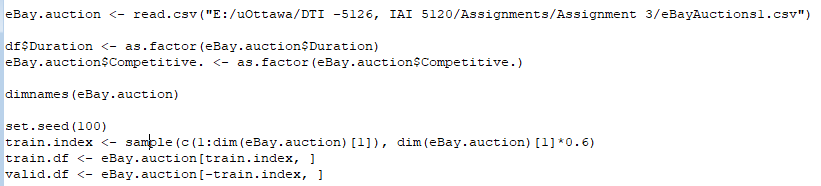
**Assignment 3- Group-28**

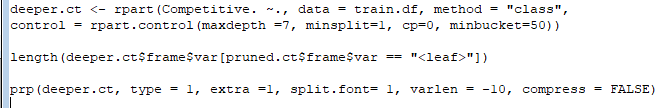
**Fahim Shahriare Anik- 300208836 & Asif Ahmed- 300253221**

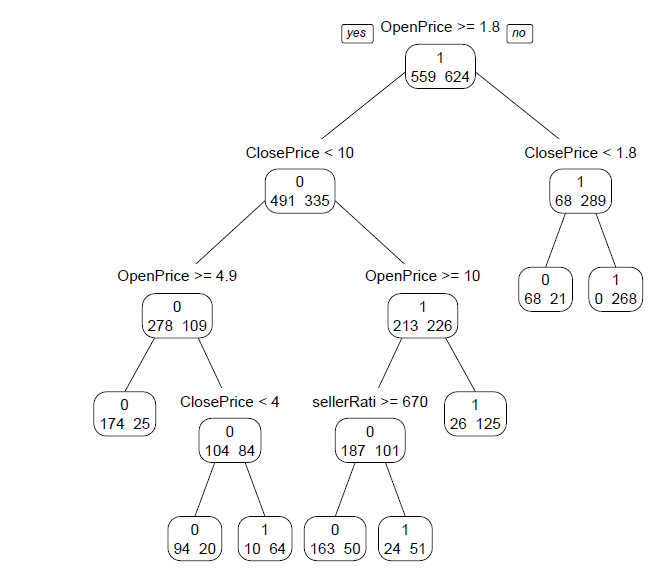
**Answer to Question 1.a:**

For this answer we first converted “Duration” and “Competitive?” to factors. Then we stored them in a variable, then we created training and validation data sets. The code is given below:

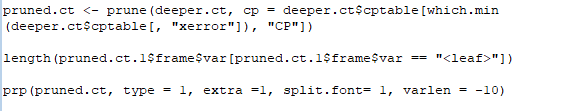


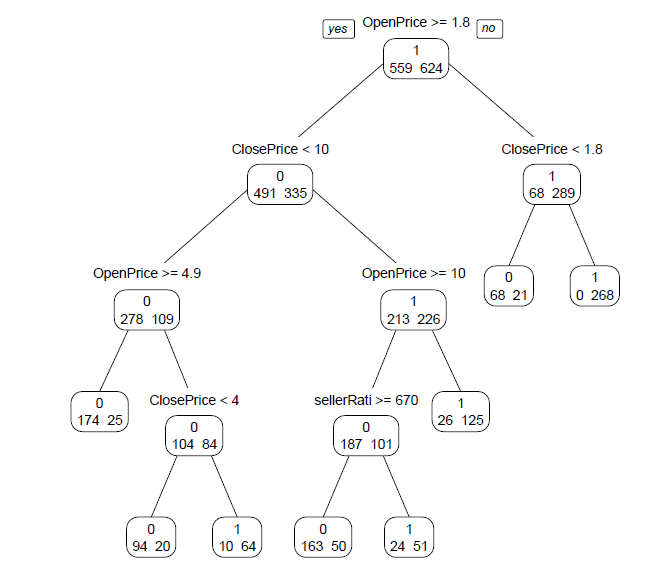
Then we created and plotted the full tree. For the full tree we used the value of “cp=0” to increase complexity so that we obtain all the splits and nodes. The rest were adjusted according to the question. The code and tree are given below:





