Pattern Recognition and Machine Learning **Ayush Oksiya**
Assignment 2 **Roll No : 150121028**

Problem 1

Dataset Description: Each row in the dataset represents a sample of biopsied tissue. The tissue for each sample is imaged and 10 characteristics of the nuclei of cells present in each image are characterized. These characteristics are: Radius, Texture, Perimeter, Area, Smoothness, Compactness, Concavity, Number of concave portions of contour, Symmetry, Fractal dimension. Each sample used in the dataset is a feature vector of length 30. The first 10 entries in this feature vector are the mean of the characteristics listed above for each image. The second 10 are the standard deviation and last 10 are the largest value of each of these characteristics present in each image.

- **Training data:** ' trainX.csv' consisting of 455 samples, 30 attributes. The label associated with each sample is provided in ' trainY.csv'. A label of value 1 indicates the sample was for malignant (cancerous) tissue, 0 indicates the sample was for benign tissue.
- **Test data:** 'testX.csv' consisting of 57 samples, 30 attributes. The label associated with each sample is provided in 'testY.csv'.

Problem: Use decision trees to classify the test data. Estimate the misclassification rates of both classes and populate the 2x2 confusion matrix.(i) Report the following: (a)Plot of decision tree model. (b)the total number of nodes in the tree. (c)the total number of leaf nodes in the tree. (ii) Train your binary decision tree with increasing sizes of training set, say 10%, 20%, ..., 100%. and test the trees with the test set. Make a plot to show how training and test accuracies vary with number of training samples.

Results:
In this problem we are provided with the data for breast cancer with attributes like Radius, Texture, etc. And we are supposed to classify test data and predict if the given combination of attributes represents the possibility of Breast cancer.

Now, we are training the tree using Training data stored in the file named "trainX.csv" which contains 455 samples and 30 attributes for each sample, and the corresponding labels (1 - Yes, 0- No) are stored in file named "trainY.csv".

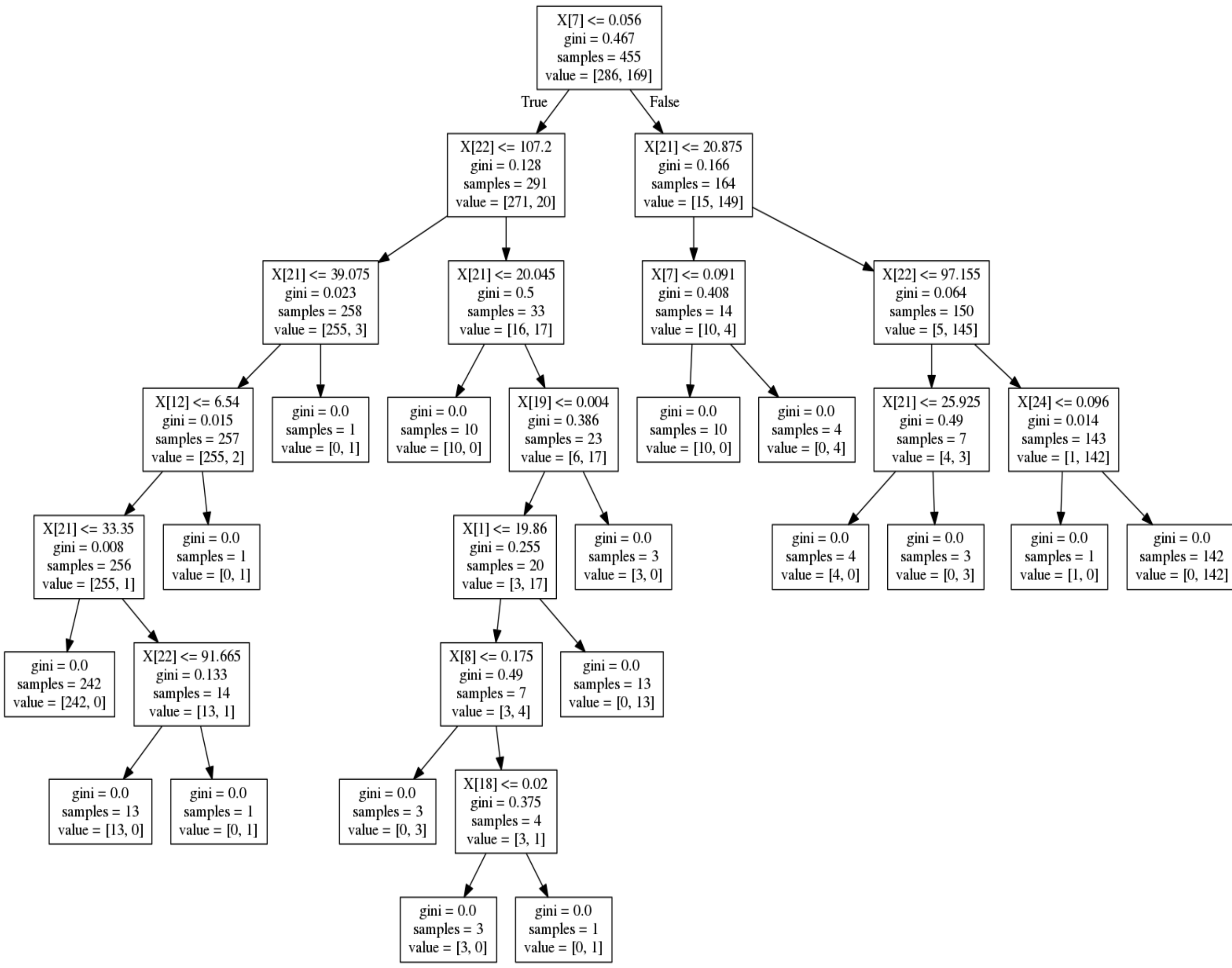
For the first case-I we are asked to train tree using entire training data.
After training the tree with given training data and predicting the labels for test data we get confusion matrix as :

