Data Mining
Assignment Solution 2

Vivek Patani

February 26, 2016

Please read readme.md for code execution directions, I have tried to maintain a generic structure for each question in coding terms.

1. Question 1 Solution

Download data set Iris and answer the following questions:

1.1 Section 1

Calculate the average value and standard deviation for each of the four features.

Average for Petal Width: 1.2
Variance for Petal Width: 0.58
Standard Deviation for Petal Width: 0.762

Average for Petal Length: 3.76
Variance for Petal Length: 3.09
Standard Deviation for Petal Length: 1.758

Average for Sepal Length: 5.84
Variance for Sepal Length: 0.68
Standard Deviation for Sepal Length: 0.82

Average for Sepal Width: 3.05
Variance for Sepal Width: 0.19
Standard Deviation for Sepal Width: 0.436

Code Located in q1/q1.py

1.2 Section 2

Repeat the previous step but separately for each type of flower.

Average for Sepal Width & Iris Setosa: 3.42
Variance for Sepal Width & Iris Setosa: 0.14
Standard Deviation for Sepal Width & Iris Setosa: 0.37416573867739417

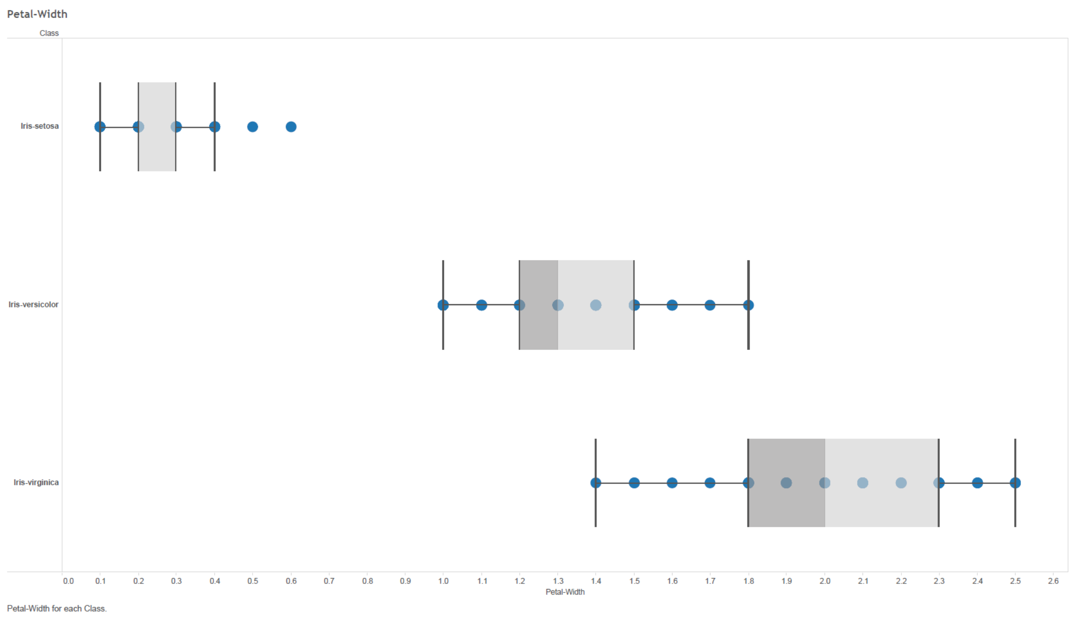
Average for Sepal Length & Iris Setosa: 5.01

```
Variance for Sepal Length & Iris Setosa: 0.12
Standard Deviation for Sepal Length & Iris Setosa: 0.34641016151377546

Average for Petal Width & Iris Setosa: 0.24
Variance for Petal Width & Iris Setosa: 0.01
Standard Deviation for Petal Width & Iris Setosa: 0.1

Average for Petal Length & Iris Setosa: 1.46
Variance for Petal Length & Iris Setosa: 0.03
Standard Deviation for Petal Length & Iris Setosa: 0.17320508075688773

Average for Sepal Width & Iris Versicolor: 2.77
Variance for Sepal Width & Iris Versicolor: 0.1
Standard Deviation for Sepal Width & Iris Versicolor: 0.31622776601683794

Average for Sepal Length & Iris Versicolor: 5.94
Variance for Sepal Length & Iris Versicolor: 0.26
Standard Deviation for Sepal Length & Iris Versicolor: 0.5099019513592785

Average for Petal Width & Iris Versicolor: 1.33
Variance for Petal Width & Iris Versicolor: 0.04
Standard Deviation for Petal Width & Iris Versicolor: 0.2

Average for Petal Length & Iris Versicolor: 4.26
Variance for Petal Length & Iris Versicolor: 0.22
Standard Deviation for Petal Length & Iris Versicolor: 0.469041575982343

Average for Sepal Width & Iris Virginica: 2.97
Variance for Sepal Width & Iris Virginica: 0.1
Standard Deviation for Sepal Width & Iris Virginica: 0.31622776601683794

Average for Sepal Length & Iris Virginica: 6.59
Variance for Sepal Length & Iris Virginica: 0.4
Standard Deviation for Sepal Length & Iris Virginica: 0.6324555320336759

Average for Petal Width & Iris Virginica: 2.03
Variance for Petal Width & Iris Virginica: 0.07
Standard Deviation for Petal Width & Iris Virginica: 0.2645751311064591

Average for Petal Length & Iris Virginica: 5.55
```

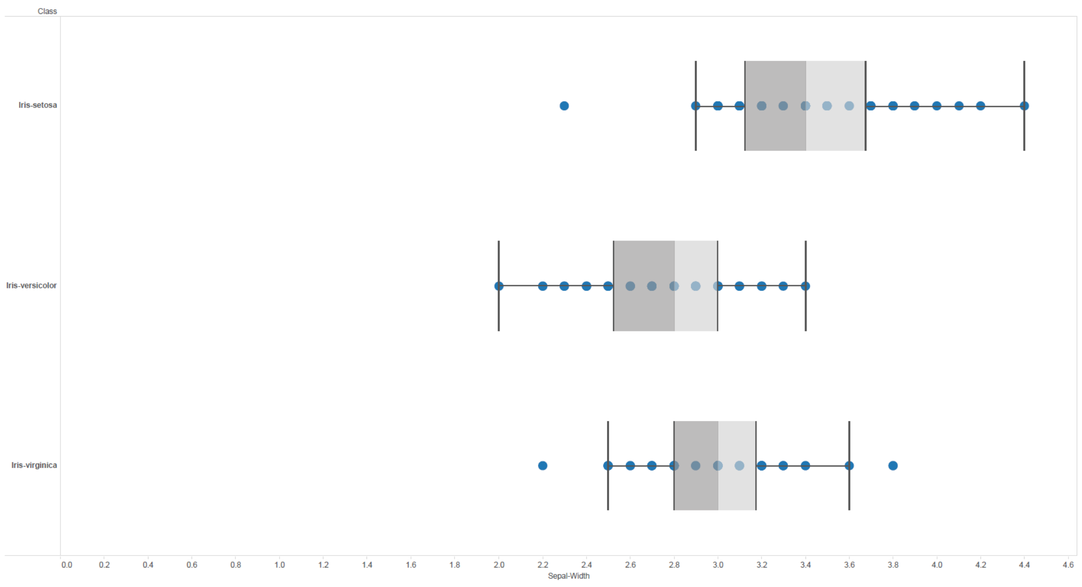
Variance for Petal Length & Iris Virginica: 0.3
Standard Deviation for Petal Length & Iris Virginica: 0.5477225575051661

1.3 Section 3

Draw four box plots, one for each feature, such that each figure shows three boxes, one for each type of flower. Properly label your figures and axes in all box plots. Make sure that the box plots look professional and appear in high resolution. Experiment with thickness of lines, font styles/sizes, etc. and describe what you tried and what looked the most professional.