For 1, we fit a tree using the best pruned Tree, with minimum number of records to be 50 in a terminal and maximum level of depth is 7. The code and Tree is given below:





The results in term of rules are given below:

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 4.9)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice < 4.9) AND (ClosePrice < 4)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice < 4.9) AND (ClosePrice >= 4)

THEN Class = 1 (competitive)

IF (Open Price >= 1.8) AND (ClosePrice >= 10) AND (OpenPrice < 10)

THEN Class = 1 (competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10) AND (SellerRating >= 670)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10) AND (SellerRating < 670)

THEN Class = 1 (competitive)

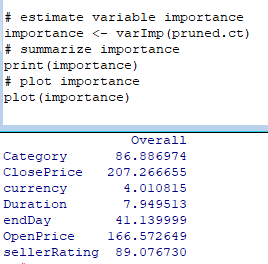
IF (Open Price < 1.8) AND (ClosePrice < 1.8)

THEN Class = 0 (non-competitive)

IF (Open Price < 1.8) AND (ClosePrice >= 1.8)

THEN Class = 0 (non-competitive)

For clarity of presentation I would check the importance of the variables as well as the splits present in the tree. Therefore, if we check the importance of the predictors in this split we get:



If there was any split I would like to reduce (even though this tree is well balanced and not overfitted), Based on the importance and the tree design we can say that “sellerRating” compared to “ClosePrice” has less importance and therefore, I would remove the rules:

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10) AND (SellerRating >= 670)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10) AND (SellerRating < 670)

THEN Class = 1 (competitive)

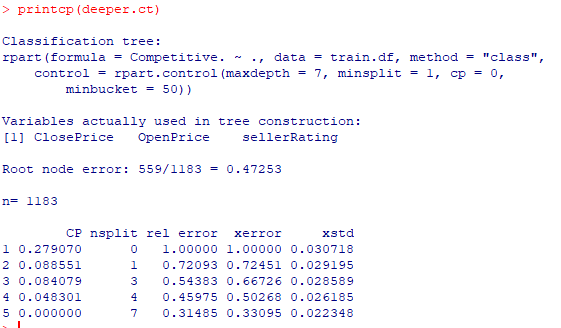
Can be generalized to:

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10)

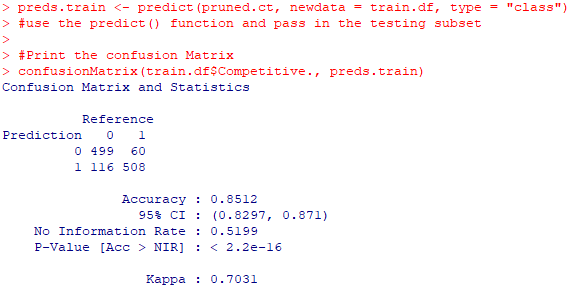
THEN Class = 0 (non-competitve)

**Answer to Question 1.b:**

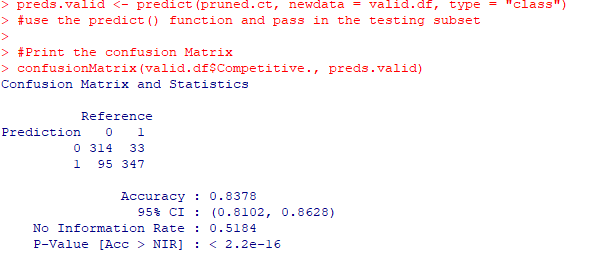
For this question if we cross validate the number of splits with the error and check the error rates and then use confusion matrix to measure the accuracy we will be able to tell if it is practical or not. Below is the code and results of both:



