CS564 Machine Learning

Project 1: Decision Trees

Introduction

Decision trees are one of the predictive modelling methods and are a "supervised" machine learning practice. In supervised learning, input and corresponding output is fed into the model using training data. Once the model is trained, it can be used to predict results for other data sets. In decision trees, data is continuously split according to a certain factor until a final decision class is obtained.

Building decision tree using ID3 with Entropy and Information Gain

In this project, we use the ID3 algorithm to construct a decision tree using the training data and predict results for the given test data. According to this algorithm, every unvisited attribute is visited in each iteration and one best attribute is selected as the splitting factor. The best attribute is selected based on ***Entropy*** *(measure of disorder)* or ***Information gain*** – lowest entropy or largest information gain.

Project implementation details:

- Iterate through the data set and select the splitting attribute by calculating minimum entropy and maximum information gain.
- At every iteration, we are dropping the previously selected attribute(column) from the dataset and repeating the above step.
- At the end, if there are any attributes left out in the dataset which could not be brought under common decision class, we are assigning the most probable outcome to that branch.

Building decision tree using CART with Gini Impurity

Gini impurity is used by Classification and Regression Tree Algorithm (CART) to determine the splitting factor.

- Iterate through the data set and select the splitting attribute by calculating minimum Gini impurity
- At every iteration, we are dropping the previously selected attribute(column) from the dataset and repeating the above step.

Building decision tree using Misclassification Error

Misclassification error is one of the common cost functions used to build a decision tree, like entropy and Gini impurity.

- Iterate through the data set and select the splitting attribute by calculating minimum error
- At every iteration, we are dropping the previously selected attribute(column) from the dataset and repeating the above step.

Decision tree and post pruning

In this method, we are building the decision tree using ID3 algorithm and then prune the tree using bottom up approach.

- At every iteration, we remove the node at the end and check accuracy.
- If there is an increase in accuracy, we accept the removal.
- Else we reject the removal and repeat the process with other nodes.

Decision tree with Chi square split stopping

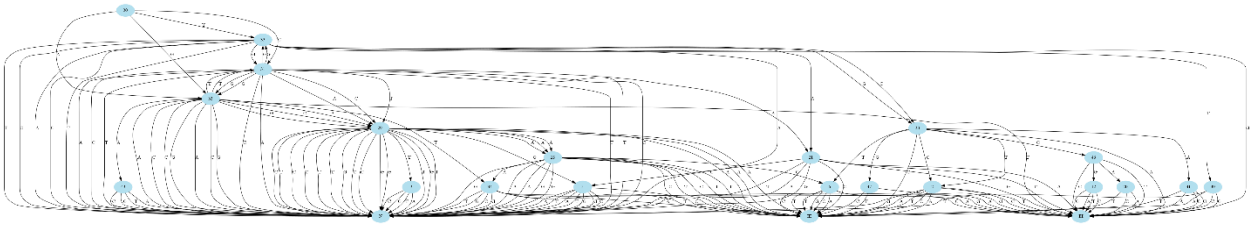
In this method, we have used chi-square test to stop growing the tree earlier (pre-pruning). We have used two confidence intervals – 99% (p-value = 0.01) and 95% (p-value = 0.05) for the test and degree of freedom.

- Calculate the chi-square value at each iteration using actual and expected probability values
- Compare the chi-square value from above step with corresponding p-value obtained from table (with confidence interval and DOF)
- If chi-square Value < p-value, we accept the split and if chi-square Value > p-value, we reject the split.

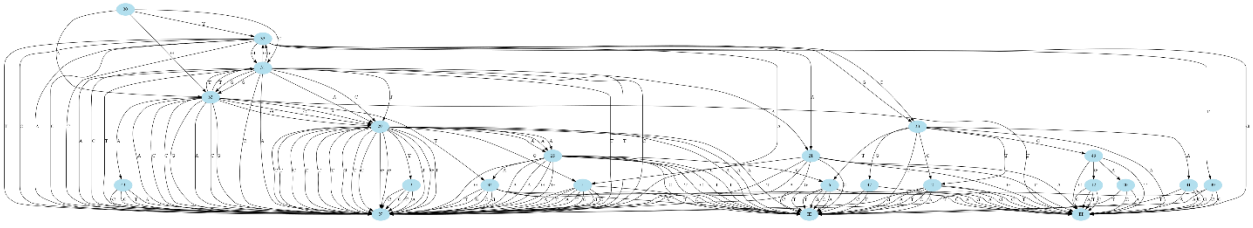
	Percentage								
	2.5	*5*	*50*	*75*	*90*	*95*	*97.5*	*99*	*99.9*
D	*Upper one-sided* α								
	.975	*.95*	*.50*	*.25*	*.10*	*.05*	*.025*	*.01*	*.001*
1	.001	.004	.455	1.32	2.71	3.84	5.02	6.63	10.83
2	.051	.103	1.39	2.77	4.61	5.99	7.38	9.21	13.82
3	.216	.352	2.37	4.11	6.25	7.82	9.35	11.34	16.27
4	.484	.711	3.36	5.39	7.78	9.49	11.14	13.28	18.47
5	.831	1.15	4.35	6.63	9.24	11.07	12.83	15.09	20.52
6	1.24	1.64	5.35	7.84	10.64	12.59	14.45	16.81	22.46
7	1.69	2.17	6.35	9.04	12.02	14.07	16.01	18.47	24.32
8	2.18	2.73	7.34	10.22	13.36	15.51	17.53	20.09	26.12
9	2.70	3.33	8.34	11.39	14.68	16.92	19.02	21.67	27.88

