Practice Exam – Bike Sharing Demand (SOA PA 12/7/20)

Instructions to Candidates: Please remember to avoid using your own name within this document or when naming your file. There is no limit on page count.

As indicated in the instructions, work on each task should be presented in the designated section for that task.

Task 1 – Examine each variable and make appropriate adjustments

We performed adjustments to the variables as follows:

- The **season** variable is left as an integer because seasons can be compared to one another. For example, spring comes after fall which comes after summer.
- The **year** variable has only two values. If I left this as numeric, a GLM would fit a single coefficient which would be multiplied by either 0 or 1. This would have the same effect as if it were converted to a factor variable which had only two values. I chose to convert this to a factor so that the base level will be the one with the most observations.
- **Holiday** is converted to a factor for the same reason that year was.
- **Weekday** will be significantly different on workdays as compared to week ends and so I converted this to a factor. If there was an unequal distribution of records across **weekday** then I would simplify the factor levels.
- The **weather** situation has no order. The values are clear or partly cloudy, mist, or snow, and so this was converted to a factor.

For each of these factors, the base level was set to be the value which has the most observations.

```
     season           year           hour               holiday      weekday
 Min.   :1.000   year2012:8732   Min.   : 0.00   not_holiday:16876   sat:2511
 1st Qu.:2.000   year2011:8644   1st Qu.: 6.00   holiday    :  500   sun:2502
 Median :3.000                   Median :12.00                       mon:2478
 Mean   :2.502                   Mean   :11.55                       tue:2453
 3rd Qu.:3.000                   3rd Qu.:18.00                       wed:2474
 Max.   :4.000                   Max.   :23.00                       thu:2471
                                                                     fri:2487
 weathersit      temp          humidity       windspeed      bikes_per_hour
 1:11413    Min.   :0.020   Min.   :0.0000   Min.   :0.0000   Min.   :  1.0
 2: 4544    1st Qu.:0.340   1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.: 40.0
 3: 1419    Median :0.500   Median :0.6300   Median :0.1940   Median :142.0
            Mean   :0.497   Mean   :0.6272   Mean   :0.1901   Mean   :189.5
            3rd Qu.:0.660   3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.:281.0
            Max.   :1.000   Max.   :1.0000   Max.   :0.8507   Max.   :977.0
```

Task 2 – Consider a new variable

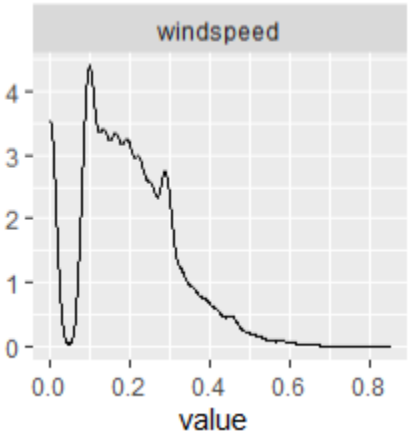
The data already contains weekdays as a factor variable and so there is overlapping information with the **workday**. This is essentially a simplified factoring of the **weekday**. One disadvantage is that the matrix would be rank deficient if both variables were included at the same time because there is a collinearity between the two. Once the day of the week is known then we also know if it is a workday or not. For GLMs, there would be a convergence issue because the determinant of the model matrix would be zero.

One advantage is that fewer coefficients would be used in the GLM. For tree-models, there would be fewer factor levels. This would reduce the variance of the model and make it more difficult to overfit.

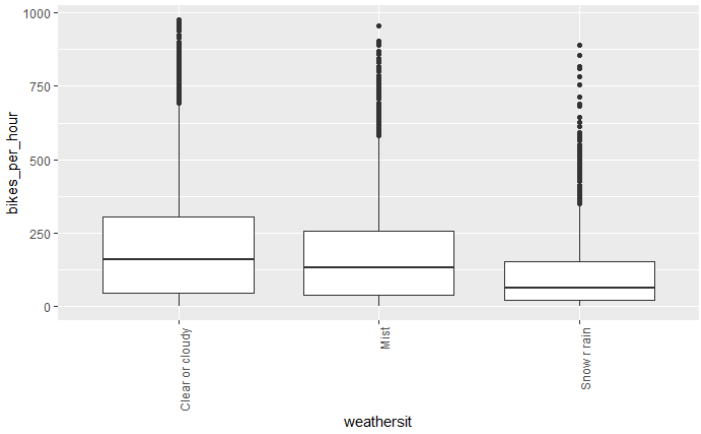
Task 3 – Write an overview of the data for your actuarial manager

The data consists of 17,376 observations and 9 predictor variables which are related to the target, the number of bicycles rented in an hour. We will use these variables to build a model that will allow ABC to predict the demand for their rental program based on the **season, year, hour, holiday, weekday, weather situation, temperature, humidity, and windspeed**.

The **windspeed** distribution below shows that most days are not windy. This distribution is right skewed and has a max of 67 and a mean of 12.7. The raw data values have been divided by the maximum (67).



