SMAI (CSE 471)
Spring-2019
Assignment-1

Prakash Nath Jha
2018201013

Decision Tree Implementation Details:

Decision tree in this case has been implemented as a binary tree where at each node of the tree we are only taking yes or no decision i.e. either to go to left child or right child.

Nodes of the tree is represented by two type of class objects one for internal node and one for leaf node. Internal node structure contains left child, right child, attribute name on which split has taken, value of the feature, type of feature which can either be “categorical” or “continuous”, positive count which stores number of rows for which our label i.e. feature “left” gives “1” and negative count which stores number of rows for which our label gives “0”. Leaf node structure contains prediction variable which stores whether the final output is “0” or “1”. Leaf node also contains positive and negative row count just like internal node structure.

For categorigcal features we first find information gain produced by each feature and based on that we choose the feature with maximum information gain and we split the dataset on that feature’s unique value which provides maximum purity. All the rows which contains that particular value in that particular feature column goes as the left child and all other as the right child. This process continues until we hit the base condition where we finally make leaf node and set the prediction value of the leaf node to the maximum of the unique values present in the label feature.

For continuous features we do the same as the categorical feature but finding the split criteria is different. For continuous feature we first sort the feature values and for every consecutive values pair we find the mid point and calculate the information gain and whichever split offers the maximum information gain is taken as the split criterion.

For pruning dataset we have used parameter like maximum depth, maximum nodes, minimum rows based on which we prune the tree so as to minimise overfitting.

In our case its been observed that the tree gives maximum accuracy when we set impurity measure as “Gini” , maximum nodes around 400 maximum depth as 8 and minimum rows until we stop spliting dataset as 3.

1. Attributes choosen for training: Work_accident, promotion_last_5years, sales, salary.

   Results:

True Positive: 0
True Negative: 1713
False Positive: 0
False Negative: 535

Accuracy : 76.201%
Recall : 0 (since true positive = 0)
Precision, F1 Score not defined as both true positive and false positive is 0

Scikit Learn library results:

True Positive: 0
True Negative: 1696
False Positive: 0
False Negative: 552

Observation : If we consider only categorical features it can be observed that the data becomes ambigous as for the same combination of values of categorical features we find both positive and negative value of label i.e. “left” feature. We can see that even scikit learn library produces similar result which reinforces above conclusion that just categorical features are not sufficent enough for making good prediction.

2. Features choosen for training: All features including continous and categorical features.

   Results:

   True Positive: 508
   True Negative: 1689
   False Positive: 24
   False Negative: 27

   Accuracy = 97.731%
   Precision = 95.488%
   Recall = 94.953%
   F1 Score = 0.9522

   Scikit Learn library results:

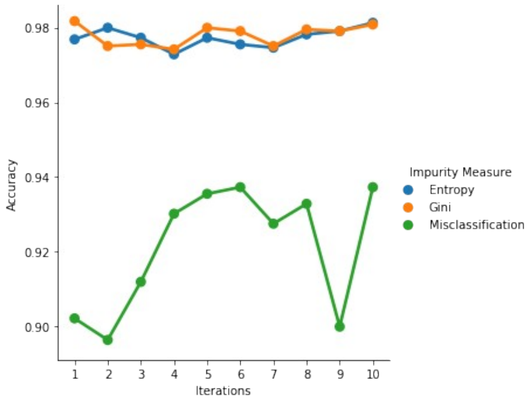
   True Positive: 486
   True Negative: 1705
   False Positive: 31
   False Negative: 26

   Accuracy: 97.763%
   Precision: 94%
   Recall: 94.921%
   F1 Score: 0.9445

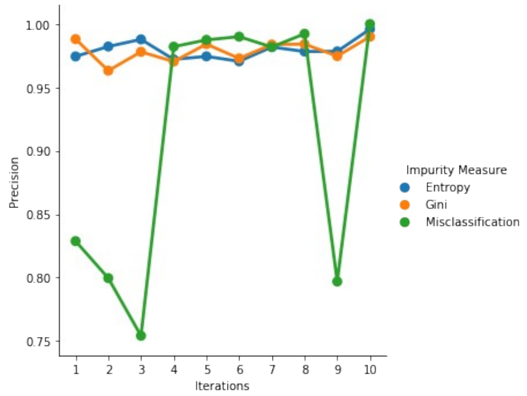
Observation: Best accuracy was obtained when Gini impurity was used as impurity measure which is also the same used by Scikit learn library internally. Misclassification impurity measure gave the least performance while Entropy measure was almost similar to Gini.

