# Understanding the demand on bike sharing

Term project report

Weihong Situ & Xiaowen Huang

Math 448: Introduction to Statistical Learning and Data Mining

San Francisco State University

Faculty: Dr. Tao He

May 24, 2020

# Contents

# 1. Executive summary

Capital bikeshare is a company that provides bicycle sharing services in Washington, D.C. and neighbors. In the recent years, the bike rental service has spread around the globe. Since it promotes good prevention of the climate change problem, its demand increases over the years. This type of services is well-recognized in big cities. Our objective for this project is to make prediction about the daily demand on bike sharing based on the environmental and seasonal settings. These factors include season, temperature (Celsius), working day, holiday, weather condition, and others.

We downloaded the dataset "Bike Sharing Dataset" from the UCI machine learning database. The dataset contains the daily count of bike rentals between January 1, 2011 and December 31, 2012 in Capital bikeshare system with the corresponding weather and seasonal information. Just a one or two variables were added from other sources because they were not included in the original dataset extracted from the Capital bikeshare database. On the other hand, some independent variables are labeled as categorical predictors, however, not all of them have relative relationship. Therefore, we have to convert them to dummy variables in order to proceed to apply methods to fit the best model.

In this project, we start by visualizing the relationship between some categorical variables with the dependent variable. So, we plot four graphs to check the relationship between season, year, weather situation, and holiday versus the total bike rental counts, respectively. There is an indication of an overall increasing trend on the total amount of bike rentals during Summer and Fall. Similarly, the increasing trend happens when we make the comparison between 2011 and 2012. Also, there are results that are close to our expectations. For instance, there is a decreasing trend on the total amount of bike rentals as the weather situation gets worse. At the same time, the demand on bike rentals decreases on the days of either a holiday or weekend.

We used three methods which include the linear regression model with 10-fold CV, the decision tree regression model, and the random forest model. In this case, we compare the RMSE and MAE of each model, and select the model with the lowest RMSE and MAE. Out of the three methods, the random forest model results the lowest RMSE and MAE. Similarly, it presents with an adjusted $R^2$ close to 90%. The second-best model is the linear regression model with 10-fold CV, and the last one is the decision tree model.

With regard to future work, we conclude to work with other methods that are not used in this project. Perhaps, we can find a better model that fits the data better. Moreover, we would like to work with a larger dataset in order to present a model that best and accurately predict the demand on bike rentals.

## 2. Introduction

In this project, we focus on finding a model that best predict the daily demand on bike sharing based on the environmental and seasonal settings. To start off, we want to check the relationship between the categorical variables and the dependent variable. These plots provide us a sign of how the data are spread through the different seasons of the year, and the year itself. Also, we are interested in the effect of weather situation and the holidays versus the daily amount of bike rentals. We will split the data into training data (70%) and testing data (30%) in order to find a more accurate model. We will implement three methods to predict the model, such as linear regression with 10-fold cv, the decision tree regression model, and the random forest model. In this case, we will focus primarily on the value of RMSE and MAE. We select the model that has the lowest RMSE and MAE. Also, we will interpret the value of $R^2$ of each model.

## 3. Data

UCI machine learning respiratory provides a collection of databases to the machine learning community for the empirical analysis of machine learning algorithms. It demonstrates two datasets related to our topic. One aggregates the data on hourly basis and the second is based on daily basis. We decided to analyze the data that were recorded daily.

The dataset includes two-year historical log corresponding to 2011 and 2012 from the Capital Bikeshare system. It is also available publicly, http://capitalbikeshare.com/system-data. The weather information are extracted from http://www.freemeteo.com, and the holiday information is extracted from http://dchr.dc.gov/page/holiday-schedule.

There are 16 variables and 731 observations. No missing value is present in the dataset. Among the sixteen variables, the variables instant, dteday, casual, and registered were not considered significant to fit the model. The variable instant refers the record index, the variable dteday refers to the date, the variable casual refers to the count of casual users, and the variable registered refers to the count of registered users. Since we have set the goal of analyzing the demand on bike sharing during 2011- and 2012-time frame, the variable dteday becomes not helpful. Similarly, the variables casual and registered are not significant because we want to study the demand on rental bikes as a total count that has already included both casual and registered customers.