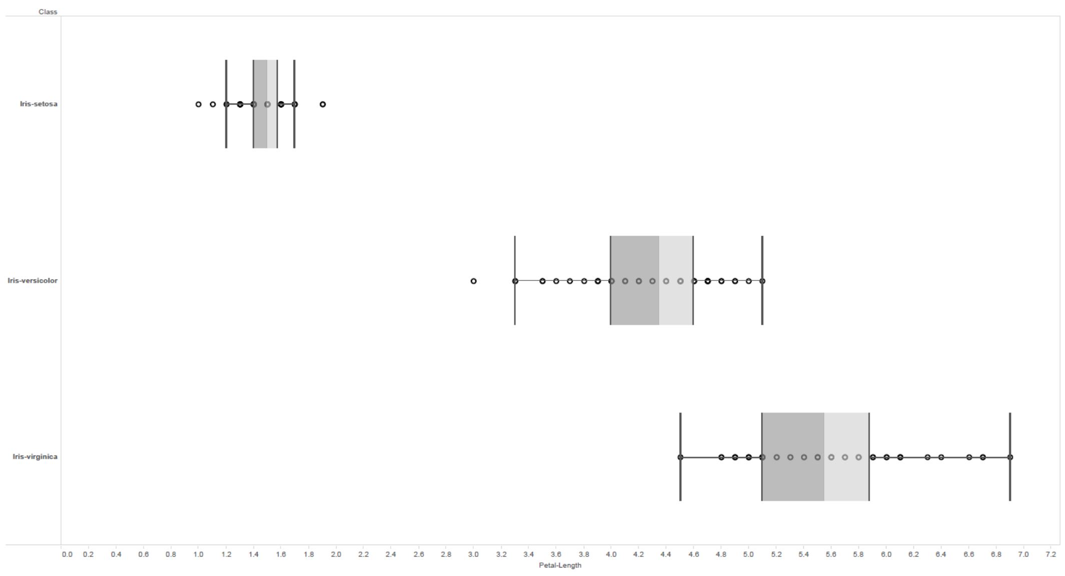


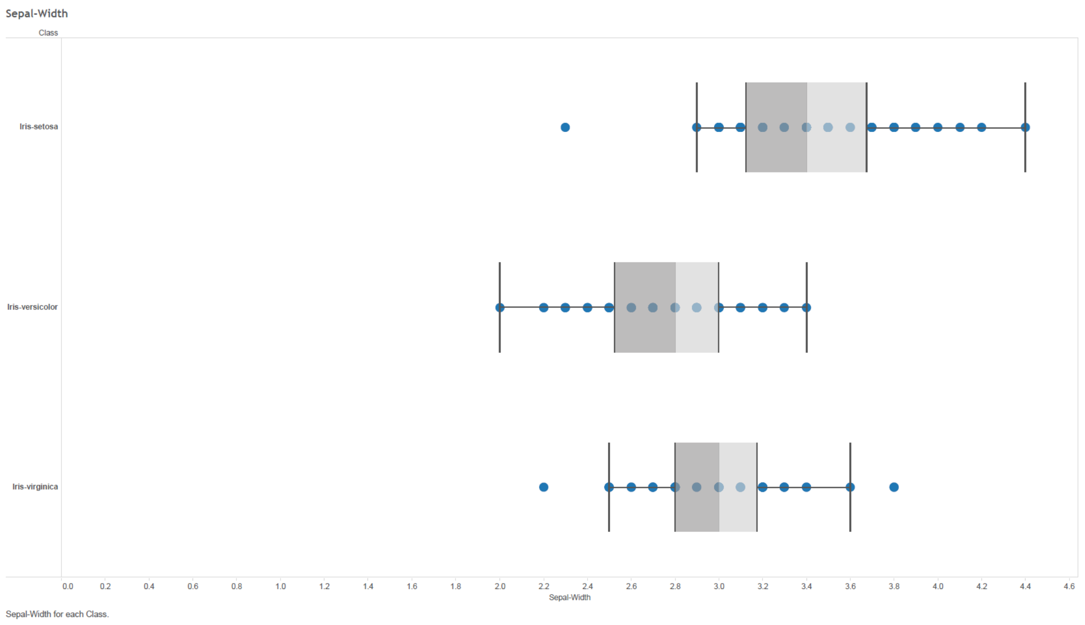
Sepal-Width



Sepal-Width for each Class.

Petal-Length





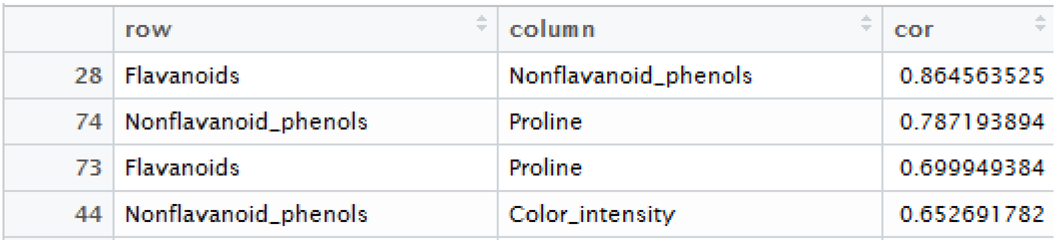
2. Question 2 Solution

Download data set Wine and answer the following questions

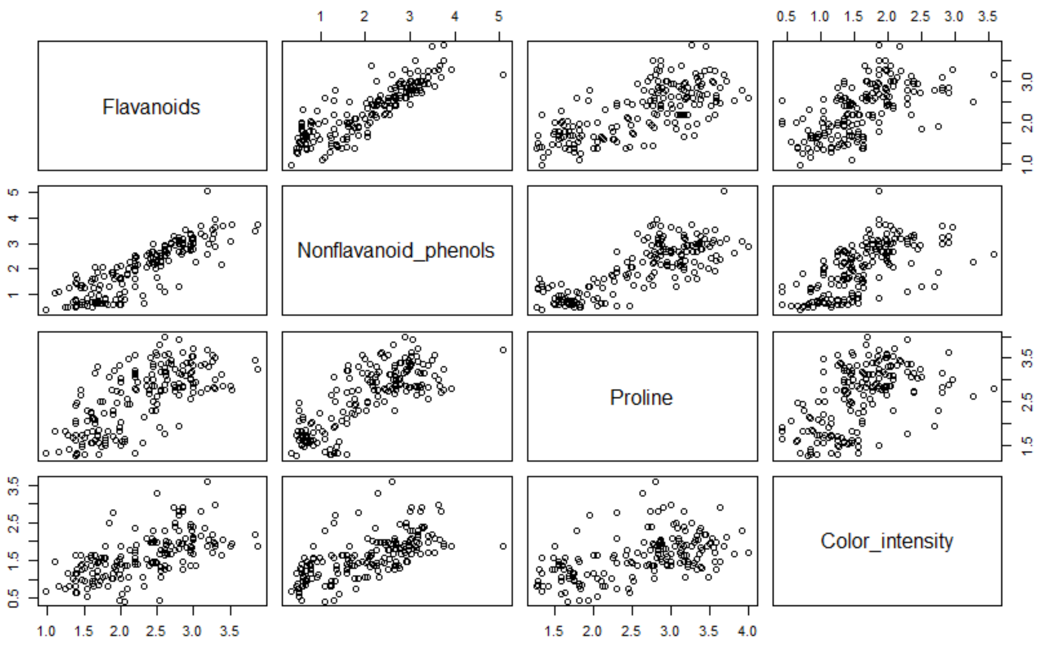
2.1 Section 1

Provide pairwise scatter plots for four most correlated and four least correlated pairs of features, using Pearson's correlation coefficient. Label all axes in all your plots and select fonts of appropriate style and size. Experiment with different ways to plot these scatter plots and choose the one most visually appealing and most professionally looking.

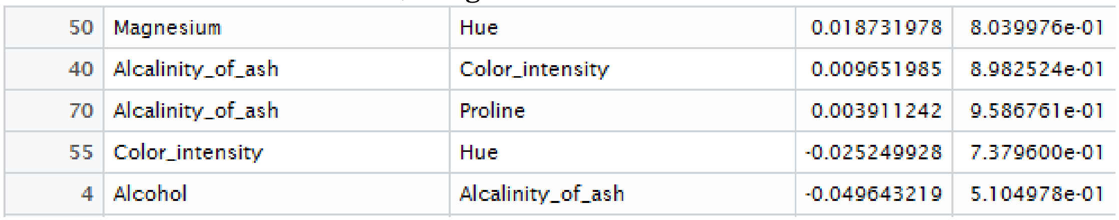
The data is preprocessed through a Python Script and R for visualisations. Actually the python script is redundant since everything was done in R later on in R itself because Math Plot lib wasn't working. It was easier through R. The four most correlated pairs were:
1.) Flavanoids - Non Flavanoid Phenols: 0.86
2.) Non Flavanoid Phenols - Proline: 0.78
3.) Flavanoids - Proline: 0.69
4.) Non flavanoid Phenols - Color Intensity: 0.65



