Forecast Use of a City Bikeshare System

Section I: Introduction

- The goal of this final project is to to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C. Under the Capital Bikeshare program, people can rent a bike from one location and return it to a different place. The process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city.

- Specifically, the aims of this project is utilize historical usage patterns of Bike sharing system as well as weather data to forecast bike rental demand. The data were generated by the Bike sharing systems in which the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Specifically, the dataset consists of hourly rental data spanning two years. The actual source of data set is from Kaggle (www.kaggle.com) which provided the data as the material for a competition,

Section 2: Exploratory Data Analysis

Section 2.1: Data Description:

- The training set is comprised of the first 19 days of each month, while the test set is comprised of 20th to the end of the month. The objective is to predict the total count of bikes rented during each hour covered by the test set by only information available prior to the rental period.
- Dimension of Training Set: 10886 (rows) observations X 12 (columns) variables
- Dimension of Testing Set: 6493 (rows) observations X 9 (columns) variables
- Original Variables Description

datetime - hourly date + timestamp
season - 1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday - whether the day is considered a holiday
workingday - whether the day is neither a weekend nor holiday
weather - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp - temperature in Celsius
atemp - "feels like" temperature in Celsius
humidity - relative humidity
windspeed - wind speed
casual - number of non-registered user rentals initiated

registered - number of registered user rentals initiated
count - number of total rentals

Section 2.2 Graphical Data Analysis

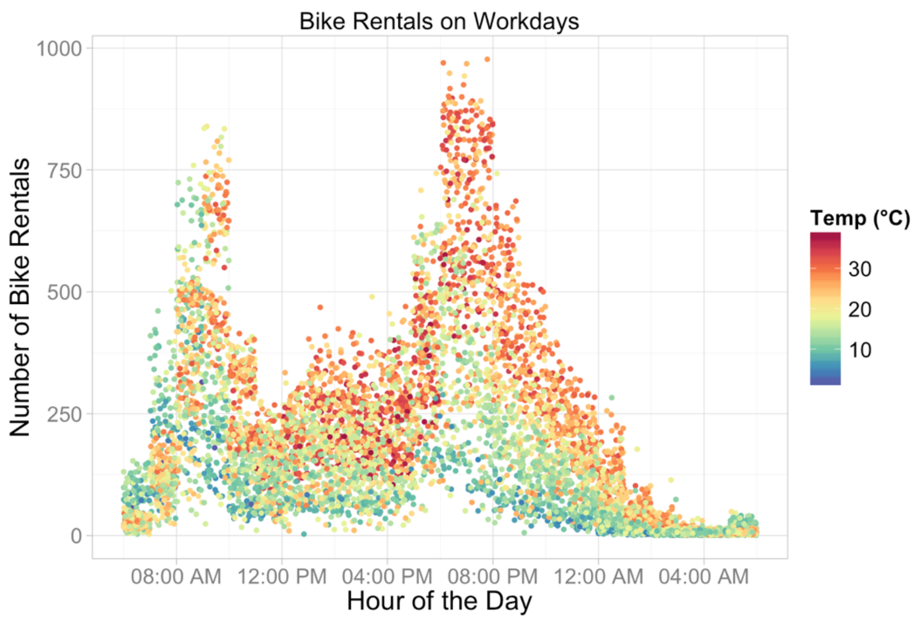


Figure 1: Color Plot of Bike Rental by Hour of the Day (Workdays)