3. Comparision of Gini , Entropy, Misclassification in terms of accuracy,precision, recall, f1 score of the decision tree.

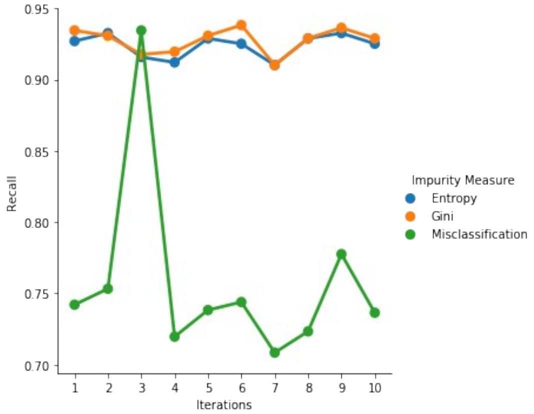
Accuracy:



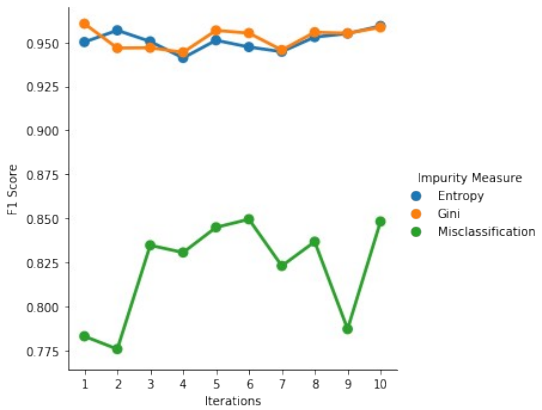
Precision:



Recall :



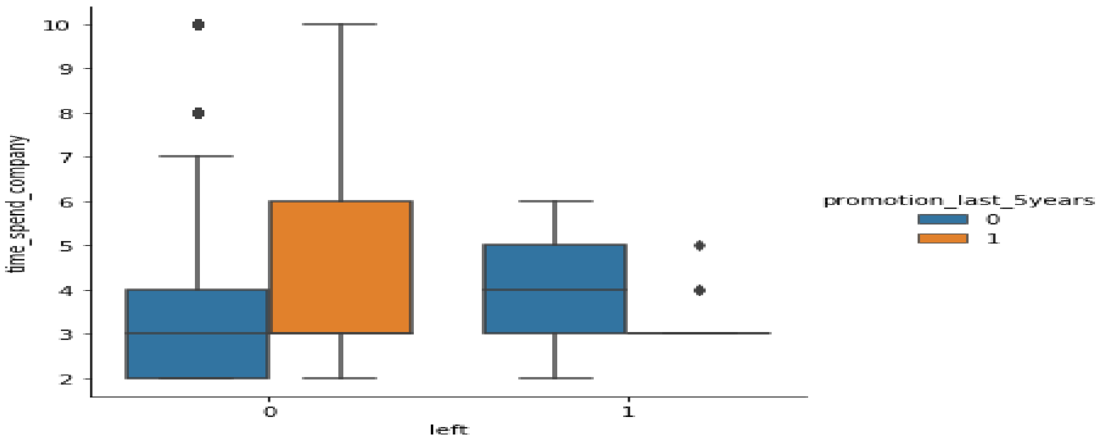
F1 Score :



As from the above plot we can infer that Gini and Entropy offers better accuracy, recall , F1 Score and precision as comapred to Misclassification. Decision tree performance was poorest when Misclassification rate was choosen as impurity measure in comparision to Gini and Entropy.