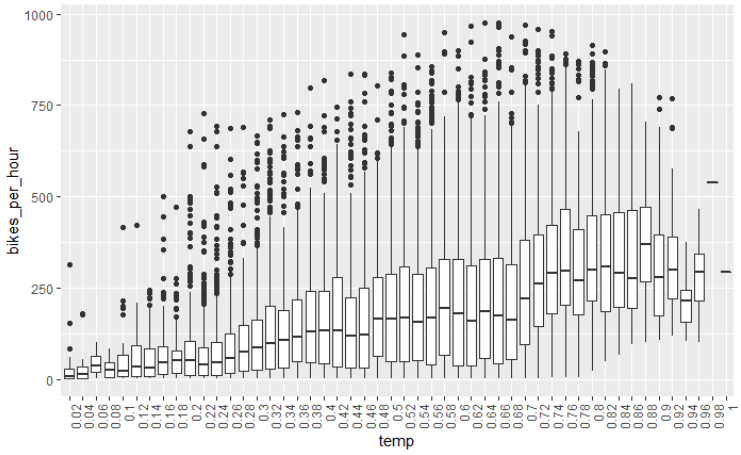
Poor **weather** makes riding bikes hazardous. In addition to the damage that rain and water can cause to clothing, vehicles have less visibility of the road and are less likely to see bicyclists. The mean number of riders during rain or snow is about half of that during clear or cloudy weather (111.6 as compared to 204.9). The boxplot below shows that distribution of riders during snow or rain (far right) is lower than clear or cloudy or misty weather.



In addition to safety, comfort plays a critical role in the demand for renting bikes. Unlike cars which have onboard heaters and air conditioning, bike riders are at the mercy of the temperature that day.

During hot or cold weather, we see fewer people renting. The correlation between the number of riders and the temp is positive 0.41. This can be seen in the increasing trend in the median values of the boxplot below. While there are fewer riders at higher temperatures, as the number of riders decreases sharply at about 0.8 (29 degrees Celsius), the number of riders increase from -9 degrees all the way up to about 39.

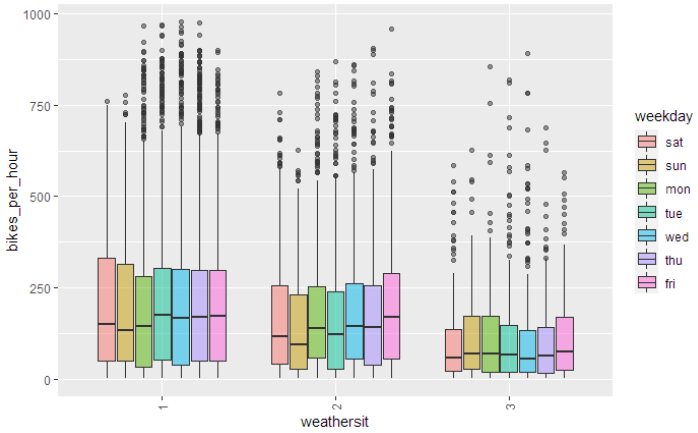
The values of this variable in the graph can only interpreted as meaning that a higher value corresponds to a higher temp.



Task 4 – Select an interaction to consider for your model

An interaction is when the impact that a variable has on the target depends on another variable which is used in the model.

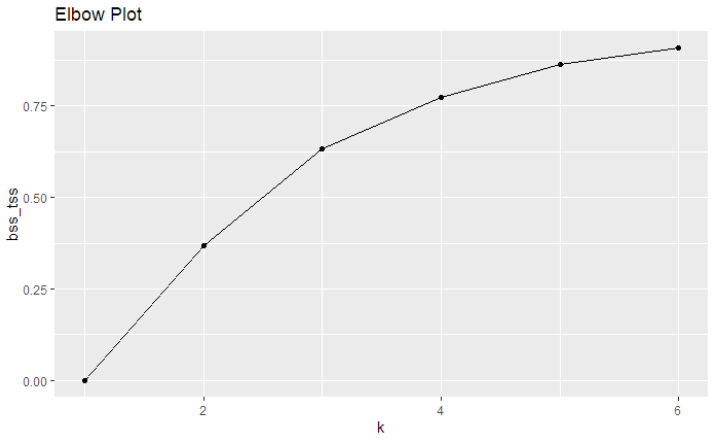
The graph below shows the interaction between the weather situation and the day of the week. The x-axis values of **weathersit** 1 and 2 are for clear or misty weather whereas the far right set of boxplots for **weathersit** 3 is for rainy or snowy weather. The orange bar represents Saturdays, and when it is not raining or snowing, these are popular days for ridings bikes as can be seen by the vertical height of the bar. During rain or snow, however, Saturdays are the least popular day. This makes sense because most people do not work during the week and therefore this represents leisure time which they could choose to spend elsewhere if the weather is not nice.



When **weathersit** is 1 or 2 on Saturdays, the median bikes per hour is about 149. In comparison, during rain or snow, this is only 57.

