Project 6 Report

Jose Alfaro

April 24, 2019

Question: 1

Part: A

We performed an exploratory data analysis on the cereal dataset in which we saw that Fiber and Potassium had the highest correlation in the data set with a correlation of 0.929. We also noted that Sodium and Fat were the least correlated variables in the dataset with a correlation of 0.00955. Moreover, we created scatterplots of the data which allowed us to visualize potential trends and patterns more closely. Here we observed a good mix between discrete and continuous data within the dataset. These observations can be seen in greater detail in *figure 1*.

Part: B

After looking at the ranges of our dataset, it became clear the we needed to standardize the data since some variables such as Calories and Potassium had drastically larger ranges than variables such as Protein and Fiber. Thus, these variables with larger ranges would dominated those with smaller ranges.

A summary of the non-standardized data:

```
##     Calories        Protein           Fat             Sodium
##  Min.   : 50.0   Min.   :1.000   Min.   :0.0000   Min.   :  0.0
##  1st Qu.:100.0   1st Qu.:2.000   1st Qu.:0.0000   1st Qu.:145.0
##  Median :110.0   Median :2.000   Median :1.0000   Median :190.0
##  Mean   :107.9   Mean   :2.465   Mean   :0.9767   Mean   :180.5
##  3rd Qu.:110.0   3rd Qu.:3.000   3rd Qu.:1.5000   3rd Qu.:220.0
##  Max.   :160.0   Max.   :6.000   Max.   :3.0000   Max.   :320.0
##      Fiber       Carbohydrates       Sugar          Potassium
##  Min.   :0.000   Min.   : 1.00   Min.   : 0.000   Min.   : 15.00
##  1st Qu.:0.500   1st Qu.:12.00   1st Qu.: 3.000   1st Qu.: 37.50
##  Median :1.000   Median :14.00   Median : 8.000   Median : 60.00
##  Mean   :1.714   Mean   :14.26   Mean   : 7.605   Mean   : 84.42
##  3rd Qu.:2.850   3rd Qu.:17.00   3rd Qu.:12.000   3rd Qu.:110.00
##  Max.   :9.000   Max.   :22.00   Max.   :15.000   Max.   :320.00
```

A summary of the standardized data:

```
##     Calories          Protein             Fat               Sodium
##  Min.   :-3.0528   Min.   :-1.1991   Min.   :-1.21874   Min.   :-2.2782
##  1st Qu.:-0.4168   1st Qu.:-0.3807   1st Qu.:-1.21874   1st Qu.:-0.4477
##  Median : 0.1103   Median :-0.3807   Median : 0.02902   Median : 0.1204
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.1103   3rd Qu.: 0.4378   3rd Qu.: 0.65290   3rd Qu.: 0.4991
##  Max.   : 2.7463   Max.   : 2.8932   Max.   : 2.52453   Max.   : 1.7615
##      Fiber         Carbohydrates          Sugar            Potassium
##  Min.   :-0.9526   Min.   :-3.11376   Min.   :-1.67640   Min.   :-1.0501
##  1st Qu.:-0.6747   1st Qu.:-0.52989   1st Qu.:-1.01507   1st Qu.:-0.7097
##  Median :-0.3968   Median :-0.06009   Median : 0.08715   Median :-0.3694
```

```
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.6314   3rd Qu.: 0.64460   3rd Qu.: 0.96893   3rd Qu.: 0.3870
##  Max.   : 4.0494   Max.   : 1.81909   Max.   : 1.63026   Max.   : 3.5635
```

Part: C

We decided to use a metric-based (Euclidean distance) distance method to cluster the cereals since we are interested in clustering cereals based on similar characteristics as opposed to their relationships to the predictors.

Part: D

After standardizing the data, we hierarchically clustered the cereals using complete linkage and Euclidean distance. This led us to produce the dendogram shown in *figure 2*. From here, we can see that there are 4 main clusters in the dendogram which are displayed in red.

Part: E

We repeated the steps from part D, however, this time we clustered the cereal via K-means clustering as opposed to a hierarchical clustering. We chose to conduct K-means clustering with K = 2, 3, and 4. In *figure 3*, we see that K = 2 clusters the cereals well, but the groups seem to be very large with high amounts of variability. In *figure 4* we see that the clusters with K = 3 also clusters the cereals well and reduces the issue of high variability within groups. Lastly, the clusters with K = 4 (*figure 5*) seems to force the clusters into four groups since there are overlaps between the clusters. Thus, it seems that 3 clusters does the best job in grouping the cereals since the variability within clusters seems to be reduced and groups are well defined without being coerced into smaller clusters.

Part: F

From the results of the hierarchical and K-means clustering, we decided that the hierarchical clustering technique is better in this case since we can reproduce these same results every time and we do not need to have prior information about the number of clusters beforehand. On the other hand, K-means clustering gave us different results every time the algorithm was ran due to the randomness of the starting centroids and we had to have a prior idea of how many clusters we needed before implementing the algorithm. It seems that if we were to have more data, then K-means would be more stable and thus possibly a better clustering technique, however this is not the case. Therefore, due to the clearness and reproducibility of the hierarchical clustering, we would recommend this over K-means for this data.

Question: 2

Part: A

After standardizing the predictors and splitting the data in a similar fashion as the previous project, we fit a regression tree and computed its confusion matrix. Here we see that only three variables determine the shape of the tree (PPERSAUT, PBRAND, MOPLLAAG). The output is displayed in *figure 6*. From here, we see that this method predicts that every Caravan customer will not buy the insurance plan. The confusion matrix for this can be seen in *table 1*. From here we see that:

$$Sensitivity = \frac{0}{59} = 0\%$$

$$Specificity = \frac{941}{941} = 100\%$$

$$MR = \frac{59}{1000} = 5.9\%$$

$$Test\ Error\ Rate : 0.059$$

Part: B

We found the optimal size for the pruning tree to be 4 which is the same size as the regression tree in part A. Moreover, the tree produced was exactly identical to the one in part A. This implies that the most important predictors are again PPERSAUT, PBRAND, and MOPLLAAG. The confusion matrix for the pruned tree can be found in *table 2* From this table we see that:

$$Sensitivity = \frac{0}{59} = 0\%$$

$$Specificity = \frac{941}{941} = 100\%$$

$$MR = \frac{59}{1000} = 5.9\%$$

$$Test\ Error\ Rate : 0.059$$

Part: C

We conducted a bagging approach via random forests with 1,000 trees and specified the number of variables sampled as candidates as 85. We then constructed a confusion matrix in which we saw the following (*table 3*):

$$Sensitivity = \frac{15}{108} = 13.89\%$$

$$Specificity = \frac{4440}{4714} = 94.19\%$$

$$MR = \frac{367}{4822} = 7.61\%$$

$$Test\ Error\ Rate : 0.08$$

It is important to note here that the number of observations drastically increased from the last two tree-based method since we specified the use of 1,000 trees in the random forest function. Furthermore, we can see that the most important predictors using this method are MOSTYPE, PBRAND, and PPERSAUT since they had the highest mean decrease accuracy and mean decrease gini index (*figure 7*). In other words, these are the top variables that had least amount of node impurity.