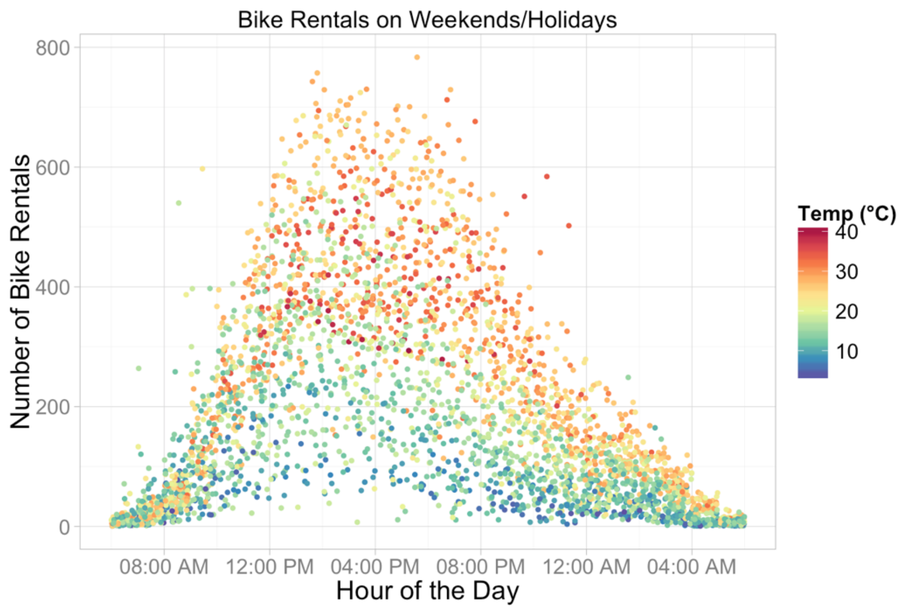


Figure 2: Color Plot of Bike Rental by Hour of the Day (Weekends/Holidays)

Figure 1 and Figure 2 identify some patterns between hour and temp on Bike Rental on Weekdays and Weekends respectively.

From figure 1, we can see that the bike rental peaks at around 9 am and 7pm which is not surprising because this is the peak traffic hours for people travelling to and from work. For weekends, we can infer from figure 2 that bike rental peaks at around 4pm. By looking at the color distribution of Figure 1 and 2, it is obviously that temp and counts are positively correlated.

Both figure 1 and 2 show no linear relationship between Hour and Bike rental, hence it makes sense to set 'hour' elements extracted from *datetime* variable.