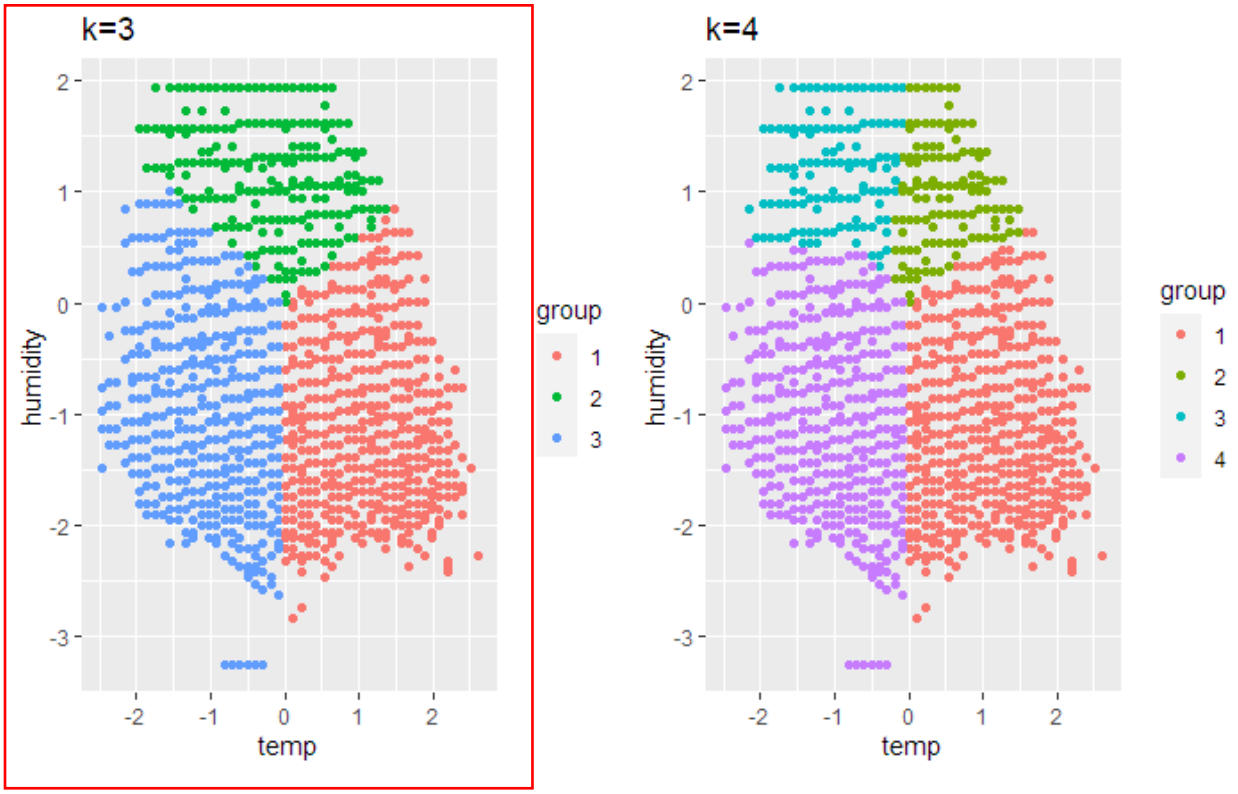
Task 5 – Perform a k-means cluster analysis

K-means is an unsupervised learning algorithm which groups observations into clusters based on the Euclidean distance. The method begins by randomly assigning cluster centers to the numerical variables. Then for each of the points, the distances to the centers are computed using the distance formula and the point is assigned to the cluster which is closest. Then the cluster centers are re-computed by taking the mean of each of the values in that cluster. This process is repeated until the cluster center stop moving. There is a parameter called n.start which controls the number of times that the algorithm is run. If only a few runs are used, then the result can converge to a local minimum of the within cluster deviance instead of the global minimum and this results in unstable cluster assignments. The code was set up to use 10 starting values.



K-means relies on the number of clusters to be specified in advance. The above graph shows the between cluster total sum of squares on the y-axis and the number of clusters on the x-axis. There is now obvious “elbow” in this graph and so we choose the value of 3. This segments the data into three groups.

The graph on the left below shows this.



Based on the graphs above, we do not recommend using the cluster variable because the groups are not distinct. If the clusters were adding predictive value, then there would be distinct regions of points for each of the colors. This is the same regardless of whether we choose 3 – 5 clusters. It is worth noting that there are several points where humidity is between -1 and -3 and temp is between -1 and 0 which appear to be outliers. It does not make physical sense that the humidity would be the same on each of these days and so we suggest looking into these points to see if there is an error in the data.

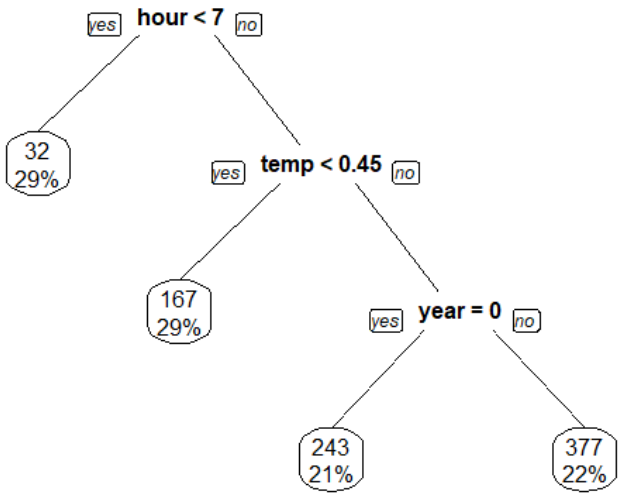
Task 6 – Construct a decision tree

The code is set to split the data into 70% training and 30% testing. We examined the mean bikes per hour in both the training and test sets as well as the combined data and sawt that this is about 189.

The default code for training the tree uses a low cp value of 0.001 and a minbucket of 20. The CP is the complexity parameter and controls the variance of the model. A lower CP indicates that the tree will have more branches and variables. In tuning the tree, first a deep tree with many branches is fit. Then a pruning algorithm is applied to create subtrees which remove branches that do not result in the model error being reduced by a factor of CP. Any split that does not decrease the overall lack of fit by a factor of cp is not attempted.

The min bucket is the minimum number of observations which are allowed in a terminal (leaf) node. If fewer than 20 observations are split off in a branch, then this branch becomes a terminal node.

The tree was far to complicated to be used for this problem and so we increased the CP to 0.05 which results in this tree which has four terminal nodes.