Part: D

We repeated the random forest approach, however, this time we specified 1,000 tress and changed the number of variables sampled at each split to be $\sqrt{m}$ (~ 9). We then created a confusion matrix in which we saw the following (*table 4*):

$$Sensitivity = \frac{9}{48} = 18.75\%$$
$$Specificity = \frac{4494}{4774} = 94.13\%$$
$$MR = \frac{319}{4822} = 6.62\%$$
$$Test\ Error\ Rate : 0.069$$

We note that the number of observations drastically increased once again from the first two tree-based method since we specified the use of 1,000 trees in the random forest function. Furthermore, we can see that the most important predictors using this method are MOSTYPE, PBRAND, and PPERSAUT since they had the highest mean decrease accuracy and mean decrease gini index (*figure 8*). In other words, these are the top variables that had least amount of node impurity. Note that these are the same important predictors selected in Part D.

Part: E

We continued our study of the data by implementing a boosting technique to compare with our previous models. Here, we implemented a boosting technique with 1,000 trees, interaction depth 1, and a shrinkage parameter of $\lambda = 0.01$. We then constructed a confusion matrix and saw the following (*table 5*):

$$Sensitivity = \frac{13}{59} = 22.03\%$$
$$Specificity = \frac{917}{941} = 97.45\%$$
$$MR = \frac{70}{1000} = 7\%$$
$$Test\ Error\ Rate : 0.07$$

We also note that the relative influence plot produced from this method showed that PBRAND was the most important variable. This can be seen in *figure 9*.

Part: F

We proceeded to implement the KNN algorithm. After running the KNN algorithm for K values 1 through 86, we saw that the "U-shaped" curve happened when the test error rate was at around K=9 (*figure 10*). To check our intuition, we decided to check for the K value associated with the minimal test error. We then concluded that $K = 9$ was indeed the optimal K value for this dataset and was associated with a 0.058 test error rate. After choosing the optimal K value, we re-ran the algorithm with $K = 9$ and created the confusion matrix found in *table 6*. From here, we saw that:

$$Sensitivity = \frac{1}{59} = 1.69\%$$
$$Specificity = \frac{941}{941} = 100\%$$
$$MR = \frac{58}{1000} = 5.8\%$$
$$Test\ Error\ Rate : 0.058$$

Part: G

Lastly, we constructed a logistic regression model with purchase as the response variable. From here, we based our predictions based off the predicted probability values. From this model, we created a confusion matrix which can be found in *table 7*. From here, we saw that:

$$Sensitivity = \frac{15}{59} = 25.42\%$$

$$Specificity = \frac{897}{941} = 95.32\%$$

$$MR = \frac{88}{1000} = 8.8\%$$

$$Test\ Error\ Rate : 0.088$$

We also note that by running several t-tests only a handful of variables are found to be significant with the most significant being PBRAND and PPERSAUT.

Part: H

When comparing the results from parts A through G, we kept in mind that our goal was to maximize sensitivity while minimizing the misclassification rate as well as the test error rate. With that in mind, we decided that the optimal method for this dataset is the random forest approach with 1,000 trees and the number of variables sampled at each split to be $\sqrt{m}$ (method used in part D) because it had a good balance of low misclassification rate, low test error rate, and high sensitivity. It also included the most important variable found by all models which was PBRAND.

Question: 3

Part: A

We fit a regression tree to the Hitters dataset by hand in which we took log(Salary) to be the response and Years and Hits to be our predictors. We created a for-loop in which we calculated the SSE and split value between the predictors and the response. After calculating these values, we split the data based on the smallest amount of SSE produced by the variables. We then subsetted the data based on the split preformed and iterated the procedure until we could not split further. This produced the tree that can be found in *figure 11*. This tree is almost identical to the one produced by the tree function in R, however, upon further investigation, we found that the tree function rounds values to the third decimal place which could be why we got slightly different values for our leafs. All in all, the trees look almost identical.

Part: B

Figure 12 shows what the regression tree looks like when done via R.

Question: 4

Part: A

We fit a classification tree to the Hitters dataset by hand in which we took log(Salary) to be the response and Years and Hits to be our predictors. First, we calculated the deviance at the root to be 409.95 and concluded

that a split was possible. We then calculated the total deviance for all the Thal levels (*table 8*) to be 323.388 and calculated the total deviance for all of the Ca levels (*table 9*) to be 333.927. We then decided to split based on the Thal variable since it had the lowest deviance. More specifically, we chose the second level of Thal to be the left branch of the tree and the other levels to be the right branch since the second level of Thal had the highest deviance out of the three levels. On the left branch, we subsetted the data so that only Thal level 2 is considered and calculated the deviance for the different levels of Ca with a fixed Thal = 2 level. From here, we saw that Ca level 0 had the highest deviance (*table 10*), therefore we made it the left branch and all other levels the right branch.

For the righthand branch, we only considered the data that considered the levels for Thal that excluded 2. From here, we calculated the deviance of the Ca variable and saw that level 0 had the highest deviance (*table 11*) and thus we split the tree here with level 0 to the left. We then noted that Ca level 2 consisted of a pure node since it had a perfect proportion of 1, thus we split Ca level 2 to the left of the sub-branch and the other levels to the right.

All in all, the tree looks identical to the one produced by code.

Part: B

Figure 13 shows what the classification tree looks like when done via R.