Confusion Matrix :

$$M = \begin{bmatrix} 28 & 4 \\ 5 & 20 \end{bmatrix}$$

from confusion matrix,
The Misclassification rate for class-I is 12 %
The Misclassification rate for class-II is 20 %

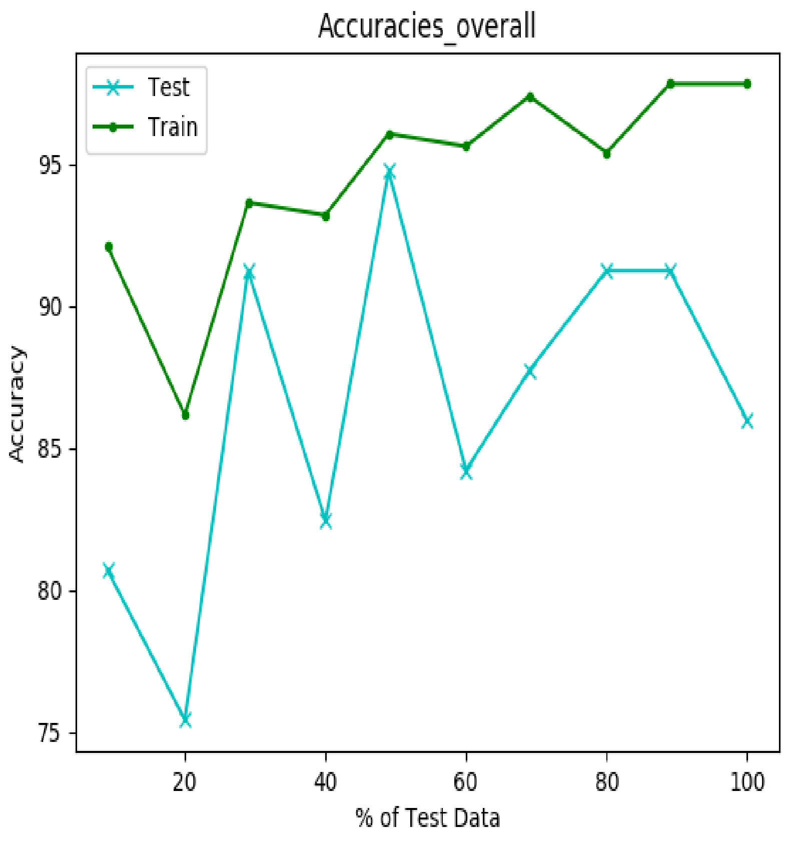
The plot of decision tree (*done using graphviz*):