Therefore, the variables that we are working with are:

season: season, it categorizes 1 for springe, 2 for summer, 3 for fall, and 4 for winter.

year: year, it categorizes 0 for 2011, and 1 for 2012.

mnth: month, it categorizes 1 for January, 2 for February, and respectively.

holiday: indicates whether the day is a holiday or not

weekday: day of the week, it labels 0 for Sunday, 1 for Monday, and respectively.

workingday: it indicates 1 for if the day is neither weekend nor holiday, and 0 for otherwise.

weathersit: it indicates the weather situation, 1 for clear to partly cloudy, 2 for mist+cloudy to mist, 3 for light snow to light rain, and 4 for heavy rain to snow and fog.

temp: normalized temperature in Celsius. The values are divided to 41(max)

atemp: normalized feeling temperature in Celsius. The values are divided to 50 (max)

hum: normalized humidity. The values are divided to 100 (max)

windspeed: normalized wind speed. The values are divided to 67(max)

cnt: count of total rental bikes including both casual and registered.

# 4. Visualization

Here are some visualizations of the behavior of some independent and dependent variables.

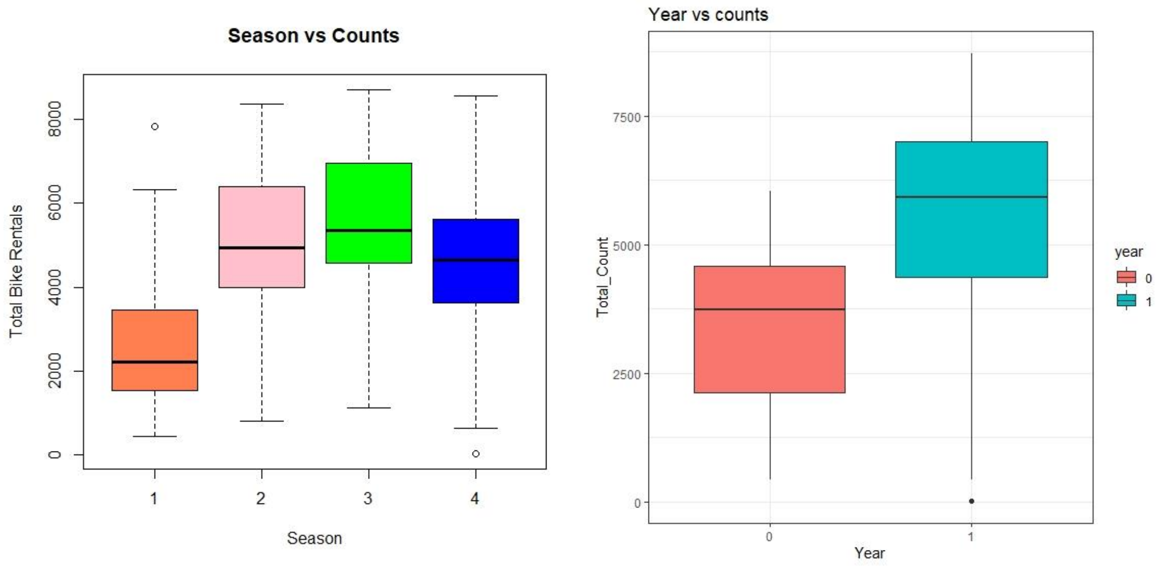


*Figure 1: Distribution of categorical variables plots – left: the relationship between season and total bike rentals plot; right: the relationship between year and total bike rentals plot.*

The boxplot on the left demonstrates the relationship between each season of the year and the total bike rentals within 2011 and 2012. Letting 1 for Spring, 2 for Summer, 3 for Fall, and 4 for Winter. There is an indication that the average numbers of bike rentals are the highest during Summer and Fall. On the other hand, the average of numbers of bike rentals is the lowest during Spring.

If we compare the average of bike rentals between 2011 and 2012, there is an indication of overall trend increasing from 2011 to 2012.