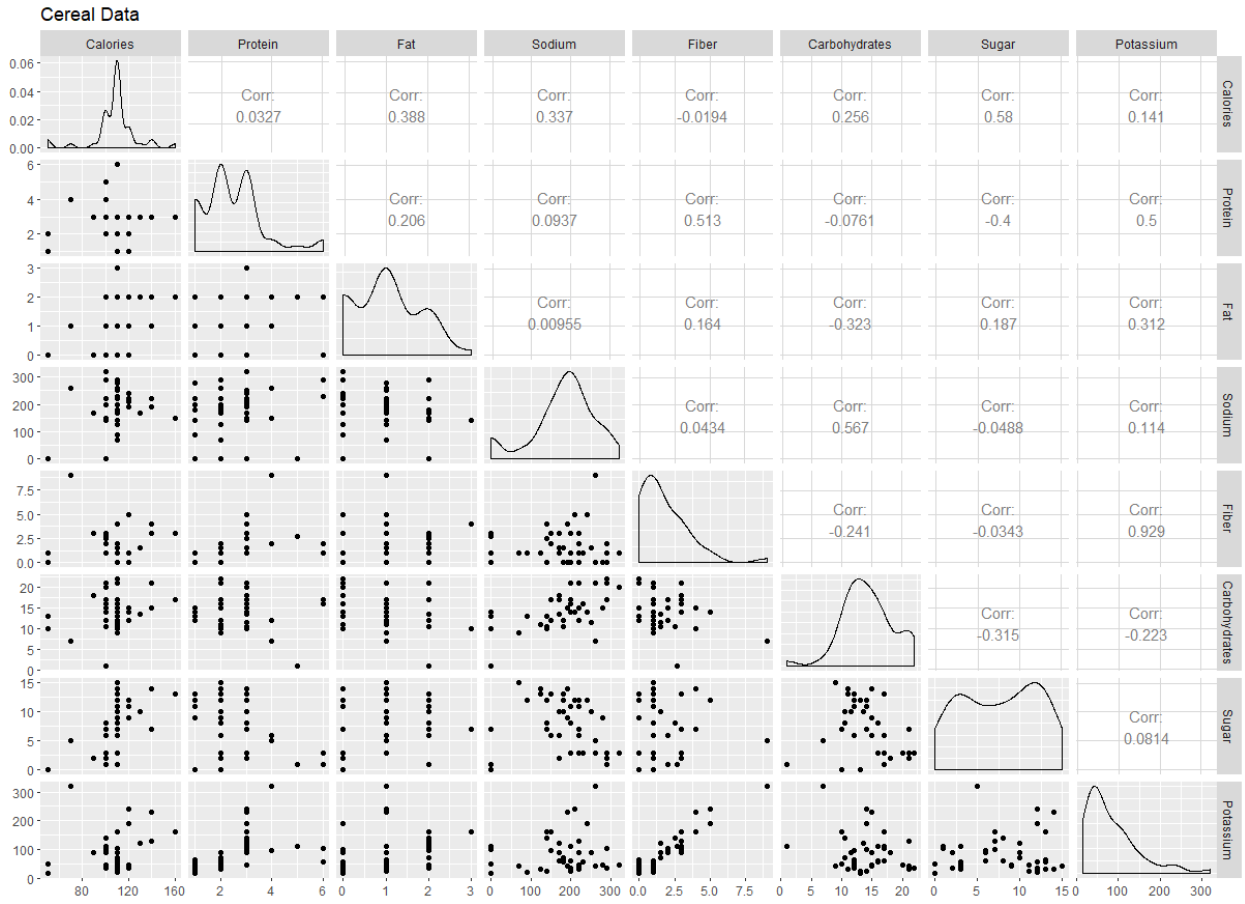

Figure 1: Exploratory Data Analysis of Cereal Data

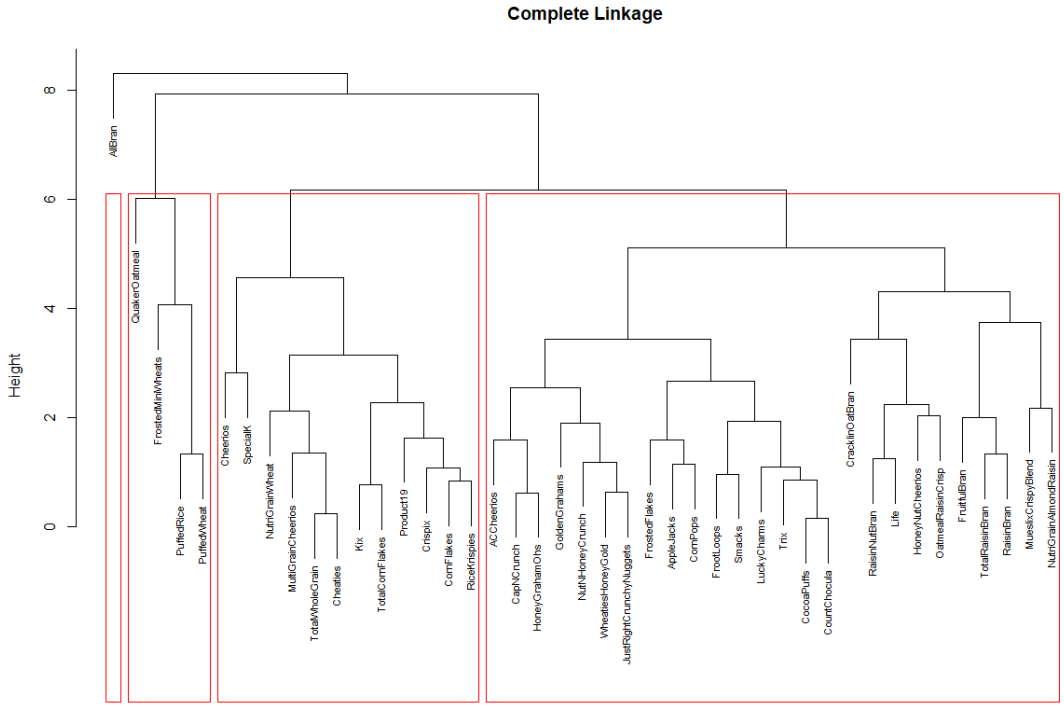

Figure 2: Heirarchical Clustering Dendogram

