Experiment No. 5

Aim: Data Clustering Using Decision Tree Algorithm for Business Intelligence

Objective: to introduce students to the practical implementation of data clustering using the decision tree algorithm for business intelligence applications.

By the end of the session, students should be able to understand the concepts of decision tree clustering and apply it to analyze and interpret business datasets.

Theory:

Machine learning algorithms are used in almost every sector of business to solve critical problems and build intelligent systems and processes. Supervised machine learning algorithms, specifically, are used for solving classification and regression problems. one of the most popularly used supervised learning algorithms: decision trees in Python.

What is a Decision Tree?

A decision tree is a tree-based supervised learning method used to predict the output of a target variable. Supervised learning uses labeled data (data with known output variables) to make predictions with the help of regression and classification algorithms. Supervised learning algorithms act as a supervisor for training a model with a defined output variable. It learns from simple decision rules using the various data features. Decision trees in Python can be used to solve both classification and regression problems—they are frequently used in determining odds.

Advantages of Using Decision Trees

- Decision trees are simple to understand, interpret, and visualize
- They can effectively handle both numerical and categorical data
- They can determine the worst, best, and expected values for several scenarios
- Decision trees require little data preparation and data normalization
- They perform well, even if the actual model violates the assumptions

Important Terms Used in Decision Trees

1. Entropy: Entropy is the measure of uncertainty or randomness in a data set. Entropy handles how a decision tree splits the data.

It is calculated using the following formula:

$$\sum_{i=1}^{k} P(value_i).log_2(P(value_i)$$

2. Information Gain: The information gain measures the decrease in entropy after the data set is split.

It is calculated as follows:

IG(Y, X) = Entropy (Y) - Entropy (Y | X)

3. Gini Index: The Gini Index is used to determine the correct variable for splitting nodes. It measures how often a randomly chosen variable would be incorrectly identified.

4. Root Node: The root node is always the top node of a decision tree. It represents the entire population or data sample, and it can be further divided into different sets.

5. Decision Node: Decision nodes are subnodes that can be split into different subnodes; they contain at least two branches.

6. Leaf Node: A leaf node in a decision tree carries the final results. These nodes, which are also known as terminal nodes, cannot be split any further.

Building a Decision Tree in Python

We'll now predict if a consumer is likely to repay a loan using the decision tree algorithm in Python. The data set contains a wide range of information for making this prediction, including the initial payment amount, last payment amount, credit score, house number, and whether the individual was able to repay the loan.

2. Dataset Exploration:

- Load the dataset for the clustering task.
- Explore the dataset to understand its structure, features, and distribution.
- Preprocess the dataset if necessary (e.g., handling missing values, scaling features).

3. Data Preparation:

- Select relevant features for clustering.
- Standardize or normalize the data if needed.

4. Building the Decision Tree Model:

- Import the necessary libraries (scikit-learn).
- Create an instance of the DecisionTreeClassifier for clustering.
- Train the decision tree model using the dataset.

5. Visualizing Clusters:

- Visualize the decision tree structure to understand how data is partitioned into clusters.
- Plot the clusters using appropriate visualization techniques (e.g., scatter plot, dendrogram).

6. Interpretation and Analysis:

- Analyze the clusters generated by the decision tree algorithm.
- Interpret the results in the context of business intelligence objectives (e.g., customer segmentation, market analysis).
- Discuss the implications of the clustering results for business decision-making.

7. Fine-tuning the Model:

- Discuss parameters such as tree depth, criterion, and splitting strategy.
- Experiment with different parameter values to optimize the clustering performance.

Hyperparameters for decision tree

The performance of a machine learning model can be improved by tuning its hyperparameters. Hyperparameters are those parameters that the user has to set in advance. They are not learned by the data during training.

Some of the most common hyperparameters for a decision tree are:

Criterion: This parameter determines how the impurity of a split will be measured. Possibilities are ‘gini’ or ‘entropy.

Splitter: How the decision tree searches the features for a split. The default is set to ‘best’ meaning for each node, the algorithm considers all the features and chooses the best split. If it is set to random, then a random subset of features will be considered. The split will be made by the best feature within the random subset. The size of the random subset is determined by the ‘max_features’ parameter.

Max_Depth: This determines how deep the tree will be. The default is none and this often results in overfitting. The max_depth parameter is one of the ways in which we can regularize the tree, or limit the way it grows to prevent over-fitting.

Min_samples_split: The minimum number of samples a node must contain to consider splitting. The default is set to 2. This again is another parameter to regularize the decision tree.

Max_features: The number of features to consider when looking for the best split. By default, the decision tree will consider all available features to make the best split.

