This model solution is provided so that candidates may better prepare for future sittings of Exam PA. It includes both a sample solution, in plain text, and commentary from those grading the exam, in italics. In many cases there is a range of fully satisfactory approaches. This solution presents one such approach, with commentary on some alternatives, but there are valid alternatives not discussed here.

Exam PA December 7, 2020 Project Report Template

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.

Also be sure all the documents you are working on have December 7 attached.

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

Task 1 – Select factor variables (5 points)

There are several reasons to convert a numeric variable to a factor variable, many demonstrated below. Other reasonable choices could have been made here.

Most candidates did well on this task, but some only justified the variables changed to a factor variable without justifying retaining the other variables as numeric. Some also did not include the summary report as requested.

The use of bold for identifying variable names when they are common words is not required but can help clarify the writing.

The following variables are each converted to a factor variable:

- **weathersit**: the three values have no natural order and the difference between values has no numeric meaning
- **season**: with all four levels retained and renamed based on their respective seasons. While seasons have some sense of order, there is not a compelling reason for the bike usage to align with this order. Converting to a factor variable gives more degrees of freedom to better fit the target variable.
- **weekday**: with all seven levels retained and renamed based on their respective days of the week. While weekdays have some sense of order, the cumulative difference in number of days is not expected to be predictive and the extremes, Sunday (0) and Saturday (6), may indeed have similar responses.
- **holiday**: as “Not Holiday” or “Holiday” as applicable, to better represent its nature as a binary variable. Values between 0 and 1 would not have meaning.

The remaining variables (**year**, **hour**, **temp**, **humidity**, **windspeed**, **bikes_per_hour**) are retained as numeric variables as they take on many values and the numerical difference in values is meaningful. The **hour** variable is retained as a numeric variable despite being a cyclical time element like the season and weekday variables. Converting from 1 variable to 23 variables may induce high variance (overfitting) and worsen the predictive power of fitted models. Also, while **year** could be treated as a factor variable given it only has two values, extrapolating to a future year is a sensible interpretation.