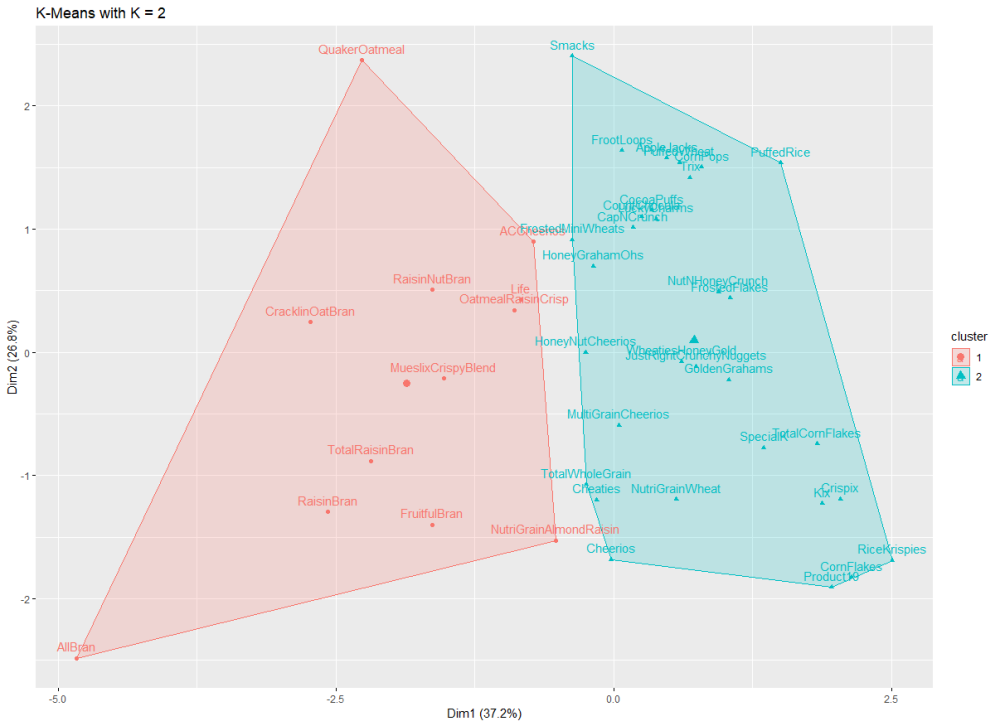

Figure 3: K-Means with K = 2

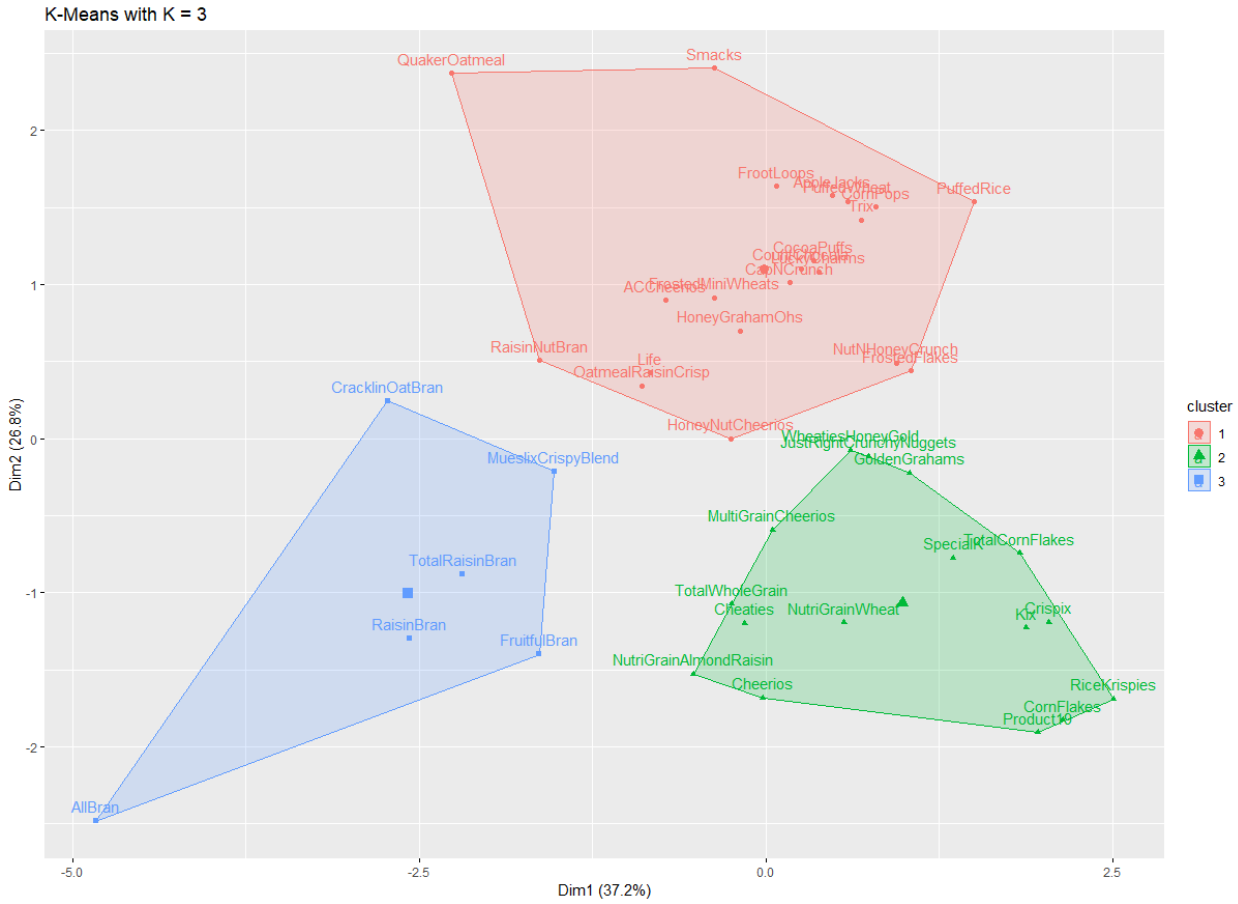

Figure 4: K-Means with K = 3

