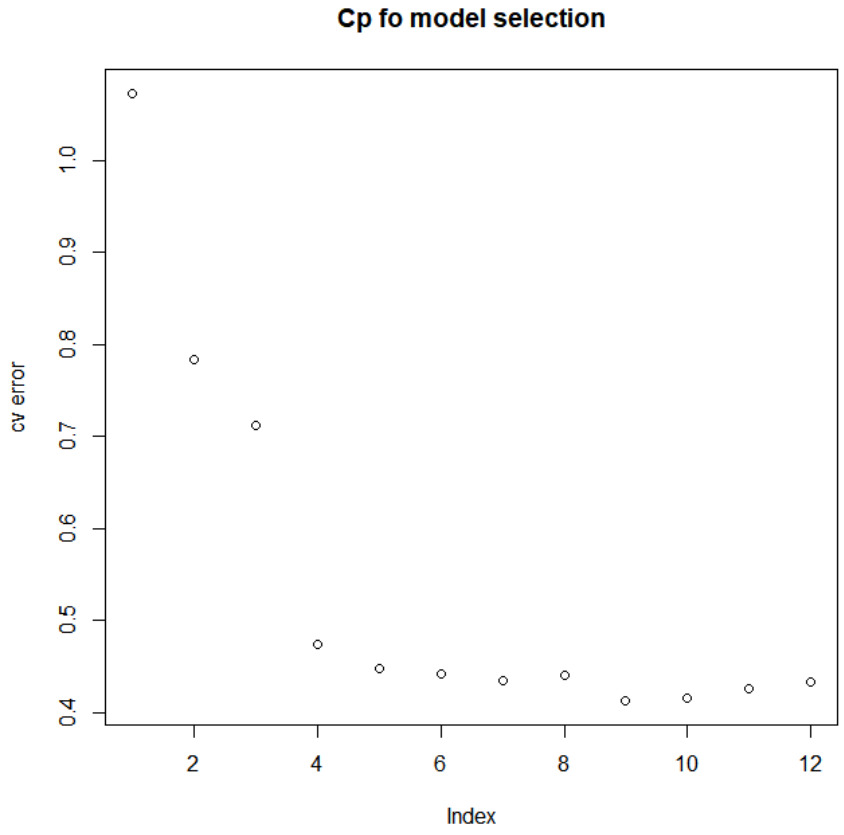
HW5 – Report

Rohith Reddy Kolla | rkolla@buffalo.edu

1. Considering the vehicles dataset, there's two columns for the class of the vehicle – one is the class digit and the other is the name of the class. The class digit column is removed and the data is fitted using rpart with a control of minsplit = 20 (for an appropriate-size tree) and cp = 0.

```
> names(vehicle)
 [1] "Comp"          "Circ"          "Dcirc"         "RR"
 [5] "PrAxisAR"      "MaxLAR"        "ScatterR"      "Elong"
 [9] "PrAxisRect"    "MaxLRect"      "SvarMajAxis"   "SvarMinAxis"
[13] "SradGyration"  "SkewMajAxis"   "SkewMinAxis"   "KurtMinAxis"
[17] "KurtMajAxis"   "Hratio"        "classdigit"    "class"
```

The graph below shows the cv error based on the complexity parameter. Cp of 0.072 at index 9 seems to be ideal.

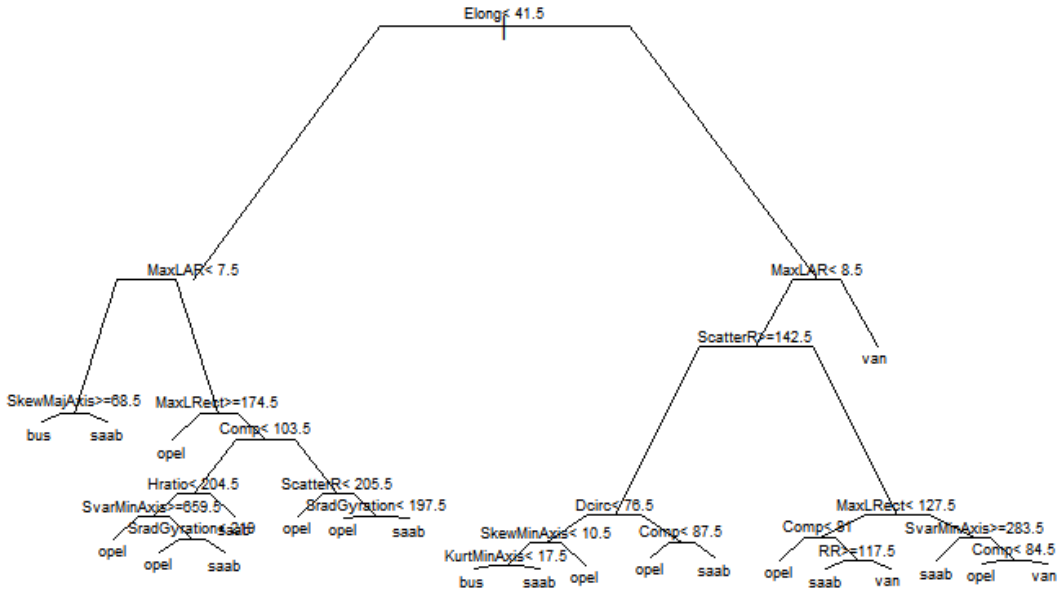



```
> fit$cptable[,4]
        1         2         3         4         5         6
1.0721154 0.7836538 0.7115385 0.4735577 0.4471154 0.4423077
        7         8         9        10        11        12
0.4350962 0.4399038 0.4134615 0.4158654 0.4254808 0.4326923
```