We interpret the above tree diagram beginning at the top and working downwards. First, the tree asks people if they have rented bikes from hour 0 to hour 6, which is the night and early morning. The number of bikes rented per hour during this time is approximately 32, and this represents 29% of the records. If the bike is rented from hour 7 to hour 23, and if the temp is below 30.6, then the predicted value is 167. This could represent people who are commuting during working hours only during warm weather but prefer to drive or take a train during cold weather. If the temp is above 30.6 at this stage of the tree, and if the year is 2011, then the predicted value is 243 and 377 if the year is 2012. This year variable does appear in the tree which indicates that the ABC bike sharing programing is changing from year to year. In a decision tree, the variables higher up are more important. This allows us to read out that hour is more important than temp which is more important than year. If year did not appear in the tree, then the bike rental program would have been the same in 2011 as it was in 2012.

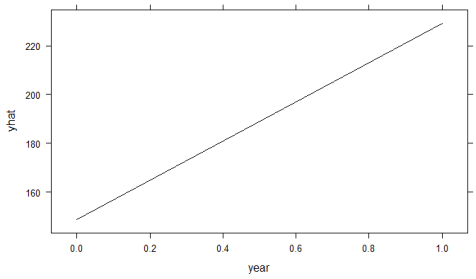
Task 7 – Construct a boosted decision tree

It is important to consider whether the business problem is to gain inference about data or to make predictions without needing to understand the model. The Project Statement says that ABC is interested only in prediction and so this allows for gradient boosted trees as well as bagged trees. For this problem, we have a mix of numeric and categorical variables. The numeric variables may have non-linear relationships with the target and boosted trees can use these variables directly as opposed to a linear model which requires manual transformations. There are correlations between the variables such as humidity and hour (-0.27) and temp and season (0.31) the p-values for one of the correlated variables tends to be low because one assumption of GLMs is that the variables are independent; whereas with boosted trees, the variables are used together in the model although they are incorporated sequentially into different trees. For a single tree, correlated variables are automatically selected out.

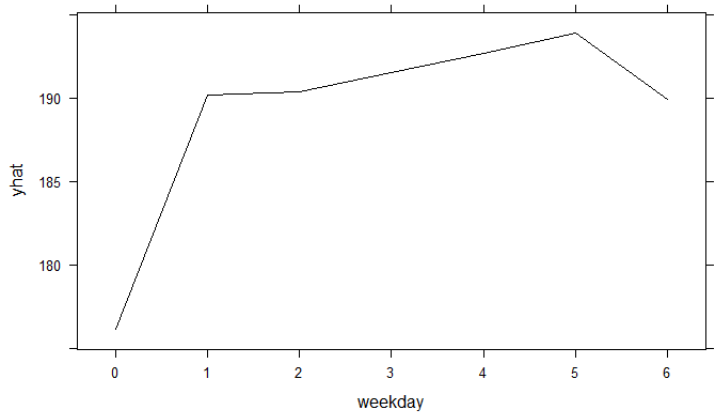
The table below shows the variable importance. This is a numeric way of measuring which variables are predictive of the target. The value is always between 0 and 100 and the rank ordering of the values represents their contributes to each of the trees used. This is taken as the average contribution over all the boosting iterations. The two most important factors are **year** and **weekday**.

	var <chr>	rel.inf <dbl>
hour	hour	62.9523206
temp	temp	13.6558933
year	year	9.3605387
weekday	weekday	6.1972214
season	season	3.6882132
humidity	humidity	2.2458125
weathersit	weathersit	1.6799971
holiday	holiday	0.1104851
windspeed	windspeed	0.1095182

The partial dependence plot (PDP) for year is below. Although R is creating a line graph, a bar graph with two bars, one at 0 and one at 1.0 would be more appropriate. The partial dependence is the effect of the variable after integrating out all the other variables in the model. This is done by considering all the pairwise combinations of year 0 and year 1 with the other values in the training data and then looking at the average of the number of bikes per hour. The result agrees with what we saw from the single decision tree because 2011 (year = 0) is lower than 2012 (year = 1).



The **weekday** PDP graph shows that weekends **weekday** = 0 (Sundays) and **weekday** = 1 (Saturdays). This agrees with our intuition because people who commute to work using rental bikes do not work during the weekends. For Monday – Friday, the number of bikes is approximately constant between 190 and 200. Saturdays are more popular than Sundays (170 as compared to 190).



A gradient boosted tree (GBM) sequentially fits trees to predict the target. The algorithm begins by predicting the average of the target and then calculates the residuals as the differences between the value of Y and the average of Y. Then a second tree is fit to predict this residual. This process continues for 1000 times. At each step, the predictions are updated so that larger residuals have more weight. The learning rate controls how quickly the predictions change with respect to the number of boosting iterations. These two models use a high learning rate (0.1) and a low learning rate (0.01). As the learning rate is lower, more trees are required. In general, the optimal method of tuning a GBM is to use early stopping which sets a low learning rate and then continues to add more trees until the error stops decreasing. That is out of scope of this project.

The first tree has a mean squared error (MSE) of 6062 on the test set which is worse than the second tree which has an MSE of 2,356. For this reason, we chose to use 0.01.

A single decision tree partitions the feature space into rectangular regions and then makes predictions which are the average of training samples. Because the number of bike rentals is always positive, the averages will also be. Therefore, decision trees will only make positive predictions. A boosted tree, on the other hand, makes predictions for the residuals. Therefore, if a tree overcompensates for a positive residual by subtracting from the prediction, the result can be negative.