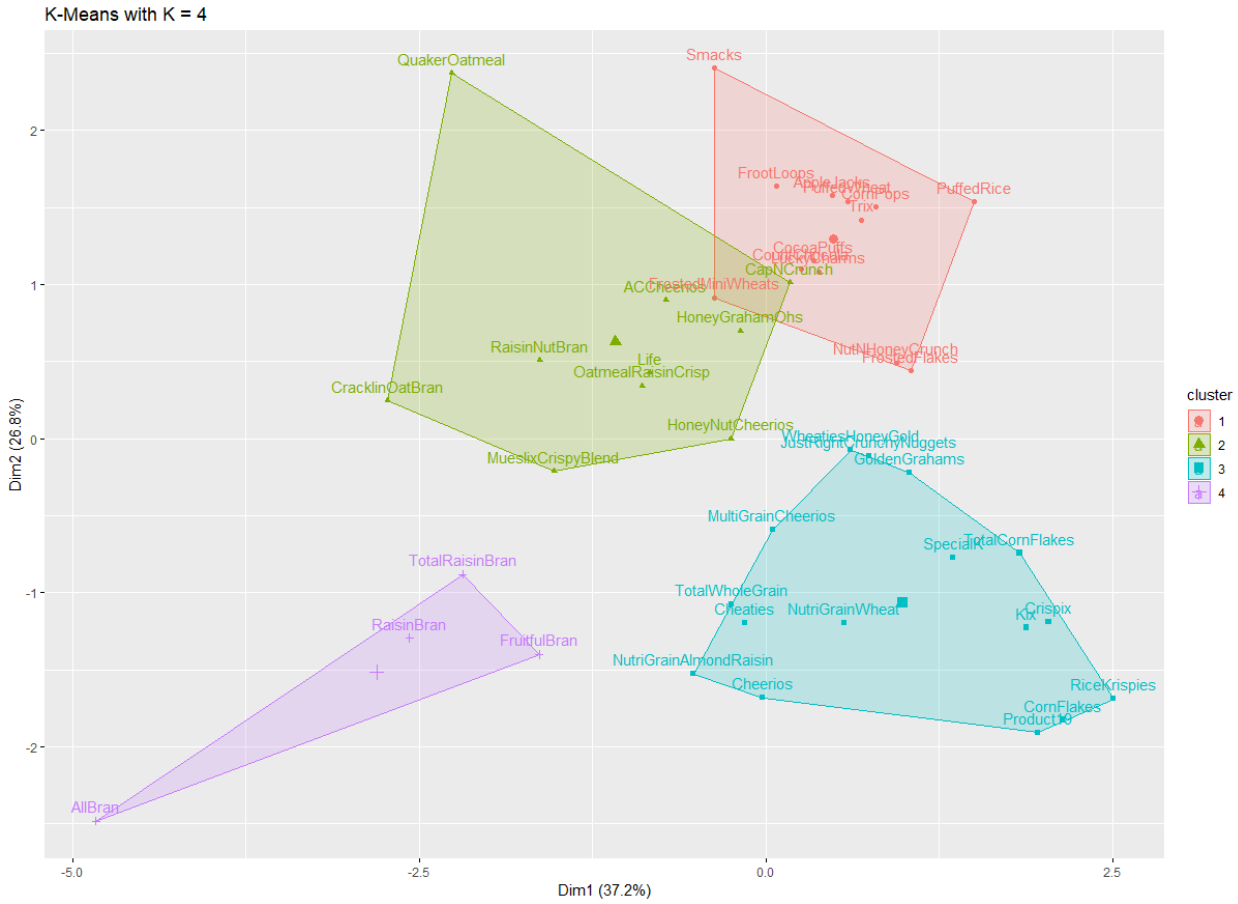

Figure 5: K-Means with K = 4

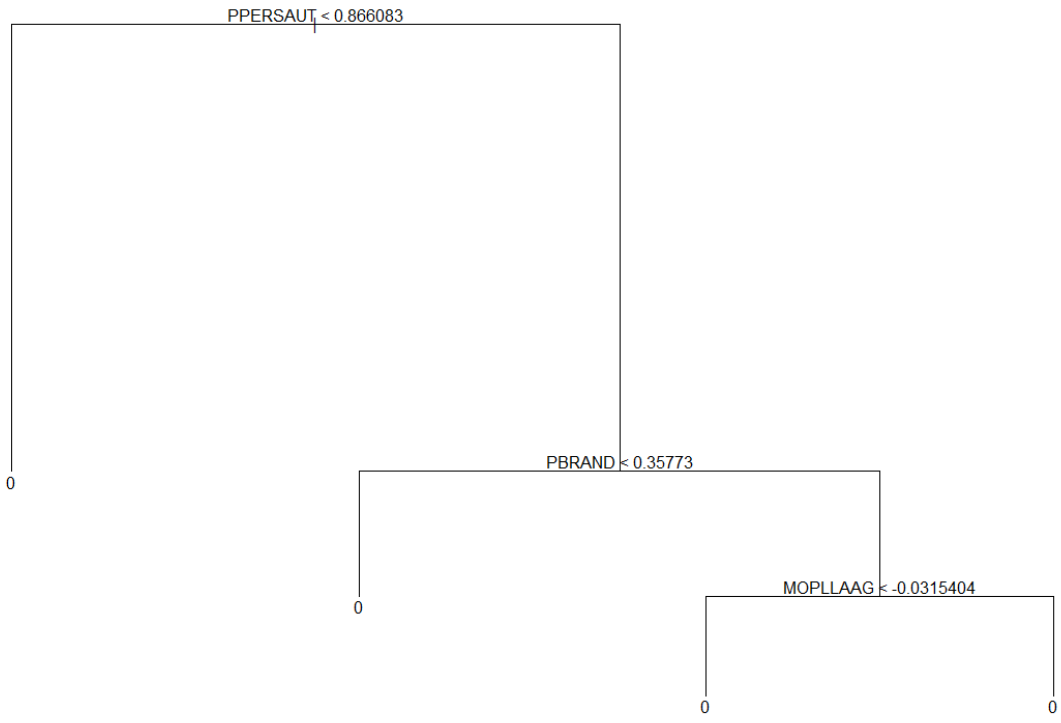

Figure 6: Caravan Data Regression Tree

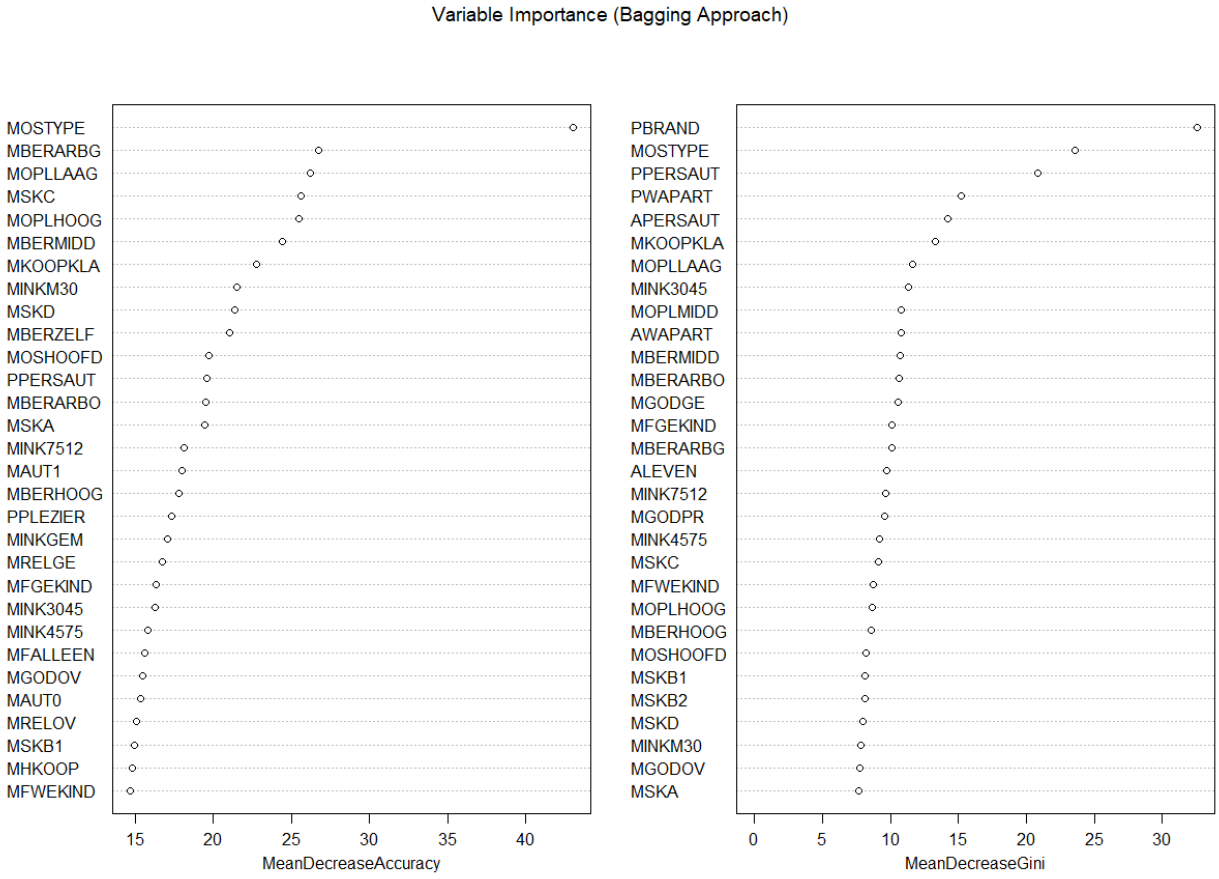