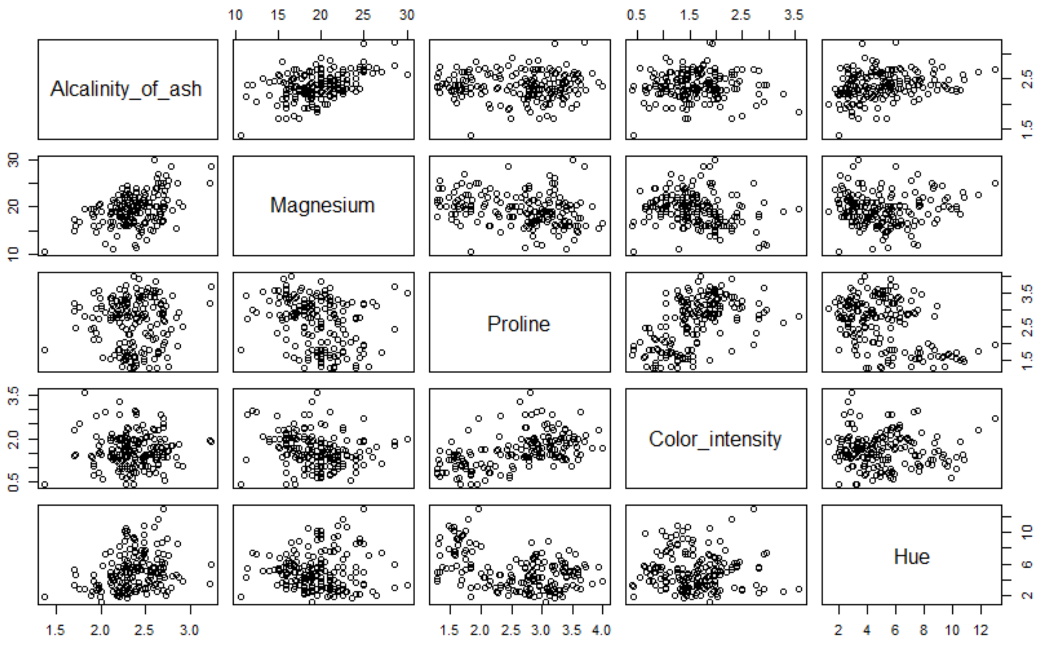
	row	column	cor
28	Flavanoids	Nonflavanoid_phenols	0.864563525
74	Nonflavanoid_phenols	Proline	0.787193894
73	Flavanoids	Proline	0.699949384
44	Nonflavanoid_phenols	Color_intensity	0.652691782



The four least correlated pairs were:
1.) Alcanility of Ash - Proline: 0.03
2.) Alcanility of Ash - Color Intensity: 0.09
3.) Color Intensity - Hue: -0.02
4.) Magnesium - Hue: 0.18



50	Magnesium	Hue	0.018731978	8.039976e-01
40	Alcalinity_of_ash	Color_intensity	0.009651985	8.982524e-01
70	Alcalinity_of_ash	Proline	0.003911242	9.586761e-01
55	Color_intensity	Hue	-0.025249928	7.379600e-01
4	Alcohol	Alcalinity_of_ash	-0.049643219	5.104978e-01



Code in q2/a

2.2 Section 2

Use Euclidean distance to find the closest example to every example available in the data set (exclude the class variable). Calculate the percentage of points whose closest neighbours have the same class label (for data set as a whole and also for each class).

The algorithm is straight forward and following a Brute Force method. The algorithm compares the data points to every other points in the set. This is then, checked upon each condition by keeping a count. The count is then incremented checking based on each condition whether if it lies in the same class at all or not. We find the minimum

Formula (Overall) = Total Matching / Total No of Computation

Formula (Each) = Total Of that Class / Total No of Computation for that class total.

Statistics
Complete Dataset: 76.8 %

Class 1: 88.1 %

Class 2: 76 %

Class 3: 64.5 %

Code in q2/b-c

2.3 Section 3

Repeat the previous step but after the data set is normalized using first 0-1 normalization and then z-score normalization. Investigate the reasons for discrepancy and provide evidence to support every one of your claims. Provide the code you used for normalizing and visualizing the data.

0 - 1 Normalization

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

```
#0-1 Normalisation
normalise  <- function(x) {
    final_data1 <- ((x - min(x))/(max(x) -min(x)))
  return (final_data)
  }
```

Statistics
Full Set Accuracy: 95%
Class 1 Set Accuracy: 100%
Class 2 Accuracy: 87.8%
Class 3 Accuracy: 100%
Z Score Normalization

$$x' := (x - x_{\min})/(x_{\max} - x_{\min})$$

```
#Z Score Normalisation
z_score<-function(x){
    scale(x)
    final_data <- ((x - mean(x)) / sd(x))
    return (final_data)
  }
}
```

Statistics
Full Set Accuracy: 95%
Class 1 Set Accuracy: 99.9%
Class 2 Accuracy: 88.3%
Class 3 Accuracy: 100%

The idea of **Normalization** comes to data mining when we need to actually render the data in a specific range. It is used to make more sense out of the data, say when we have to actually data that is non - linear and to bring them to a linear state to form a relationship is one instance where we can use normalisation. There are a few techniques used for normalisation such min-max norm, 0-1 normalisation, Decimal Scaling etc. The drawback of normalization is that the data comes really close to each other and may sometimes cross boundaries. So if in a dataset where points have a few sparse data points, while normalising they may come really close and would give you an unclean result. Normalisation should be performed only when required.

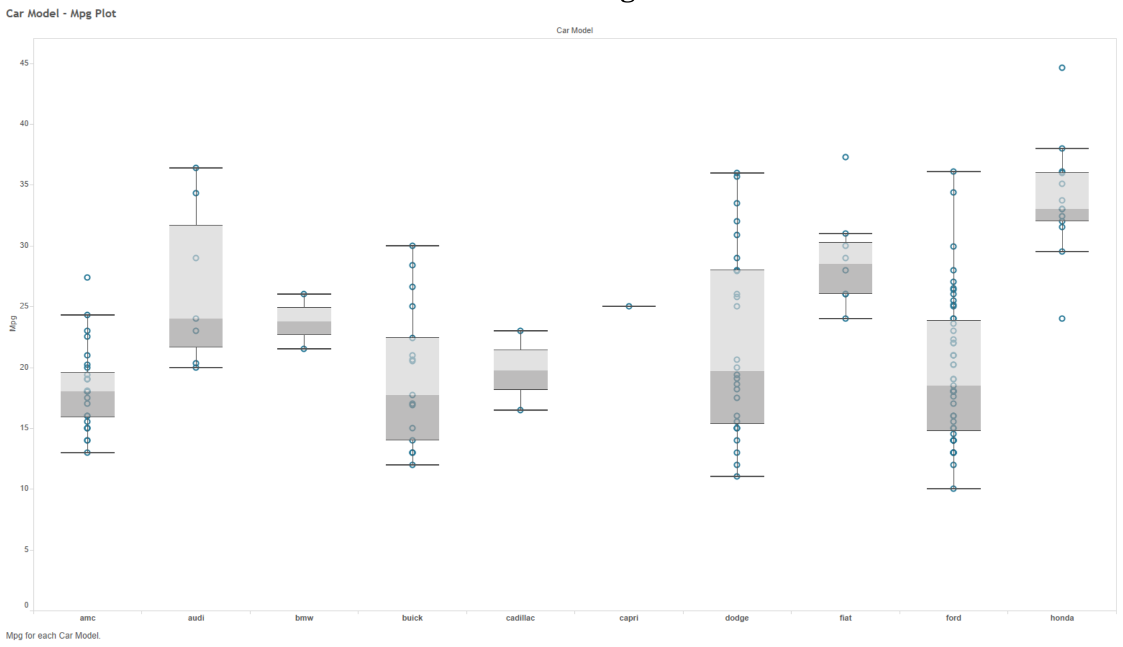
3. Question 3 Solution

Data exploration is often the first step in many data analysis tasks. Visualizing relationships between features as well as between features and the target variable(s), for example, can be exploited to design a good model or to understand why a particular model works. There are many software packages developed to make this step easier. In this question you will experiment with Tableau.

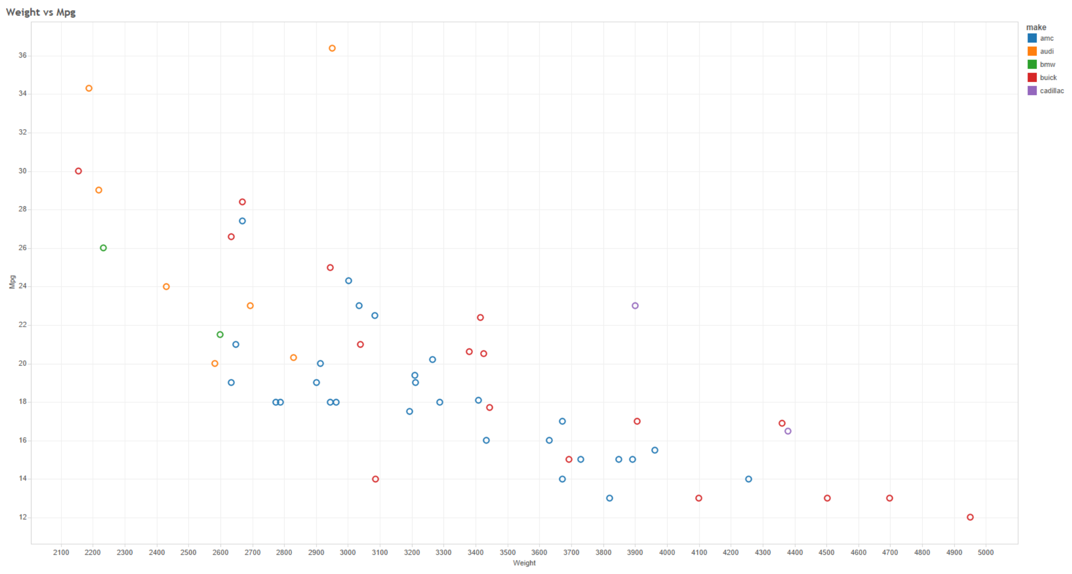
3.1 Section 1

Download and study the Auto MPG data set from the UCI Machine Learning Repository. Import the data set into Tableau. Create a new feature make (Honda, Toyota, . . .) that contains the make of the automobile (extract this feature automatically from other features through Tableau) and in a single figure generate box plots of mpg for 10 makes of your choice. Then, for 5 makes of your choice create scatter plots of weight versus mpg. Include all figures in your submission and comment on what you observe.