The factor variables were releveled to make the most common level the baseline level. The summary after these changes is below:

```
   season          year             hour               holiday          weekday
Summer:4496   Min.   :0.0000   Min.   : 0.00   Not Holiday:16876   Saturday :2511
Winter:4239   1st Qu.:0.0000   1st Qu.: 6.00   Holiday    :  500   Sunday   :2502
Spring:4409   Median :1.0000   Median :12.00                       Monday   :2478
Fall  :4232   Mean   :0.5025   Mean   :11.55                       Tuesday  :2453
              3rd Qu.:1.0000   3rd Qu.:18.00                       Wednesday:2474
              Max.   :1.0000   Max.   :23.00                       Thursday :2471
                                                                   Friday   :2487
              weathersit         temp          humidity        windspeed      bikes_per_hour
Clear/Partly Cloudy:11413   Min.   :0.020   Min.   :0.0000   Min.   :0.0000   Min.   :  1.0
Mist               : 4544   1st Qu.:0.340   1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.: 40.0
Rain/Snow          : 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 (3 points)

Only some candidates made the key observation that the new variable does not add information and a rank-deficient set of predictors ensues, a disadvantage in itself. Stronger candidates considered the advantage and disadvantage of the result once this is corrected for.

The binary **workday** variable would not add information to the data. If added, the set of predictor variables would be rank deficient, producing errors when fitting certain types of models, so including **workday** also involves removing either the **weekday** or **holiday** variable.

One advantage of including **workday** and creating a set of variables that simply classifies days as workdays, holidays, and weekends is that it would be easier to understand, use, and communicate. Bike usage may vary directly with the workday status, but this is hard to see when the definition of workday always involves two variables.

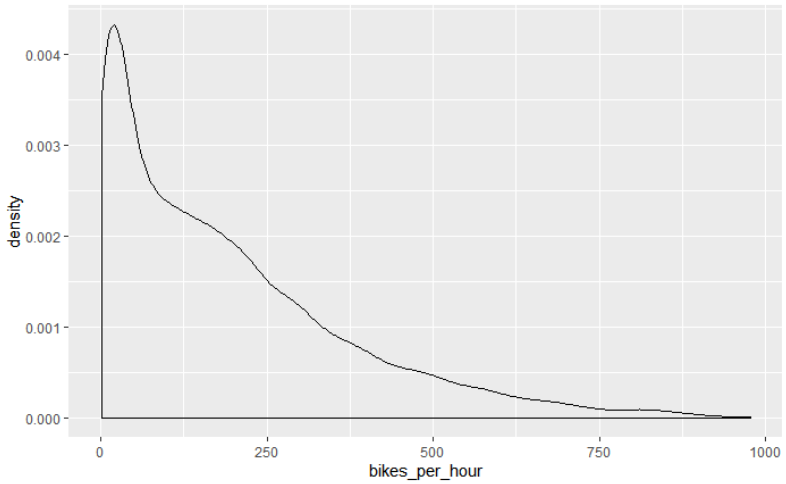
One disadvantage of including **workday** is that, if **weekday** were removed to compensate, information would be lost, going from twelve combined levels to just three. This may reduce the predictive power of fitted models.

Task 3 – Write an overview of the data for your actuarial manager (12 points)

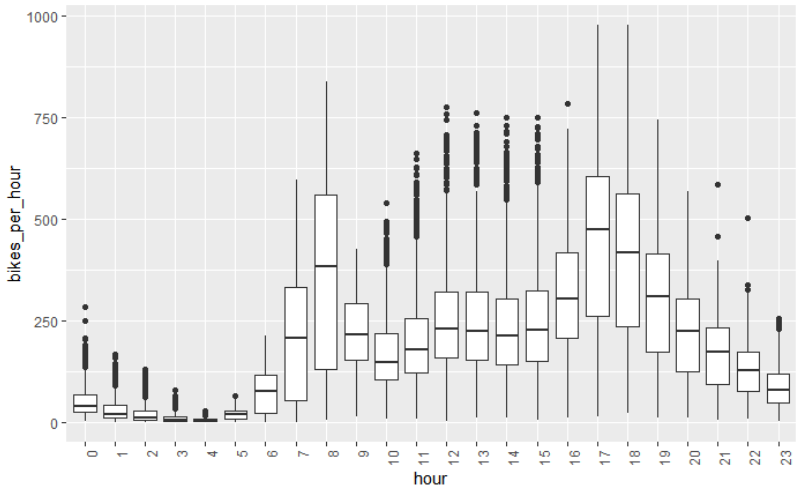
Presentation matters here and helps with clear communication. The soft limit of one page is intended to give candidates a scale for work required and encourage thought on what information is more important to include. Stronger candidates balanced a brief summary of the entire dataset with more in-depth coverage of key variables and relationships, including modeling implications of the data. Some candidates had poorly labeled graphs that did not indicate which variables were involved.

To predict bike usage, a dataset with 17,376 records and ten variables, including five time variables, four weather variables, and the target variable **bikes_per_hour**, is provided. The data contains no missing or seemingly erroneous values. While the original location of the data is not disclosed, it includes hourly observations for the entirety of 2011 and 2012. Each hour of each day of the week is distinct and has between 92 and 105 weeks of observations, with slightly fewer observations on weekdays between 3 and 5 a.m., presumably due to regular maintenance. Rounding out the presence of time variables,

calendar seasons are noted but months are not. Holidays that occur between Monday and Friday are also noted.



The target variable **bikes_per_hour** ranges from 1 to 977 but has significant right skew, as seen above. The average usage across all observations is 189 bikes per hour, but the median usage is 142.



Of the time variables, the most illustrative is **hour**. A box plot of **bikes_per_hour** against **hour** is shown to the right. The peaks occur at hour 8 and again around hours 17 and 18. These suggest that commuters to work are using bikes from ABC. In between these peaks, bike usage has considerable variance but median usage is fairly consistent. After the evening commute and into nighttime hours, bike usage steadily declines, with the lowest usage occurring at hour 4.

Among the weather variables, the factor variable **weathersit** illustrates another dynamic of bike usage:

weathersit	Mean	Median	n
Clear/Partly Cloudy	205	159	11413
Mist	175	133	4544
Rain/Snow	112	63	1419

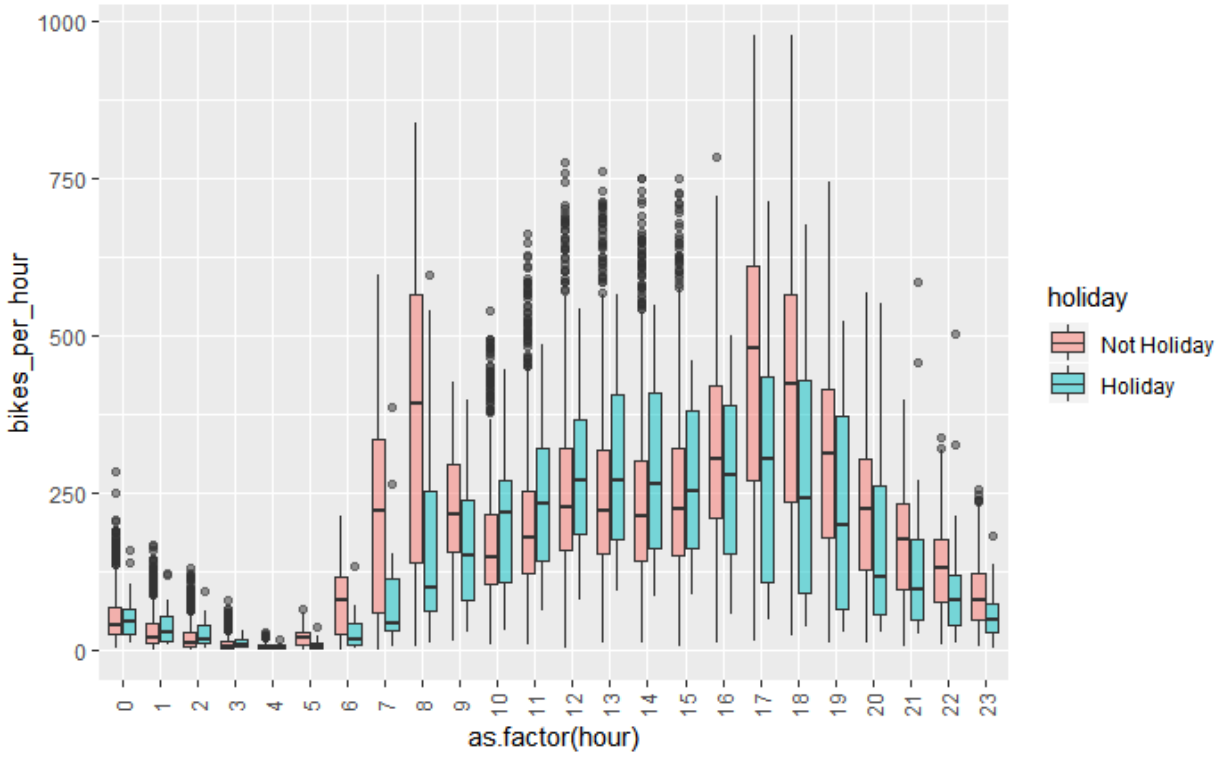
Adverse weather conditions, particularly rain/snow, results in significantly lower mean and median values for bike usage. **temp** is also observed to have a strong impact on bike usage, with generally increasing usage with warmer temperatures, though the highest bike usage figures, above 750, rarely occur once temperatures increase above 30 degrees Celsius. **temp**, **hour**, and **season** have complex interdependencies which should be considered when modeling.

Task 4 – Select an interaction to consider for your model (6 points)

Explaining, choosing, and justifying interactions requires precise language involving three variables—the two variables said to be interacting as well as the target variable. Some candidates confused interactions with correlations, which only involve two variables. Some candidates explained interactions well but struggled to justify their chosen interaction due to a poor choice of variables or not taking into account the primary impact of each variable on the target prior to the interaction effect. Stronger candidates included reasoning for why the interaction would exist.

An interaction is when the dependency of the target variable on a predictor variable is itself dependent on a third variable. In other words, when the target variable relates to two predictor variables differently than expected based on combining how it relates to each predictor variable independently, an interaction effect is present.

The variables **hour** and **holiday** present a compelling interaction, as seen in the following box plot:



When not a holiday, typical bike usage peaks at hours aligned with commuting workers. When a holiday, the morning peak does not occur and bike usage gradually increases until the afternoon. This makes sense because workers are not commuting on holidays.

In general, bike usage is somewhat lower on holidays than non-holidays:

Holiday (All Hours)	Mean	Median	n
Not Holiday	190	144	16876
Holiday	157	97	500

However, this pattern is more pronounced when filtering to hours 5-9:

Holiday (Hours 5-9)	Mean	Median	n
Not Holiday	180	118	3518
Holiday	93	44	105

And flipped in hours 10-14:

Holiday (Hours 10-14)	Mean	Median	n
Not Holiday	225	197	3535
Holiday	265	251	105

The above comparison further demonstrates how the dependence of **bikes_per_hour** on **holiday** varies greatly depending on **hour**.

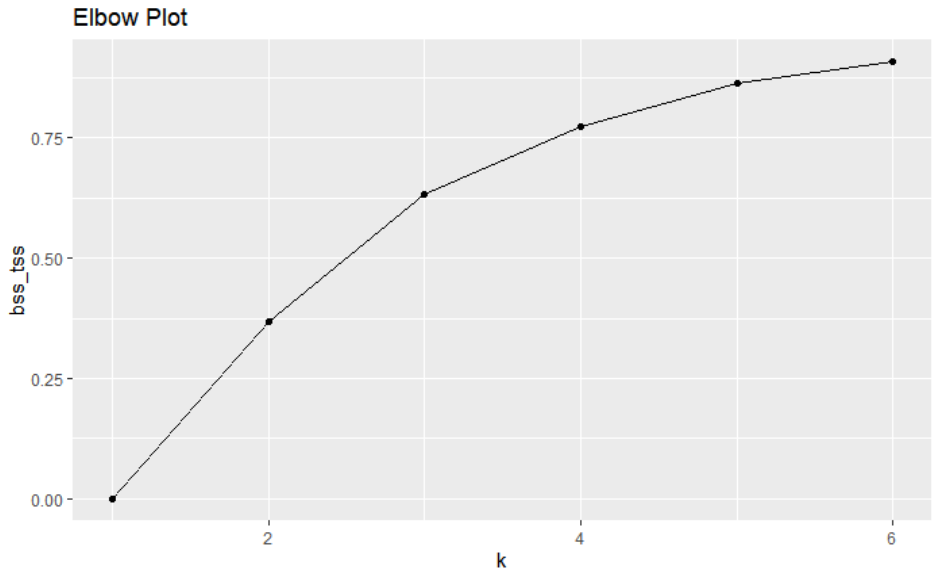
Task 5 – Perform a k-means cluster analysis (9 points)

Many candidates were able to give helpful information regarding k-means cluster analysis and elbow plots but only some were able to give a full explanation. Most candidates recognized that the data was not a compelling candidate for clustering, particularly given the importance of ***temp****.*

With k-means cluster analysis, an unsupervised learning technique, the goal is to assign records into one of k groups or clusters such that members of each group are overall more similar to one another than they are to members of other groups. The number of groups, k, is specified at the beginning and the group members are determined through an iterative process. Initially, k random centers are chosen and the group assignment for each record is determined by which of these centers is closest. New centers for each group are calculated based on its members, and then group membership is redetermined based on these new centers. This process continues until the centers and group membership are stable or stopped by an iteration limit.

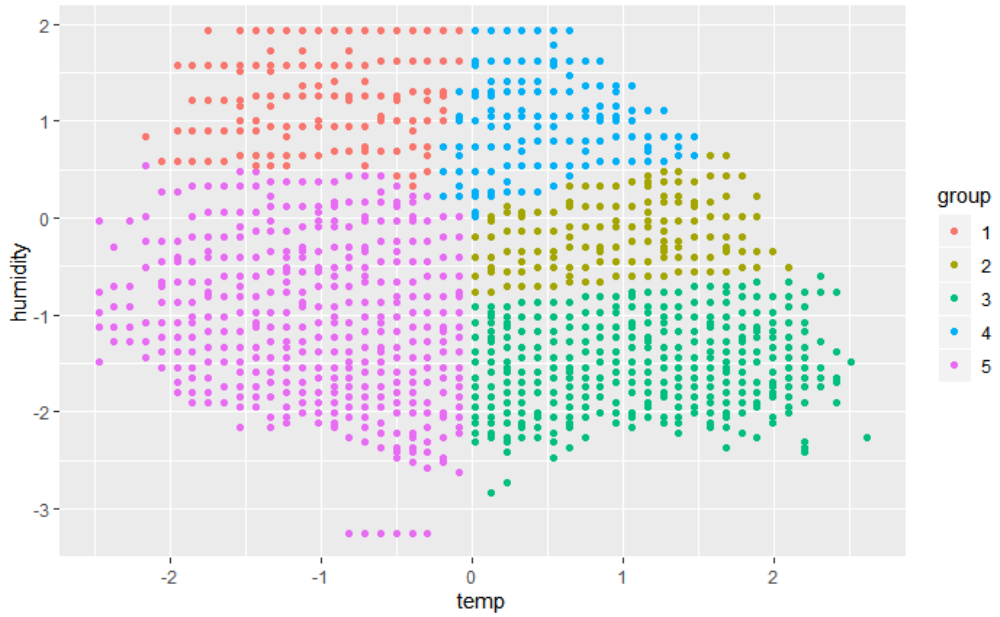
In an elbow plot, the proportion of variance explained by the variance between the k centers is calculated and plotted for successive values of k. Increases in k generally lead to increases in the proportion of variance explained, but the size of each increase typically decreases with each additional cluster. Where the incremental proportion of variance explained suddenly decreases with the addition of another cluster, the plot shows an “elbow” for the sudden change of direction, and the number of clusters just to the left of this, before the less helpful cluster is added, is considered a good, parsimonious choice for k.

When k-means cluster analysis is applied on **temp** and **humidity**, the following elbow plot appears:



The elbow plot does not provide depict an obvious choice for the number of clusters. Visually, three and five clusters seem like marginally better choices. Since the portion of variance explained is still significantly higher at five compared to three, a new cluster variable using five clusters is created.

The following shows colored scatterplots of the clusters themselves:



It is not recommended that the new cluster variable be used to replace the **temp** and **humidity** variables. To be most useful, the clusters would represent similar groups where observations for each group are close to each other but separate from other groups, which is not the case here. Also, the groups fall near simple divisions of **temp** and **humidity** which could be captured fairly easily by successively grouping the original variables. Finally, **temp** shows a gradual relationship with **bikes_per_hour** that would be lost with the discrete clusters.

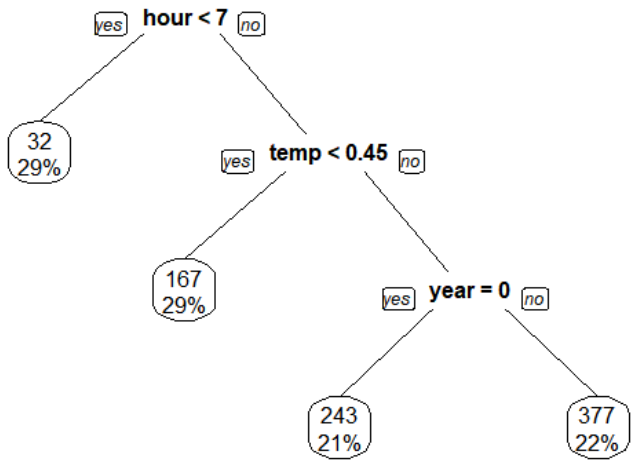
Task 6 – Construct a decision tree (7 points)

*Most candidates did well on this task. When describing splits beyond the first, some candidates failed to note how they do not apply everywhere but only for that branch of the tree. Some candidates did not adequately interpret normalized variables like **temp**. Stronger candidates directly compared to the unseen nodes and gave compelling reasons on why the split makes sense, rather than just reading the tree.*

Having split the data into train and test data using the provided code, an unpruned decision tree was fit to the train data, producing the following complexity parameter (CP) table (only first six rows shown):