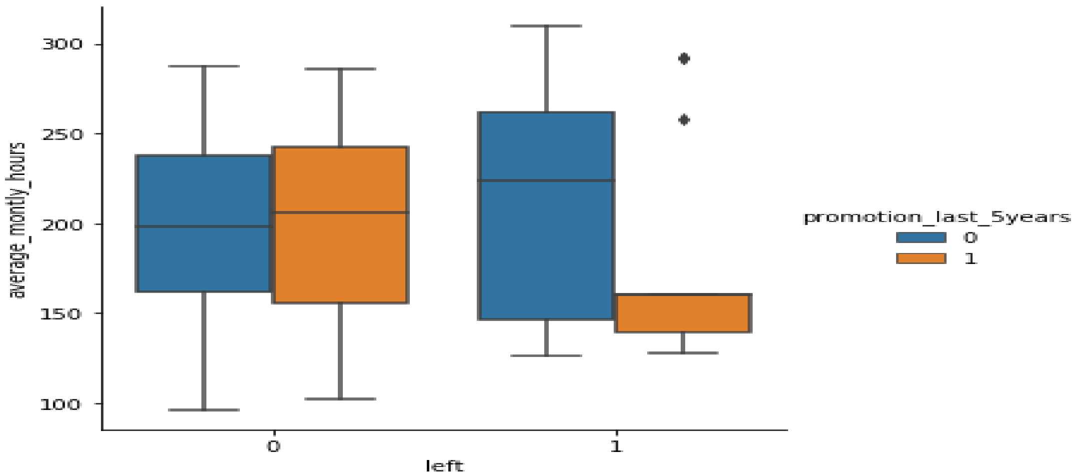
4. Data Visualisation:

4.1 time_spend_company vs promotiom_last_5years with respect to feature left



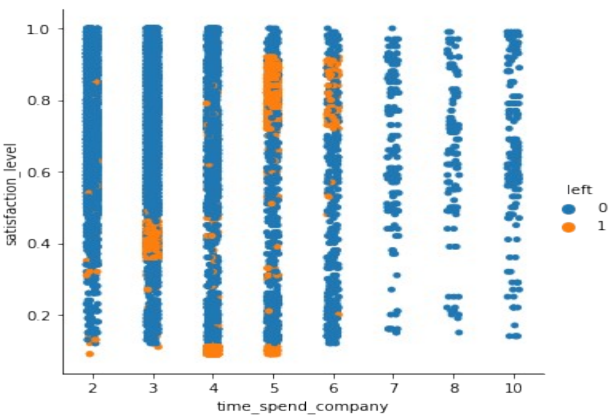
From the above graph we can infer that employees who left the company were mostly those whose median time spend in company was more than other and were still without promotions in last 5 years.

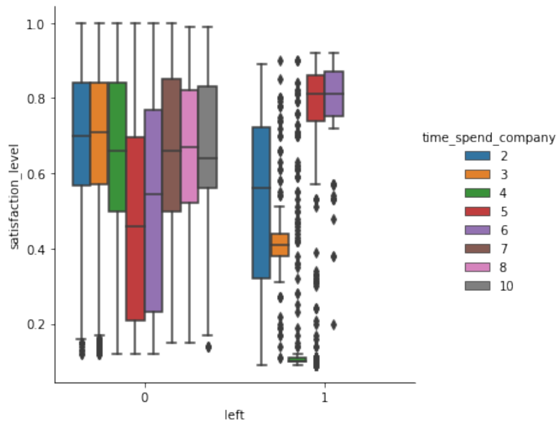
4.2. average_monthly_hours vs promotion_last_5years with respect to feature left



From the above graph we can infer that employees who left the company were mostly those whose median monthly_hours spent in company was more than other and were still without promotions in last 5 years.

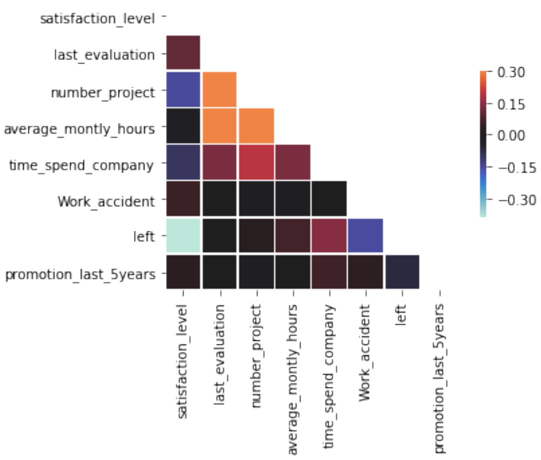
4.3. satisfaction_level vs time_spend_company with respect to feature left





From the above graph we can infer that employees who worked for around 2 to 3 years on average with with slightly less satisfaction level as compared to there counter-part where the majority of people who left the company. We can also infer from the graph that employees with 5 to 6 years of experience of working for that company with high satisfcation level have left the company where as same group of employees with 5 to 6 years of experince with less satisfaction level choosed to stay with company which is difficult to explain.