This is a Box Plot for 10 makes.
The data here tells us that Honda gives a good MPG and cars such as Buic, Dodge and Ford provide a wide range of MPG's which tells us that they must be producing a wide range of cars and probably a good amount of models ranging from basic to high end.



This is a Scatter Plot for 5 makes.
This plot describes a rather negative correlation between Weight and MPG. This is also true mostly in real life and which is quite a logical thing. The legends display the 5 makes for which we are drawing conclusions.

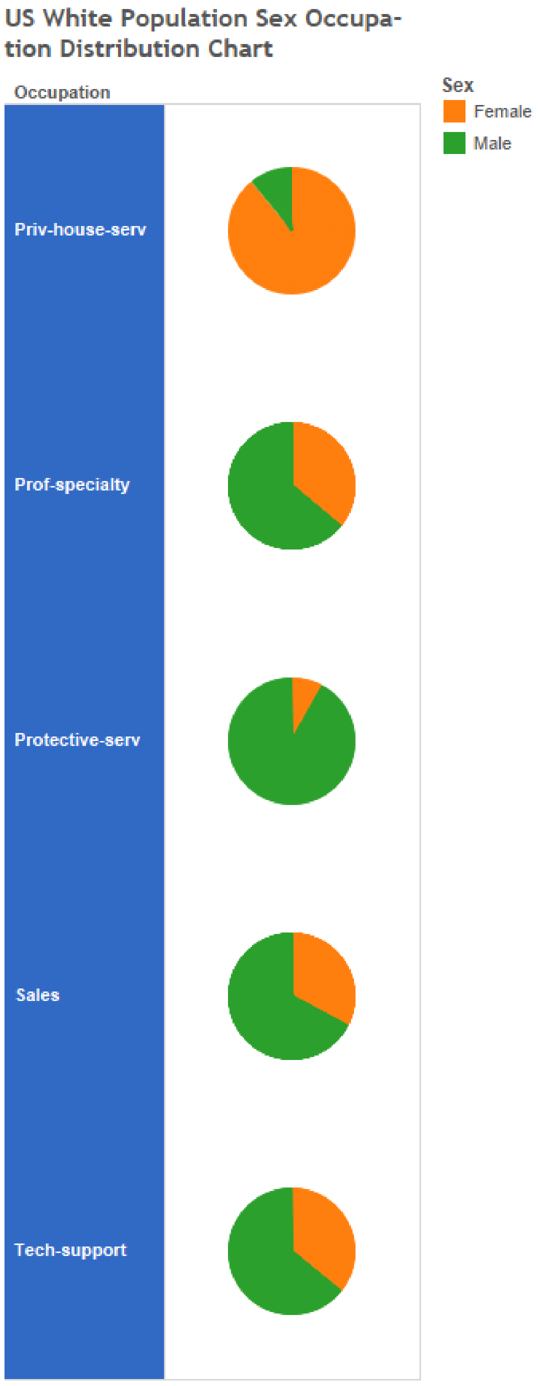


Weight vs. Mpg. Color shows details about make.

3.2 Section 2

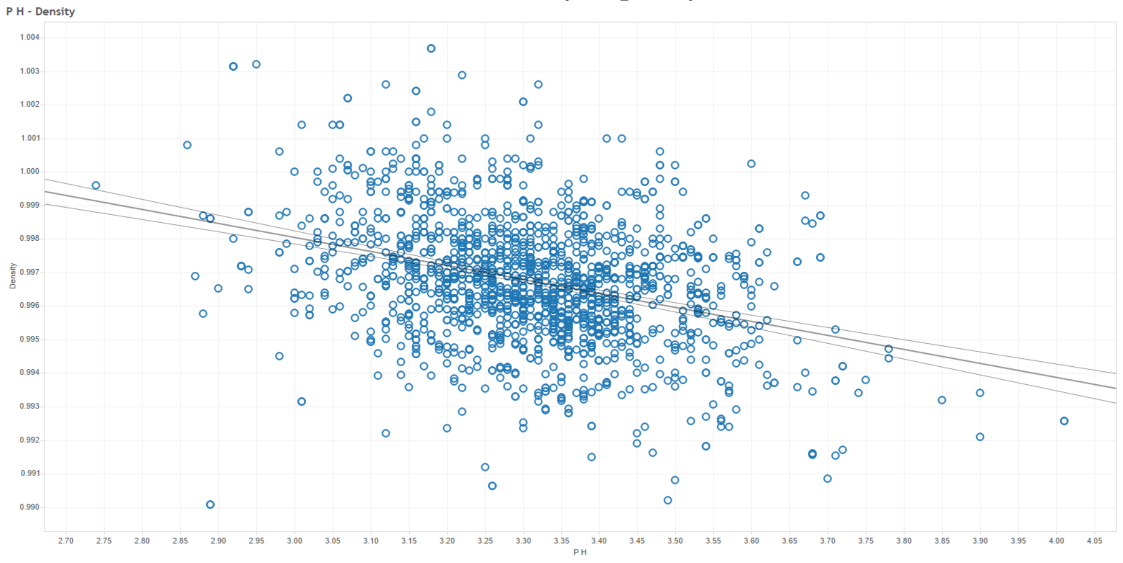
Pick 3 data sets of your choice from UCI Machine Learning Repository. Visualize each the data set in meaningful ways that show hidden patterns. Experiment with colors, size, shapes, filters,groups and sets. Feel free to experiment with other advanced functionalities of Tableau.

The data set used here is the adult dataset from the UCI ML repo. (Link: http://mlr.cs.umass.edu/ml/datasets/Adult)
We observe the distribution of sex among specific working classes and a few other filters are applied. We see that while private housing service has more females others are more dominated by males. This tells us that we should encourage more females in the other departments to create equality in each work domain.



Sex (color) broken down by Occupation. The data is filtered on Hours-per-Week, Race and native-country. The Hours-per-Week filter keeps non-Null values only. The Race filter keeps White. The native-country filter keeps United-States. The view is filtered on Occupation, which keeps Priv-house-serv, Prof-specialty, Protective-serv, Sales and Tech-support.

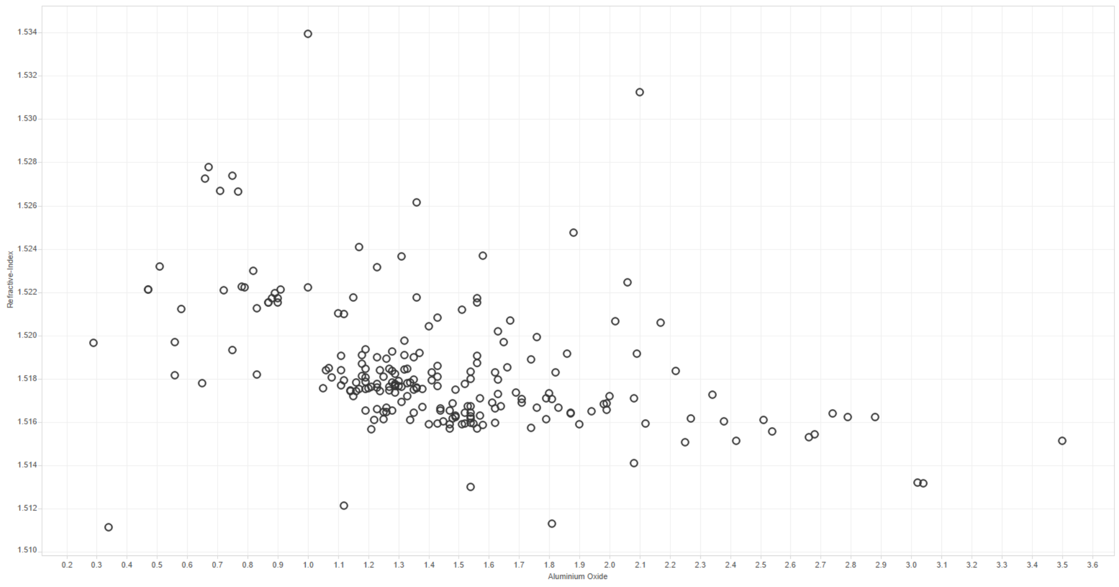
This data is the wine dataset taken from UCI ML Repo.
(Link: http://archive.ics.uci.edu/ml/datasets/Wine)
The PH and density a rather negative correlation among them. But the interesting observation is that most of the values are pretty closely clustered and tells us that the ratio of PH and density is pretty much constant.