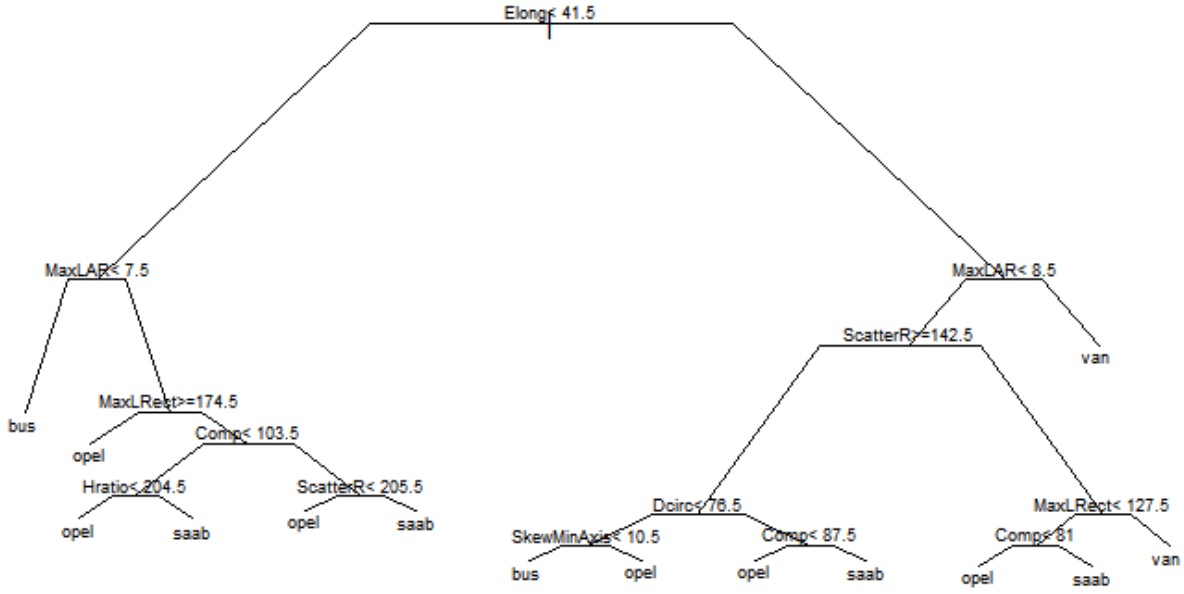
cv error as shown in the graph

The full classification tree generated from the fit and the pruned classification tree generated considering the ideal cp are shown below. The full tree is a bit too large especially compared to the neat and simplistic pruned tree.

