Assignment-1 Report
(Decision Tree)

Name: Kishan Shankar Singhal
Roll No: 2018201023
M.Tech PG-1 CSE

Question 1-

Part 1:

Used id3 algorithm in which attribute with maximum info gain is selected as root node. This process is recursively done until the set in a given sub-tree is homogeneous (i.e. it contains objects belonging to the same category). Following are the different measures:-

```
The accuracy is:  0.775355871886121

Recall:  0.0019880715705765406

Precision:  1.0

F1-Score:  0.003968253968253968

True pos:  1

False pos:  0

True neg:  1742

False neg:  502
```

(Using scikit learn different values are:)

```
Accuracy : 0.7732974910394266
Confusion Matrix : [[1726    0]
                    [ 506    0]]
              precision    recall  f1-score   support

           0       0.77      1.00      0.87      1726
           1       0.00      0.00      0.00       506

   micro avg       0.77      0.77      0.77      2232
   macro avg       0.39      0.50      0.44      2232
weighted avg       0.60      0.77      0.67      2232
```

Part-2:

Used CART algorithm in which:
for continuous data- Divided the values into mean of previous and next values then values less are taken on left of tree and greater values on right.
for categorical data- Similarly if values are exactly equal with any category then it is taken on left and all others on right.

Following are the different measures:-

```
The accuracy is:  0.9697508896797153

Recall:  1.0

Precision:  0.9980237154150198

F1-Score:  0.9990108803165183
```

(Using scikit learn different values are:)

```
Accuracy : 0.9763709317877842
Confusion Matrix :[[1659   29]
                   [  24  531]]
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1688
           1       0.95      0.96      0.95       555

   micro avg       0.98      0.98      0.98      2243
   macro avg       0.97      0.97      0.97      2243
weighted avg       0.98      0.98      0.98      2243
```

Part-3:

Making decision tree using different impurity measures.

Following are the values of Accuracy, Recall, Precision and F1-score:

1. Using Entropy:

```
The accuracy is:  0.9679715302491103

Recall:  1.0

Precision:  0.9980544747081712

F1-Score:  0.9990262901655307
```

(Using scikit learn different values are:)

```
Accuracy : 0.9763709317877842
Confusion Matrix : [[1659   29]
                    [24  531]]

              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1688
           1       0.95      0.96      0.95       555

   micro avg       0.98      0.98      0.98      2243
   macro avg       0.97      0.97      0.97      2243
weighted avg       0.98      0.98      0.98      2243
```

2. Using Gini Index:

```
The accuracy is:  0.9715302491103203

Recall:  1.0

Precision:  0.9980430528375733

F1-Score:  0.999020568070519
```

(Using scikit learn different values are:)

```
Accuracy : 0.9763709317877842
Confusion Matrix :[[1659   29]
                   [  24  531]]
              precision    recall  f1-score   support
```

```
              0       0.99      0.98      0.98      1688
              1       0.95      0.96      0.95       555

   micro avg       0.98      0.98      0.98      2243
   macro avg       0.97      0.97      0.97      2243
weighted avg       0.98      0.98      0.98      2243
```

3. Using Misclassification:

```
The accuracy is:  0.969306049822064

Recall:  0.9980353634577603

Precision:  1.0

F1-Score:  0.9990167158308751
```

