First we use cross validation to select the tunning parameter: *n.trees*(B) and *interaction.depth*(d).

```
Resampling results across tuning parameters:

  interaction.depth  n.trees  RMSE      Rsquared   MAE
  1                   50      405.6640  0.6344703  294.5278
  1                  100      370.9196  0.6769457  273.1623
  1                  150      354.4846  0.7007607  262.5709
  2                   50      321.0580  0.7637308  226.3765
  2                  100      285.3172  0.8079997  198.1435
  2                  150      266.1354  0.8303039  186.0786
  3                   50      282.5084  0.8135964  196.5184
  3                  100      247.3841  0.8531189  167.2324
  3                  150      228.0083  0.8743530  154.4716
```

Figure 17. CV result for selection of tuning parameter in Gradient Boosting

We find that *n.trees* = 150, *interaction.depth* = 3 gives the smallest RMSE, so we select them to build our gradient boosting model. And we use λ=0.1 (shrinkage=0.1) by default; it is good enough.

We use *n.trees* = 150, *interaction.depth* = 3 and *shrinkage* = 0.1 to build our gradient boosting model:

```
boost.bike = gbm(count ~ .,
                 data=training,
                 distribution="gaussian",
                 n.trees=150,
                 interaction.depth=3,
                 n.minobsinnode = 10,
                 shrinkage = 0.1)
summary(boost.bike)
```

And we use summary() function to see the relative importance of each variable:

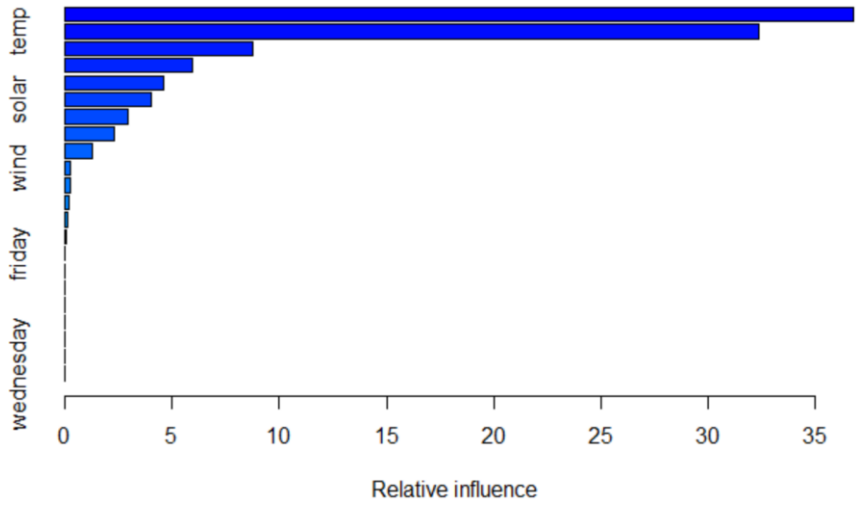


Figure 18. The relative influence of variables in Gradient Boosting

We can see that *temp*(temperature) and hour are the 2 most important variables.

We want to show the marginal effect of *temp* and *hour* on the response after integrating out the other variables.

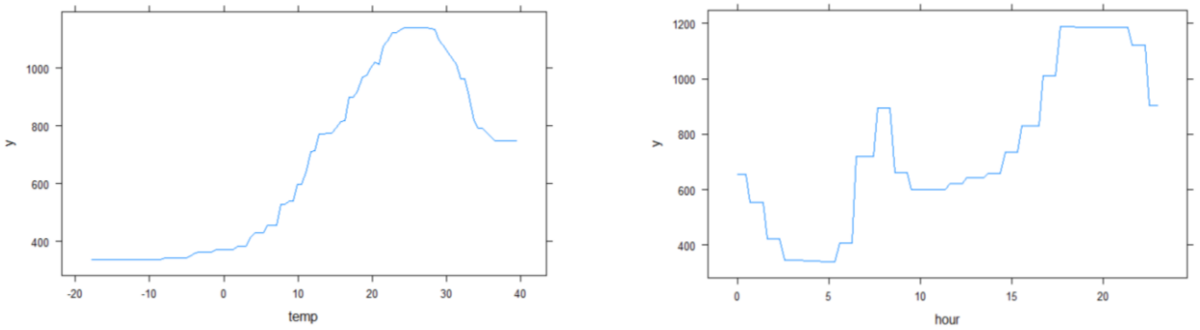


Figure 19. Marginal effect plots

The marginal effect plot shows:

1. when temp is around 25℃, the demand for bikes is the largest.
2. people have the greatest demand for bikes during commuting hours(around 8am and 7pm)

After applying CV on the training set, the estimated MSE of gradient boosting is 51987.78.