Figure 7: Variable Importance via Bagging Approach

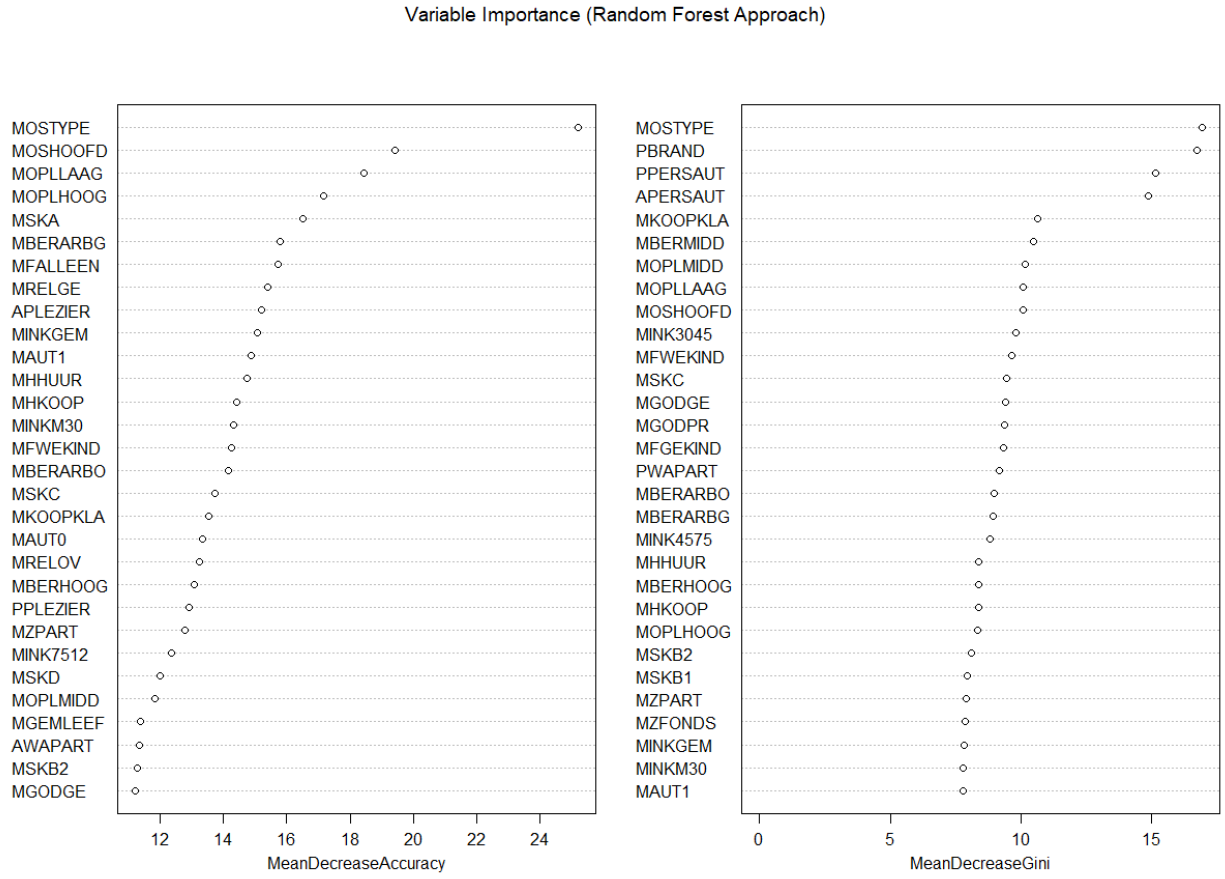

Figure 8: Variable Importance via Random Forest Approach