Iris dataset:
The dataset consists of 150 samples of iris flowers, each with four features:

- Sepal length (in centimeters)
- Sepal width (in centimeters)

- Petal length (in centimeters)
- Petal width (in centimeters)

The goal of the dataset is to classify iris flowers into one of three species:

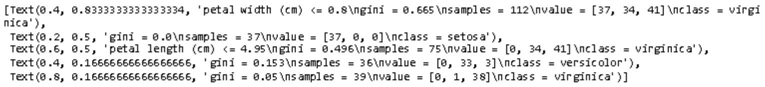
- Setosa
- Versicolor
- Virginica

Applications: Decision Tree Applications

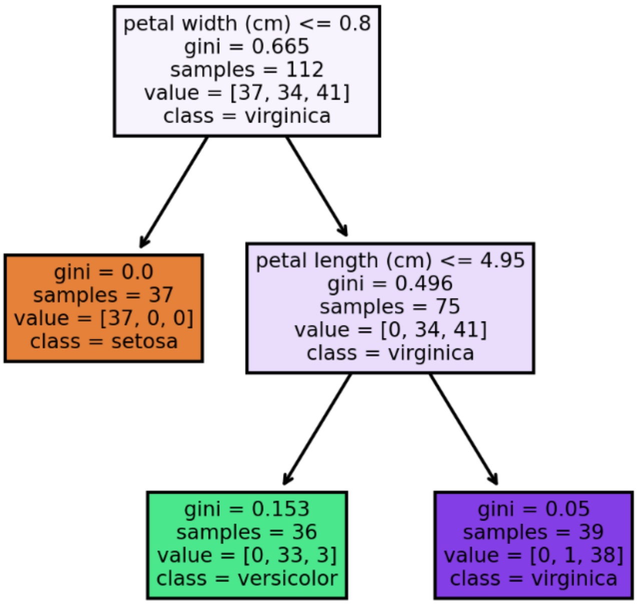
1. A decision tree is used to determine whether an applicant is likely to default on a loan.
2. It can be used to determine the odds of an individual developing a specific disease.
3. It can help ecommerce companies in predicting whether a consumer is likely to purchase a specific product.
4. Decision trees can also be used to find customer churn rates.

Input: iris dataset

Output:



```
[Text(0.4, 0.8333333333333334, 'petal width (cm) <= 0.8\ngini = 0.665\nsamples = 112\nvalue = [37, 34, 41]\nclass = virgi
nica'),
 Text(0.2, 0.5, 'gini = 0.0\nsamples = 37\nvalue = [37, 0, 0]\nclass = setosa'),
 Text(0.6, 0.5, 'petal length (cm) <= 4.95\ngini = 0.496\nsamples = 75\nvalue = [0, 34, 41]\nclass = virginica'),
 Text(0.4, 0.16666666666666666, 'gini = 0.153\nsamples = 36\nvalue = [0, 33, 3]\nclass = versicolor'),
 Text(0.8, 0.16666666666666666, 'gini = 0.05\nsamples = 39\nvalue = [0, 1, 38]\nclass = virginica')]
```



Conclusion:

In conclusion, the Decision Tree algorithm offers a powerful tool for data clustering in business intelligence, allowing for the effective analysis and interpretation of complex datasets to derive actionable insights.

Outcome:

The practical implementation of Decision Tree clustering facilitates a deeper understanding of business datasets, enabling students to apply decision tree algorithms for segmentation and analysis tasks in real-world scenarios

Questions:

1. How does the Decision Tree algorithm contribute to business intelligence applications?
2. What are the advantages of using Decision Trees for data clustering in comparison to other methods?
3. Can you explain the significance of hyperparameters in fine-tuning a Decision Tree model?
4. What are some real-world applications of Decision Trees in business and industry?

Answers:

1. The Decision Tree algorithm is integral to business intelligence applications because it offers a transparent and interpretable way to analyze data. By visually representing decision paths, it helps in understanding complex relationships within the data, aiding in better decision-making processes for businesses.

2. Decision Trees excel in data clustering due to several advantages over other methods. Firstly, they are simple and easy to interpret, which is crucial for understanding the structure of the data. Secondly, Decision Trees can handle both numerical and categorical data efficiently, unlike some other clustering methods that might struggle with categorical variables. Finally, Decision Trees inherently partition the data into clusters based on decision rules, making them effective for clustering tasks.

3. Hyperparameters play a significant role in fine-tuning Decision Tree models. These parameters control the complexity and behavior of the tree, such as its depth, minimum samples per leaf, and splitting criteria. By adjusting these hyperparameters, practitioners can optimize the model's performance, prevent overfitting, and enhance its ability to generalize to unseen data.

4. Decision Trees find wide-ranging applications across industries. In finance, they are used for credit scoring to assess the creditworthiness of individuals. In healthcare, Decision Trees aid in medical diagnosis by analyzing patient data to identify potential diseases or conditions.

In marketing, they help in customer segmentation, allowing businesses to tailor marketing strategies to specific customer groups. In manufacturing, Decision Trees are utilized for quality control to identify factors affecting product defects and optimize production processes accordingly.