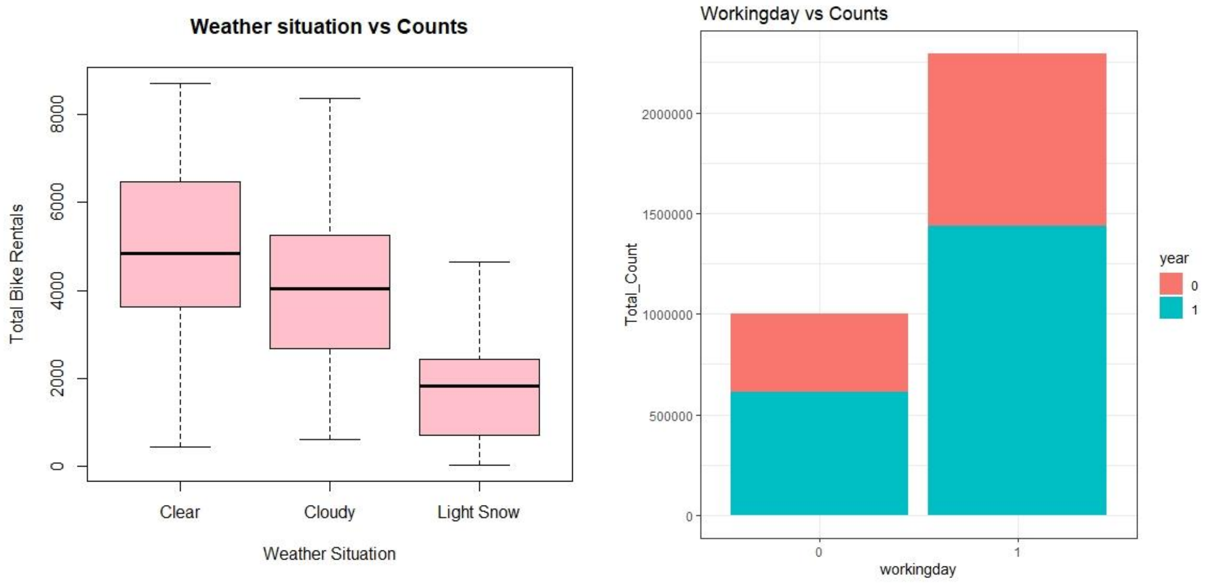


*Figure 2: Distribution of categorical variables plots – left: the relationship between weather situation and total bike rentals plot; right: the relationship between working day and total bike rentals plot.*

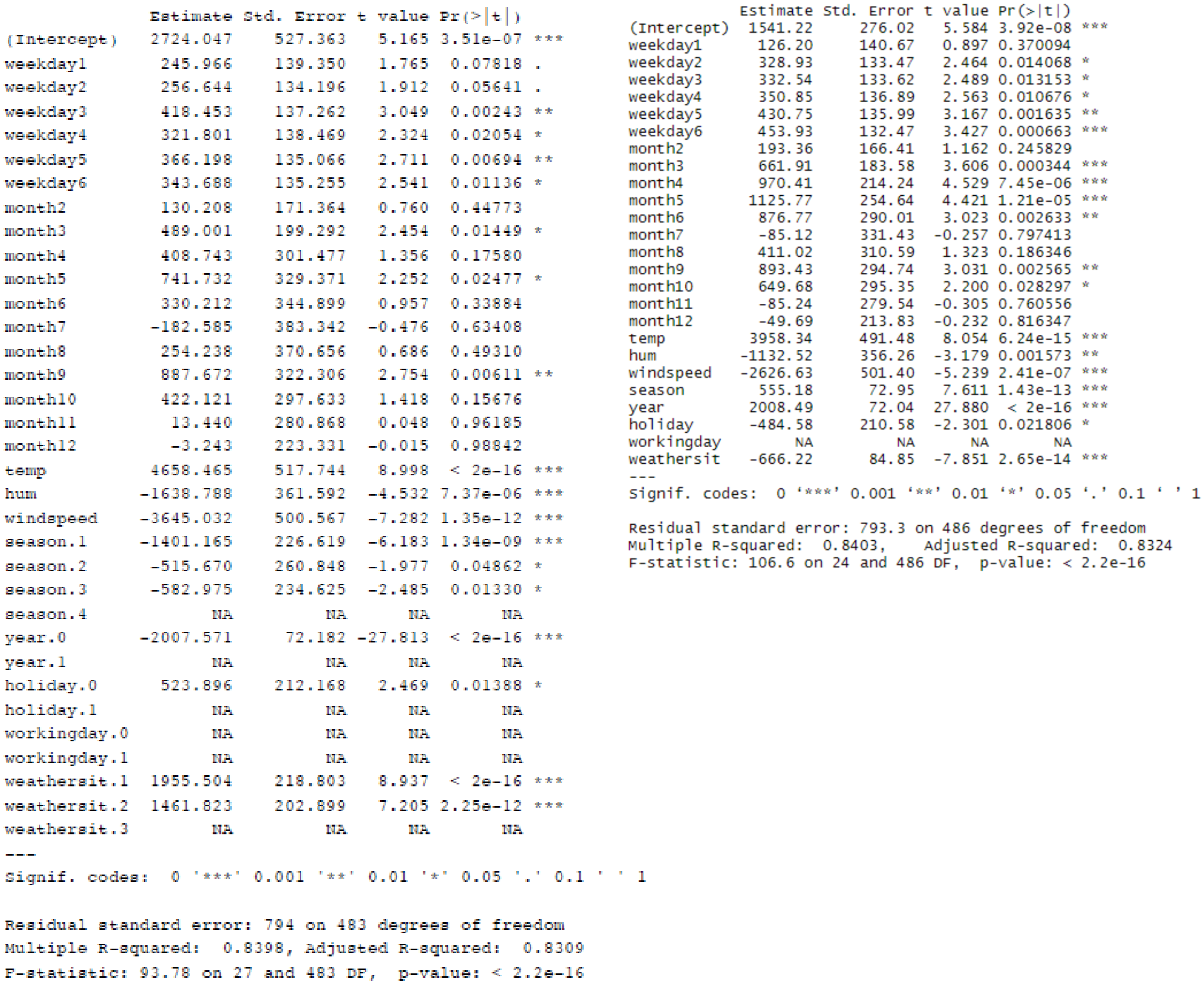
The boxplot on the left shows the average of numbers of bike rentals corresponding to four different weather situations. There is a decreasing trend as the weather gets worse. Also, the weather situation of heavy rain and ice pallets is not shown in the boxplot, which means there is no indicator of bike rentals under that weather condition. The graph on the right demonstrates the relationship between working day and the total bike rentals. The variable working day indicates 1 if the day is neither weekend nor holiday, otherwise is 0. There is an indication that the numbers of bike rentals increase on working days.