Full Tree



Pruned Tree

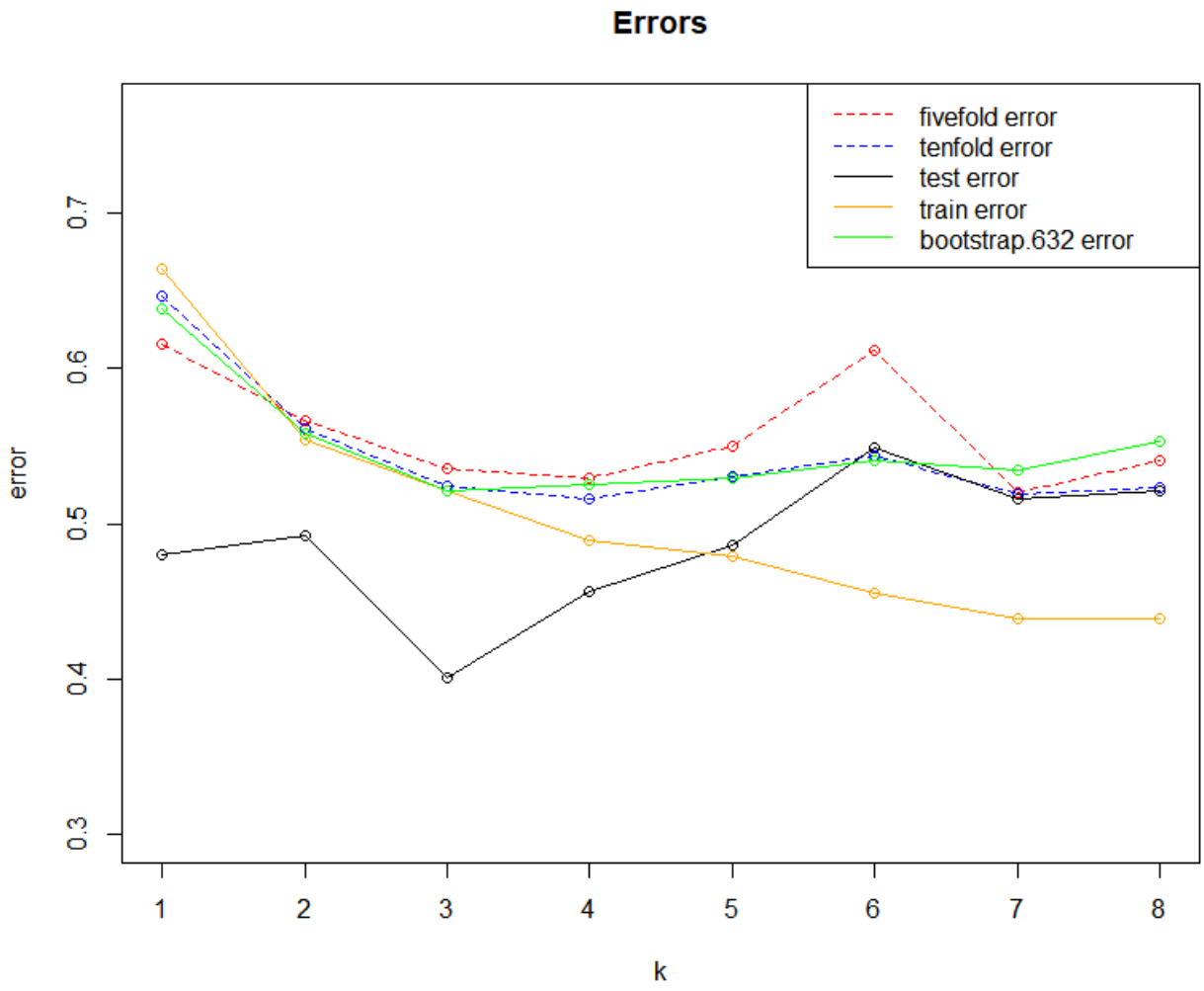


2. The prostate dataset in the ElemStatLearn package contains 97 instances with 10 variables. The variables are log cancer volume (lcavol), log prostate weight (lweight), age, log of the amount of benign prostatic hyperplasia(lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of Gleason scores 4 or 5 (pgg45), level of prostate-specific antigen (lpsa). Additionally, there is a variable train which we use to split the dataset into training and test sets.

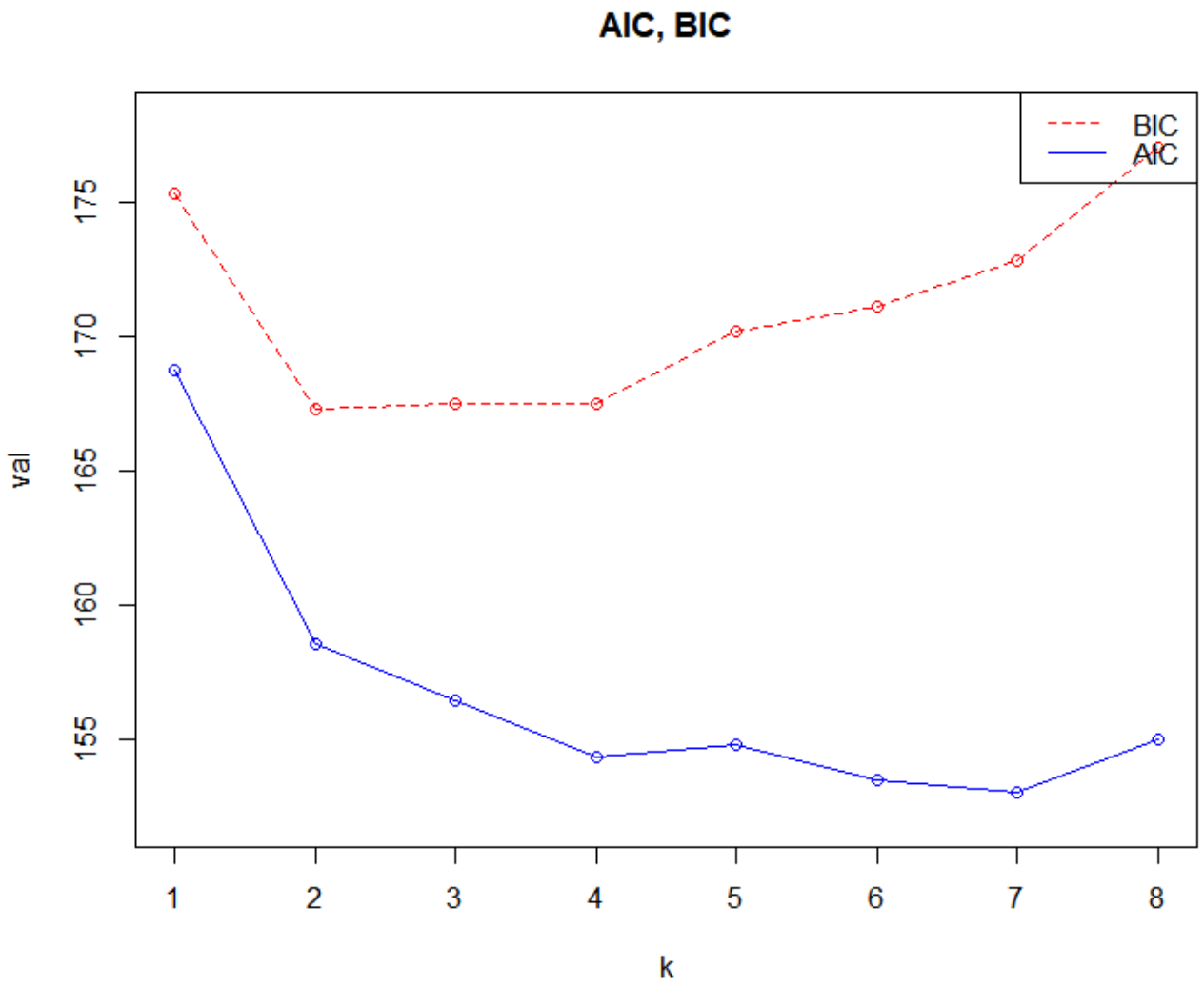
Applying best subset selection using the exhaustive method with regsubsets() gives the results shown below. lcavol seems to be the most important attribute followed by lweight whereas gleason is the least important.

		lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45
1	(1)	"*"	" "	" "	" "	" "	" "	" "	" "
2	(1)	"*"	"*"	" "	" "	" "	" "	" "	" "
3	(1)	"*"	"*"	" "	" "	"*"	" "	" "	" "
4	(1)	"*"	"*"	" "	"*"	"*"	" "	" "	" "
5	(1)	"*"	"*"	" "	"*"	"*"	" "	" "	"*"
6	(1)	"*"	"*"	" "	"*"	"*"	"*"	" "	"*"
7	(1)	"*"	"*"	"*"	"*"	"*"	"*"	" "	"*"
8	(1)	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"

The training error, testing error, fivefold cross validation error, tenfold cross validation error and bootstrap .632 estimates of prediction error are computed and plotted below based on the k value along with AIC and BIC.



```
> fivefold.error.store
[1] 0.6153974 0.5666190 0.5357967 0.5294412 0.5500207 0.6117516 0.5197335 0.5403121
> tenfold.error.store
[1] 0.6469070 0.5607572 0.5242319 0.5159559 0.5301621 0.5439952 0.5188209 0.5234649
> test.error.store
[1] 0.4797387 0.4924823 0.4005308 0.4563321 0.4859242 0.5485933 0.5165135 0.5212740
> train.error.store
[1] 0.6646057 0.5536096 0.5210112 0.4897760 0.4786485 0.4558176 0.4393627 0.4391998
> bootstrap.error.store
[1] 0.6383859 0.5586159 0.5210811 0.5251599 0.5294382 0.5411101 0.5347242 0.5528109
```