Task 8 – Compare distribution choices for a generalized linear model (GLM)

Poisson with Log Link

The Poisson model uses a log link function which is the canonical function for this family. The target is a counting value and so this agrees with the Poisson distribution choice. The log link has the added benefits of being easy to interpret because the model can be translated into a product of relativities and being able to interpret the values of the coefficients as having a percentage-change impact on the target. The mean will always be positive because the inverse of a log is the exponential function which has a positive range.

```
Call:
glm(formula = bikes_per_hour ~ ., family = poisson(link = "log"),
    data = data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-24.962   -8.661   -2.992    3.960   38.176

Coefficients:
                          Estimate Std. Error  z value Pr(>|z|)
(Intercept)             4.2306508  0.0059068  716.231  < 2e-16 ***
season2                 0.1511775  0.0019232   78.607  < 2e-16 ***
season1                -0.0496080  0.0031669  -15.665  < 2e-16 ***
season4                 0.4109726  0.0023787  172.774  < 2e-16 ***
year2011               -0.4253061  0.0013673 -311.055  < 2e-16 ***
hour                    0.0446378  0.0001095  407.532  < 2e-16 ***
holidayholiday         -0.1713550  0.0045152  -37.951  < 2e-16 ***
weekdayFri              0.0101044  0.0024103    4.192 2.76e-05 ***
weekdayWed             -0.0125564  0.0024535   -5.118 3.09e-07 ***
weekdayMon             -0.0195864  0.0025306   -7.740 9.95e-15 ***
weekdaySun             -0.0492539  0.0024966  -19.729  < 2e-16 ***
weekdayTue             -0.0146127  0.0024500   -5.964 2.46e-09 ***
weekdayThu             -0.0012140  0.0024362   -0.498    0.618
weathersitMist          0.0725231  0.0016312   44.461  < 2e-16 ***
weathersitSnow r rain -0.2378218  0.0033096  -71.859  < 2e-16 ***
temp                    1.8875775  0.0057478  328.398  < 2e-16 ***
humidity               -0.9238978  0.0042351 -218.152  < 2e-16 ***
windspeed               0.2412924  0.0057204   42.181  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2015403  on 12164  degrees of freedom
Residual deviance: 1154021  on 12147  degrees of freedom
AIC: 1231675
```

Test MSE: 20,021.94

AIC: 1,231,675

Gamma with Inverse Link

A gamma distribution is continuous and strictly positive. The target variable is discrete, however, target variables which span a wide range of numbers, using a continuous distribution is acceptable because the results are approximately the same. The link function of inverse is reasonable because this is the canonical link for a Gamma distribution. One disadvantage is that there is a possibility for negative predicted values because the inverse mean function has negative numbers in its range. The inverse function is monotonic which means that we can interpret the signs of the coefficients as either increasing or decreasing the predictions.

```
Call:
glm(formula = data.train$bikes_per_hour ~ ., family = Gamma(link = "inverse"),
    data = data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7867  -0.9796  -0.2061   0.3395   2.7246

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.092e-02  3.289e-04  33.213  < 2e-16 ***
season2              -6.224e-04  8.839e-05  -7.042 2.00e-12 ***
season1               1.154e-03  2.002e-04   5.765 8.38e-09 ***
season4              -1.826e-03  1.205e-04 -15.152  < 2e-16 ***
year2011              1.738e-03  8.616e-05  20.173  < 2e-16 ***
hour                 -2.395e-04  6.939e-06 -34.509  < 2e-16 ***
holidayholiday        8.580e-04  2.724e-04   3.150  0.00164 **
weekdayFri           -9.721e-05  1.150e-04  -0.845  0.39793
weekdayWed            1.518e-07  1.157e-04   0.001  0.99895
weekdayMon           -1.080e-04  1.319e-04  -0.818  0.41320
weekdaySun            4.249e-05  1.304e-04   0.326  0.74446
weekdayTue           -9.965e-05  1.216e-04  -0.819  0.41272
weekdayThu           -8.810e-05  1.183e-04  -0.745  0.45654
weathersitMist       -3.332e-04  8.495e-05  -3.922 8.83e-05 ***
weathersitSnow r rain  1.746e-03  2.525e-04   6.914 4.94e-12 ***
temp                 -7.544e-03  2.947e-04 -25.600  < 2e-16 ***
humidity              3.811e-03  2.365e-04  16.115  < 2e-16 ***
windspeed            -1.387e-03  2.887e-04  -4.803 1.58e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.8101203)

    Null deviance: 17211  on 12164  degrees of freedom
Residual deviance: 12891  on 12147  degrees of freedom
AIC: 147690
```

Test MSE: 127,311.3

AIC: 147,690

We wish to maximize the penalized log likelihood (AIC) and minimize the Mean Squared Error (MSE) and the first Poisson model accomplishes this by having a lower MSE and higher AIC.

To measure the effect that **season** has on the number of bikes, we look at the mean value and then change the coefficients to be from summer (which is the intercept term) to winter. Before beginning, I checked that the base level is summer because this has the most observations. Then we apply the inverse link function, which is the exponential in the first model and the inverse in the second model. This converts the linear predictor to the predicted mean number of bikes per hour.

The overall average number of bikes for the first model is about 105 and then 100 in the winter.

Poisson with Log Link	
Summer	105.2
Winter	100.1
	-5.1

The second model makes an overall average of 109 and then 97 in the winter.

Gamma with Inverse Link	
Summer	108.9
Winter	96.7
	-12.1

This makes intuitive sense because during warmer weather we expect that more people will ride their bikes than in cold weather. To measure the temp impact, we first converted the temp values to be on the same scale as the normalized values in the model.

temp	normalized temp
10	0.40
20	0.60

Then we applied the inverse link functions to the linear predictors. In both models when the temperature is higher there are more predicted bikes. This agrees with our exploratory results. The Poisson model, we use the exponential function, and we use the inverse function in the Gamma model.

