Name: Mujahid Ahmad

Assignment Project

Decision Trees Classifier:

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical tree structure, which consists of a root node, branches, internal nodes, and leaf nodes.

Accuracy: It measures the proportion of correctly classified instances among all instances. It's a simple and intuitive metric but can be misleading with imbalanced datasets.
Formula: (TP + TN) / (TP + TN + FP + FN)

Recall (Sensitivity): It measures the ability of the model to correctly identify positive instances. It's particularly important when the cost of false negatives is high, like in medical diagnoses.
Formula: TP / (TP + FN)

Precision: It measures the proportion of true positive predictions among the instances predicted as positive. It's important when the cost of false positives is high, as in spam email classification.
Formula: TP / (TP + FP)

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balanced view of the model's performance, considering both false positives and false negatives.
Formula: 2 * (Precision * Recall) / (Precision + Recall) Time Taken: Time taken refers to the time required for model training (including hyperparameter tuning) and making predictions on new data.

Advantages

1. Simple to understand, interpret and visualize.
2. Decision trees implicitly perform feature selection.
3. Can handle both numerical and categorical data. Can also handle multi-output problems.
4. Decision trees require relatively little effort from users for data preparation.
5. Nonlinear relationships between parameters do not affect tree performance.

Disadvantages

1. Decision tree learners can create over-complex trees that do not generalize the data well, i.e, they can easily lead to overfitting of the data.
2. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called variance, which needs to be lowered by methods like bagging and boosting.

3. Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

Attributes:

- **Information Gain**
- **Gini Index**

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy (each feature)

I.G = $-[\left(\frac{P}{P}+N\right) Log_2(P/P+N) + \left(\frac{N}{P}+N\right) Log_2(N/P+N)]$

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no) OR

The mathematical formula for entropy is:

$$E(S) = -p_{(+)}\log p_{(+)} - p_{(-)}\log p_{(-)}$$

Where,

- **S= Total number of samples**
- **P(yes)= probability of yes**
- **P(no)= probability of no**

2. Gini Index:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

Gini Index= 1- $\sum_j P_j^2$

Gain = I.G – E(S)

Random Forest Classifier:

Random forest is a supervised learning algorithm. The “forest” it builds is an ensemble of decision trees, usually trained with the bagging method. The general idea of the bagging method. is that a combination of learning models increases the overall result.

Advantages:

1. It reduces overfitting in decision trees and helps to improve the accuracy.
2. It is flexible to both classification and regression problems.
3. It works well with both categorical and continuous values.
4. It automates missing values present in the data.
5. Normalizing data is not required as it uses a rule-based approach.

Disadvantages:

1. It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
2. It also requires much time for training as it combines a lot of decision trees to determine the class.
3. Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

Implementation in Scikit-learn:

For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

- ni_j= the importance of node j
- w_j = weighted number of samples reaching node j
- C_j= the impurity value of node j
- left(j) = child node from left split on node j

- right(j) = child node from right split on node j

The importance for each feature on a decision tree is then calculated as:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

- fi_i= the importance of feature i
- ni_j= the importance of node j

These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

The final feature importance, at the Random Forest level, is its average over all the trees. The sum of the feature's importance value on each tree is calculated and divided by the total number of trees:

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

- $RFfi_i$= the importance of feature i calculated from all trees in the Random Forest model
- $normfi_{ij}$= the normalized feature importance for i in tree j
- T = total number of trees

Implementation in Spark

For each decision tree, Spark calculates a feature's importance by summing the gain, scaled by the number of samples passing through the node:

$$fi_i = \sum_{j:nodes\ j\ splits\ on\ feature\ i} s_j C_j$$

- fi_i = the importance of feature i
- s_j = number of samples reaching node j
- C_j = the impurity value of node j

To calculate the final feature importance at the Random Forest level, first the feature importance for each tree is normalized in relation to the tree:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

- $normfi_i$ = the normalized importance of feature i
- fi_i = the importance of feature i

Then feature importance values from each tree are summed normalized:

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\sum_{j \in all\ features, k \in all\ trees} normfi_{jk}}$$

- $RFfi_i$= the importance of feature i calculated from all trees in the Random Forest model
- $normfi_j$= the normalized feature importance for i in tree j

Extra Trees Classifier:

Extra Trees (Extremely Randomized Trees) is an ensemble machine learning algorithm that's closely related to Random Forest. It's used primarily for classification and regression tasks. Like any algorithm, Extra Trees has its own set of advantages and disadvantages:

Advantages of Extra Trees Classifier:

1. Reduced Overfitting: Extra Trees introduces additional randomness by using random splits for the features at each node, which can help reduce overfitting compared to regular decision trees.

2. Less Sensitive to Hyperparameters: Extra Trees is less sensitive to hyperparameters compared to Random Forests. You don't need to fine-tune the parameters extensively.

3. Handles Noisy Data: Extra Trees can handle noisy and irrelevant features better than traditional decision trees due to its randomized feature selection.

4. Feature Importance: Extra Trees provides a measure of feature importance, which can help you understand which features contribute most to the model's predictions.

Disadvantages of Extra Trees Classifier:

1. Lack of Interpretability: Just like other ensemble methods, the final model of Extra Trees is not easily interpretable. It's a black-box model, making it harder to understand the decision-making process.

2. Overfitting with Small Datasets: Despite being generally less prone to overfitting compared to individual decision trees, Extra Trees can still overfit if the dataset is too small or if the number of trees is too high.

3. Imbalanced Data: Extra Trees might struggle with imbalanced datasets since it doesn't handle class imbalances as effectively as other algorithms specifically designed for this, like Balanced Random Forests.

4. Slower Prediction: Predictions can be slower compared to some simpler models due to the aggregation of multiple

	Decision Trees (DT)	**Random Forest (RF)**	**Extra Trees (EXT)**
Accuracy	0.9701492537313433	0.9738805970149254	0.9738805970149254
Recall	0.9848484848484849	0.9848484848484849	0.9848484848484849
Precision	0.9558823529411765	0.9629629629629629	0.9629629629629629
F1 Score	0.9701492537313432	0.9737827715355806	0.9737827715355806
Time Taken	0.00539789997856132	0.4138319999910891	0.2652988999907393

Conclusion:

The **Decision Trees** offers a straightforward and interpretable solution, often preferred by Data Scientists. Upon comparison, **the Decision Tree Classifier** is sufficient for producing effective results. The accuracy of Decision Tree Classifier is high, operates efficiently, Similarly, the time perform efficiently of **Decision Tree Classifier** is lower than Random Forest Classifier and Extra Tree Classifier. Consequently, selecting the Decision Tree Classifier emerges as a more favorable option compared to the alternatives. Taking an overall perspective, the Decision Tree Classifier demonstrates its excellence with an impressive 97% accuracy rate and quicker processing times. Consequently, selecting the Decision Tree Classifier emerges as a more favorable option compared to the alternatives. Taking an overall perspective, the Decision Tree Classifier demonstrates its excellence with an impressive 97% accuracy rate and quicker processing times.