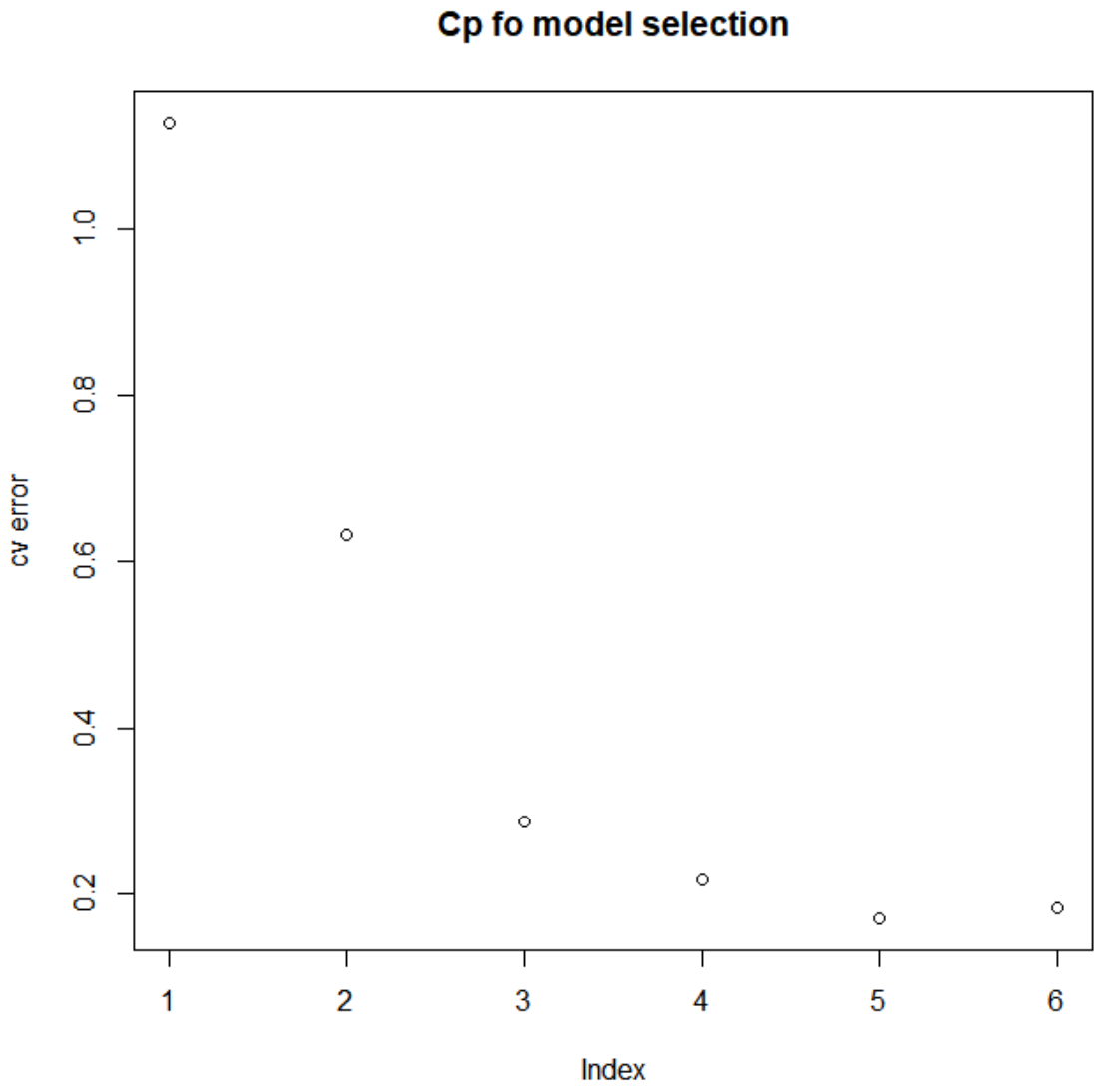

```
> aic.store
[1] 168.7642 158.5210 156.4548 154.3127 154.7729 153.4984 153.0350 155.0101
> bic.store
[1] 175.3782 167.3397 167.4783 167.5408 170.2058 171.1359 172.8772 177.0570
```

3.a) The wine data from the UCI machine learning repository has data containing 13 attributes of 178 wines belonging to 3 classes - class 1 are Barolo wines with 59 instances, class 2 are Grignolino wines with 71 instances and class 3 are Barbera wines with 48 instances.

The 13 attributes are - 1) Alcohol, 2) Malic acid, 3) Ash, 4) Alcalinity of ash, 5) Magnesium, 6) Total phenols, 7) Flavanoids, 8) Nonflavanoid phenols, 9) Proanthocyanins, 10)Color intensity, 11)Hue, 12)OD280/OD315 of diluted wines, 13)Proline

The data is split into a training set containing 138 instances and a test set containing 40 instances.

The training data is fitted using rpart with a control of minsplit = 3 (since the dataset is quite small) and cp = 0. The graph below shows the cv error based on the complexity parameter. cp of 0.0114 at index 5 seems to be ideal.



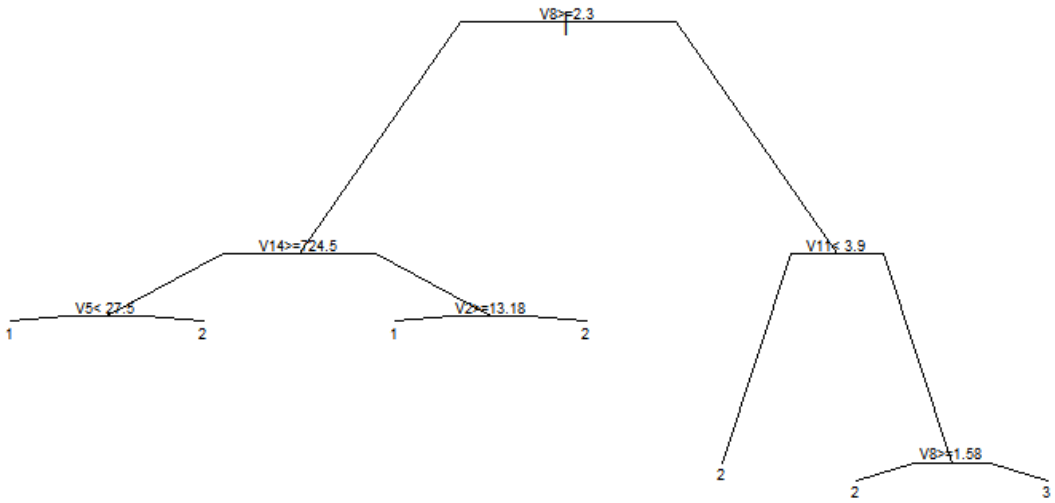
1	2	3	4	5	6
1.1264368	0.6321839	0.2873563	0.2183908	0.1724138	0.1839080

cv error as shown in the graph

The full tree and pruned tree generated considering the ideal cp are shown in the plots below along with their accuracy on the test set.