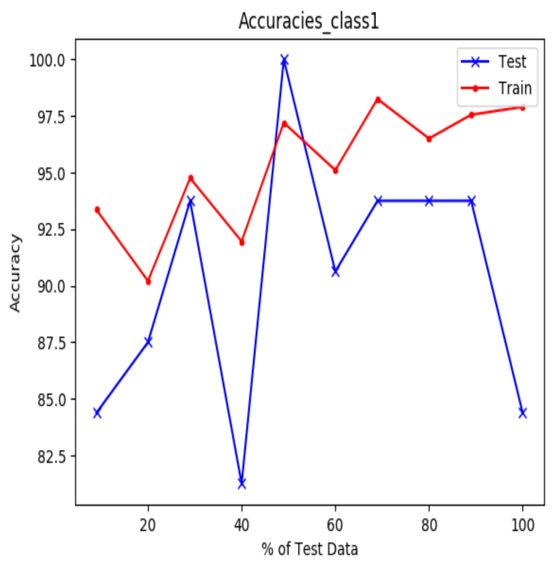
(b)Total no of nodes in the tree : 33
(c)Total no of leaf nodes in the tree: 17

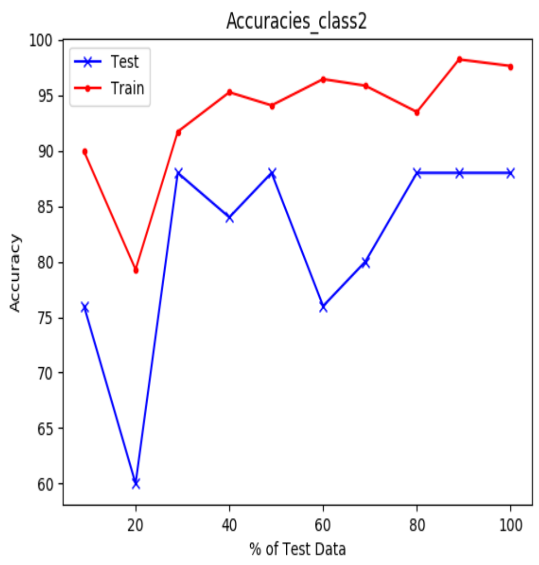
For the first case-II we are asked to train tree using some percentage of training data and test the obtained tree on test data and also plot the accuracies for both test data and training data by varying the percentage of train data used

Plot showing variation of training and test accuracies with number of training samples:



for each class separately :





As we can see the Accuracy for Train data saturates to 100 % when we use 100 % of training data, whereas it decreases for test data as we tahe 100 % training data which is expected as there might be overfitting of training data.

We get best accuracy on unknown(Test data somewhere in between 0 % and 100 %). which is optimal as we avoid overfitting here while maintaining the good accuracy on train data.

Problem 2

Dataset Description: The data set is derived from a two-year usage log of a Washington, D.C. bike-sharing system called Captial Bike Sharing (CBS). Bike sharing systems are variants of traditional bicycle rentals, where the process of renting and returning is heavily automated; typically, bikes can be rented at one location and returned at another without ever having to deal with a human being. The goal is to predict the daily level of bicycle rentals from environmental and seasonal variables using decision trees.