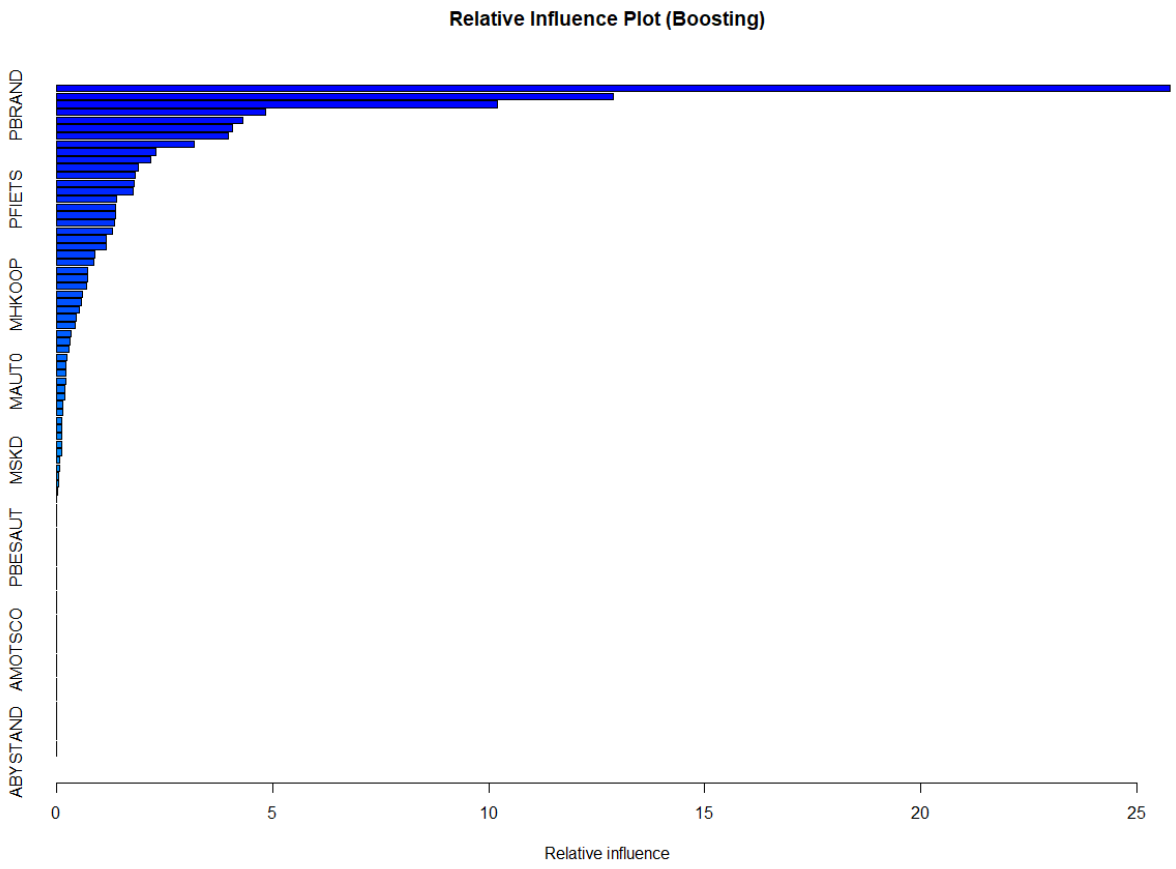

Figure 9: Relative Influence Plot via Boosting (Part E)

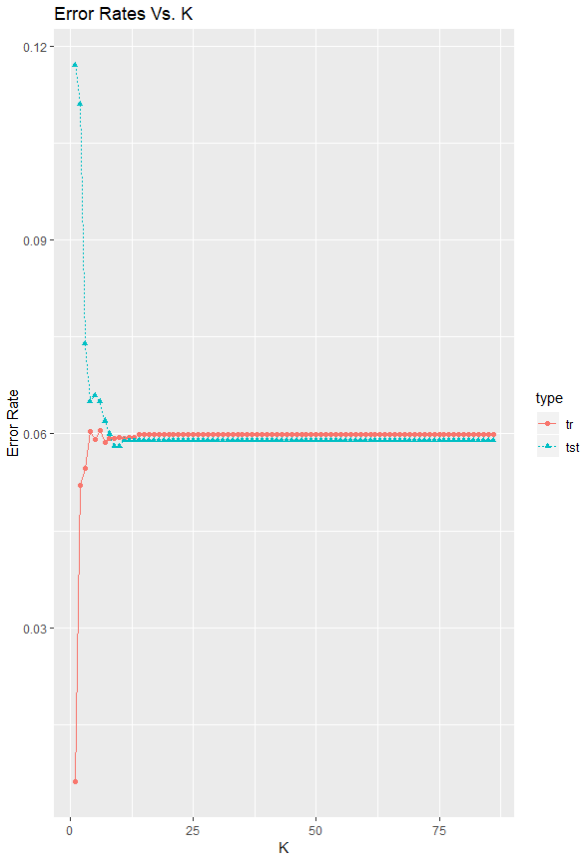

Figure 10: Error Rates Vs. K (KNN)