# 5. Model selection

First, we proceed to split the data into training data (70%) and testing data (30%). Also, we have to do some other changes. Since we have some categorical predictors that don’t contain any relative relationship, we have to transfer them to dummy variables.

## 5.1. Linear model with 10-fold CV prediction

The first method we used was 10-fold cross validation prediction. We started by converting the following variables to factors: dteday, yr, mnth, season, holiday, weekday, workingday, and weathersit. Fitting the model by having those independent variables as factors results that the estimate for some variables is NA. For instance, the estimates for season.4, year.1, holiday.1, workingday.0, workingday.1, and weathersit.3 are NA. Therefore, we decided to keep the variables season, yr, holiday, workingday, and weathersit as they are. From the right-side output, most of the variables are significant to the contribution of the model. Similarly, the both multiple $R^2$ and adjusted $R^2$ improved after we reduced the number of factor variables. We have that the adjusted $R^2$ is 0.8324 on average for 10-fold CV, which means that dependent variable can predict 83.24% of the variance in the target variable contributed by the independent variables.



```
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2724.047    527.363   5.165 3.51e-07 ***
weekday1       245.966    139.350   1.765  0.07818 .  
weekday2       256.644    134.196   1.912  0.05641 .  
weekday3       418.453    137.262   3.049  0.00243 ** 
weekday4       321.801    138.469   2.324  0.02054 *  
weekday5       366.198    135.066   2.711  0.00694 ** 
weekday6       343.688    135.255   2.541  0.01136 *  
month2         130.208    171.364   0.760  0.44773    
month3         489.001    199.292   2.454  0.01449 *  
month4         408.743    301.477   1.356  0.17580    
month5         741.732    329.371   2.252  0.02477 *  
month6         330.212    344.899   0.957  0.33884    
month7        -182.585    383.342  -0.476  0.63408    
month8         254.238    370.656   0.686  0.49310    
month9         887.672    322.306   2.754  0.00611 ** 
month10        422.121    297.633   1.418  0.15676    
month11         13.440    280.868   0.048  0.96185    
month12         -3.243    223.331  -0.015  0.98842    
temp          4658.465    517.744   8.998  < 2e-16 ***
hum          -1638.788    361.592  -4.532 7.37e-06 ***
windspeed    -3645.032    500.567  -7.282 1.35e-12 ***
season.1     -1401.165    226.619  -6.183 1.34e-09 ***
season.2      -515.670    260.848  -1.977  0.04862 *  
season.3      -582.975    234.625  -2.485  0.01330 *  
season.4            NA         NA      NA       NA    
year.0       -2007.571     72.182 -27.813  < 2e-16 ***
year.1              NA         NA      NA       NA    
holiday.0      523.896    212.168   2.469  0.01388 *  
holiday.1           NA         NA      NA       NA    
workingday.0        NA         NA      NA       NA    
workingday.1        NA         NA      NA       NA    
weathersit.1  1955.504    218.803   8.937  < 2e-16 ***
weathersit.2  1461.823    202.899   7.205 2.25e-12 ***
weathersit.3        NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 794 on 483 degrees of freedom
Multiple R-squared:  0.8398,	Adjusted R-squared:  0.8309 
F-statistic: 93.78 on 27 and 483 DF,  p-value: < 2.2e-16
```