- **Data:** 'bikes.csv' has the following attributes:

  - date: The full date, in year month-day format.
  - season: Season of the year, 1 to 4
  - year: Year, 0=2011, 1=2012
  - month: Month (1 to 12)
  - holiday: Whether the day is holiday or not
  - weekday: Day of the week (coded by 0-6)
  - workingday: 1 for working days, 0 for weekends and holidays
  - weather: Weather, coded as follows:
  1. Clear to partly cloudy
  2. Mist but no heavier precipitation
  3. Light rain or snow, possibly with thunder
  4. Heavy rain or snow
  - temp: Normalized temperature in Celsius. The values are derived via (t - t_{min})/(t_{max})
  - t_{min}; t_{min} = -8, t_{max} = +39.
  - humidity: Normalized humidity (actual humidity divided by 100).
  - windspeed: Normalized wind speed (actual wind speed in miles per hour divided by 67).
  - count: Count of total bike rentals that day, including both casual and registered users.

Problem: Build a regression tree predicting daily bike rentals from all available variables. (i) Report the following: (a) Plot of regression tree (b) The total number of leaf nodes in the tree, (c) Into how many different groups of days does the tree divide the data, (d) Which variables appear in the tree, (e) Which variables are important, (f) The MSE. (ii) Now re-code the months so that January and February share one code, May through October shares another, and March, April, November and December share a third. Re-estimate the regression tree. How does the tree change (if at all)? What is the MSE? Did it improve the fit?

Results :

Here we are provided with the data of an bike-sharing system, with attributes which include season, year,month,etc. and corresponding count (i.e. no of bikes rented).

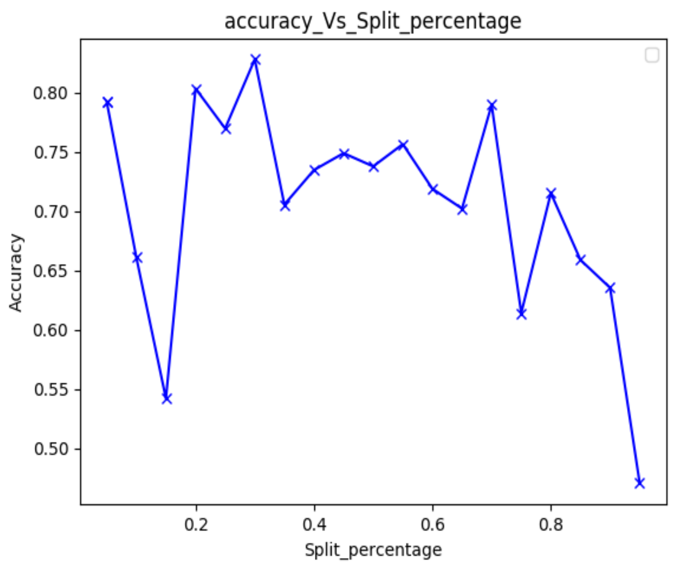
We are required to fit the given data to a regression tree.
Now, if we go on and directly make a regression tree with all the given data there will be high Over-fitting for sure and we might end up generating a leaf for each count.

Now problem with overfitting is the tree will be 100 % accurate for the training data but will do terribly bad on test data which is yet unknown.

So we need to find out a way to train a regressor tree such that it should not overfit the training data and also perform good on both training data and test data(which will be unknown for tree till now).

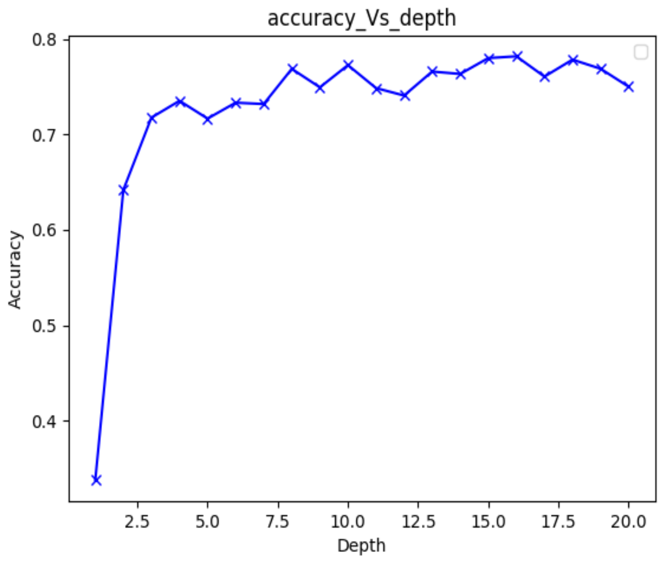
To get those pre-conditions on tree, my initial approach was to split data in "2" parts, test and train respectively.
To decide on how to split(i.e. % train and % test) i checked the variation in accuracy for different combination of test and train percentages.