Results

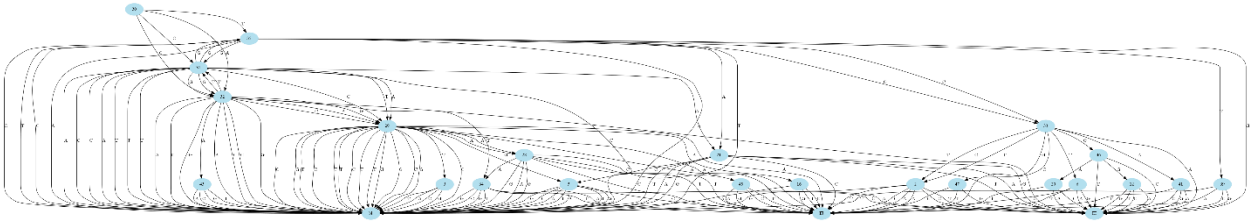
CART with 95% confidence interval:



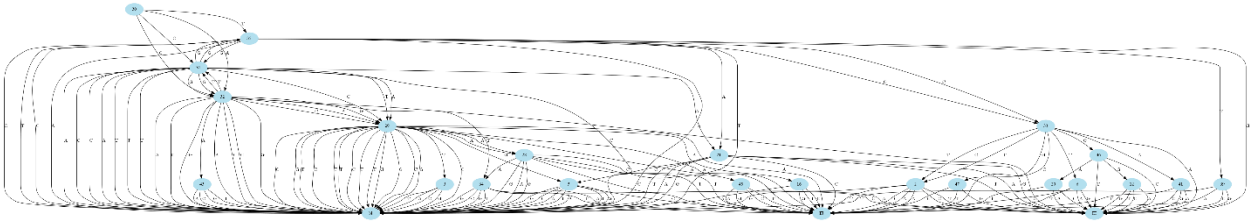
CART with 99% confidence interval:



ID3(E & IG) with 95% confidence interval:



ID3(E & IG) with 99% confidence interval:



Training dataset and calculating accuracy

From the available dataset with Type result, $3/4^{th}$ portion is used to train our decision tree. The rest of the data is used to predict result class and is compared to the actual result set we have from the data, to get an estimate of accuracy.

Prediction using decision tree

Once the decision tree is built using one of the above methods, we use the predict method to get results for the test data. In this function, test data and decision tree are passed as parameters. For every row in the data set, tree is traversed, and decision class is obtained. This function lets us traverse the tree while looking for a match in the branch searched (A, G, C, T) and the branch of the tree.

Implementation

The algorithm is implemented in python 3.8.1. The libraries used are Pandas (handling dataset), Graphviz (Visualizing the tree), NumPy (calculating log values).

Tree:

The tree data structure is stored as an object of class tree. Tree object consists:

- *tree.data* (Node)
- *branches* (List of branch objects)

--The branch object consists of a *branch value* and a *tree* object

Build tree function:

This function finds highest Information gain/lowest Gini impurity/lowest Misclassification Error, depending on the algorithm used and calls itself recursively passing the sub dataset to the function thus building a tree.

Accuracy values obtained *(from Kaggle)*

ID3 (Entropy & Information Gain):

- 89.6%

ID3 (Entropy & Information Gain) and post pruning:

- 87.931%

CART with Gini Impurity:

- 87.216%

CART with Gini Impurity and post pruning: 90.6

Misclassification Error: 87.6 our

Misclassification Error and post pruning: 90.87 our

ID3 (Entropy & Information Gain) with Chi square split stopping and post pruning:

- 90.2461% *(*95% confidence interval)
- 91.386% (99% confidence interval)

ID3 (Entropy & Information Gain) with Chi square split stopping:

- 88.46% (99% confidence interval)

CART (Gini Impurity) with Chi square split stopping and post pruning:

- 91.806% *(*95% confidence interval)
- **92.857%** (99% confidence interval)

CART (Gini Impurity) with Chi square split stopping:

- 87.18%, 88.2% our code (99% confidence interval)

Misclassification Error with Chi square split stopping and post pruning:

- 91.6% our, 89.705% (99% confidence interval)

Result:

From the tests we have observed that, best results are seen while using CART (Based on Gini index) and implementing Chi Square test (with 99% confidence) to stop splitting and post pruning the resulting tree

with an accuracy of 92.857%. The next best results where observed from the same configuration but with 95% confidence interval.

ID3 using entropy gave the next best outcome, resulting in an accuracy of 91.38% under the configuration of Chi square test and post pruning the tree.

The decision tree built using the miss classification error resulted in the least useful tree which resulted in an accuracy of 89.7%.