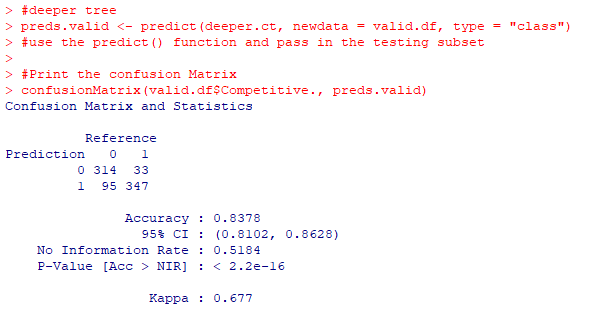
This is the code for the complexity matrix that was cross validated, as we can see with 7 splits, the cp value is reduced to 0, however it gives us the lowest xerror value, and adding the xerror and xstd gives us (0.33095+0.022348 = 0.353298) which falls under the One Standard Error region. Below we have the confusion matrix to evaluate the accuracy of the tree:

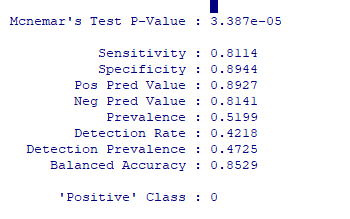


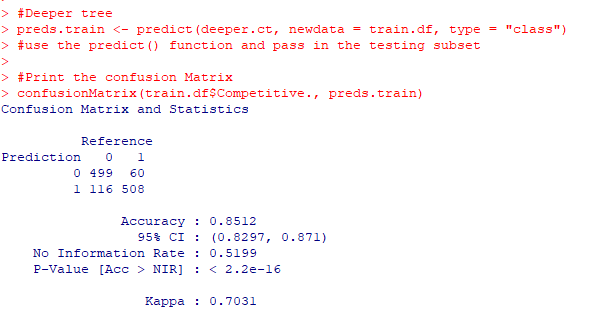
This is for the Training set, for the validation set we have:

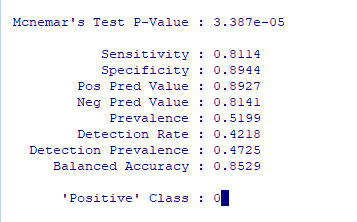


Therefore, as we can see that the accuracy is 83.78% for the pruned tree, and as we can see from the code below from the original tree that it is the same, therefore, we can say that this is a good model to predict the outcome of a new auction.









These results tell us that based on the accuracy of the pruned tree and original tree and compared to the cross validation with number of splits, this gives us the lowest error and the highest accuracy compared to other splits. Also, based on the sensitivity and specifity, the values are all above 80% show that we can detect higher amounts of values with True negative compared to True positive.

**Answer to Question 1.c:**

When the opening price is greater or equal to 1.8 and the closing price is less than 10, then it as to a non-competitive bidding.

We can see that when the opening price is more than or equal to 10 and the seller rate is less than 670 then it is a competitive bidding. However, when the seller Rate is more than that, then bidding becomes non-competitive. It clearly shows, that seller rating matters upto one point, when the opening price of an item is more than 11, however, it does not matter after a certain point which is 670

Likewise, when the opening price is less 1.8 and the closing price is greater than 1.8 then it becomes a competitive bidding.

One of the situations we see is that the impurity of the split reduces as we go down the splits and move more towards the leaf nodes. With the use of Gini Index we can see that compared to the split at ClosePrice < 10, the impurity at ClosePrice < 4 is much lower. Also, the splits were more even and therefore this shows that the tree was good in terms of quality.

At Close Price <10:

Gini\_Index =

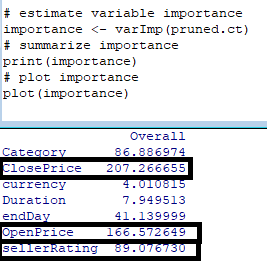
At ClosePrice <4:

Gini\_Index =

As you can see the split at the lower level is closer to 0.5 compared to the split at the higher level, this shows a better splitting and a more even distribution of the tree.