The graph of accuracy Vs percentage test :



Then creating regression trees for varying maximum depth using train set and prediction the test data and hence finding the optimal depth for which tree will still perform good for unknown data.

The graph of accuracy Vs the max depth for this method :



After this i tried the "min impurity decrease" and checked the variation of Mean Squared Error(MSE) by varying min impurity decrease value. And hence got the maximum value of it for which there was relatively low Mean Square Error.

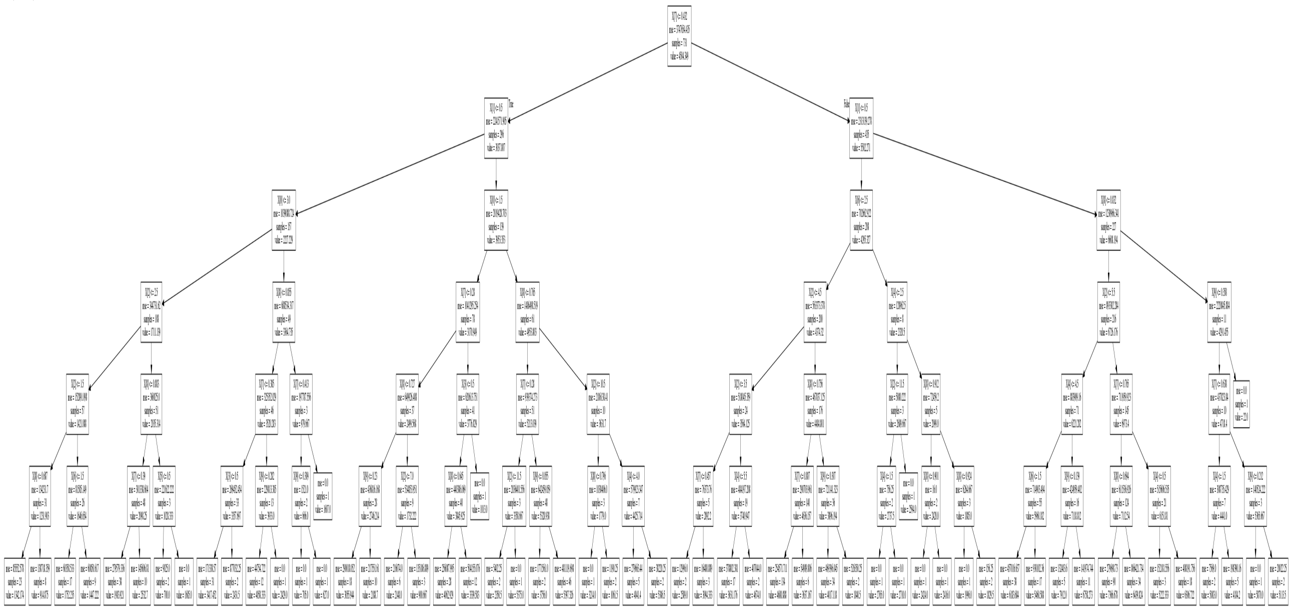
Then i used K-Fold cross validation to find out max depth, here i created 10 splits and ran the cross validation to get max depth such that the mean square error is minimum and also the depth(and hence no. of nodes) is minimum to perform good on unknown data.

The Mean Squared Error For all 3 types : 1934.0689, 4034.1237, 316697.3191
The accuracy/score : 99.94, 99.89, 91.54
The no of nodes : 651, 405, 58

Then on creating the regression tree from all three cases and comparing the mean squared error and no. of nodes for all three cases, i found that deciding the max depth using K-Fold cross validation and creating decision tree on entire data was best way to do it(as we are getting *sim* 92 % accuracy with much significant less no. of nodes).