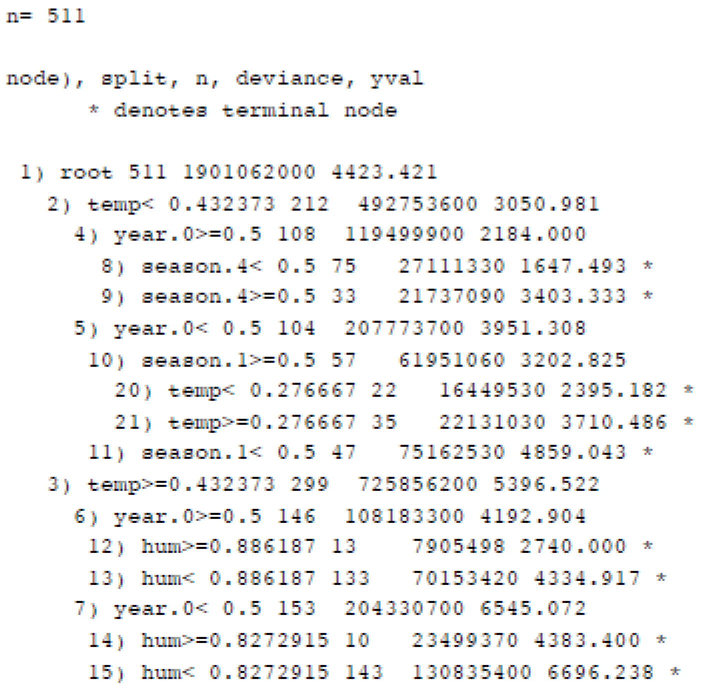
```
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1541.22     276.02   5.584 3.92e-08 ***
weekday1      126.20     140.67   0.897 0.370094    
weekday2      328.93     133.47   2.464 0.014068 *  
weekday3      332.54     133.62   2.489 0.013153 *  
weekday4      350.85     136.89   2.563 0.010676 *  
weekday5      430.75     135.99   3.167 0.001635 ** 
weekday6      453.93     132.47   3.427 0.000663 ***
month2        193.36     166.41   1.162 0.245829    
month3        661.91     183.58   3.606 0.000344 ***
month4        970.41     214.24   4.529 7.45e-06 ***
month5       1125.77     254.64   4.421 1.21e-05 ***
month6        876.77     290.01   3.023 0.002633 ** 
month7        -85.12     331.43  -0.257 0.797413    
month8        411.02     310.59   1.323 0.186346    
month9        893.43     294.74   3.031 0.002565 ** 
month10       649.68     295.35   2.200 0.028297 *  
month11       -85.24     279.54  -0.305 0.760556    
month12       -49.69     213.83  -0.232 0.816347    
temp         3958.34     491.48   8.054 6.24e-15 ***
hum         -1132.52     356.26  -3.179 0.001573 ** 
windspeed   -2626.63     501.40  -5.239 2.41e-07 ***
season        555.18      72.95   7.611 1.43e-13 ***
year         2008.49      72.04  27.880  < 2e-16 ***
holiday      -484.58     210.58  -2.301 0.021806 *  
workingday        NA         NA      NA       NA    
weathersit   -666.22      84.85  -7.851 2.65e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 793.3 on 486 degrees of freedom
Multiple R-squared:  0.8403,	Adjusted R-squared:  0.8324 
F-statistic: 106.6 on 24 and 486 DF,  p-value: < 2.2e-16
```

*Table 1: Summary of linear regression with 8 factor variables (left); summary of linear regression with 3 factor variables (right)*

we use the testing data to test the model and calculate the RMSE and MAE. In this case, the result of RMSE and MAE is slightly larger because the value of our dependent variable is relatively large and there are some outliers in the model. The fact that there is no obvious relationship between some variables, has caused the RMSE and MAE to be amplified.  We have that RMSE = 773.6467, and MAE = 563.7594.

## 5.2. Decision tree regression model

The next method we used was decision tree regression model. The root has 511 observations, and based on the condition, it splits the data to the following sub-bunches.