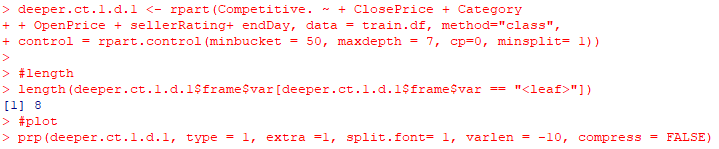
In addition to this we can say that another finding but not so interesting would be that the split quantity is also evenly spread, for the split at OpenPrice >= 1.8 - the left side had 559 and right side had 624 values.

Another expected output that was interesting is the fact that most of the splits were splits that had High importance:

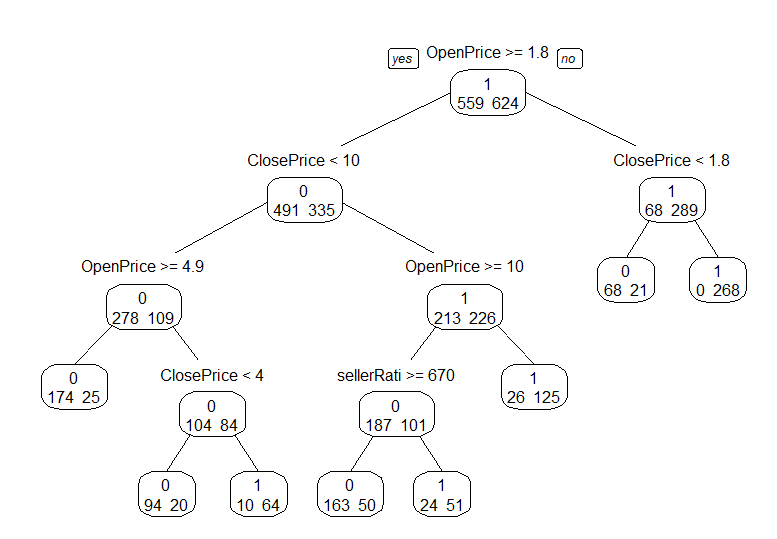


**Answer to Question 1.d:**

Based on the importance from the tree we can say that “currency” and “duration” are not of high importance and therefore we can remove those when making the classification tree in this question as all the other predictors are important. The code below and followed by the tree shows the new classification tree equation:



The image below will show the tree. With the equation above, as you can see the “Duration” and “currency” is not included.



As we can see from the image above it is the same as the pruned tree in the first part of the question (1.a) therefore, the rules are still the same. They are:

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 4.9)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice < 4.9) AND (ClosePrice < 4)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice < 4.9) AND (ClosePrice >= 4)

THEN Class = 1 (competitive)

IF (Open Price >= 1.8) AND (ClosePrice >= 10) AND (OpenPrice < 10)

THEN Class = 1 (competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10) AND (SellerRating >= 670)

THEN Class = 0 (non-competitive)

IF (Open Price >= 1.8) AND (ClosePrice < 10) AND (OpenPrice >= 10) AND (SellerRating < 670)

THEN Class = 1 (competitive)

IF (Open Price < 1.8) AND (ClosePrice < 1.8)

THEN Class = 0 (non-competitive)

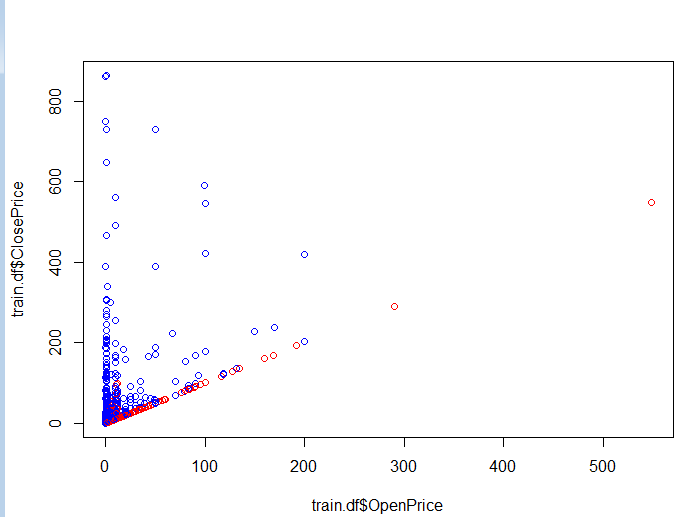
IF (Open Price < 1.8) AND (ClosePrice >= 1.8)