Regression tree using pre conditions derived from case 3.

(a)*Plot of regression tree :*



(b)Total no of leaf nodes in the tree : 115

(c)The tree divides data in "3" different groups of days, weekdays, workingdays and holidays.

(d)Variables which appear in the tree with increasing order of their importance are : $X[5] < X[3] < X[4] < X[6] < X[9] < X[2] < X[8] < X[0] < X[1] < X[7]$

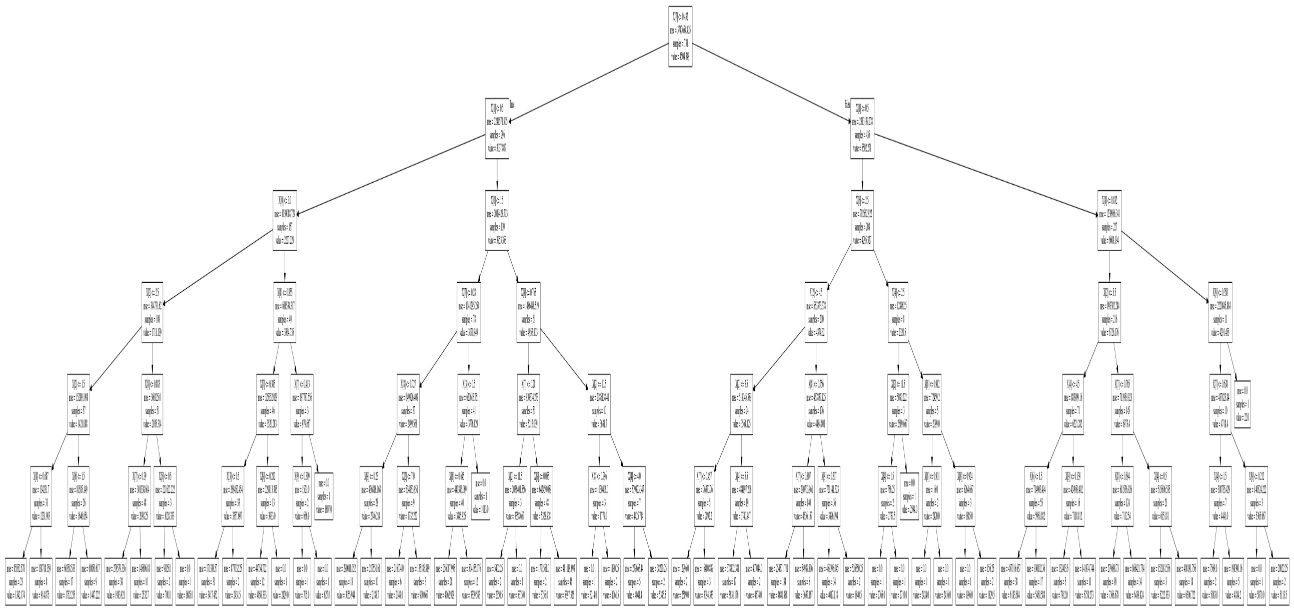
As we can see the variables which are of least importance are workingdays, holidays, weekdays. and which are most important are temperature, season,year, etc as expected by the encoding of given data set.

(e)Variables with most Importance are : X[7], X[1], X[0] i.e. Temperature, Year and season.
(f)The Mean Square Error(MSE) is <u>316691.319152</u>

For the second part of this question we are asked to re-code the months such that January and February are identical and so on.
In order to do that i have given same attribute to those months which are identical.

Now i used same pre conditions as above to get the regression tree on this new training set with new attributes corresponding to months.

New regression tree :



The tree now changes very little (insignificantly) at the very bottom i.e. leaf part.

This is expected as Month is basically "X[2]" which 5^{th} significant and extremely insignificant when compared to X[7] and slightly significant when compared to X[5]

The Mean Squared Error(MSE) now is : 334406.431911
The no. of nodes now is 58.

As the no of nodes remains unchanged but the MSE is increased it did not improve the fit, but neither did it changed MSE by a very large number. Which was expected as it is almost irrelevant for the regression tree.