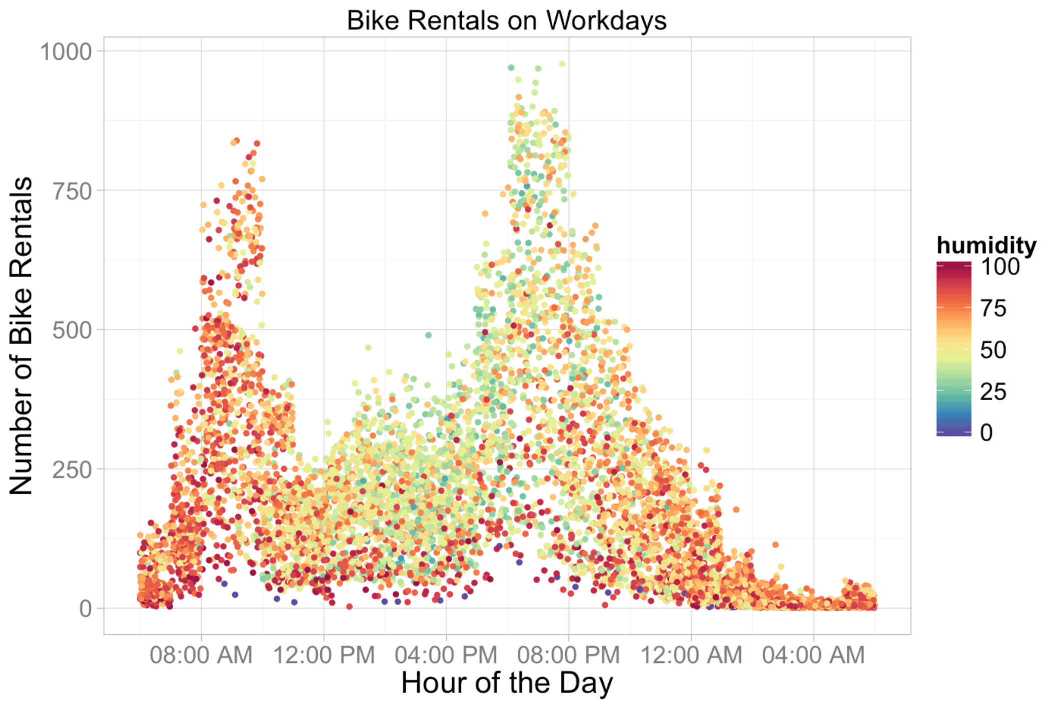


Figure 3: Color Plot for Humidity of Bike Rental by Hour of the Day (Workdays)

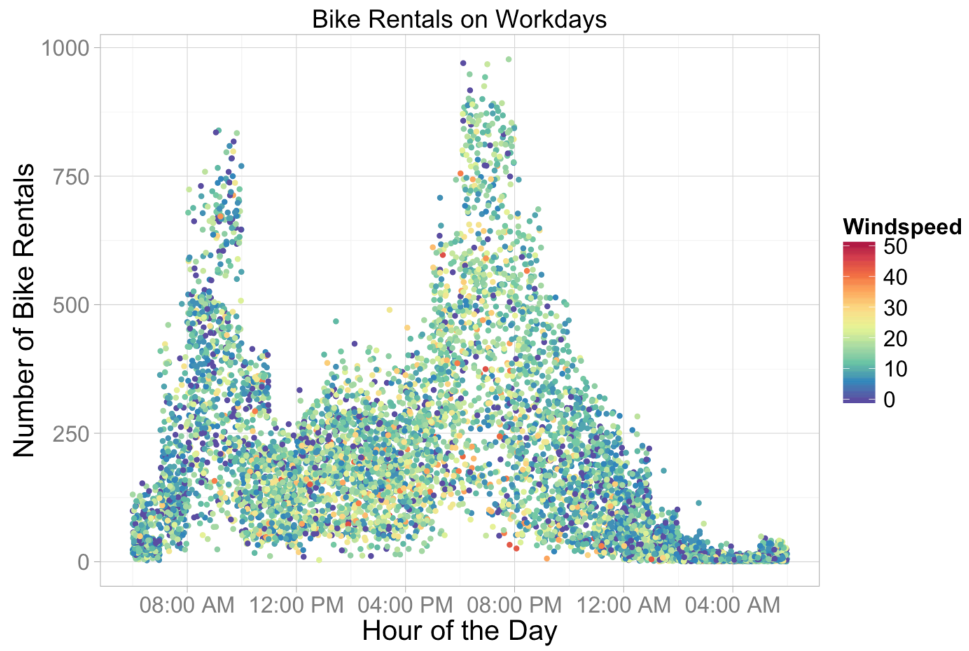


Figure 4: Color Plot for Windspeed of Bike Rental by Hour of the Day (Workdays)

Relationship between Bike Rental and Humidity:
Figure 3 shows that wind speed and counts seems don't have strong correlation; we can see that there is negative correlation between Humidity and Bike Rental.

Relationship between Bike Rental and Windspeed:
From figure 4, the color distribution on the plot evenly, therefore it looks like there is no strong correlation between Bike Rentals and Windspeed.