```
           CP nsplit rel error    xerror        xstd
1  0.304604914      0 1.0000000 1.0001128 0.016610474
2  0.110481143      1 0.6953951 0.6954729 0.013217062
3  0.057776682      2 0.5849139 0.5876316 0.011393520
4  0.042345685      3 0.5271373 0.5313812 0.009787735
5  0.032256585      4 0.4847916 0.4893667 0.009046308
6  0.021641928      5 0.4525350 0.4567516 0.008408229
```

To prune the decision tree to four terminal nodes means three splits, corresponding to row 4 above. The pruned tree was fit using CP of 0.05, resulting in the following tree:



The first split differentiates hours 0-6 on the left from hours 7-23 on the right. The early morning hours on the left have far lower bike usage, 32 per hour, than the remaining hours, at 253 per hour (not shown). This makes sense as most people are asleep during these overnight hours rather than renting bikes.

The second split differentiates temperatures under 12.6 degrees Celsius (or about 55 degrees Fahrenheit, each converted from the normalized value of 0.45) and those at or above this mark when hour is between 7 and 23. The predicted usage is 167 in cold weather and 312 in warm weather during these hours. This makes sense that, when most people are awake, riding a bike in warmer weather is more comfortable and generally preferable to doing so in colder weather.

The third split differentiates usage in 2011 (**year** = 0) and 2012 for hours 7-23 and temperatures of at least 12.6 degrees Celsius, with predicted values of 243 and 377 respectively. It appears that bike rental from ABC during these favorable times and conditions became more popular from the first year provided to the second, perhaps due to increased awareness of ABC bikes. The tree does not deny that 2012 was more popular than 2011 in other situations, but this is the situation in which the growth in usage is most distinct.

Task 7 – Construct a boosted decision tree (12 points)

Few candidates showed mastery of all relevant details of boosted decision trees, though most candidates received partial credit. Many candidates could not clearly articulate how the individual models relate to one another, and some candidates confused boosting and bagging. Some candidates did not realize how a smaller shrinkage parameter can produce poorer predictions for the same number of trees. When identifying important variables, some candidates did not see that not every variable is labeled and chose the first and third most important items.

Boosting and this business problem

Boosting builds up an ensemble model, taking the aggregate prediction of many individual models or learners, by training models in series, each successive model building on the deficiencies of the prior model. Unlike bagging, another ensemble method, the individual models are not independent. The technique gains accuracy not by the particular predictions of any of its individual models, often called weak learners, but by its iterative process for improving the aggregate performance of the models in total.

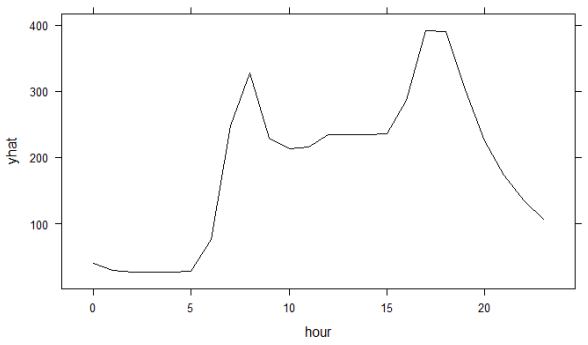
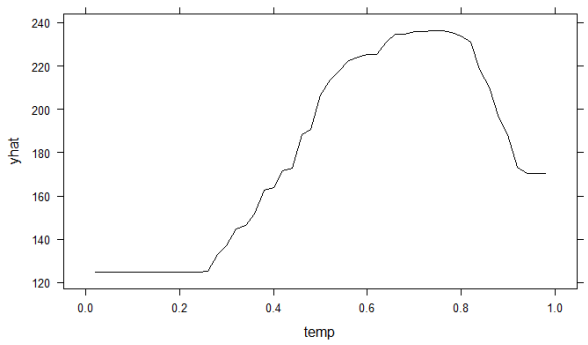
The first model is trained on unweighted data, and then the second model is trained on the residuals produced by the first model. The third model is trained based on the residuals of the first two models taken together, and so on. The boosting process is typically stopped after a set number of iterations, and the sum of all model output is used.

To prevent overfitting within this iterative process, a shrinkage parameter is applied to individual models so that the aggregate performance of the models approaches the training data in a controlled manner and avoids being overly sensitive to the structure of any one model.

Boosting is appropriate for this business problem because its predictions, by directly addressing the errors of prior model fittings, are typically more accurate than those of other predictive modeling techniques. Being a more complex ensemble method, it is difficult to gather insight into how the model is making these accurate predictions, but this seems relatively unimportant to ABC.

Partial dependence plots

For the boosted decision tree with 1000 decision trees of depth 4 and shrinkage parameter of 0.01, the two most important variables are **hour** (62% relative importance) and **temp** (13%). Their partial dependence plots are shown below (see next page):

hour	temp
	

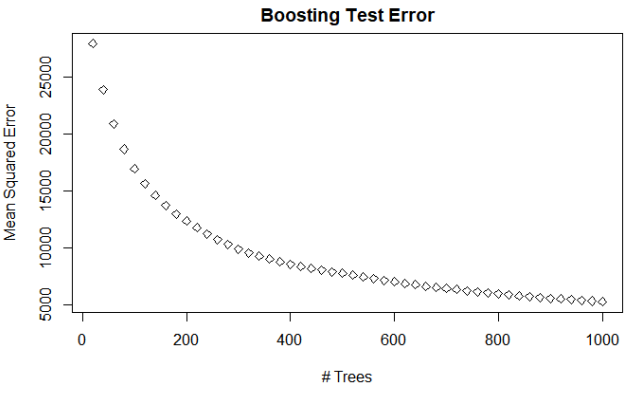
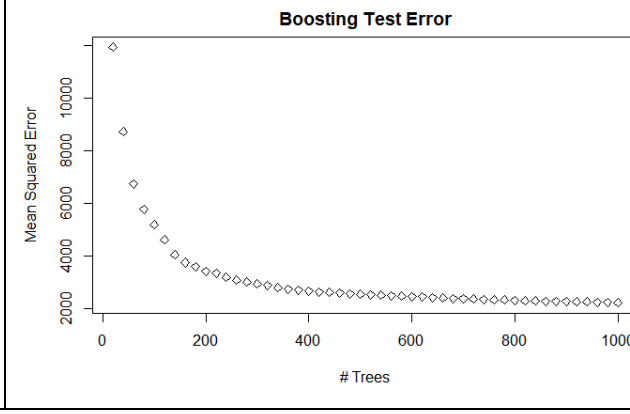
Partial dependence plots use the expected value of the prediction at the variable value shown when paired with all values of the other variables as found in the training data. The yhat values can be compared to the overall average mean bike usage of 189 per hour in the train data.

For **hour**, the two-peaked pattern corresponding to worker commute times is seen, with lowest bike usage during the morning hours. For **temp**, bike usage generally increases with temperature until reaching about 0.8 (around 30 degrees Celsius), where a steep decline begins.

The flat sections on the left side of each depict regions where the weak learners are not making distinctions even after the boosting process, likely due to sparse data in those regions. Also, each plot assumes that the other variable and all other predictor variables not shown are independent from the featured variable, creating situations that may not exist in the data such as the hottest temperatures being equally likely at any season or hour.

Shrinkage parameters

Two boosted trees were fit up to 1000 iterations each with shrinkage parameters of 0.01 and 0.1. The mean squared errors on predictions on the test data using the first n trees, n stepping by 20 from 20 to 1000, are shown below.

Shrinkage = 0.01	Shrinkage = 0.1
	

While the learning curve initially looks similar, the y-axis shows that the higher shrinkage parameter, representing greater weight for each weak learner, produces a substantially better prediction as measured by mean squared error. After 1000 iterations, the boosting at 0.01 shrinkage is still making substantial progress in reducing the error and may eventually be as accurate as that at 0.1 shrinkage, but the latter got there much faster without overfitting, which would have been indicated by a rising curve on this test data. Also, the higher shrinkage led to a larger range of predictions on test data, being -19 to 705 at 0.01 shrinkage and -79 to 868 at 0.1 shrinkage. Because the higher shrinkage parameter gives more weight to each model fitting the successive residuals, the predictions tend to be more spread out given the same number of trees.

Negative predictions

In a decision tree for regression, the prediction is based on the values present in each terminal node and uses an average result, so predictions are interpolated from the available data and fall within its range. In a boosted tree, the prediction starts out like the decision tree for the first model but then each successive model is effectively an adjustment made to the aggregate performance of the models in total. Because these adjustments are average adjustments made to varying large blocks of records with each successive model, a particular record's adjustments may accumulate to produce a prediction beyond the range of the data.

Task 8 – Compare distribution choices for a generalized linear model (10 points)

*Most candidates did well with the more familiar log link function associated with the Poisson distribution but struggled to recognize or explain the impact of the canonical inverse link function associated with the gamma distribution. Some candidates did not incorporate the normalization of the **temp** variable nor recognize how extreme their results were as a result.*

Choice of distribution and link function

The Poisson distribution, with its canonical log link function, is a reasonable choice for this data and business problem. The **bikes_per_hour** target variable only has non-negative integer values, so the Poisson loss function can be applied when comparing the predicted mean to the target variable during fitting. The log link function, besides being canonical, allows the predicted mean to vary multiplicatively rather than linearly with the coefficients for each predictor variable, more naturally fitting the right-skewed distribution of the target variable. Having a percentage increase or decrease in bike rentals due to a change in a particular predictor variable makes intuitive sense and prevents negative predictions when conditions for low bike usage are present.

The gamma distribution, with its canonical inverse link function, is also a reasonable choice but less well adapted to this data and business problem. There is not material harm in applying the gamma distribution function to only integer values when the values span a large range. The data matches its support of strictly non-negative values, though it is conceivable that future data could include zero bike rentals. The inverse link function, besides being canonical, allows the predicted mean to vary hyperbolically rather than linearly with the coefficients for each predictor variable. Unlike the log link function, the inverse link function can result in negative predictions, typically massive and unusable when they occur.

Poisson fitting

The summary for the Poisson fitting without interaction or cluster predictor variables is below:

```
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)         3.8053447  0.0059902  635.263  < 2e-16 ***
seasonWinter       -0.0496080  0.0031669  -15.665  < 2e-16 ***
seasonSpring        0.1511775  0.0019232   78.607  < 2e-16 ***
seasonFall          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 ***
weekdaySunday      -0.0492539  0.0024966  -19.729  < 2e-16 ***
weekdayMonday      -0.0195864  0.0025306   -7.740 9.95e-15 ***
weekdayTuesday     -0.0146127  0.0024500   -5.964 2.46e-09 ***
weekdayWednesday   -0.0125564  0.0024535   -5.118 3.09e-07 ***
weekdayThursday    -0.0012140  0.0024362   -0.498    0.618
weekdayFriday       0.0101044  0.0024103    4.192 2.76e-05 ***
weathersitMist      0.0725231  0.0016312   44.461  < 2e-16 ***
weathersitRain/Snow -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
```

For the effect of **season** on expected bike usage, all other variables being equal, Summer is included in the intercept while Winter has a coefficient of -0.0496. Thus, the linear predictor for Summer prior to applying the inverse of the log link function will be 0.0496 higher than that for Winter. Applying the inverse of the log link function, this effect becomes $e^{0.0496} = 1.051$, meaning that bike usage is 5% higher in Summer than it is in Winter, assuming all else being equal.

For the effect of temperature on expected bike usage, all other variables being equal, just the difference in temperature is required. The increase of 10 degrees Celsius corresponds to 10/(39 - (-9)) = 0.2083 for the scaled **temp** variable. The linear predictor is then 0.2083 * 1.888 = 0.3933, and the effect on the predicted mean $e^{0.3933} = 1.482$, so bike usage will increase by 48% with a 10 degree increase in degrees Celsius, assuming all else being equal. This is a far stronger impact than Summer compared to Winter.

Gamma fitting

The summary for the gamma fitting without interaction or cluster predictor variables is below:

```
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)          1.266e-02  3.298e-04  38.395  < 2e-16 ***
seasonWinter         1.154e-03  2.002e-04   5.765 8.38e-09 ***
seasonSpring        -6.224e-04  8.839e-05  -7.042 2.00e-12 ***
seasonFall          -1.826e-03  1.205e-04 -15.152  < 2e-16 ***
year                -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 **
weekdaySunday        4.249e-05  1.304e-04   0.326  0.74446
weekdayMonday       -1.080e-04  1.319e-04  -0.818  0.41320
weekdayTuesday      -9.965e-05  1.216e-04  -0.819  0.41272
weekdayWednesday     1.518e-07  1.157e-04   0.001  0.99895
weekdayThursday     -8.810e-05  1.183e-04  -0.745  0.45654
weekdayFriday       -9.721e-05  1.150e-04  -0.845  0.39793
weathersitMist      -3.332e-04  8.495e-05  -3.922 8.83e-05 ***
weathersitRain/Snow  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
```

For the effect of **season** on expected bike usage, all other variables being equal, Summer is included in the intercept while Winter has a coefficient of 0.001154. Thus, the linear predictor for Summer prior to applying the inverse link function will be -0.001154 lower than that for Winter. The effect on bike usage is most easily described by example. In the table below, an assumed predicted mean for Winter is converted to its linear predictor using the inverse link function, the effect of Summer applied by adding -0.001154, and the predicted mean for Summer is determined by inverting yet again.

Winter predicted mean	100	200	500
Winter linear predictor	0.010000	0.005000	0.002000
Summer linear predictor	0.008846	0.003846	0.000846
Summer predicted mean	113	260	1182

The relative multiplicative effect of Summer vs. Winter, which was constant with the log link function, increases significantly as the Winter predicted mean increases.

For the temperature effect of increasing 10 degrees Celsius, the same 0.2083 change in **temp** variable is multiplied by the -0.007544 to result in a -0.001571 impact on the linear predictor. The example impacts are calculated similarly, resulting in the following:

Winter predicted mean	100	200	500
Winter linear predictor	0.010000	0.005000	0.002000
Summer linear predictor	0.008429	0.003429	0.000429
Summer predicted mean	119	292	2331

The impact of Summer vs. Winter and +10 degrees Celsius are similar during lower bike usage times but strikingly different during higher usage times.

Task 9 – Evaluate the interaction term (6 points)

Candidates generally did well on this question. A few candidates did not provide a clear recommendation or did not justify their recommendation. The standard answer regarding how decision tree splits produce interactions is appropriate, but the presence of an interaction term can affect decision tree fitting.

An interaction term is generally not needed when evaluating a decision tree because, after the initial split of data, each subsequent split of data affects only a portion of the data. When the splits are based on different predictor variables, an interaction effect is automatically modeled. While it is possible that the greedy decision tree algorithm may miss an interaction effect that it would find were that interaction present as a distinct predictor variable, this is uncommon, and exploratory data analysis does not suggest such an interaction is present.

The summary function without the interaction term is repeated here:

```
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)       3.8053447  0.0059902  635.263  < 2e-16 ***
seasonWinter     -0.0496080  0.0031669  -15.665  < 2e-16 ***
seasonSpring      0.1511775  0.0019232   78.607  < 2e-16 ***
seasonFall        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 ***
weekdaySunday    -0.0492539  0.0024966  -19.729  < 2e-16 ***
weekdayMonday    -0.0195864  0.0025306   -7.740 9.95e-15 ***
weekdayTuesday   -0.0146127  0.0024500   -5.964 2.46e-09 ***
weekdayWednesday -0.0125564  0.0024535   -5.118 3.09e-07 ***
weekdayThursday  -0.0012140  0.0024362   -0.498    0.618
weekdayFriday     0.0101044  0.0024103    4.192 2.76e-05 ***
weathersitMist    0.0725231  0.0016312   44.461  < 2e-16 ***
```

```
weathersitRain/Snow -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
```

The summary function with the interaction term is given here:

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-24.956   -8.663   -2.993    3.960   38.172

Coefficients:
                     Estimate Std. Error  z value Pr(>|z|)
(Intercept)         3.8059339  0.0060023  634.084  < 2e-16 ***
seasonWinter       -0.0496561  0.0031670  -15.679  < 2e-16 ***
seasonSpring        0.1511361  0.0019234   78.577  < 2e-16 ***
seasonFall          0.4109363  0.0023788  172.751  < 2e-16 ***
year                0.4252920  0.0013673  311.037  < 2e-16 ***
hour                0.0446130  0.0001107  403.004  < 2e-16 ***
holidayHoliday     -0.1867196  0.0109544  -17.045  < 2e-16 ***
weekdaySunday      -0.0492583  0.0024966  -19.730  < 2e-16 ***
weekdayMonday      -0.0195989  0.0025306   -7.745 9.58e-15 ***
weekdayTuesday     -0.0146067  0.0024500   -5.962 2.49e-09 ***
weekdayWednesday   -0.0125560  0.0024535   -5.118 3.09e-07 ***
weekdayThursday    -0.0012096  0.0024362   -0.497    0.620
weekdayFriday       0.0101188  0.0024103    4.198 2.69e-05 ***
weathersitMist      0.0724883  0.0016313   44.435  < 2e-16 ***
weathersitRain/Snow -0.2378807  0.0033098  -71.872  < 2e-16 ***
temp                1.8873616  0.0057495  328.266  < 2e-16 ***
humidity           -0.9239775  0.0042354 -218.154  < 2e-16 ***
windspeed           0.2411663  0.0057210   42.154  < 2e-16 ***
hour:holidayHoliday 0.0010875  0.0007054    1.542    0.123
---
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: 1154018  on 12146  degrees of freedom
AIC: 1231674
```