Mean of Poisson at 10 C	222.1
Mean of Poisson at 20 C	329.1
Mean of Gamma at 10 C	161.3
Mean of Gamma at 20 C	216.1

Task 9 – Evaluate the interaction term

Decision trees automatically capture interaction effects and so we do not need to specify this as we do with GLMs. The tree algorithm recursively splits the data using different yes or no questions, and each question can be understood as either increasing or decreasing the running total for the predicted value. An interaction is when a variable has a different impact on the target depending on the values of other variables in the model, which in the case of a tree is just the variables which are higher up in the tree. For example, if variable A were interacting with a variable B, then there would be first a split based on A, and then for each of these values of A, the B would have a different impact on the predicted number of bikes.

Model without interaction

```
Call:
glm(formula = bikes_per_hour ~ ., family = poisson(link = "log"),
    data = data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-24.962   -8.661   -2.992    3.960   38.176

Coefficients:
                       Estimate Std. Error  z value Pr(>|z|)
(Intercept)           4.6559569  0.0061350  758.921  < 2e-16 ***
season2               0.1511775  0.0019232   78.607  < 2e-16 ***
season1              -0.0496080  0.0031669  -15.665  < 2e-16 ***
season4               0.4109726  0.0023787  172.774  < 2e-16 ***
year                 -0.4253061  0.0013673 -311.055  < 2e-16 ***
hour                  0.0446378  0.0001095  407.532  < 2e-16 ***
holidayholiday       -0.1713550  0.0045152  -37.951  < 2e-16 ***
weekdayFri            0.0101044  0.0024103    4.192 2.76e-05 ***
weekdayWed           -0.0125564  0.0024535   -5.118 3.09e-07 ***
weekdayMon           -0.0195864  0.0025306   -7.740 9.95e-15 ***
weekdaySun           -0.0492539  0.0024966  -19.729  < 2e-16 ***
weekdayTue           -0.0146127  0.0024500   -5.964 2.46e-09 ***
weekdayThu           -0.0012140  0.0024362   -0.498    0.618
weathersitMist        0.0725231  0.0016312   44.461  < 2e-16 ***
weathersitSnow r rain -0.2378218  0.0033096  -71.859  < 2e-16 ***
temp                  1.8875775  0.0057478  328.398  < 2e-16 ***
humidity             -0.9238978  0.0042351 -218.152  < 2e-16 ***
windspeed             0.2412924  0.0057204   42.181  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2015403  on 12164  degrees of freedom
Residual deviance: 1154021  on 12147  degrees of freedom
AIC: 1231675
```

Model with interaction

```
Call:
glm(formula = bikes_per_hour ~ . + weekday * weathersit, family = poisson(link = "log"),
    data = data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-25.402   -8.629   -2.998    3.964   38.148

Coefficients:
                                    Estimate Std. Error  z value Pr(>|z|)
(Intercept)                        4.6793642  0.0062445  749.360  < 2e-16 ***
season2                            0.1536723  0.0019295   79.643  < 2e-16 ***
season1                           -0.0520807  0.0031813  -16.371  < 2e-16 ***
season4                            0.4097403  0.0023916  171.328  < 2e-16 ***
year                              -0.4244643  0.0013713 -309.540  < 2e-16 ***
hour                               0.0447554  0.0001099  407.188  < 2e-16 ***
holidayholiday                    -0.1680025  0.0045187  -37.180  < 2e-16 ***
weekdayFri                        -0.0425030  0.0028717  -14.800  < 2e-16 ***
weekdayWed                        -0.0551117  0.0029280  -18.822  < 2e-16 ***
weekdayMon                        -0.0633498  0.0030475  -20.787  < 2e-16 ***
weekdaySun                        -0.0587937  0.0028862  -20.371  < 2e-16 ***
weekdayTue                        -0.0322087  0.0029260  -11.008  < 2e-16 ***
weekdayThu                        -0.0258216  0.0028580   -9.035  < 2e-16 ***
weathersitMist                    -0.0391211  0.0041206   -9.494  < 2e-16 ***
weathersitSnow r rain             -0.2292993  0.0087818  -26.111  < 2e-16 ***
temp                               1.8843038  0.0057688  326.636  < 2e-16 ***
humidity                          -0.9202672  0.0042485 -216.608  < 2e-16 ***
windspeed                          0.2437791  0.0057342   42.513  < 2e-16 ***
weekdayFri:weathersitMist          0.2120078  0.0056073   37.809  < 2e-16 ***
weekdayWed:weathersitMist          0.1540993  0.0057934   26.599  < 2e-16 ***
weekdayMon:weathersitMist          0.1490963  0.0056453   26.411  < 2e-16 ***
weekdaySun:weathersitMist          0.0525193  0.0062195    8.444  < 2e-16 ***
weekdayTue:weathersitMist          0.0793586  0.0057365   13.834  < 2e-16 ***
weekdayThu:weathersitMist          0.1130791  0.0058336   19.384  < 2e-16 ***
weekdayFri:weathersitSnow r rain  -0.0301778  0.0123956   -2.435   0.0149 *
weekdayWed:weathersitSnow r rain   0.0670277  0.0111180    6.029 1.65e-09 ***
weekdayMon:weathersitSnow r rain   0.1021217  0.0121989    8.371  < 2e-16 ***
weekdaySun:weathersitSnow r rain  -0.1365802  0.0122511  -11.148  < 2e-16 ***
weekdayTue:weathersitSnow r rain  -0.0221394  0.0115301   -1.920   0.0548 .
weekdayThu:weathersitSnow r rain  -0.0746899  0.0125183   -5.966 2.42e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2015403  on 12164  degrees of freedom
Residual deviance: 1151646  on 12135  degrees of freedom
AIC: 1229324
```

We compare the performance for each based on the penalized log likelihood (AIC) and the mean squared error (MSE). The AIC takes into consideration the number of parameters, where a higher number of parameters is penalized. This is higher for the more parsimonious model, the one without the extra interaction coefficients. The MSE on the test set, however, is lower (better) with the interaction.