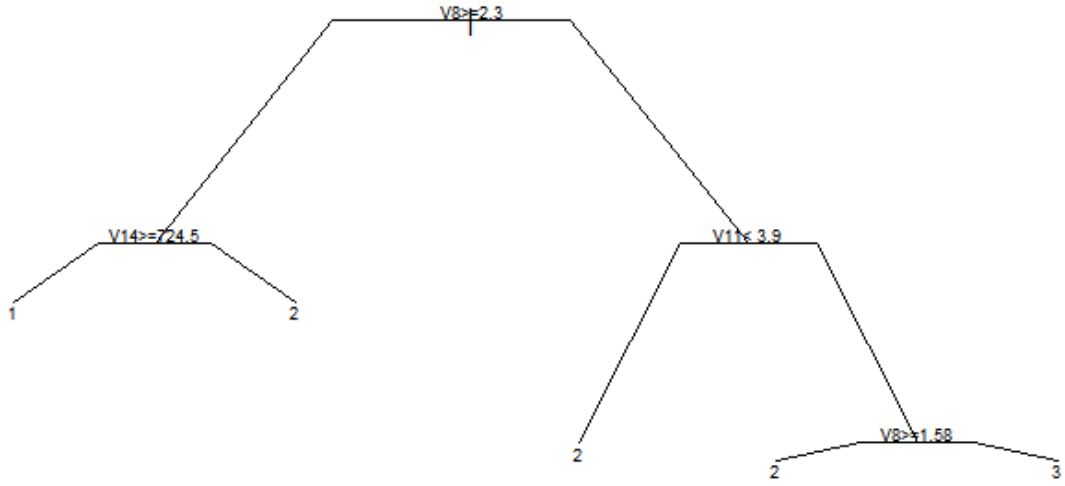
```
Accuracy : 0.9
```

Full Tree




```
Accuracy : 0.925
```

Pruned Tree



The pruned tree manages to provide better prediction accuracy with this particular randomly sampled test set.

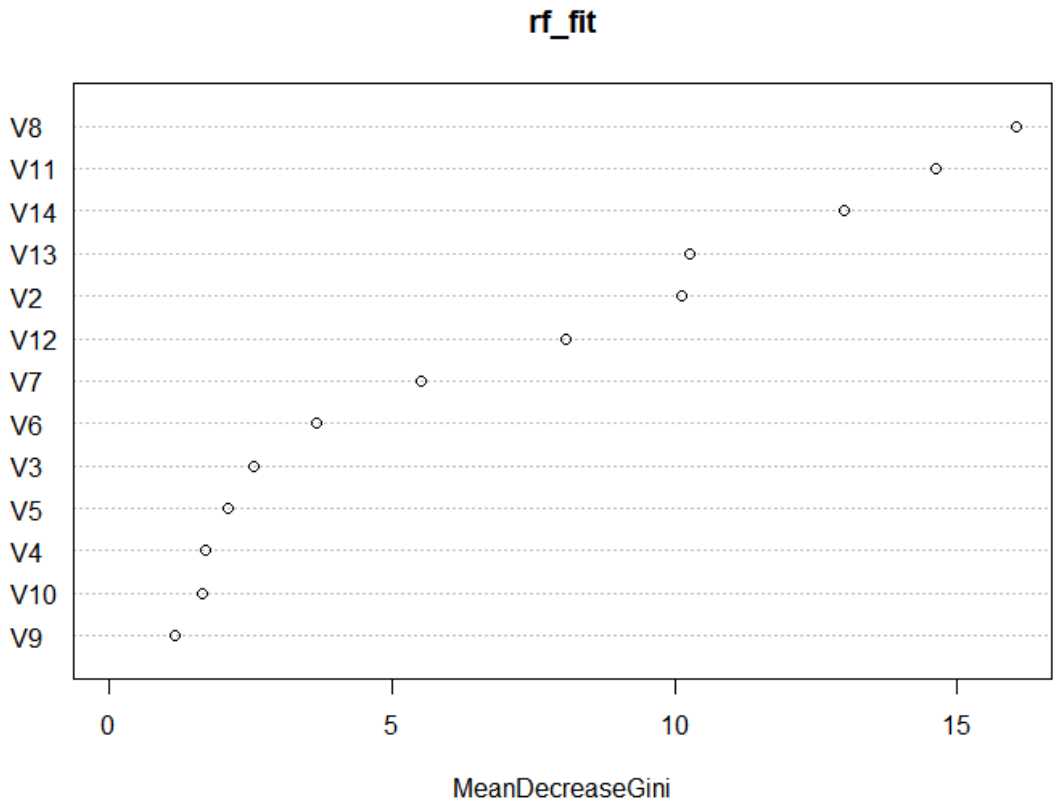
The ensemble technique used is random forest which is applied by fitting the training data using randomForest() with n.tree = 1000. This method is able to provide better classification accuracy on the test set than the full tree and pruned tree shown above.

```
          Reference
Prediction  1  2  3
         1  9  0  0
         2  0 19  0
         3  0  1 11

Overall Statistics

               Accuracy : 0.975
```

The varImpPlot shown below provides an idea as to what the most important variables are with V8 – Flavanoids being the most important and V9 – Non Flavanoid phenols being the least important.



