SMAI ASSIGNMENT-1

SUBMITTED BY: AISHWARYA SHIVACHANDRA

ROLL NO. : 2018202005

1. Train decision tree only on categorical data. Report precision, recall, f1 score and accuracy.

Algorithm implemented: Firstly, split the given dataset using 80-20% train-validation split. Then, from the given dataset, considering only categorical data and taking attribute **'left'** as target attribute:
calculated entropy for dataset and each of the attributes
and used this information to obtain maximum information gain and start building the decision tree.

Results and observations:
Precision: 1.0
Recall: 0.001834862
F1-score: 0.003663003
Accuracy: 0.758007117438

2. Train the decision tree with categorical and numerical features. Report precision, recall, f1 score and accuracy.

Algorithm implemented: Firstly, split the given dataset using 80-20% train-validation split. To train the decision tree, numerical as well as categorical data has been used. Then, from the given dataset, considering all attributes and taking **'left'** as target attribute:
calculated entropy for dataset and each of the attributes.
for numerical attributes, obtained the split point which gives minimum entropy.
Then used this information to obtain maximum information gain and used this node with max Information Gain as the node for building the decision tree.

Results and observations:
Precision: 0.95063985
Recall: 0.95412844
F1-score: 0.9523809523
Accuracy: 0.758007117438

3. Contrast the effectiveness of Misclassification rate, Gini, Entropy as impurity measures in terms of precision, recall and accuracy

Algorithm implemented: Using the approach used in Q2 above, we again build decision tree to calculate precision, recall and accuracy. Also, find Maximum information gain and thus build decision tree using Gini Index and Missclassification rate respectively to see the contrast between the three methods.

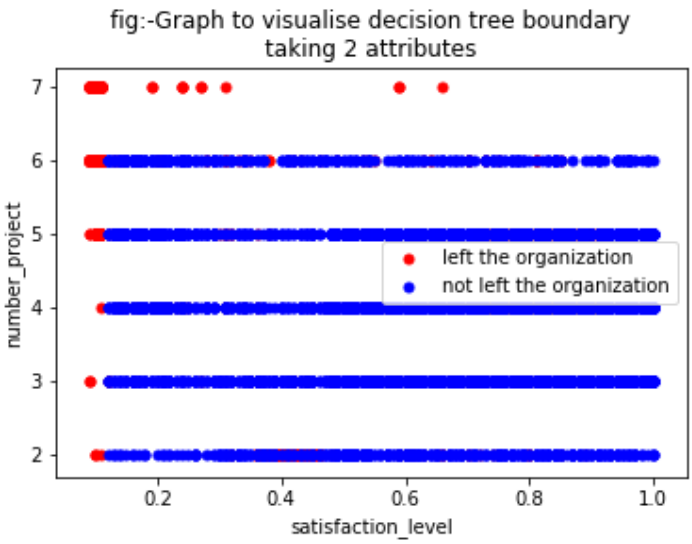
Results and observations: =>Using Gini Index as impurity measure to find Information gain and build tree, we found:
Precision: 0.972709551657
Recall: 0.915596330275
F1: 0.943289224953
Accuracy: 0.973309608541

=>Using Entropy as impurity measure to find Information gain and build tree, we found:
Precision: 0.950639853748
Recall: 0.954128440367
F1: 0.952380952381
Accuracy: 0.976868327402

=>Using Misclassification as impurity measure to find Information gain and build tree, we found:
Precision: 0.950099800399
Recall: 0.873394495413
F1: 0.910133843212
accuracy: 0.958185053381

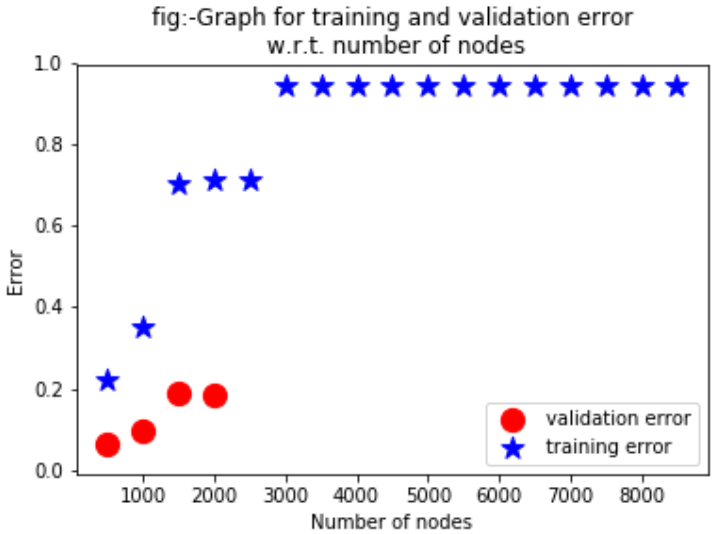
4. Visualise training data on a 2-dimensional plot taking the two features with maximum information gain