The interaction term, which introduced an adjustment to the linear predictor for **hour** when it is a holiday, should not be included. While there is difference in the distribution of **bikes_per_hour** on non-holidays and holidays, this interaction term is insufficient to capture the various differences in the shapes. Also, adding the interaction term increased the mean squared error (MSE) on the test data from 20,022 to 20,023, providing no improvement in predictive power, the main objective for ABC.

Task 10 – Evaluate the cluster variable in the GLM (4 points)

Candidates generally did well on this task, with a few candidates failing to provide a clear recommendation despite discussing pros and cons of including the cluster variable. Stronger candidates include in their justification the sensibility of comparative predictions.

The summary function with the cluster variable and without its contributor variables is shown below:

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-28.774   -8.894   -3.022    4.211   41.095

Coefficients:
                      Estimate Std. Error  z value Pr(>|z|)
(Intercept)          3.8608068  0.0037467 1030.466  < 2e-16 ***
seasonWinter        -0.3244545  0.0026616 -121.901  < 2e-16 ***
seasonSpring        -0.0100869  0.0017501   -5.764 8.23e-09 ***
seasonFall           0.1923535  0.0020309   94.712  < 2e-16 ***
year                 0.4369920  0.0013656  320.004  < 2e-16 ***
hour                 0.0477261  0.0001068  446.847  < 2e-16 ***
holidayHoliday      -0.1149461  0.0045138  -25.465  < 2e-16 ***
weekdaySunday       -0.0676845  0.0024962  -27.115  < 2e-16 ***
weekdayMonday       -0.0425196  0.0025319  -16.794  < 2e-16 ***
weekdayTuesday      -0.0130309  0.0024491   -5.321 1.03e-07 ***
weekdayWednesday    -0.0201217  0.0024520   -8.206 2.28e-16 ***
weekdayThursday     -0.0078978  0.0024366   -3.241  0.00119 **
weekdayFriday        0.0148746  0.0024115    6.168 6.91e-10 ***
weathersitMist       0.0474616  0.0016098   29.483  < 2e-16 ***
weathersitRain/Snow -0.2964236  0.0033014  -89.786  < 2e-16 ***
windspeed            0.3842668  0.0056683   67.793  < 2e-16 ***
cluster2             0.7581044  0.0030648  247.355  < 2e-16 ***
cluster3             0.9645947  0.0031151  309.649  < 2e-16 ***
cluster4             0.3875939  0.0029967  129.340  < 2e-16 ***
cluster5             0.2413202  0.0028720   84.026  < 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: 1189997  on 12145  degrees of freedom
AIC: 1267655
```

The cluster variable should not be included in place of the **temp** and **humidity** variables. This substitution increased the test MSE from 20,022 to 20,477, a significant worsening of the highly valued predictive power. Also, because the clusters are not well separated, the sudden jumps in predictions from one cluster to another is not sensible for miniscule differences in temperature or humidity. For example, in going from cluster 5 (low temp, low to medium humidity) to cluster 3 (high temp, low humidity), the multiplicative impact on the predicted mean is $e^{0.965-0.241} = e^{0.724} = 2.06$, or more than doubling.

Task 11 – Select the final model to present to the client (6 points)

Many candidates did well on this task, though some did not address the relative unimportance of interpretation given the business problem. Stronger candidates noted that the intercept-only model has not just larger errors but imbalanced errors, giving the most erroneous predictions at the times when bike usage is highest and most important to ABC.

The table below contains the train and test MSE, from best (lowest) to worst (highest) of the latter, for all the models considered:

Train MSE	Test MSE	Model
1,715	2,228	Boosted decision tree with shrinkage of 0.1 on 1000 depth 4 trees (Task 6)
4,968	5,282	Boosted decision tree with shrinkage of 0.01 on 1000 depth 4 trees (Task 6)
19,782	20,022	GLM with Poisson distribution on all original variables (Task 8)
19,782	20,023	GLM with Poisson distribution including hour * holiday interaction (Task 9)
20,162	20,477	GLM with Poisson distribution substituting cluster for temp/humidity (Task 10)
32,177	127,311	GLM with gamma distribution on all original variables (Task 8)

The boosted decision tree with shrinkage of 0.1 is recommended based on its vastly superior predictive power overall as indicated by the much lower test MSE, as ABC is most interested in predictive power. The boosted decision tree does have drawbacks compared to other models in being harder to explain and interpret, and its negative predictions will need to be addressed, likely by flooring these at zero, but these drawbacks are not enough to offset the massive reduction in the mean square error.

The summary function for the intercept-only model is given here:

```
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
```

This model predicts bike usage of $e^{5.243}$ = 189 bikes per hour, the average of the data it was trained on, regardless of time, day, or weather. On test data not used to train the model, it has a mean square error (MSE), a common measure of model accuracy, of 33,546. In contrast, the chosen boosted decision tree model has a test MSE of 2,228. In addition, the simpler model will do particularly poorly during high usage times, likely the most important prediction for ABC and where the error of the intercept-only model is greatest.

Task 12 – Write an executive summary for the client (20 points)

The presentation in the executive summary needs to be significantly different from that of the tasks above. It should provide key takeaways and use technical terms only as needed and with sufficient explanation. Discussion of the business problem and the data supporting it should include deeper insights on the nature of the data and cautions regarding reliance on the data. The discussion of models should focus on predictive power given this business problem, but it is worth noting the reduction in interpretability. High-level conclusions about what drives the model predictions, linking these to common sense notions, e.g. preference for daylight hours and warm but not hot temperatures, should be used instead of technical listings that are not sufficiently interpreted. A concrete example of the value added by the model is specifically called for. The summary should have a conclusion and next steps.

Many candidates did well when describing the data, though some candidates devoted too much of the summary to this section. Candidates generally did less well when describing the modeling process and often did not adequately explain why a particular model was chosen. Some candidates missed the opportunity to connect the modeling results to common sense regarding bike usage.

To: ABC Bike Sharing

From: Actuarial Analyst

You requested that we develop a model to predict the number of bike rentals in a given hour to help you with the distribution of bikes.

Preparing for Modeling

The dataset used to develop the model has 17,376 records with information on the number of bikes per hour as well as the season, year (2011 or 2012), hour of the day, day of the week, weather conditions (clear/partly cloudy, misting, or rain/snow), temperature, humidity, and windspeed.

Preliminary data exploration provided some initial insights into how the number of bike rentals per hour varies based on the other variables.

Daily peaks in rentals per hour occur at 8 a.m. and again around 5-6 p.m. This suggests that bike rentals are being used for the morning and evening commutes. However, there is a longer time with elevated rentals around the evening commute compared to the morning commute, and there is also a higher peak. This likely reflects higher rental use for non-commute late afternoon or early evening social activities. As one would expect, nighttime rentals are very low.

Other intuitive observations include 1) lower bike rentals in the winter, 2) lower bike rentals when it is raining or snowing, and 3) lower bike rentals in low temperatures.

Because there is a distinct pattern of bike rentals throughout the day for holidays compared to non-holidays, we developed a new interaction variable, a variable that can reflect the dependency of bike rentals per hour on whether it is or is not a holiday. While not impacting ordinary day-to-day distribution for bikes, an observed lack of morning demand for bikes on holidays may mean mid-day restocking is not needed and you may be able to require fewer employees to work holidays.

The Model

Three types of model were considered for this project, decision trees, Generalized Linear Models (GLMs), and boosted decision trees. Decision trees use a series of if-else statements to make a prediction. GLMs use a mathematical formula to make a prediction. Boosted decision trees use a sequential series of decision trees, with each successive tree targeting the deficiencies of the prior tree. The decision tree did not provide useful insights, only predicting low bike rentals in the early morning and for lower daytime temperatures and noting that daytime, warmer weather bike rentals had grown from 2011 to 2012. Several GLMs and boosted trees were developed and compared.

Our final recommended model is a boosted tree. To select this model, we considered 1) the model performance (as measured by MSE – a commonly used metric for measuring model accuracy) and 2) the interpretability of the model results. The test MSE, or mean squared error on the test data set, was significantly lower for the boosted tree than for the other models, indicating it was able to more closely predict actual bikes per hour on data that was not included when developing the model. While GLMs are easy to interpret, since they can be expressed as a mathematical formula, they had significantly worse model performance. While the boosted tree does not have a straightforward, formulaic model representation, we can still gain significant insights from a review of the model. It also naturally reflects interactions between variables, like the interaction we discussed above between the hour of the day and whether the day is a holiday.

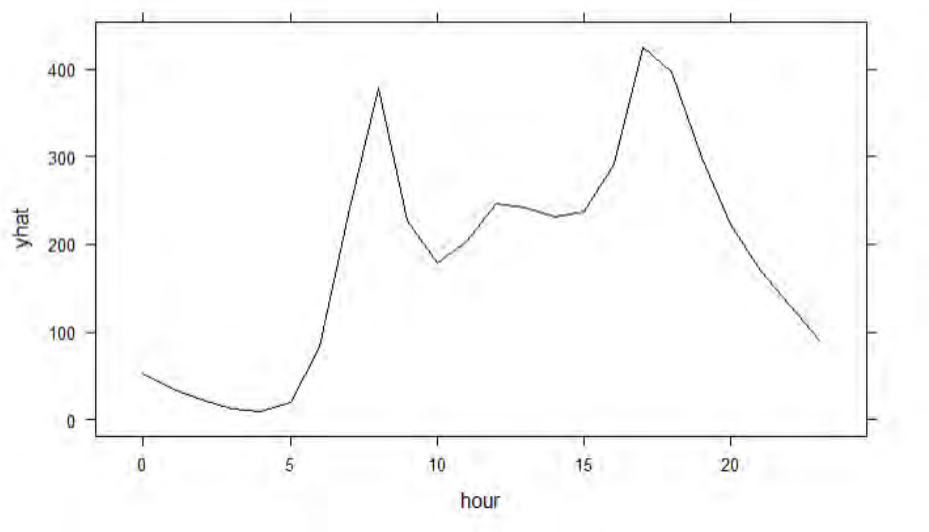
The overall average bike usage in the data used to develop our models is 189. To evaluate how much the model can help predict bike usage compared to this naïve estimate, we compare the average error size of the model predictions (averaging the absolute difference between the predicted bikes per hour based on the boosted decision tree and the actual bikes per hour) to that if we always predicted 189 bikes per hour. The naïve prediction that the number of bikes per hour will always be 189 has an average error size of 143 bikes per hour, showing a significant amount of variability in the data around the average, with the largest errors occurring at high bike usage times. In contrast, the recommended boosted tree gives an average error size of 31 bikes per hour.

For the selected boosted decision tree, the relative contribution of each variable to the model's prediction is shown below.

Variable	Relative Importance
Hour of the Day	59%
Temperature	13%
Day of the Week	10%
Year (2011 or 2012)	9%
Season	3%
Humidity	3%
Weather Situation	2%
Holiday	1%
Windspeed	1%

We see that the hour of the day provides the majority of the contribution to the model predictions, followed by temperature and day of the week.

Across all observations used to build the model, the average predicted bikes per hour ("yhat" in the graph below) based on the hour of the day ("hour") is shown below.



We see that the model predictions have captured the morning and evening commute peaks that we observed during our initial data exploration, as well as the very low overnight rentals. These are intuitive results based on general commute and sleep patterns, and they can be used to inform the timing of bike distribution.

Because bike usage appears to be driven by commuters, ensuring availability along common commute routes may increase usage. Based on usage patterns by day, we can recommend that overnight redistribution of bikes is not needed, as long as bikes have been distributed after the evening rush subsides, redistributing in the 8-11 pm time range. Finally, the data set provided for analysis did not include location information. To better assist you in planning distribution, we would like to analyze additional information that includes pick-up and drop-off locations, to evaluate when you may need to redistribute bikes between locations.