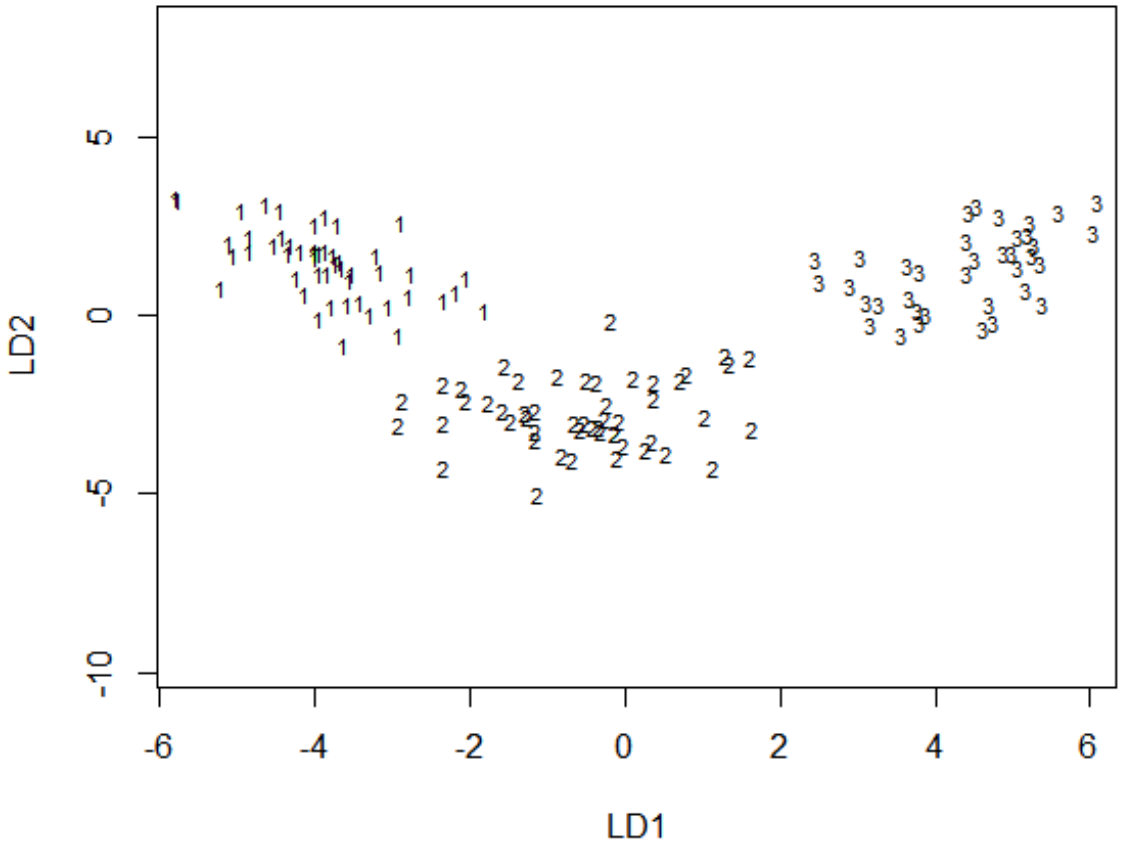
b) An LDA model is constructed on the training data set and its predictions of the test set along with the accuracy are shown below.

```
          Reference
Prediction  1  2  3
         1  9  1  0
         2  0 19  1
         3  0  0 10

Overall Statistics

               Accuracy : 0.95
```

The model is able to predict the test set with better accuracy than the full and pruned tree but worse than the random forest method.



4) The covertype dataset is a relatively large dataset comprising of 581012 instances and 55 attributes. The attribute details are shown below (from UCI machine learning repository).