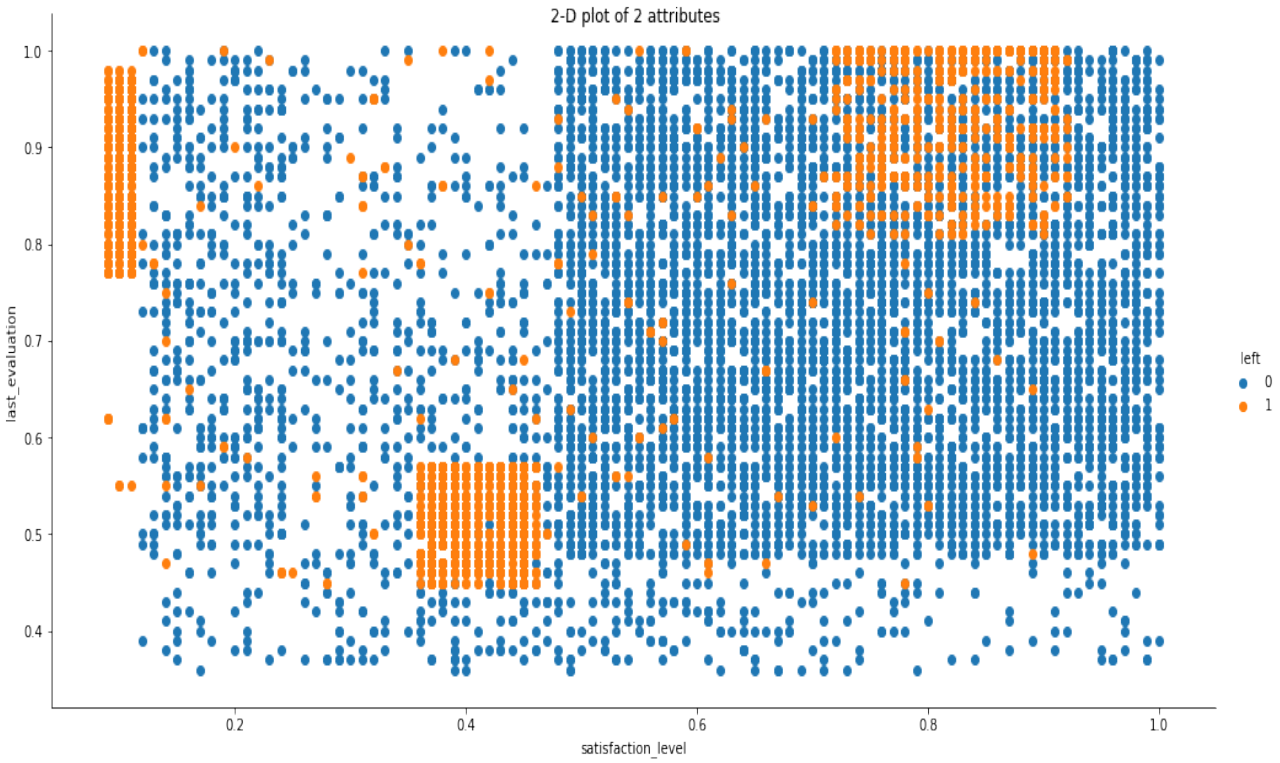
(Using scikit learn different values are:)

```
Accuracy : 0.9763709317877842
Confusion Matrix :[[1659   29]
                   [  24  531]]
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1688
           1       0.95      0.96      0.95       555

   micro avg       0.98      0.98      0.98      2243
   macro avg       0.97      0.97      0.97      2243
weighted avg       0.98      0.98      0.98      2243
```

Part-4:

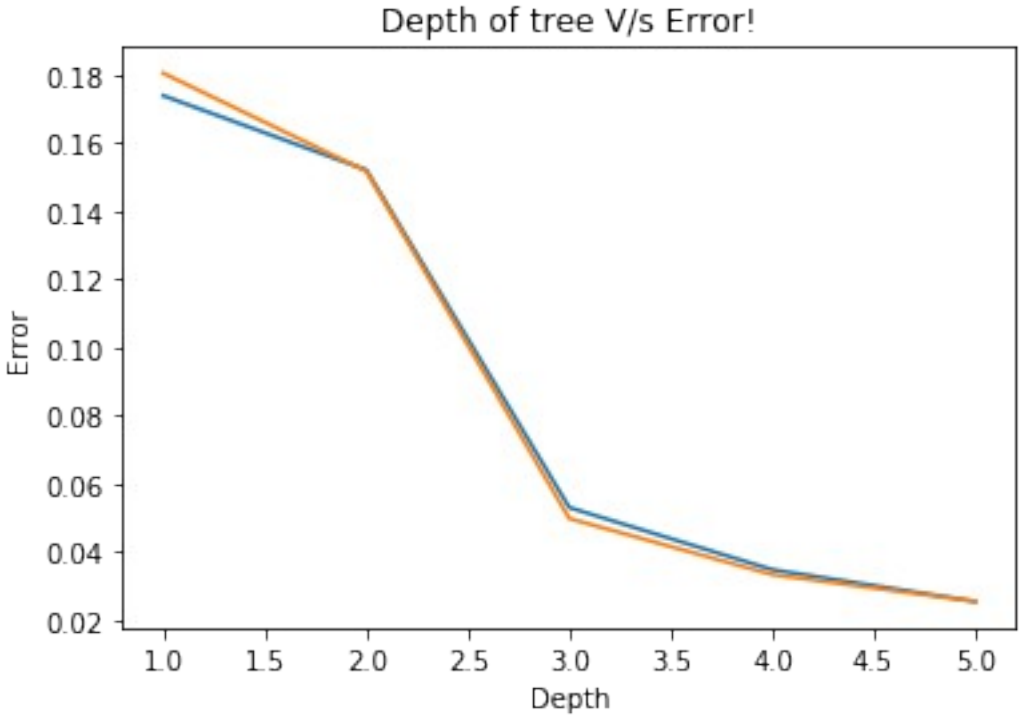
Visualise training data on a 2-D plot using scatter plot:

1. On one axis: attribute taken is ‘satisfaction_level’
2. On other axis: attribute taken is ‘last_evaluation’

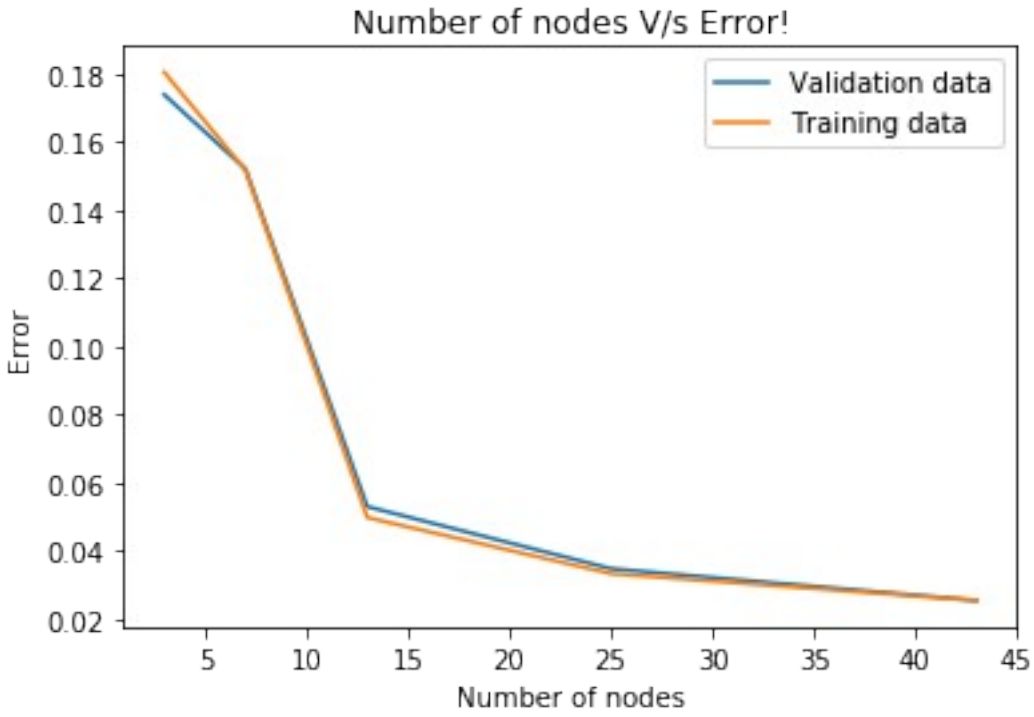


Part-5:

a) Graph of training and validation error with respect to depth of decision tree:



b) Graph of training and validation error with respect to Number of nodes of tree:



Part-6:

Explain how decision tree is suitable handle missing values in data:-

There can be various ways to handle missing values in the data set:-

1.Like Simply ignore the missing values (like ID3 and other old algorithms does)

2.Treat all missing values as a separate group(by filling any special symbol or something like label all missing values as "Random")

3.For categorical attribute, Find the mode of that column(attribute with the biggest number of instances) and fill it in place of missing values.

4.For continuous data attribute, Find the mean or median of entire attribute and fill it in place of missing values. Then build the decision tree.

5.If the attribute with missing value is less important(i.e. with very less entropy), then ignore missing values from it.

6.If data set is large and attribute is important then ignore that row having missing value.

7.Decide one path to take on missing values and take always that path only (like move to left most child if missing value found).

8.Lazy Decision Tree (Reduced Feature Models/Known Value Strategy):- here the prediction model is constructed at testing time based on the available test instance values. This is also known as ‘Known values strategy’. During tree construction it uses only attributes whose values are known at testing. Hence it naturally handles the missing values at testing.