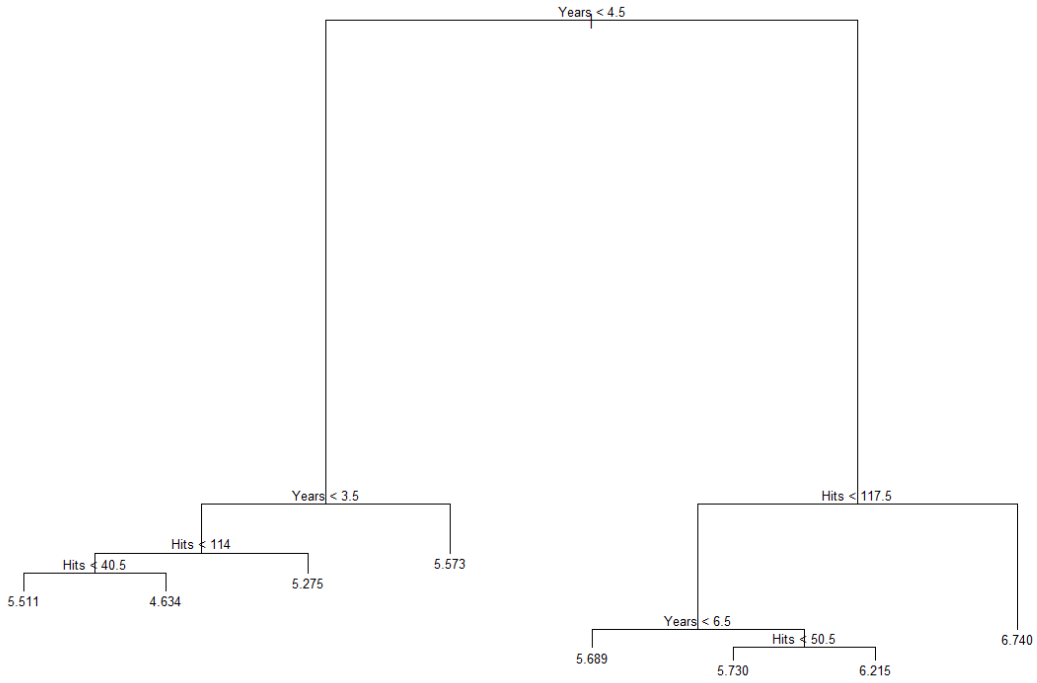

Figure 11: Regression Tree by Hand

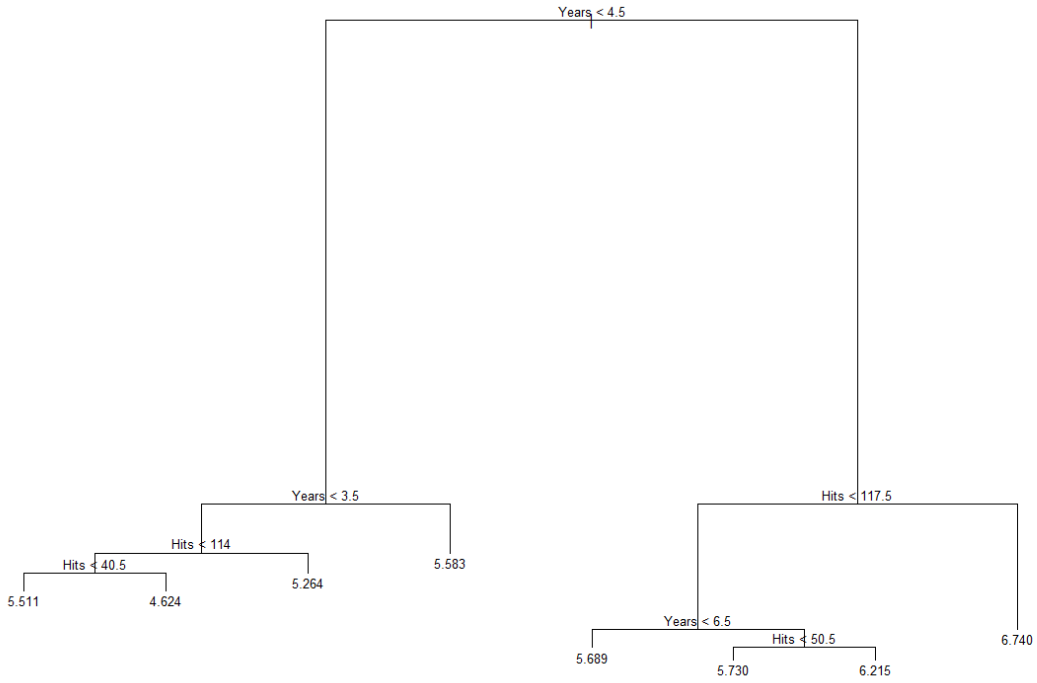

Figure 12: Regression Tree via Code

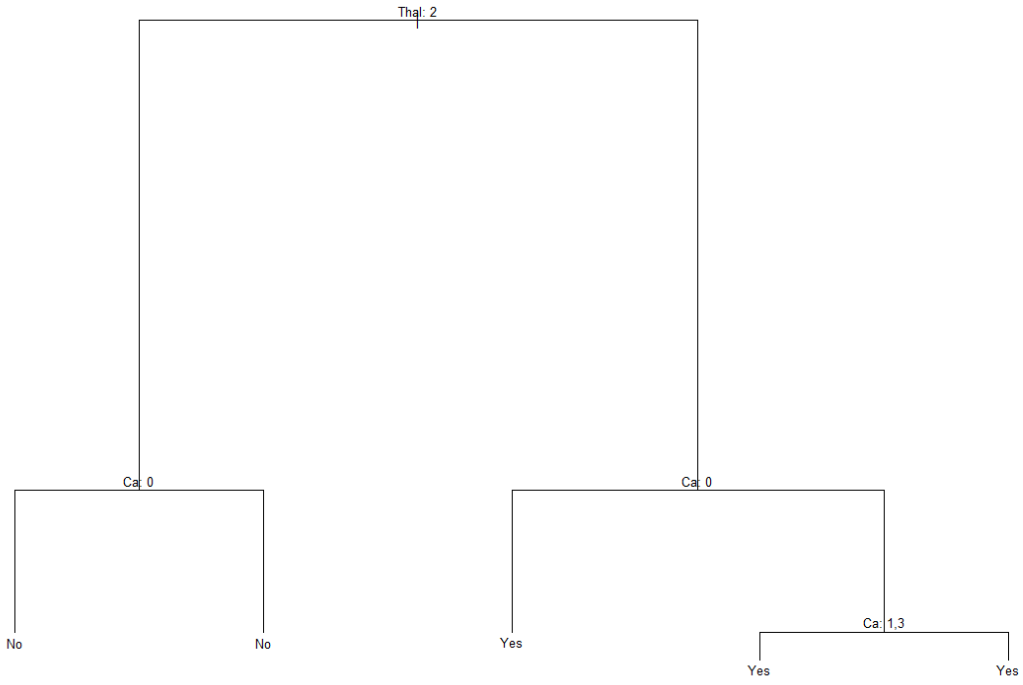

Figure 13: Classification Tree via Code

		Actual	
		No	Yes
Predicted	No	941	59
	Yes	0	0

Table 1: Confustion Matrix for Caravan Regression Tree

		Actual	
		No	Yes
Predicted	No	941	59
	Yes	0	0

Table 2: Confustion Matrix for Pruned Caravan Regression Tree

		Actual	
		No	Yes
Predicted	No	4440	93
	Yes	274	15

Table 3: Confusion Matrix for Bagging Approach

		Actual	
		No	Yes
Predicted	No	4494	39
	Yes	280	9

Table 4: Confusion Matrix for Random Forest Approach

		Actual	
		No	Yes
Predicted	No	917	46
	Yes	24	13

Table 5: Confusion Matrix for Bagging (Part E)

		Actual	
		No	Yes
Predicted	No	941	58
	Yes	0	1

Table 6: Confusion Matrix for KNN

		Actual	
		No	Yes
Predicted	No	897	44
	Yes	44	15

Table 7: Confustion Matrix for Logistic Regression

Thal Level	No. Yes	No. No	P. Yes	P. No	Deviance
1	12	6	0.667	0.333	22.915
2	37	127	0.226	0.774	175.125
3	88	27	0.765	0.2348	125.348
Total	137	160	.461	.539	323.388

Table 8: Deviance Values for Thal Levels

Ca Level	No. Yes	No. No	P. Yes	P. No	Deviance
0	45	129	0.259	0.741	198.920
1	44	21	0.677	0.323	81.792
2	31	7	0.816	0.184	36.307
3	17	3	0.850	0.150	16.908
Total	137	160	.461	.539	333.927

Table 9: Deviance Values for Ca Levels

Thal = 2 and Ca Level	No. Yes	No. No	P. Yes	P. No	Deviance
0	13	102	0.113	0.887	81.151
1	12	17	0.414	0.586	39.336
2	7	7	0.500	0.500	19.408
3	5	1	0.833	0.167	5.407

Table 10: Deviance Values for Thal = 2 and Different Ca Values

Thal = 1/3 and Ca Level	No. Yes	No. No	P. Yes	P. No	Deviance
0	32	27	0.542	0.458	81.367
1	32	4	0.889	0.111	25.116
2	24	0	1.000	0.000	NaN
3	12	2	0.857	0.143	11.483

Table 11: Deviance Values for Thal = 1/3 and Different Ca Values