Because the goal of this project is to make predictions for the number of bike rentals, we recommend using the interaction because this result is based on the test data set which will be representative of new bike rentals whereas the AIC is based on the training data. In addition, although adding the extra coefficients will make the result more difficult to interpret, inference is not the priority of the model.

Without interaction - AIC: 1,231,675

MSE: 20,021.94

With interaction - AIC: 1,229,324

MSE: 20,010.21

Task 10 – Evaluate the cluster variable in the GLM

We ran the code that used the cluster and excluded the variables used in clustering.

```
Call:
glm(formula = bikes_per_hour ~ ., family = poisson(link = "log"),
    data = data.clustered.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-28.747   -8.933   -3.014    4.264   37.821

Coefficients:
                       Estimate Std. Error  z value Pr(>|z|)
(Intercept)           4.7561258  0.0035947 1323.092  < 2e-16 ***
season2               0.3746395  0.0024303  154.156  < 2e-16 ***
season3               0.3838539  0.0025660  149.594  < 2e-16 ***
season4               0.5368226  0.0022702  236.470  < 2e-16 ***
year2011             -0.4512159  0.0013630 -331.048  < 2e-16 ***
hour                  0.0486628  0.0001057  460.470  < 2e-16 ***
holidayholiday       -0.1314809  0.0045104  -29.150  < 2e-16 ***
weekdaySun           -0.0850363  0.0024949  -34.084  < 2e-16 ***
weekdayMon           -0.0584234  0.0025290  -23.102  < 2e-16 ***
weekdayTue           -0.0131555  0.0024482   -5.374 7.72e-08 ***
weekdayWed           -0.0191456  0.0024517   -7.809 5.76e-15 ***
weekdayThu           -0.0035399  0.0024364   -1.453    0.146
weekdayFri            0.0191203  0.0024119    7.928 2.24e-15 ***
weathersitMist        0.0195609  0.0015987   12.236  < 2e-16 ***
weathersitSnow r rain -0.3739913  0.0032800 -114.022  < 2e-16 ***
windspeed             0.4825576  0.0056373   85.600  < 2e-16 ***
cluster2             -0.5671484  0.0018628 -304.464  < 2e-16 ***
cluster3             -0.5929446  0.0020768 -285.510  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2015403  on 12164  degrees of freedom
Residual deviance: 1214302  on 12147  degrees of freedom
AIC: 1291956
```

We are purely looking at the predictive power when we decide whether to use the clustered variable or to use the original variables instead. The predictive performance is worse and so we recommend not using the cluster. The MSE is higher (worse) and the AIC is lower (worse).

Results with Cluster

Test MSE: 20,877

AIC: 1,291,956

Results without Cluster from prior tasks

Test MSE: 20,021

AIC: 1,231,675

Task 11 – Select the final model to present to the client

Note to grader: I expected to be able to find the test and train MSE for the single decision tree from task 6 but was unable to do so.

The boosted decision tree is clearly the best choice because it has the lowest error. The difference is one order of magnitude lower than the second-best model, the GLM with the interaction term. While the boosted tree is more challenging to interpret, this is not an issue for the task at hand because we do not prioritize inference. The gain in accuracy and performance is so significant that ABC does not even need to consider other types of models further.

Model	Task	Test MSE
Decision Tree	6	NA
Boosted decision tree	7	2,356
GLM with all original variables	8	20,021
GLM with all original variables using Gamma	8	127,311
GLM with an interaction term	9	20,010
GLm with clustered variables	10	20,877

The hyperparameters for the GBM are below so that the results can be reproduced.

- Shrinkage of 0.1
- N.trees of 1000
- Interaction Depth of 4
- Distribution of "gaussian"

To help us better appreciate how much value the GBM is adding, consider what would happen if we did not use the data at all and instead predicted the overall average. A metric such as Mean Squared Error does not have an intuitive explanation because the value changes depending on the scale of the target variable.

If we used an intercept only model, we would be predicting that the mean bikes per hour of $e^{5.24}$ = 188.7, which is the mean bikes per hour of the training data. This results in a test MSE of 33,546 bikes per hour. This is much higher than the GBM of only 2,356. Notice also that when we add in the data and use the GLM with all of the variables, the MSE decreases by about 13,000.

We can also look at the summary output for this model. Notice that the median residual is negative, meaning that the predictions are too high on average. This makes sense because the distribution of bikes per hour is right skewed.

```
Call:
glm(formula = bikes_per_hour ~ 1, family = poisson(link = "log"),
    data = data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-19.127  -13.192   -3.589    6.224   40.405

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.2425874  0.0006592    7953   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2015403  on 12164  degrees of freedom
Residual deviance: 2015403  on 12164  degrees of freedom
AIC: 2093023

Number of Fisher Scoring iterations: 5

[1] "Mean Squared Error"
[1] "Intercept Only, Train"
[1] 32626.55
[1] "Intercept Only, Test"
[1] 33546.25
```

Task 12 – Write an executive summary for the client

To: ABC Bike Rentals

From: Actuarial Analytics Team

We present you with a predictive analytics-based solution to your problem of managing bike rentals. Using data and machine-learning-based results, you can efficiently predict what the demand for bike rentals will be and then plan accordingly.