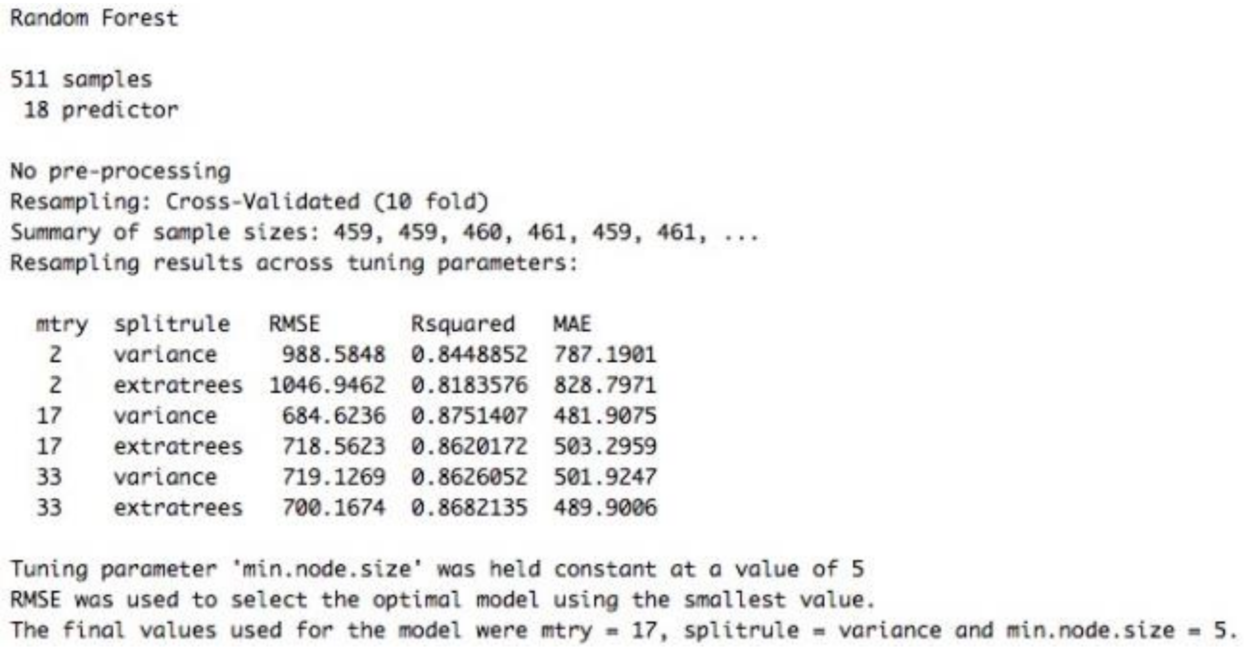
```
n= 511

node), split, n, deviance, yval
      * denotes terminal node

 1) root 511 1901062000 4423.421
   2) temp< 0.432373 212  492753600 3050.981
     4) year.0>=0.5 108  119499900 2184.000
       8) season.4< 0.5 75   27111330 1647.493 *
       9) season.4>=0.5 33   21737090 3403.333 *
     5) year.0< 0.5 104  207773700 3951.308
      10) season.1>=0.5 57   61951060 3202.825
        20) temp< 0.276667 22   16449530 2395.182 *
        21) temp>=0.276667 35   22131030 3710.486 *
      11) season.1< 0.5 47   75162530 4859.043 *
   3) temp>=0.432373 299  725856200 5396.522
     6) year.0>=0.5 146  108183300 4192.904
      12) hum>=0.886187 13    7905498 2740.000 *
      13) hum< 0.886187 133   70153420 4334.917 *
     7) year.0< 0.5 153  204330700 6545.072
      14) hum>=0.8272915 10   23499370 4383.400 *
      15) hum< 0.8272915 143  130835400 6696.238 *
```

*Table 2: Summary of decision tree model*

Also, we apply the 10-fold CV and calculate the RMSE, $R^2$, and MAE. Normally, we pick the model with the lowest RMSE. In this case, the RMSE = 879.1844, and MAE = 660.2696. Hence, we pick the linear model with 10-fold CV prediction. Also, the $R^2$ for the decision tree model is 0.61 on average for 10-fold cv., which means it can only predict 61% of the data.