5. Results and Discussion

5.1 Results

We have obtained the results of all methods mentioned before.

We used the estimated test MSE to select the best model. It can be summarized as the table:

Table 4. Estimated test MSE for all models

Model (Method)	Estimated test MSE
Linear model	185893.7
Linear model using log-transformed response	210288.8
Weighted-Least-Squared linear model	185111.3
Regression tree model	205345.7
Random forest model	31653.85
Gradient boosting model	51987.78

We do not consider the linear models as candidates for our final models because they hardly satisfy the LINE assumptions. Since the random forest model has the smallest Estimated test MSE (31653.85), we selected this model as our final model for our project. We predict on the testing data set:

```
rf.bike = randomForest(count ~ .,
                       data=testing,
                       ntree=1000,
                       mtry=12,
                       importance=TRUE)
```

The test MSE for our final model, random forest, is 8719.738, which is significantly smaller than the estimated test MSE, and thus we conclude this model has a good performance.

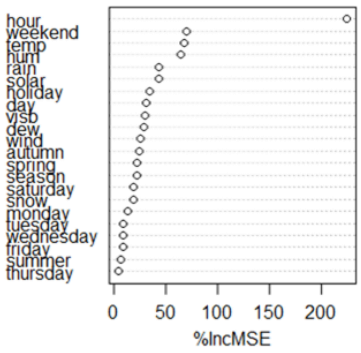


Figure 20. Features importance

After getting the model with the smallest error, our interests turn to find out which factors are most important in predicting the accurate bike rented count. By ranking each feature in our final model in an MSE-decreasing order, we find that the first 4 important factors are *hours*, *weekend*, *temperature*, and *humidity*.

5.2 Discussion

(1) Advantage of Random Forest

We believe that random forest outperforms all other models because of the following reasons:

i. High accuracy, Low variance: due to the embedded feature selection in the model generation process, the random forest can address the high model variation caused by variables that have a strong influence on the response. In this case, it can be the predictors *hours, temp,* and *humidity.* Moreover, more trees in

the random forest will help to achieve better prediction accuracy; hence the choice of 1000 trees in our case gives this model advantage over others.

ii. Ability to handle a large amount of data: random forest can handle a large amount of data with numerous variables. It shows its strength in our case with 19 explanatory variables and around 6000 observations in the training set.

iii. Process fast: random forest can handle variables quickly, making it suitable for complicated tasks. In our case, it is to predict demand for sharing bikes in each hour using 19 different variables.

(2) Inference from important features

The first 4 important factors are hours, weekend, temperature, and humidity. This shows that the exact time in a day and whether it is on the weekend affect the rental bike demand quantity most: people rent more bikes on a weekday and during rush hours. Also, people's feeling of weather will affect the demand for sharing bikes: when the temperature is appropriately comfortable, and the humidity is pleasant and delightful, people are more likely to rent bikes, so the rental companies should make adjustments with a focus on these weather conditions accordingly.

(3) Limitations

1. We split the data without considering the time variation. Although we believe that through random sampling, the effect of time series can be neglected, it may be helpful to take time variations into consideration when splitting the training and testing data sets.
2. We do not include interaction terms in the models. Some interaction patterns may not be captured by our model.
3. Due to the limitation of the space and the computation, the size of the random forest and boosting is limited, and it may cost some time and storage space to run the code.
4. The prediction and inference we have done in this project may only have limited implications since the commuting tools in the post-pandemic time may be different from before.

6. Conclusion

In order to investigate the prediction power of different weather condition variables and days information variables on the demand of sharing bikes, we first attempt to fit and improve linear regression to draw useful inferences, but the linear models all fail to satisfy the LINE assumptions. We further turn our efforts on the nonlinear models. The random forest turns out to be the best model with the least estimated MSE. We believe random forest can produce a satisfying prediction for demands of sharing bikes since it achieves a significantly low test MSE, compared with all the estimated MSEs. The inferences drawn from the random forest model demonstrate that the variables hour in a day, weekend, temperature, and humidity are most important in the prediction of sharing bikes' demand. Hence, we conclude that the sharing bikes company can adjust the placement of sharing bikes in big cities like Seoul by placing more bikes during the rushing hours on weekdays. They should also place more attention to the weather conditions since temperature and humidity very likely affect people's willingness to commute by bike.

References

Seoul bike sharing demand data set. (2020). UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand#

Solving the problem of heteroscedasticity through weighted regression. (2019, October 30). DATAMOTUS. https://datamotus.com/2019/10/30/Weighted-Least-Squares.html

V. E., S., Park, J., & Cho, Y. (2020). *Using data mining techniques for bike sharing demand prediction in metropolitan city.* Computer Communications, 153. 353-366. https://doi.org/10.1016/j.comcom.2020.02.007