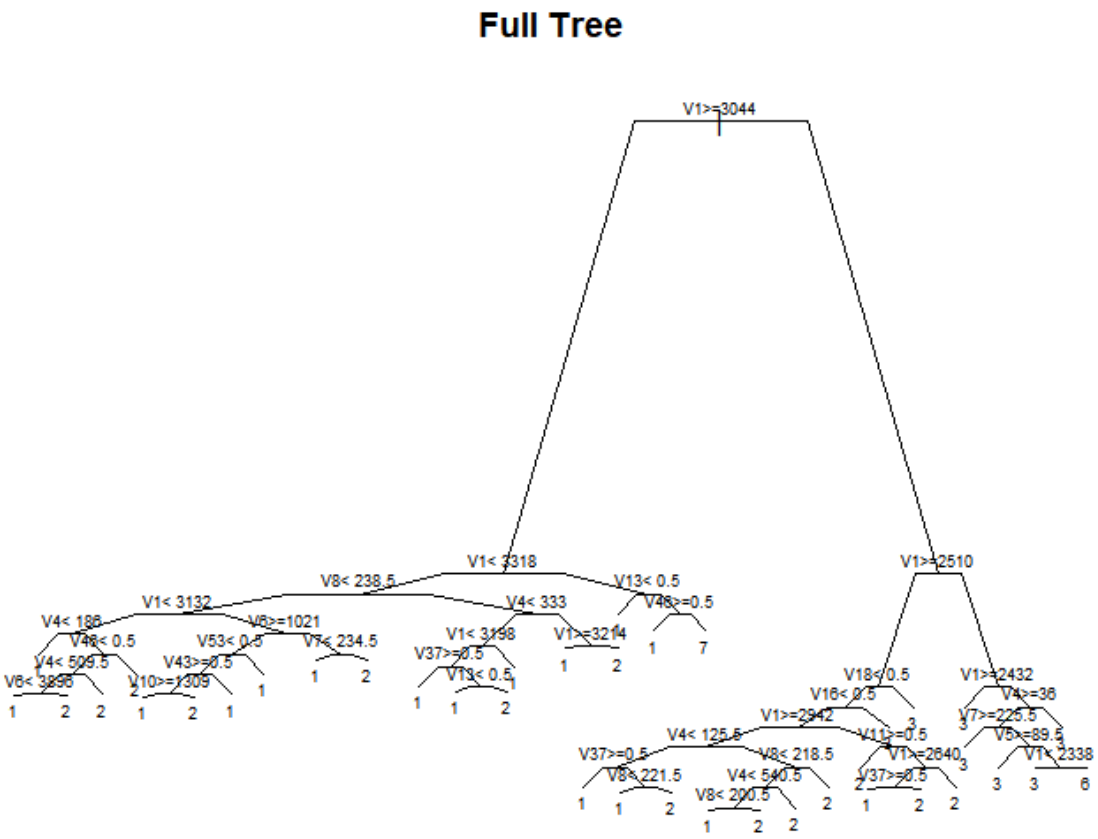
Given is the attribute name, attribute type, the measurement unit and a brief description. The forest cover type is the classification problem. The order of this listing corresponds to the order of numerals along the rows of the database.

Name / Data Type / Measurement / Description

Elevation / quantitative /meters / Elevation in meters
Aspect / quantitative / azimuth / Aspect in degrees azimuth
Slope / quantitative / degrees / Slope in degrees
Horizontal_Distance_To_Hydrology / quantitative / meters / Horz Dist to nearest surface water features
Vertical_Distance_To_Hydrology / quantitative / meters / Vert Dist to nearest surface water features
Horizontal_Distance_To_Roadways / quantitative / meters / Horz Dist to nearest roadway
Hillshade_9am / quantitative / 0 to 255 index / Hillshade index at 9am, summer solstice
Hillshade_Noon / quantitative / 0 to 255 index / Hillshade index at noon, summer soltice
Hillshade_3pm / quantitative / 0 to 255 index / Hillshade index at 3pm, summer solstice
Horizontal_Distance_To_Fire_Points / quantitative / meters / Horz Dist to nearest wildfire ignition points
Wilderness_Area (4 binary columns) / qualitative / 0 (absence) or 1 (presence) / Wilderness area designation
Soil_Type (40 binary columns) / qualitative / 0 (absence) or 1 (presence) / Soil Type designation
Cover_Type (7 types) / integer / 1 to 7 / Forest Cover Type designation

The data is split into a test set containing 200000 instances and a training set containing 381012 instances.

The training set is fitted to a classification tree using rpart with control minsplit = 7000 (due to the extremely large size of the dataset) and cp = 0. The resulting tree is shown below.



While the resulting tree is far from simplistic, it is able to provide a prediction accuracy of >70% on both the test and training set. This could most certainly improved by fine tuning how the data is split and the minsplit value.

```
          Reference
Prediction     1     2     3     4     5     6     7
         1 50476 17017     0     0   170    11  3790
         2 21303 77615  1261     1  2715  1598    83
         3    72  2486 10547   956   333  3735    30
         4     0     0     0     0     0     0     0
         5     0     0     0     0     0     0     0
         6     0    76   661    12     0   671     0
         7  1195   117     0     0     0     0  3069

Overall Statistics

               Accuracy : 0.7119
```

Test Set

```
          Reference
Prediction     1     2     3     4     5     6     7
         1 96181 32659     0     0   333    18  7485
         2 40173 148384  2315     2  5301  2963   189
         3   121   4546 19762  1747   641  7112    48
         4     0     0     0     0     0     0     0
         5     0     0     0     0     0     0     0
         6     0   138  1208    29     0  1259     0
         7  2319   263     0     0     0     0  5816

Overall Statistics

               Accuracy : 0.7123
```

Training Set