Data

Our results are reliable because they are based on 17,376 observations of bike rentals in CA from 2013. The dataset shows the season, year, hour, holiday, weekday, weather situation, temperature, humidity, and windspeed along with the number of bikes rented per hour. We conducted cleaning and exploration and made only minimal changes to the original data. You can reproduce all of these steps by following our easy-to-read documentation in the body of this report.

We understood that your main priority was to have a model that will be reliable for making predictions in the real world. To that end, we used a blind-holdout test where 30% of the data was used as a validation which proves that our results will perform similarly when you deploy the model.

After looking at your data, we found that

- Poor weather makes riding bikes hazardous.
- Most days are calm and clear with little wind.
- People are half as likely to rent a bike during rain or snow than during clear weather.
- Riders prioritize comfort as well as safety because rentals were lower during cold or hot temperatures.

- There is a relationship between the day of the week and the weather situation because during workdays, commuters have no choice but to ride a bike to their work whereas on the weekends, when their rental is for leisure, rentals are far lower during poor weather.
- During early morning, before 7 am, the number of rentals is quite low.
- During the day, when the temp is below 30.6 C, the demand is moderate and then increases warmer temperatures.
- The rental program is growing because demand was higher in 2012 as compared to 2011.

Our recommended model

There is tremendous richness to the data which you provided and this this allows us to build a highly accurate machine learning model. After looking at dozens of candidates, we found that the highest-value model is a boosted-tree ensemble. This had better performance than six other models. This type of model, known also as a GBM, sequentially learns from the data by making iterative improvements. Out of all the models tested, it is the most complicated, however, we do provide interpretations using the best methods possible.

We measured the value added by comparing our model against a naïve model. The overall average number of bikes rented per hour is 189. If we made this prediction to every single bike that was rented, the average difference between the actual number of bikes rented and this estimate would be 143. By using our model, this error is significantly reduced to 39.

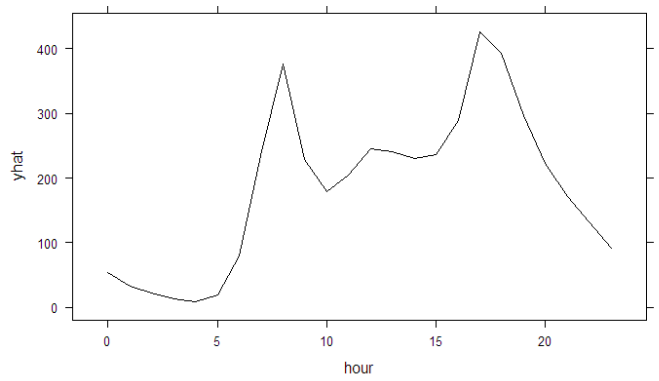
We are happy to discuss with you the other types of models which we tested. The first was a single decision tree. This has the advantage of breaking up the problem into a serious of simple yes/no questions but has the disadvantage of not making dependable predictions. Then we tested two version of boosted trees. These are based on many decision trees which are connected to make a single model. On other hand, moving to linear-based models, we tested four different GLMs. These are a way of relating multiple input variables to the number of bikers per hour using a simple spreadsheet formula.

You can be confident that these results are the best that can be achieved by comparing the improvement over not using a model at all.

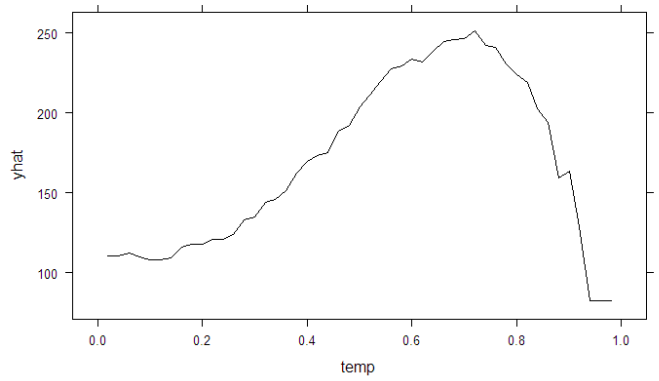
The model is not a black box because we can see that the variable importance agrees with our intuition. The most important factor is the time because no person rents bikes during the night. The second is the temperature followed by the weekday. The ranking below is based on the GBM's variable importance. This looks across each of the decision trees used and sums up the influence from each tree and then ranks them using a scale from 0 – 100. A higher value of "Predictive Influence" below means that the variable was more influential than other variables used in the model.

Variable	Predictive Influence
Hour	60
Temp	14
Weekday	10
Year	8
Season	3
Humidity	2
Weather Situation	1
Windspeed	0.6
Holiday	0.5

During mornings when people are commuting to work and during the evenings at around 5 pm, rentals are at a peak. The graph below shows the partial dependence which is the effect that hour is having on the model when averaging out the effects of the other variables.



When the temp is low and the weather is cold, people are less inclined to ride bikes. The demand gradually increases with the temperature only up to a point because people dislike weather which is too hot as well as weather which is too cold.



In conclusion, ABC can increase customer satisfaction and drive revenue growth by making more bikes available when people need them. During off hours at night, we recommend performing servicing of bikes and moving bikes from populated areas to the rental locations which will be used the following

morning. Safety should be a priority whenever dealing with pedestrian and city data because of the potential lawsuits. This analysis did not consider riding accidents or road hazards and it may be worthwhile to consider these factors by looking at hospital ER visits from accidents or road blockages from construction. Rider satisfaction can be enhanced by servicing the bikes well to ensure that the air in the tires is kept at optimal levels and that the chains are lubricated, and the demand rentals could be tied to individual bikes so that maintenance teams could track which bikes were due for servicing.