## 5.3. Random forest model

The last method we used was the random forest model. We fit the random forest model by setting the trees equals 200. Then, we use the 10-fold cv to predict the RMSE and MAE. The RMSE is close to 700, MAE is close to 500, and $R^2$ is close to 90%. Similarly, we use the testing data to test the model performance. As result, we have RMSE = 447.5723, MAE = 299.5527. This means the generalization ability of the model is good.



```
Random Forest

511 samples
 18 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 459, 459, 460, 461, 459, 461, ...
Resampling results across tuning parameters:

  mtry  splitrule   RMSE       Rsquared   MAE
   2    variance     988.5848  0.8448852  787.1901
   2    extratrees  1046.9462  0.8183576  828.7971
  17    variance     684.6236  0.8751407  481.9075
  17    extratrees   718.5623  0.8620172  503.2959
  33    variance     719.1269  0.8626052  501.9247
  33    extratrees   700.1674  0.8682135  489.9006

Tuning parameter 'min.node.size' was held constant at a value of 5
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were mtry = 17, splitrule = variance and min.node.size = 5.
```

*Table 3: Summary of the random forest model*

# 6. Conclusion

After comparing the root mean squared error (RMSE) and mean absolute error (MAE) of all the three models, the random forest model has less RMSE and MAE. Hence, we opt to use the random forest model, and conclude that the random forest model is the best for predicting the daily demand on bike sharing based on the environmental and seasonal settings.

For future improvements, we would like to use other methods to fit the model, such as Lasso, Ridge, stepwise selection, and others. Perhaps, we can find a lower RMSE and MAE. However, the random forest model is good enough now. We consider that it would be a good idea to use a larger dataset which include the log of a longer period. Since the dataset we used in this project only holds 731 observations, the resulted model may not be accurate enough to predict future bike rental counts.