THEN Class = 0 (non-competitive)

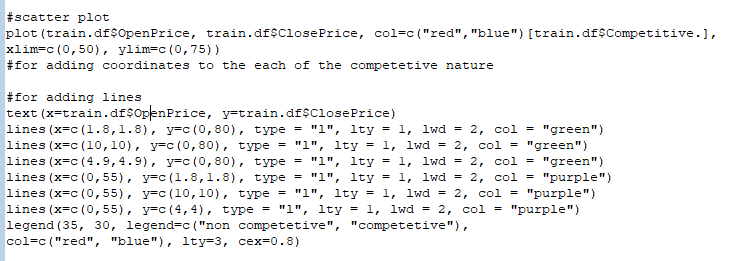
**Answer to Question 1.e:**

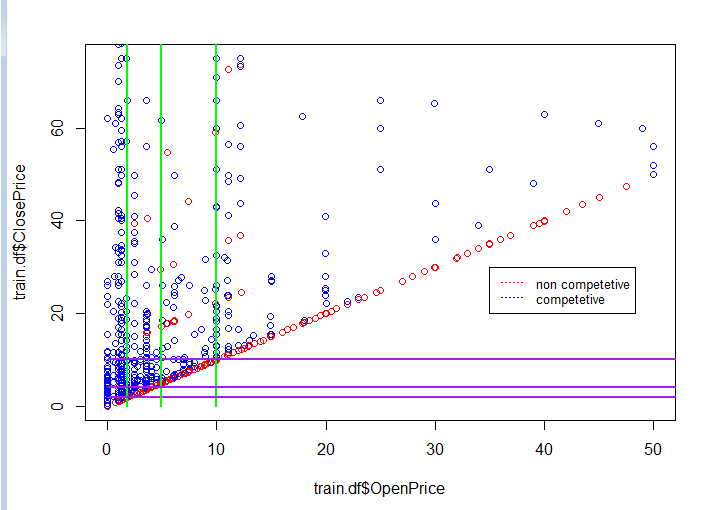
For this answer, since the scatter plot is too big, we showed the lines after constricting the plot, the unconstricted scattered plot code and image is given below:



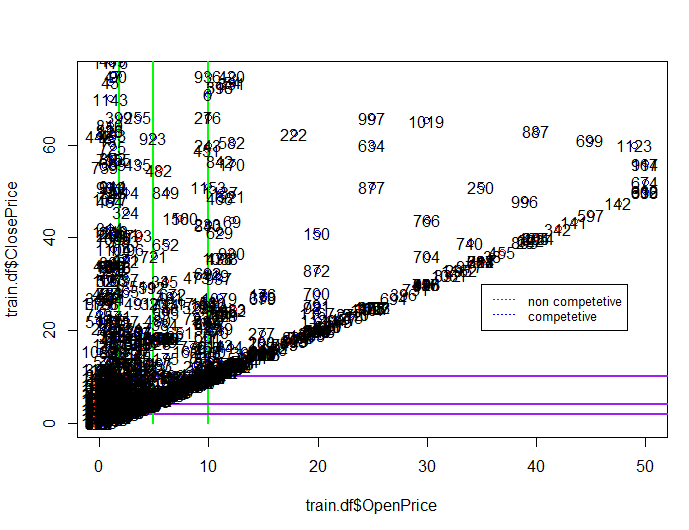


Then we used limiting ranges as you will see in the code to get this graph:





The Following image shows the coordinates of all the corresponding points as well:

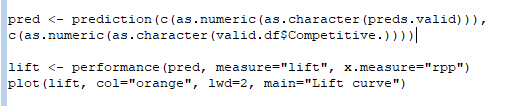


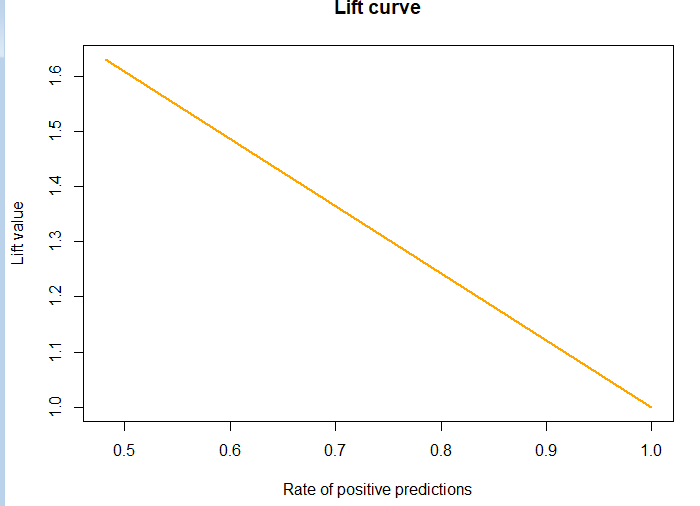
With respect to the two predictors, the splitting does seem reasonable because most of the observations are concentrated between 0<OpenPrice<10 and 0<ClosePrice<20. Therefore, lines shown are closer to those values.

Based on the graph, as we can see the split is done in a way that ensures that most of the values in the split is pure with some impurities from the other side. Although it can be better this split as there are some impurities with non-competitive values mixed with competitive values, it is the best that can be done given the distribution of data.

**Answer to Question 1.f:**

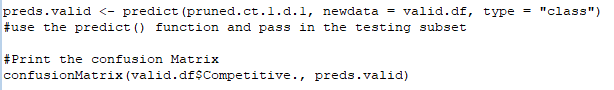
Lift chart shows the actual lift. Since the lift chart baseline starts from 1 in the y-axis, and the lift curve is straight. Therefore, the results obtained with and without the model can be represented using this line. As the rate of Positive prediction increases, the lift value falls linearly and therefore we can conclude that this is a good model for predicting the outcome. This shows that, for example, with low rates, (say 50%) we will be able to predict 1.5 times more in terms of predicting the outcome (whether it is competitive vs. Non-competitive).

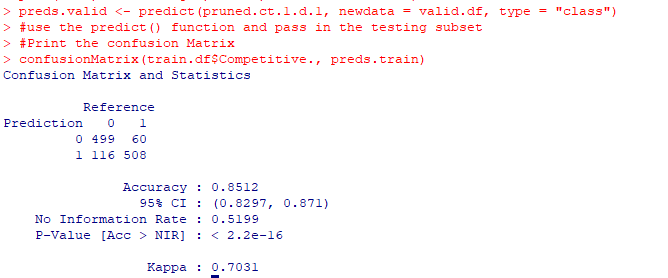


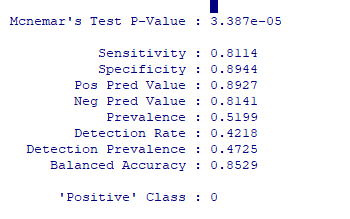


As you can see from the code and the graph,

As for the confusion matrix, below is the code and result:







This shows that the model is better at detecting “True Negative” values compared to “True positive” values as it has a higher Specificity compared to sensitivity. However, the value for Sensitivity is also not bad (81%+). Therefore, the model will have high correctness value as

Correctness = (TF+TN)/(TF+TN+FP+FN)