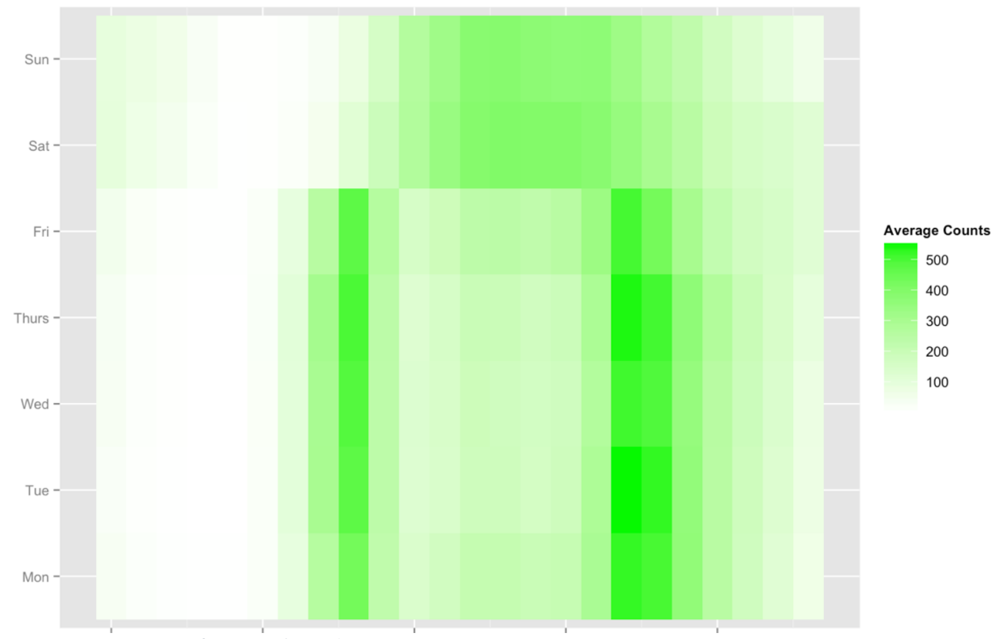


Figure 5 Heat Map of Counts by wday

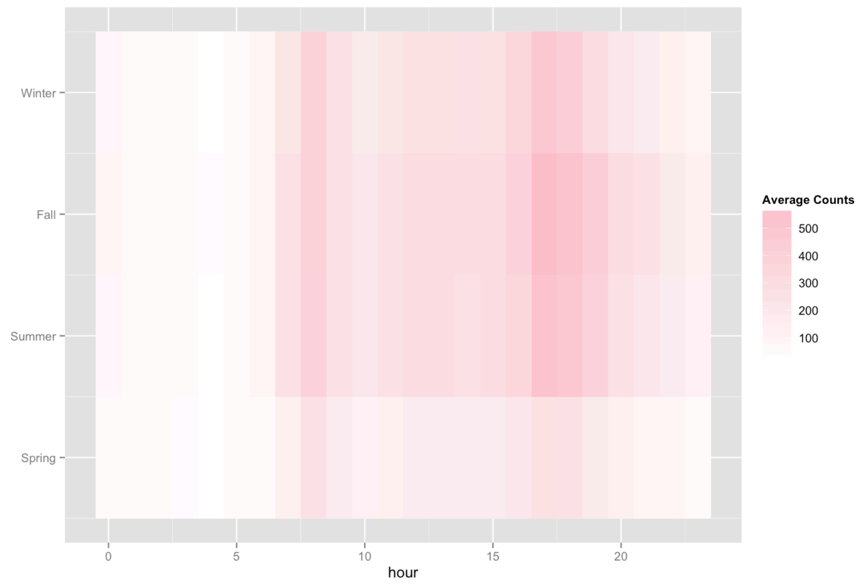


Figure 6: Heat mat of Counts by seasons

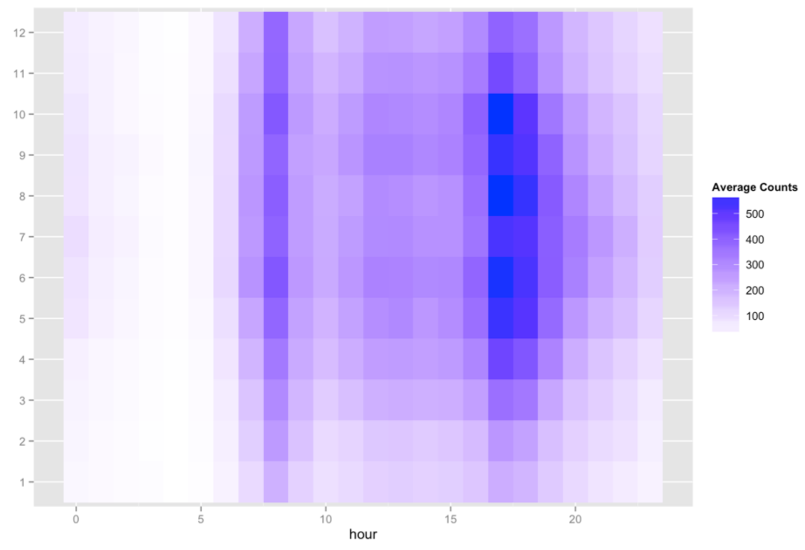


Figure 7: Heat map of Counts by months

Figure 5 shows the relationship between wday, hour of the day and count. Monday to Friday have the same pattern while Saturday and Sun shows another pattern. Hence, during model selection, we could considering dropping out the variable working day because variable wday in would be good enough to explain the difference for prediction.

Figure 6 shows the relationship between season and count, and figure 7 shows the relationship between months and count. Since the two plots have highly similar pattern, we may consider using only one of the two variables in model prediction.

2.3 Feature Engineering:

- **New Variable Added:**

Year	2011, 2012	Categorical Variable extracted from the variable *datetime*
month	1,2, …12	
wday	Monday = 1, Tuesday = 2…Sunday = 7	
hour	0,1,2…23	
lcount	Log(count +1)	Numeric Variable by transformation with log on variables *count, casual* and *registered* to match evaluation metric in Kaggle.
lcasual	Log(casual + 1)	
lregistered	Log (registered + 1)	

Table 1: Variables Added