Plotted a Graph to visualise scattering of dataset using 2 attributes with maximum information gain:
1)satisfaction_level on X-axis
2)number_project on Y-axis to visualise decision tree boundary



5. Plot a graph of training and validation error with respect to to number of nodes in the decision tree.

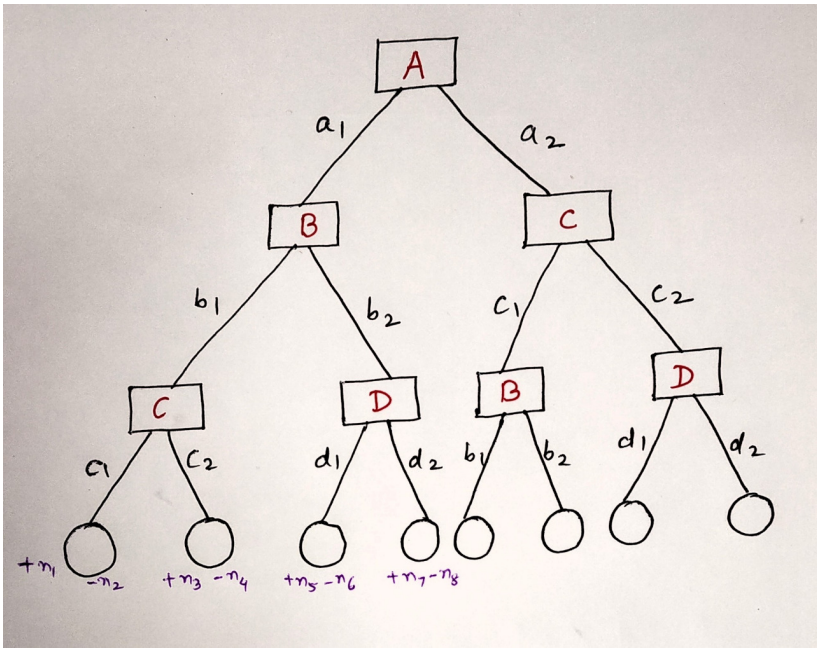
Algorithm implemented:
Starting from root of tree, calculate accuracy and thus error for the following cases:
1) for validation data,
2) for training data
for each of the above datasets, vary the number of nodes from 500 till length of the respective datasets and calculate the error for each and plot scatter graph as shown.



Results and observations:
As we increase the number of nodes to traverse, the error increases.
This happens because when the number of nodes to be traversed is large, the dataset is less splitted and thus accuracy is very low.

6. Explain how decision tree is suitable to handle missing values in data.

When some attributes are missing in test sample dataset, then we will consider all the possible values which target attribute can take from original dataset and turn-wise predict result for each of the possible value. Then from the obtained result whichever value has the highest probability will be returned as the predicted value.



example 1:
in dataset if given attributes are: **[A,B,C,D]**. At each leaf node, we calculate the number of positive results for target attribute as well as number of negative results for the same as shown in the figure as n1,n2,n3 and so on.

Given missing-value test-case is **[a1,_,c1,-]** Since the value for "B" attribute is missing, we sum up the result for all possible cases of B,(b1 and b2). At the next step, we have the value for "C" in the given test dataset so we will consider n1 and n2. Since the value for "D" attribute is missing, we consider all n5,n6,n7,n8.

Finally, if positive results for target attribute are more than the negative results, then we return "TRUE" as predicted value, otherwise return "FALSE".

example 2:
in dataset if given attributes are: **[Outlook, Temperature, Humidity, Wind, Play]** and Play is the attribute for which value has to be predicted. Given missing-value test-case is :[rain, hot, high,__]

Here we can see that value for "Wind" Attribute is missing and we found that "Wind" can take following values ["weak", "normal", "strong"] from Given Dataset. Then we'll find result for all the three possible cases i.e.,[overcast, hot, high, weak], [overcast, hot, high, normal], [overcast, hot, high, strong].

Finally, if positive results for target attribute are more than the negative results, then we return "TRUE" as predicted value, otherwise return "FALSE".