P H vs. Density. We see that there is a partial negative correlation but the majority of the values cluster around forming the mode of the data. The trend lines depict the relation which is not completely nagetive but upto a large extent. In order to make the wine likeable an it should not be too much towards the neutral side but also not too dense otherwise it will loose its finesse.

This is the Glass ID Dataset
(Link:https://archive.ics.uci.edu/ml/datasets/Glass+Identification)
The scatter plot tells us that the data is pretty sparse and we can see that there is no correlation between or rather very less amount of correlation. The mean value lies about RI:1.5 and Aluminium Oxide:1.45.

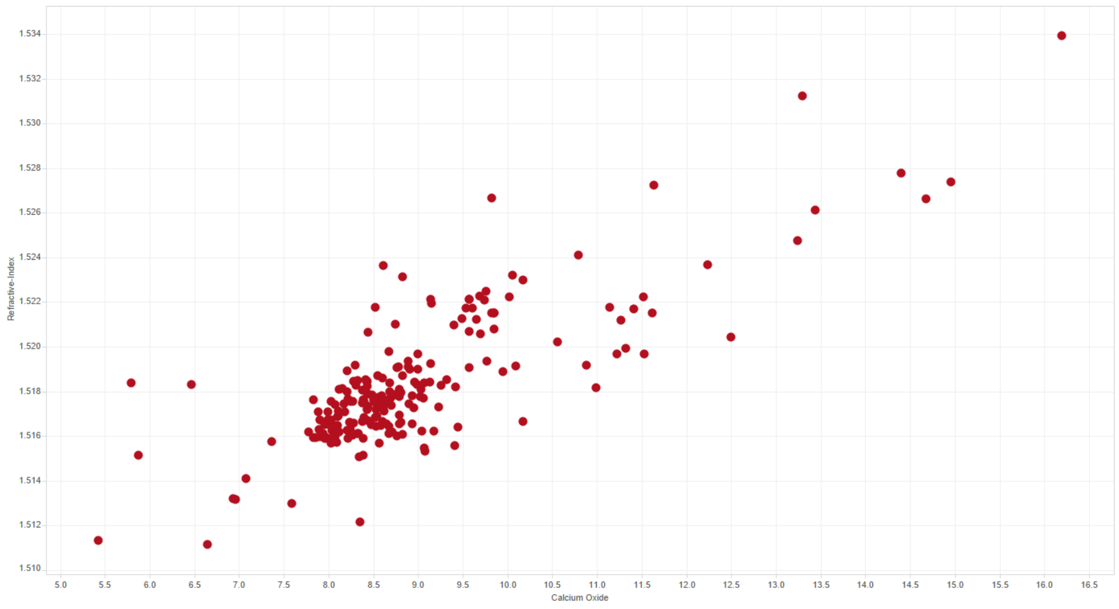
Aluminium Oxide vs. Refractive Index



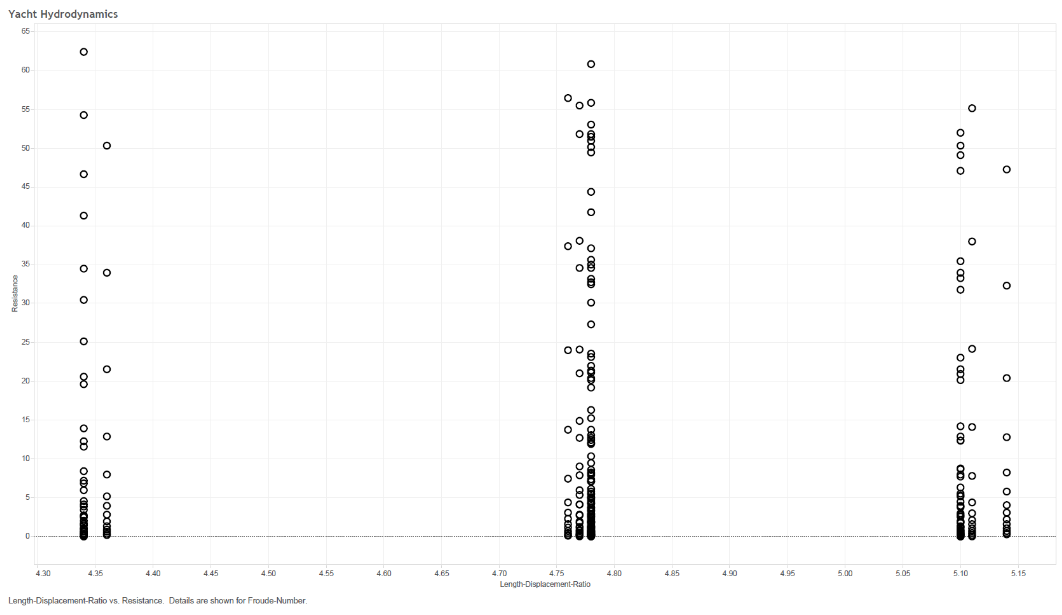
Aluminium Oxide vs. Refractive-Index.

This is the Glass ID Dataset (Link:https://archive.ics.uci.edu/ml/datasets/Glass+Identification)
The Graph tells us that the data is pretty correlated and in a positive way, so the relation is that the data is RI increases with increase in the Caclcium Oxide Content and but is limited to a certain extent.

Refractive Index - Calcium Oxide



The graph displays a positive correlation between Refractive Index of the lens and Calcium Oxide.



3.3 Section 3

Tell us about your experience with Tableau. What did you learn? What did you like/dislike about Tableau?

Good things:
1.) The loading of data is pretty easy. Also the support for various datasets is wide.
2.) The filter application is great, since it gives you all the unique values to select from all the attribute values. Elimination and Error Handling of Null values is also taken care of.
3.) Produces neat and clean visualisations with relevant legends, axis, title and caption.

Bad Things:
1.) JSON direct import is not available, need an API. I believe JSON to be well established and well structured to be included.
2.) Well it is fairly good for small datasets but takes a lot of while to import large datasets.
3.) Data Transformation is fairly limited because of the ability to manipulate.

4. Question 4 Solution

Implementing classification trees and evaluating their accuracy

4.1 Section 1

Implement the greedy algorithm that learns a classification tree given a data set. Assume that all features are numerical and properly find the best threshold for each split. Use Gini and information gain, as specified by user, to decide on the best attribute to split in every step. Stop growing the tree when all examples in a node belong to the same class or the remaining examples contain identical features.

Algorithm

1.) The primary step is to clean up the data by putting the class variable to the right hand side column (making the class the last attribute, since I started with Iris and then have done that for datasets onwards). (code: q4-data formatter.py) - Using pickle to pull and push data since it stores data as is.
2.) Defined a class Node (which contains a simple class to store the node structure). (node.py)
3.) Then we need to divide the dataset by calculating Entropy or Gini Impurity, based on that attribute we split the data. For initiation we use the 0th attribute. (Code: divide.py and impurity.py)
4.) Then we build the tree using the division based on the attribute selection and store everything on the node.(buildtree.py)
5.) Print tree is used to generate a tree picture.(printtree.py)
6.) There is a main.py script which runs the complete script (This has parts of Qustion 5 running together, script has comments on running the 4th only).
References for this: http://kldavenport.com/pure-python-decision-trees/ - This is my basic tutorial what I started off with.