- Converting Data Types: Since *weather, season* are qualitative variables which the numbers are actually labels instead of values, they are converted to categorical variables for analysis.

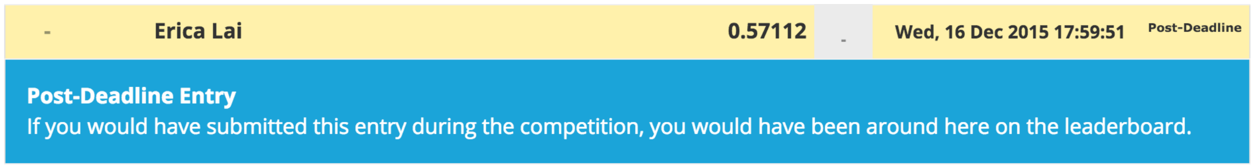
Section 3: Method

Section 3.1: Simple Model

- Model description: The Simple Model does not require much training. The prediction on the test data were based on the average counts from the same month, same hour of the day, same day of the week of the period up to the time of the test sample
- Implementation: The *group_by* function from package {dplyr} was used to group all train data by *wday* and *hour* and return the mean of *lcount* by the *summary* function:

```
by_wday_hour = group_by(trainSubset, wday, hour)
wday_hour_lcounts = summarise(by_wday_hour, lcounts=mean(lcount))
```

- The prediction on count was implemented by convert lcount where count = exp(lcount) -1
- The test Score is : 0.57112



-	Erica Lai	0.57112	-	Wed, 16 Dec 2015 17:59:51	Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

Figure 8: Result of Simple Model

Section 3.2: Linear Regression Model

- Linear regression Model allows us to summarize the relationship between the response variable and predictor variables by fitting least regression line.