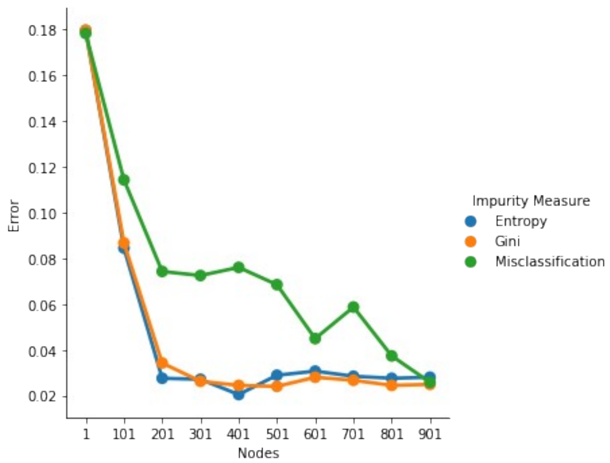
4.4 Correlation Matrix



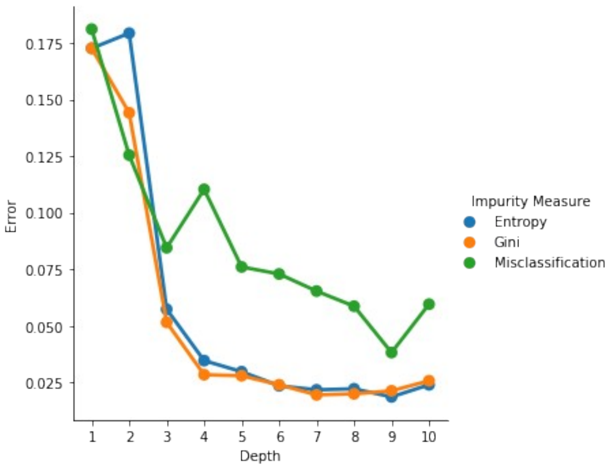
Here we can see the correlation matrix from which it clear that satisfaction level is inversely proportional to employee attrition. Lesser satisfaction level means more tendency to leave the job. We can also see positive correlation between employee atrrition and time spend in company which means employee who have spend long years at company are more prone to make job switch. Also work accident correlation with employee attrition tells that employee who have suffered work accident are likely to leave company

5.

Error vs Nodes of Decision Tree

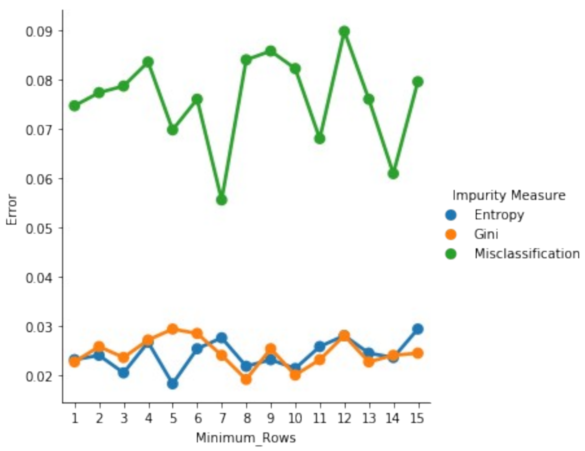


Error vs Depth of Decision Tree



From the above graph we can infer that in our decision tree minimum error occurs at depth of around 8 to 9 and at number of nodes of around 400.

Minimum number of nodes neccessary for spliting vs Error



From the above graph its evident that we get minimum error when we take minimum rows as 8 and once this threshold is hit we stop spliting.

6. How decision tree can handle missing value in test dataset

There can be many approach to handle missing values in test data which are listed below. Here I am assuming binary split decision tree.

- Randomly choose any child ; in case of binary split decision tree we randomly choose to go to left child or right child, although it won’t be a wise thing to do
- Choose the child with more number of samples so that probability of making wrong choice in minimised.
- Recurse to both the child until we reach the leaf node and we count all positive and negative answer we get and return whichever is maximum. Here we also can provide weight to children based on number of rows of dataset on left and right.