4.2 Section 2

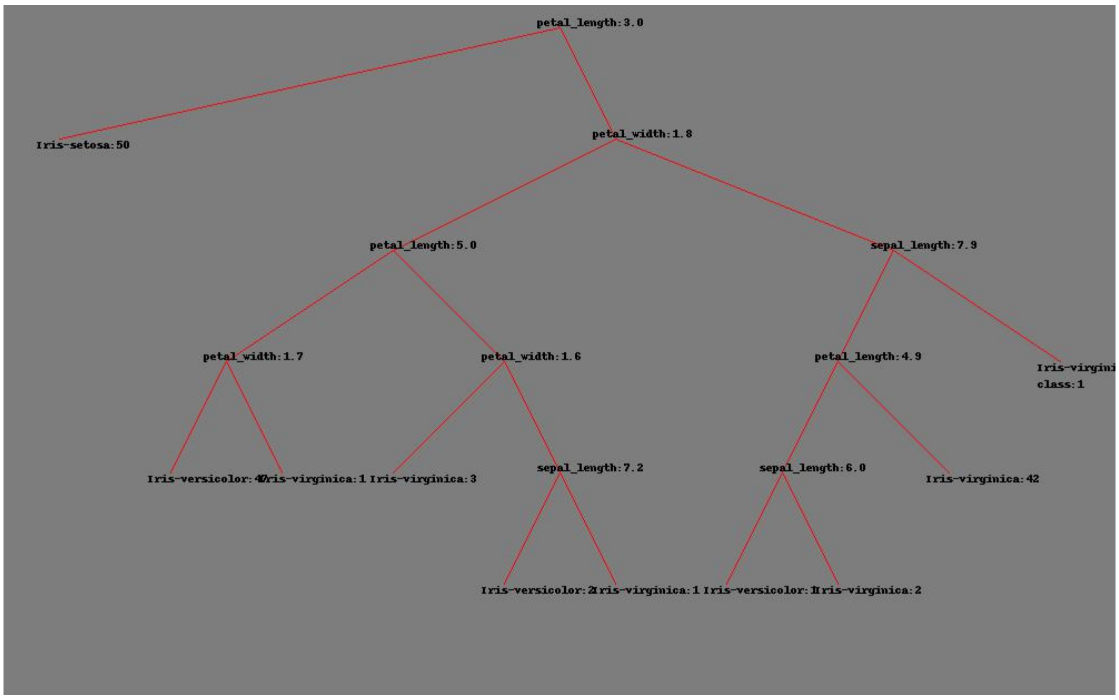
Implement 10-fold cross-validation to evaluate the accuracy of your algorithm on 10 different data sets from the UCI Machine Learning Repository. Select only those data sets where all features are numerical. In certain cases you can convert categorical features into numerical by encoding them using sparse binary representation. That is, if feature values belong to a set blue, yellow,red, green, encode this feature using 4-dimensional binary vectors such that if the feature value is blue, the encoding is (1, 0, 0, 0), if the feature value is yellow, the encoding is (0, 1, 0, 0), etc. You can also transform regression problems from the repository into classification problems by using the mean of the target variable to dichotomize the continuous target into binary class labels

1.) The ten folding is done with the help of dividing strata of data using Stratified K folding. Pythons Scikit has a library to do this. Say you have 150 values so it splits 150/K sets of test and train. Our K = 10(genkfolds.py)

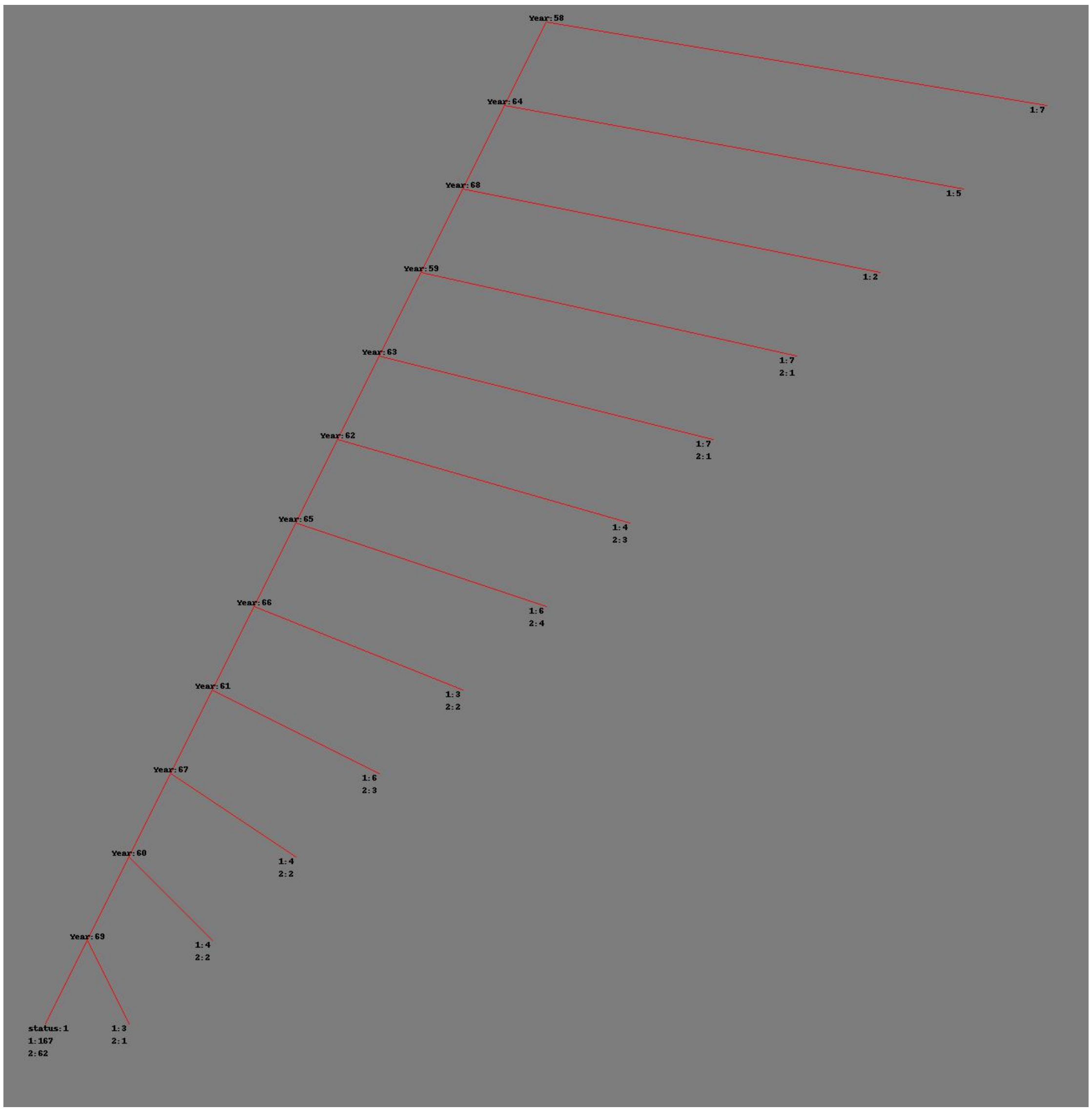
2.) In this the conversion of categorical attributes is performed to make them numeric. We check in the condition for a numeric (float or int) value or non numeric value.

3.) Classification is applied using all test data points generated and then checking for accuracy. The accuracy (accuracy.py) is checked for the number of false and true predictions.
Accuracy = Correctly Predicted Instances / Total Predictions made

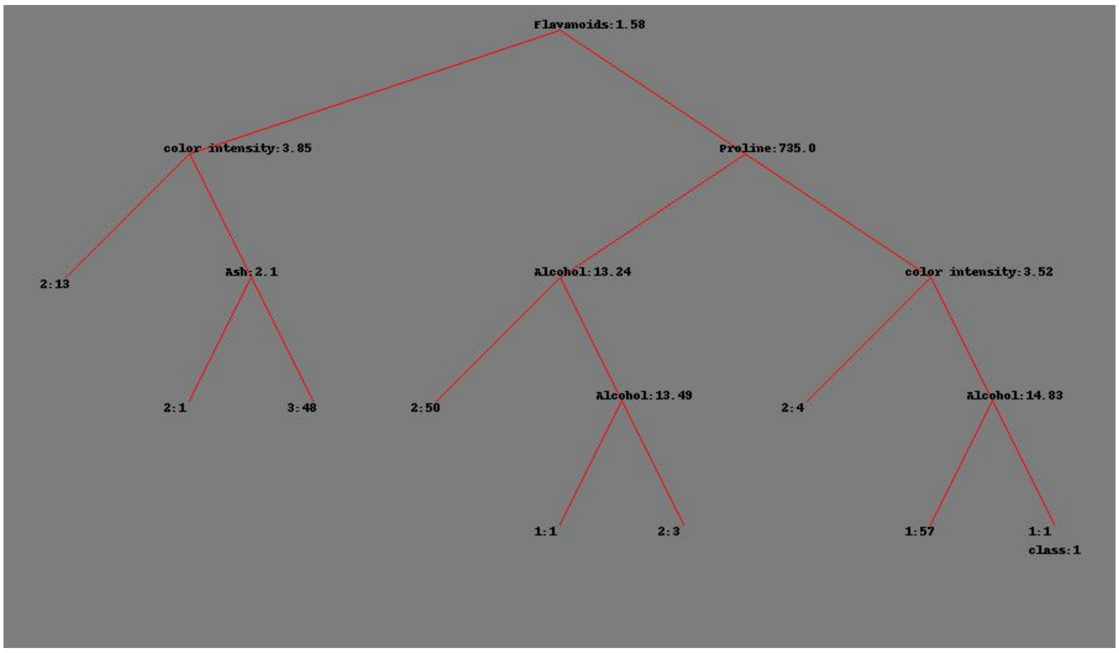
Accuracy for Iris: 99% on an average.
Accuracy for Haberman: 76% on an average.
Accuracy for Wine: 97% on an average.
Accuracy for Banknote: 95% on an average.
Accuracy for Yeast: 54% on an average.
IRIS


petal_length:3.0
Iris-setosa:50
petal_width:1.8
petal_length:5.0
sepal_length:7.9
petal_width:1.7
petal_width:1.6
petal_length:4.9
Iris-virgini
class:1
Iris-versicolor:47
Iris-virginica:1
Iris-virginica:3
sepal_length:7.2
sepal_length:6.0
Iris-virginica:42
Iris-versicolor:2
Iris-virginica:1
Iris-versicolor:1
Iris-virginica:2