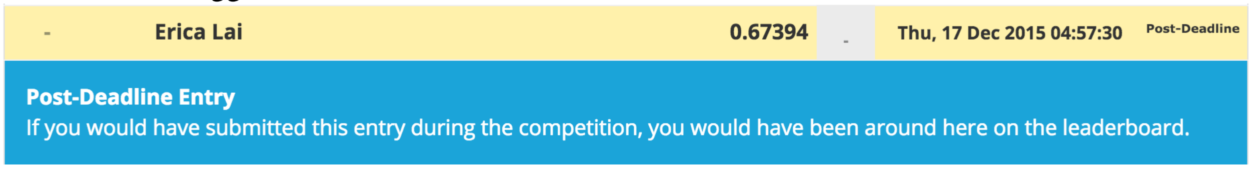
- To get a general idea of how linear regression model works in this set of data, a simple linear regression Model was first fitted by variables *workingday*, *temp, atemp, humidity, hour, wday, windspeed*, the variables *year, month, weather, season* were not used because for each batch of test data, these we cannot have all levels for these categorical variables so they do not work well in this model.
- As mentioned above, Figure 6 imples that there could be relationship between seasons and counts. However, for first 3 quarters of 2011, there are only 3 levels of seasons, which limit the functionality of the lm model and generate error message. Hence in order to add the variable *season* in consideration for model selection, a conditional statement is added in order to allow the variable season to be included in the regression for year 2012, where 4 levels of season are available in the training set.
- Since we are only required to use only past data to make predictions on the test set, a for loop is used to select training samples which are precedent to the time of the test set. A lm model was then fitted by lm function in R. To conduct model selection based on the set of dependent variables, the Akaike's Information Criterion (AIC) was used to select the best model for each training set with smallest AIC.

For each model M,

$$\text{AIC}(M) = -2 \log L(M) + 2 \cdot p(M)$$

Where L(M) is the likelihood function of the parameters in model M evaluated at the MLE (Maximum Likelihood Estimators) and p refers to the number of parameters.

- The score on Kaggle is 0.67394


- Erica Lai 0.67394 - Thu, 17 Dec 2015 04:57:30 Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

Figure 9: Result of Linear Regression Model

Section 3.3: Random Forest

- Random forests is an ensemble method of prediction. It combines k base tree models to create an improved composite model for prediction. For a given data set, k training data sets would be created to generate k learnt models, each of which would return a prediction for the test set. The ensemble will then make prediction on the test sample according to the overall prediction by the composite prediction model. In other words, Random forests are a combination of tree predictors and each tree depends random train observations sampled independently. Each decision tree is built by a random selection of attributes at each node to find the best split. By introducing randomness in best-split selection, correlation between trees built will be reduced because the split in each node are based on different predictors. Since Random forests would consider fewer features at each split during the process of decision tree building, the algorithm is more efficient especially toward large datasets. It could also estimate variable importance to identify important predictors.

- To implement Random Forests in R, the package {randomForest} implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression.
- Similar to Simple Model and Linear Regression Model mentioned above, we are only required to utilize past data to make predictions on the test set, a for loop is used to select training samples which are precedent to the time of the test set and a random forest tree is built based on all the past data of that test set.

- <u>Model Fitting</u>

  I have tried 2 approach to build model under this algorithm.

  Approach 1:
  I used lcount as the response variable and return the count by exponential function. The results are shown as M1 to M5.

  Approach 2:
  Considering the registered + casual = total count, I used lregistered and lcasual (logistics of registered and casual) as the response variables to built random forest models. The total predicted count was then return by the sum of exponential of predicted result by lregistered and lcasual as below:

  Predicted Registered = $e^{lregistered+1}$
  Predicted Casual = $e^{lcasual+1}$
  Count = Predicted Registered + Predicted Casual

  The results are shown in M6 – M8 in Table 6.

  In general, M6 – M8 give a better prediction than M1 – M5. Hence, we can conclude that using Approach 2 is more preferable than Approach 1.

