CS771 | Introduction to Machine Learning Assignment 2 - Report

Hritik Kumar
210451
hritik21@iitk.ac.in

Rishav Dev
210847
rishavd21@iitk.ac.in

Kamal Kishor
210483
kamalk21@iitk.ac.in

Deepanshu
210311
deepanshud21@iitk.ac.in

Akansha patel
210081
akanshap21@iitk.ac.in

Amisha Patel
210119
amishap21@iitk.ac.in

1 Question 1

Criteria for Decision Tree Algorithm

When building a decision tree, the algorithm will recursively split the data into subsets based on certain features and their values, in order to separate the data points into classes. At each node of the tree, the algorithm tries to find the feature and value that results in the best separation of the data points into the different classes. The entropy of a node is a measure of how mixed the data points in that node are in terms of their classification. The formula for entropy is:

$$\textbf{Entropy} = - \sum [p_i log_2(p_i)]$$

where p_i is the proportion of data points in the node that belong to class i. The entropy is minimized when all the data points in the node belong to the same class (i.e., the node is pure). We first tried with the Gini index which is an alternative measure of impurity that is often used in decision trees and Random Forest algorithms. It quantifies the level of diversity or impurity within a set of data points, like entropy. For a classification problem with K classes and N data points in a node, the Gini index is calculated as the probability of incorrectly classifying a randomly selected data point from the set. This is expressed as 1 minus the sum of the squared proportion of data points in each class.

$$\textbf{Gini} = 1 - \sum [(p_i)^2]$$

A node with a Gini index of 0 indicates that all data points belong to the same class, whereas a node with a Gini index of 1 signifies that the data points are evenly distributed across all classes, resulting in maximum impurity.

In decision trees and Random Forest algorithms, the split with the lowest Gini index is selected as the optimal split for each node, as it represents the highest degree of purity. The Gini index can be used as an alternative to entropy as a measure of impurity in Random Forest algorithms.

We also considered the Log loss index, which is commonly used in classification problems to measure the accuracy of probabilistic predictions. Unlike the Gini index, which measures impurity, the log loss index evaluates the uncertainty of predictions. For a classification problem with K classes and N data points, the log loss index is calculated by taking the negative logarithm of the predicted probabilities for the true classes:

$$\textbf{Log Loss} = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{K} \mathrm{y}_{ij} \log(p_{ji})$$

where y_{ij} is a binary indicator (0 or 1) if the class label i is the correct classification for the data point j, and p_{ji} is the predicted probability of data point j belonging to class i.

1.1 HyperParameter Tuning

To improve the performance of our decision tree model, we conducted manual hyperparameter testing. This process helps us find the optimal combination of hyperparameters, potentially enhancing various aspects such as training time, testing time, and memory usage. The parameters under consideration include:

Criterion:

We tested for **gini**, **log loss**, and **entropy** and found that the Gini criterion provides the best accuracy, training time, and testing time.

Max Depth:

We tested [**None, 10, 20, 30, 40, 50**] and found that the default value (None) gives the best accuracy.

Min Samples Split:

We tested [**2, 5, 10**] and realized that the default value (2) provides the best accuracy and memory efficiency.

Max Features:

We tried [**None, sqrt, log2**] and realized that **sqrt** provides the best training time.

Lookahead Depth:

We tested different values and found that 20 provides the best training time.

These parameters were chosen based on their ability to control the complexity and performance of the decision tree. By tuning these parameters, we aimed to find a balance between bias and variance, leading to a more robust and efficient model.