# A. Code

```
library(data.table)
library(ggplot2)
library(purrr)
library(caret)
library(rpart)
library(randomForest)
library(ranger)
library(corrgram)
data <- read.csv(file="C:/Users/Henry & Wendy/Downloads/MATH 448 project/day.csv")
str(data)

data$dteday <- as.Date(data$dteday)
#data$yr <- as.factor(data$yr)
data$mnth <-as.factor(data$mnth)
#data$season <- as.factor(data$season)
#data$holiday<- as.factor(data$holiday)
data$weekday<- as.factor(data$weekday)
#data$workingday<- as.factor(data$workingday)
#data$weathersit<- as.factor(data$weathersit)
str(data)
setnames(data,"dteday","date")
setnames(data,"yr","year")
setnames(data,"mnth","month")
setnames(data,"cnt","total_count")

### Missing value
missing <- data.frame(apply(data,2,function(x){sum(is.na(x))}))
names(missing)[1] <- "missing_value"
missing
```

```
###Visualizations
boxplot(data$total_count ~ data$season,
        main = "Season vs Counts",
        data = data,
        xlab = "Season",
        ylab = "Total Bike Rentals",
        col = c("coral", "pink", "green", "blue"))

ggplot(data,aes(x=year,y=total_count,fill=year))+theme_bw()+geom_boxplot()+
  labs(x='Year',y='Total_Count',title='Year vs counts')

boxplot(data$total_count ~ data$weathersit,
        data=data,
        main="Weather situation vs Counts",
        xlab = "Weather Situation",
        ylab = "Total Bike Rentals",
        names= c("Clear","Cloudy","Light Snow"),
        col = c("pink", "pink", "pink", "pink"))

ggplot(data,aes(x=workingday,y=total_count,fill=year))+geom_col()+theme_bw()+
  labs(x='workingday',y='Total_Count',title='Workingday vs Counts')

### split data
train_index<-sample(1:nrow(data),0.7*nrow(data))
train_data<-data[train_index,]
test_data<-data[-train_index,]
dim(train_data)
dim(test_data)


train<-subset(train_data,select=c('season','year','month','holiday',
'weekday','workingday','weathersit','temp','hum','windspeed','total_count'))
```

```
test<-subset(train_data,select=c('season','year','month','holiday',
'weekday','workingday','weathersit','temp','hum','windspeed','total_count'))


train_cat_attributes<-subset(train,select=c('season','holiday','workingday','weathersit','year'))

test_cat_attributes<-subset(test,select=c('season','holiday','workingday','weathersit','year'))

train_num_attributes<-subset(train,select=c('weekday','month','temp','hum','windspeed','total_count'))

test_num_attributes<-subset(test,select=c('weekday','month','temp','hum','windspeed','total_count'))


othervars<-c('month','weekday','temp','hum','windspeed','total_count')


###train

set.seed(888)

vars<-setdiff(colnames(train),c(train$total_count,othervars))

f <- paste('~', paste(vars, collapse = ' + '))

encoder<-dummyVars(as.formula(f), train)

encode_attributes<-predict(encoder,train)

train_encoded_attributes<-cbind(train_num_attributes,encode_attributes)

head(train_encoded_attributes,5)


### test

set.seed(999)

vars<-setdiff(colnames(test),c(test$total_count,othervars))

f<- paste('~',paste(vars,collapse='+'))

encoder<-dummyVars(as.formula(f),test)

encode_attributes<-predict(encoder,test)

test_encoded_attributes<-cbind(test_num_attributes,encode_attributes)

head(test_encoded_attributes,5)


###Excluding year, season, holiday, workingday, and weathersit to be factors ##

data$dteday <- as.Date(data$dteday)

#data$yr <- as.factor(data$yr)

data$mnth <-as.factor(data$mnth)
```

```
#data$season <- as.factor(data$season)
#data$holiday<- as.factor(data$holiday)
data$weekday<- as.factor(data$weekday)
#data$workingday<- as.factor(data$workingday)
#data$weathersit<- as.factor(data$weathersit)
str(data)
setnames(data,"dteday","date")
setnames(data,"yr","year")
setnames(data,"mnth","month")
setnames(data,"cnt","total_count")
####

### fit model(linear model)
set.seed(672)
lr_model<-lm(train_encoded_attributes$total_count~.,train_encoded_attributes[,-c(6)])
summary(lr_model)

### CV
set.seed(672)
train.control<-trainControl(method='CV',number=10)
CV_predict<-train(total_count~.,data=train_encoded_attributes,method='lm',trControl=train.control)
summary(CV_predict)

### test
options(warn=-1)
lm_predict<- predict(lr_model,test_encoded_attributes[,-c(6)])
head(lm_predict,5)
set.seed(688)
rmse<-RMSE(lm_predict, test_encoded_attributes$total_count)
print(rmse)
#Mean squared error
```

```
mae<-MAE(lm_predict, test_encoded_attributes$total_count)
print(mae)

y_test<-test_encoded_attributes$total_count
residuals<-y_test-lm_predict
plot(y_test,residuals,xlab='Observed',ylab='Residuals',main='Residual plot')
abline(0,0)

### Decision Tree  ###
set.seed(568)
rpart.control<-rpart.control(minbucket = 2,cp = 0.01,maxcompete = 3, maxsurrogate = 4, usesurrogate = 2, xval = 3,surrogatestyle = 0, maxdepth = 10)
dtr<-rpart(train_encoded_attributes$total_count~.,data=train_encoded_attributes[,-c(6)],control=rpart.control,method='anova',cp=0.01)
dtr

fit_tree = tree(train_encoded_attributes$total_count ~ . , subset = train, data=train_encoded_attributes[,-c(6)])
plot(fit_tree)
text(fit_tree, pretty=0)

### CV
options(warn=-1)
set.seed(5769)
train.control<-trainControl(method='CV',number=10)
dtr_CV_predict<-train(total_count~.,data=train_encoded_attributes,method='rpart',trControl=train.control)
dtr_CV_predict

### test
set.seed(7882)
dtr_predict<-predict(dtr,test_encoded_attributes[,-c(6)])
head(dtr_predict,5)
```

```
set.seed(6889)
rmse<-RMSE(y_test,dtr_predict)
print(rmse)
mae<-MAE(y_test,dtr_predict)
print(mae)

set.seed(6889)
rmse<-RMSE(y_test,dtr_predict)
print(rmse)
mae<-MAE(y_test,dtr_predict)
print(mae)

residuals<-y_test-dtr_predict
plot(y_test,residuals,xlab='Observed',ylab='Residuals',main='Residual plot')
abline(0,0)

### Random forest ###
set.seed(1)
rf_model<-randomForest(total_count~.,train_encoded_attributes,importance=TRUE,ntree=200)
rf_model

### CV
set.seed(2)
train.control<-trainControl(method='CV',number=10)
#Cross validation prediction
rf_CV_predict<-train(total_count~.,train_encoded_attributes,method='ranger',trControl=train.control)
rf_CV_predict

### test
set.seed(3)
rf_predict<-predict(rf_model,test_encoded_attributes[,-c(6)])
```

```
head(rf_predict,5)

### Final prediction, the best model is random forest model.
Bike_predictions=data.frame(y_test,rf_predict)
write.csv(Bike_predictions,'Bike_Renting_R.CSV',row.names=F)
Bike_predictions
```