- <u>Parameters setting</u>

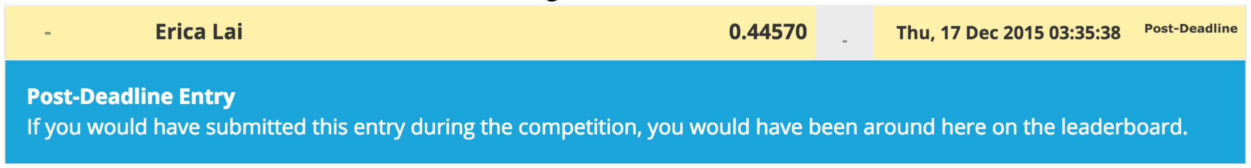
  Number of trees built(*ntree*):
  The algorithm has been run with 100, 200, 500 and 800 trees respectively. It is not surprising to see that the larger the number of tree built, the longer the run time of the algorithm. As shown from table 2, building 200 trees give a relative lower score on Kaggle, when running Random forests with 500 and 800 trees give very similar score (M2 vs M3 both give around 0.46 while M7 & M8 both give around 0.44). Hence striking the balance between runtime and prediction accuracy, it looks like building 500 trees is optimal.

  Number of variables randomly sampled as candidates at each split (*mtry*):
  By default, the number of variables selected at each split is p/3 for regression, where p is the number of predictor variables. Apart from the default value, I have also tried to

increase the size of *mtry* to 6 and 7 respectively. It is worth noting that mtry size with 7 give then best prediction, followed by mtry = 6 and 3. (M7 is better than M3 better than M4 and M4 better than M1)

- In consideration of all the factors above, we can conclude that in implementation this algorithm, tuning mtry is essential parameter in prediction, while Ntree is not critical parameter in prediction accuracy, but it is critical in computation time.

  M7 is the best model which is shown in Figure 10.



- Erica Lai 0.44570 - Thu, 17 Dec 2015 03:35:38 Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

Figure 10: Best Model for Random Forests

Model	Y	Ntree	Mtry	Runtime	Score
M1	lcount	200	3	26 mins	0.5100
M2	lcount	500	7	43 mins	0.46600
M3	lcount	800	7	62 mins	0.46487
M4	lcount	500	6	40 mins	0.47282
M5	lcount	100	7	23 mins	0.45357
M6	lcasual, lregistered	100	6	20 mins	0.44901
M7	**lcasual, lregistered**	**100**	**7**	**23 mins**	**0.44570**
M8	lcasual, lregistered	500	7	48 mins	0.44879

Table 2: Comparison of Random Forest Model

Section 4: Discussion

This project has considered 3 models for forecasting use of the bike rental in city bike share system. The 3 models include simple model, simple linear regression (with AIC for feature selection) and random forest. Among the 3 models, random forest gives the best performance, followed by the simple model and then linear regression model. The respective advantages or disadvantages of the models would be discussed below.

Comment on Simple model

Simple model has satisfactory prediction result of 0.57112. Although it is not the best prediction model, it is still preferable in the way that it is easy to implement, requirement no training and easy for interpreting the model.

Comparison of linear regression model and random forest:

Comparing the linear regression model and random forest, the reason why random forest gives a fairly better result worth discussion. One of the possible reasons could be that the response and prediction does not follow a directly linear relationship or there is interaction between different features. This is more likely to happen when there are various features in the datasets. The relationship between the response variable is conditional upon the values of other features. In that case, it is noteworthy that random forest is a tree-based model would be able to capture this kind of conditionality, but linear models only produce functions which is linear to the selected set of features.

Improvement for linear regression model:

To address this issue in linear regression model as mentioned above, one alternative is to consider interaction terms during the process of model selection. Besides, another solution could be transforming some of the variables (such as hours, weather, seasons) before fitting into the linear regression model.

Section 5: Reference:

Reading Materials for random forest:

1. https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf
2. http://statweb.stanford.edu/~jtaylo/courses/stats203/notes/selection.pdf
3. https://www.stat.berkeley.edu/~breiman/RandomForests/reg_examples/RFR.f