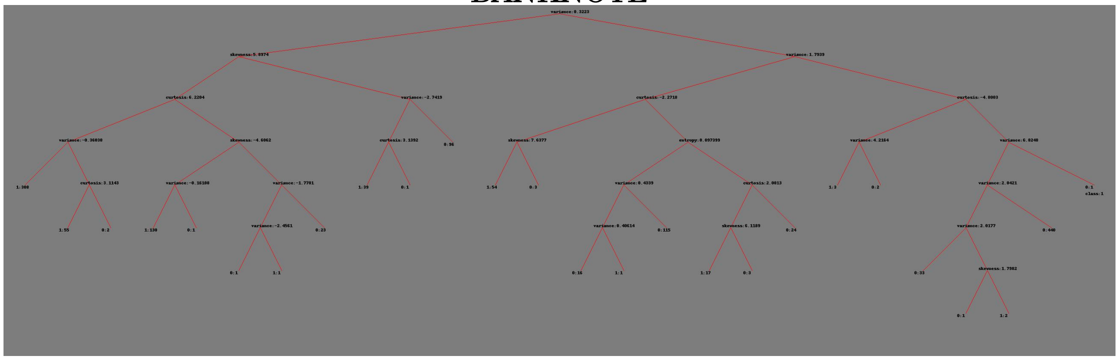
HABERMAN


Year: 58
1: 7
Year: 64
1: 5
Year: 68
1: 2
Year: 59
1: 7
2: 1
Year: 63
1: 7
2: 1
Year: 62
1: 4
2: 3
Year: 65
1: 6
2: 4
Year: 66
1: 3
2: 2
Year: 61
1: 6
2: 3
Year: 67
1: 4
2: 2
Year: 60
1: 4
2: 2
Year: 69
status: 1
1: 167
2: 62
1: 3
2: 1

WINE



BANKNOTE



4.3 Section 3

Compare Gini and information gain as splitting criteria and discuss any observation on the quality of splitting.

1.) Gini to calculate gain is much faster as compared to Entropy since it avoids logs and lambda functions.
2.) Gini is used to work on Continuous Attributes contrary to the Entropy for classes.
3.) While Gini is used for minimising Misclassification whereas Entropy for Exploratory Data Analysis.

4.) There is hardly any difference while you use gini or entropy. (Approximately maximum of 5

5. Question 5 Solution

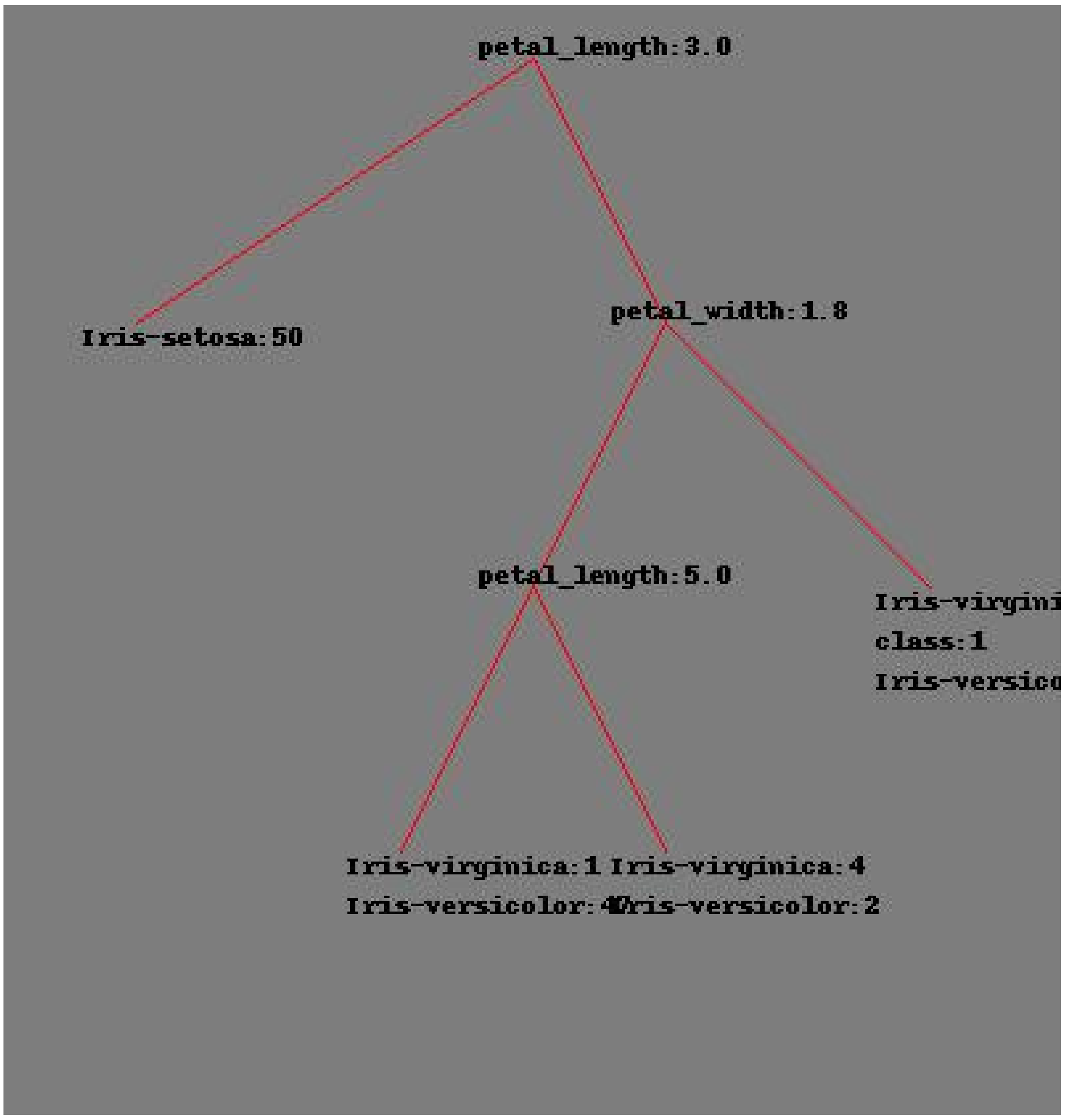
Modify your decision tree from the previous question and evaluate the overfitting prevention methods listed below. All evaluation should be carried out using 10-fold cross-validation as implemented in the previous Problem, but the performance should be evaluated using multiple measures.

