SMAI Assignment-1 Report

Name - Akshat Maheshwari (20161024)

Part-1 and Part-2:

I have written the codes as q-1-1.py and q-1-2.py for the categorical and categorical+numerical respectively.
I have given the training file as hard-code in my code. Both my codes return the accuracy, error, precision, recall and F1-score corresponding to my validation data.

For part 1,

- Max Precision = 1.0
- Max Recall = 0.05
- Max Accuracy = 0.757
- Max F1-Score = 0.004

For part 2,

- Max Precision = 0.98
- Max Recall = 0.995
- Max Accuracy = 0.98
- Max F1-Score = 0.983

Part-3:

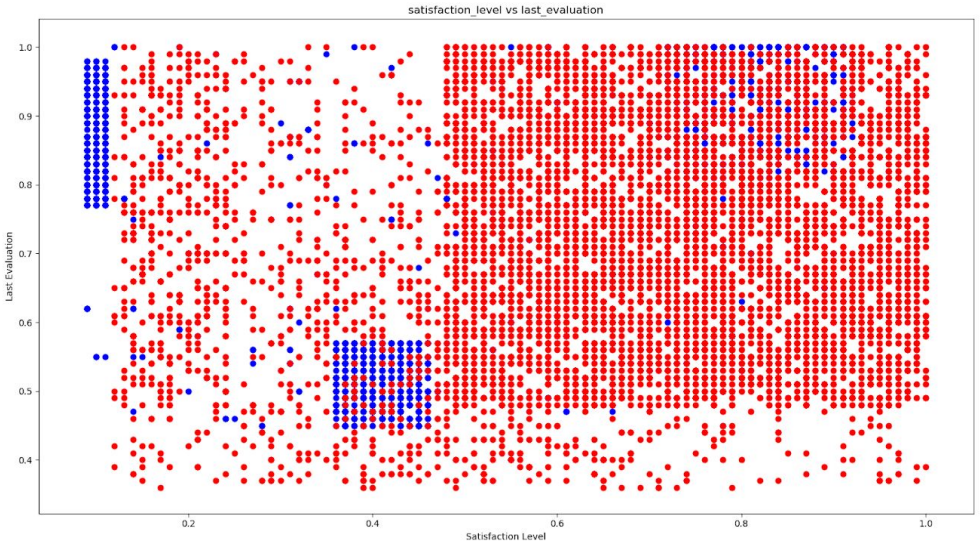
Misclassification error, Gini and Entropy are the three most commonly used impurity measures in case of decision trees.

- Entropy = $-\sum_j p_j \log_2 p_j$
- Gini = $1 - \Sigma p^2_j$
- Classification Error = $1 - \max p_j$

Gini and entropy are pretty much the same, although the formula for both is different. In general, these two are used interchangeably. The entropy is 0 if all samples of a node belong to the same class, and the entropy is maximal if we have a uniform class distribution. In other words, the entropy of a node (consist of single class) is zero because the probability is 1 and log (1) = 0. Entropy reaches maximum value when all classes in the node have equal probability.

Part-4:

With the codes, I have made the plots. The plots keep on changing because I have randomized the split of the train data into training and validation.
I have used only the part 2 code which consist of all the attributes taken into account.
Randomly I have chosen the first 2 features, satisfaction_level and last_evaluation for plotting the graph.

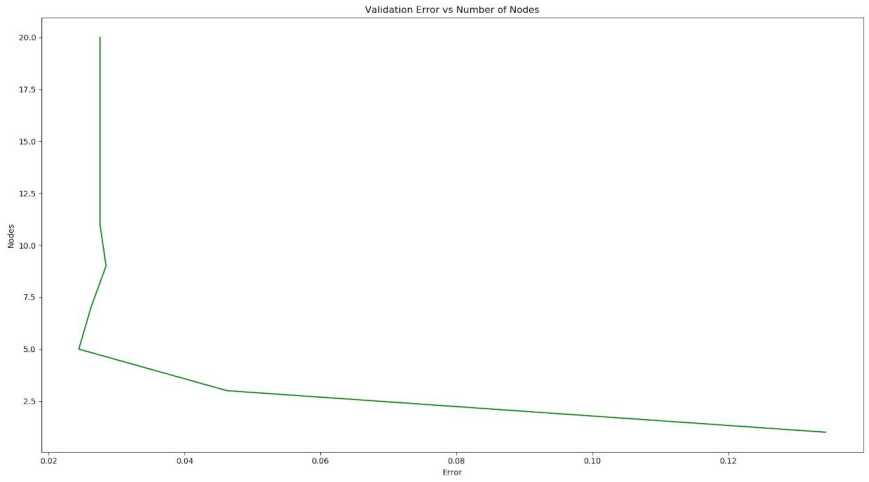


Part-5:

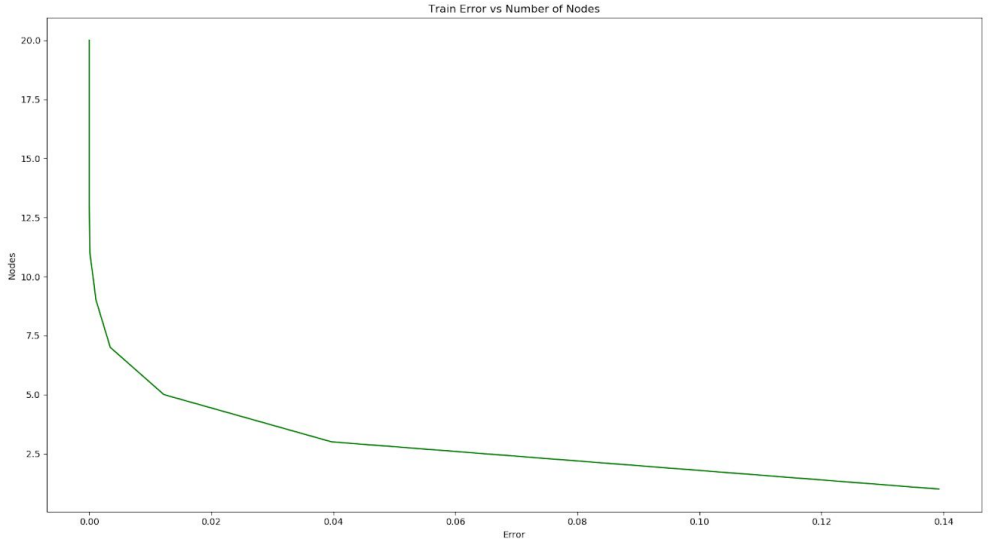
For this part, I have created linear plots for error vs the maximum number of nodes in the graph. In this part also, I have used just the part 2 code only that consists of all the attributes (both numerical and categorical).

The error first reduces to a particular minimum value and the increases a bit and then becomes almost stable.

- Validation Error vs Number of Nodes



- Train Error vs Number of Nodes



Part-6:

There are several methods used by various decision trees. Simply ignoring the missing values (like ID3 and other old algorithms does) or treating the missing values as another category (in case of a nominal feature) are not real handling missing values. However those approaches were used in the early stages of decision tree development.

Some approaches to distribute the missing value instances to the child nodes:

- all goes to the node which already has the biggest number of instances
- distribute to all children, but with diminished weights, proportional with the number of instances from each child node
- distribute randomly to only one single child node, eventually according with a categorical distribution
- build, sort and use surrogates to distribute instances to a child node, where surrogates are input features which resembles best how the test feature send data instances to left or right child node
- Or, ignore some of the rows that have the missing data