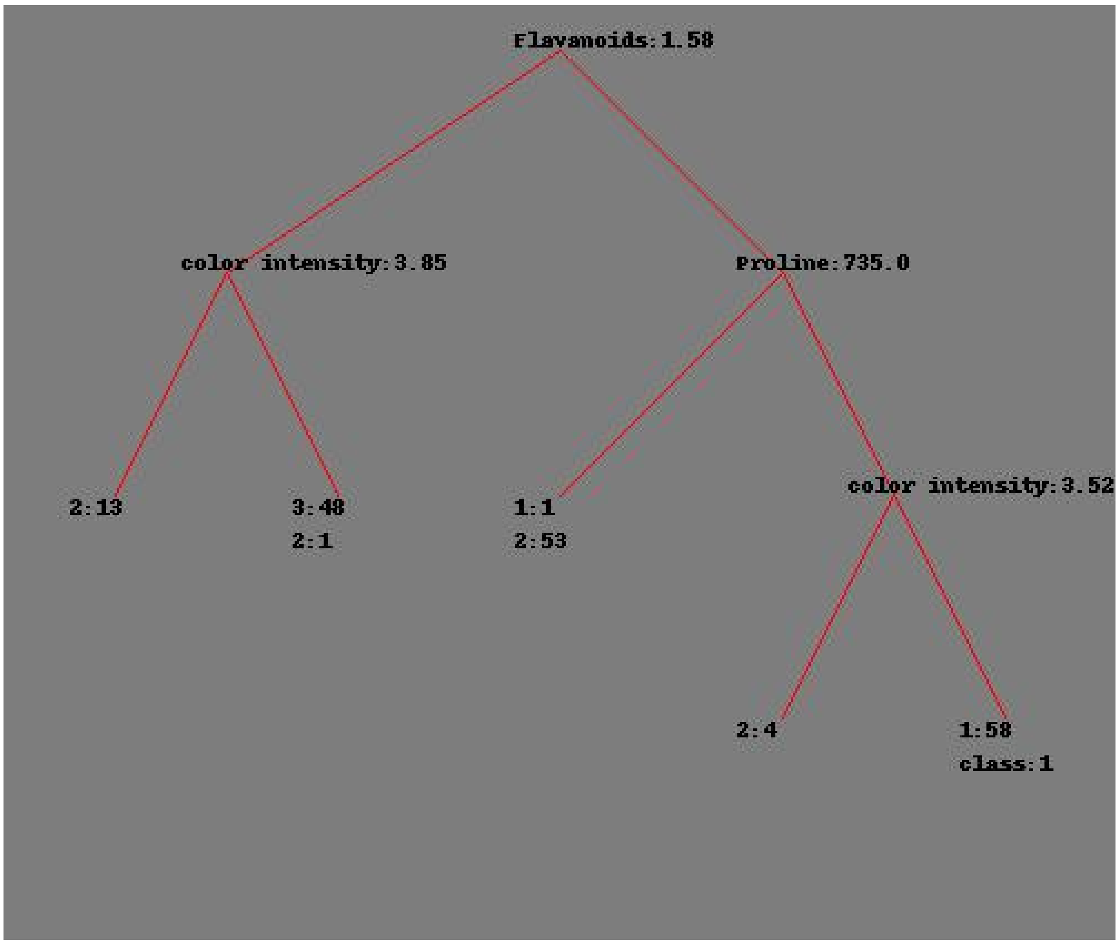
5.1 Section 1

Use pessimistic estimates of the generalization error by adding a penalty factor 0.5 for each node in the tree.

1.) There are two methods which could have been chosen at this point Post and Pre Pruning, we go ahead with post pruning due to the computational limitations and code architecture limitations. Also, the pessimistic error is calculate on basis of over all training error and the number of training records keeping in mind the penalty applicable on each leaf node.

Few of the Pruned Trees:


petal_length:3.0
Iris-setosa:50
petal_width:1.8
petal_length:5.0
Iris-virgini
class:1
Iris-versico
Iris-virginica:1
Iris-versicolor:47
Iris-virginica:4
Iris-versicolor:2



5.2 Section 2

Use a validation set that consists of 25% of the training partition

1.) The algorithm is splitting the already split data to consider that for validation set.
2.) It continues to follow the first question for the K Fold Stratified Sampling
3.) The validation test is used to test at each decision instance of the tree. As we start building the tree the issue I was stuck at was to maintain a node count. I just wanted to check the validation accuracy. In theory I should stop the tree growth as the validation set accuracy does not increment any more.

5.3 Section 3

Use the minimum description length principle, as explained in Question #8, page 201 of your Textbook.

MDL Formula:
Cost(tree,data) = Cost(tree) + Cost(data/tree)

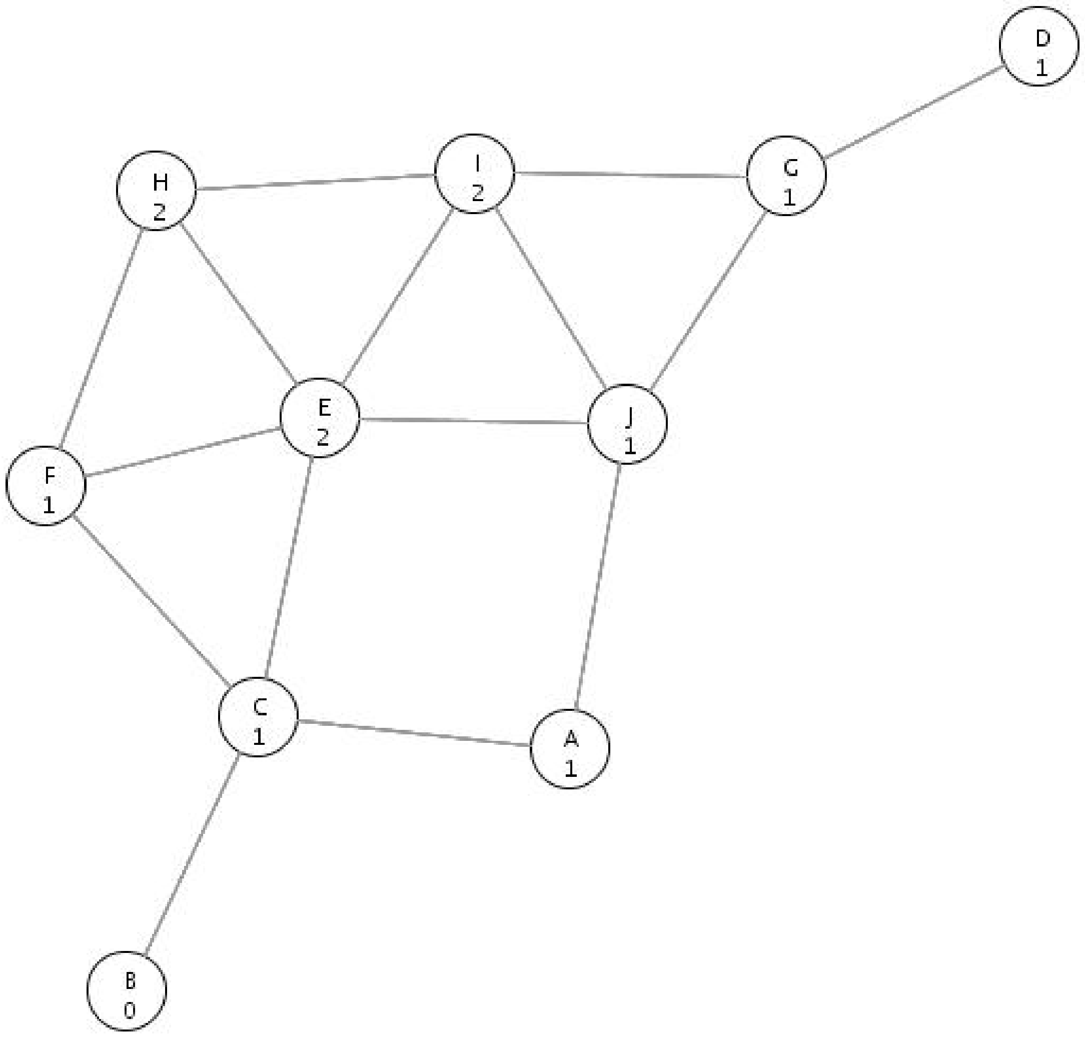
$$CostofEncodingeachAttribute = log_2(m)$$

$$CostofEncodingeachClass = log_2(k)$$

The basic logic is to find the difference between the parent and the child node as cost of encoding class and attribute is constant and we need to calculate the cost for the penalty incurred.

6. Question 6 Solution

Beyond greedy: combining beam search and classification tree construction. Extend the greedy algorithm for tree learning developed in the previous two questions to find a better classification tree as follows: At each step of node splitting, instead of picking the single best attribute to find the successor tree, keep top m successor trees based on the splits according to the top m attributes. For each of the trees in the list, generate the successor tree by splitting according to the next attribute. Then, keep top m successor trees (from all generated trees) for the next step and drop all the others. Continue the process until the stoppage criteria are fulfilled and select the best remaining tree as your model. Compare this algorithm with the greedy classification tree learning algorithm (m = 1) and comment on what you observe. Feel free to adjust this algorithm and experiment with different m's and other parameters to see whether there is improvement in accuracy.



Here we implement the Beam Search Algorithm in combination with the Breadth First Search Tree. We need to maintain M trees and select the best resulting atributes from each tree. The worst case will be that the attributes will be spread over m trees while the best cases will be when all the best attributes will be found in one place.

7. References

Discussed with Krish Mahajan. 1.) https://www.quora.com/How-is-the-market-for-

BI-tool-Tableau-What-is-its-future-Is-it-a-good-career-choice
2.) https://www.quora.com/Is-Tableau-good-from-a-career-point-of-view
3.) http://stats.stackexchange.com/questions/94886/what-is-the-relationship-between-the-gini-score-and-the-log-likelihood-ratio
4.) https://www.garysieling.com/blog/sklearn-gini-vs-entropy-criteria