Application of Decision Tree Classifier for Iris Species Classification: A Comparative Analysis with Regression Models

Pooja S
Student, Dept. of CSE
Ramaiah University of Applied Sciences
Bengaluru, India

***Abstract*— This report utilizes a Decision Tree Classifier on the well-known Iris dataset, which consists of 150 samples of iris flowers described by four features: sepal length, sepal width, petal length, and petal width. The objective is to classify these samples into one of three species: Iris-setosa, Iris-versicolor, and Iris-virginica. The Decision Tree Classifier is chosen due to its interpretability, ability to handle both numerical and categorical data, capability to model non-linear relationships, and provision of feature importance scores. The methodology includes steps for importing, cleaning, and examining the data, preprocessing, feature engineering, data splitting, model training, and evaluation. Key steps involve converting categorical labels to numeric codes, standardizing feature scales, and creating a new feature, 'petal_area', to enhance model performance. The model's effectiveness is assessed using metrics such as accuracy, precision, recall, and F1 score, which measure the quality of classification. A comparison with regression models highlights that classification is more suitable for tasks involving the assignment of instances to distinct categories, while regression is focused on predicting continuous numeric values. The Decision Tree Classifier is well-suited for the Iris dataset's classification task, emphasizing its appropriateness over regression for categorizing iris species.**

Keywords—Machine Learning, Classification, Decision Tree, Iris Dataset, Data Pre-processing, Feature Engineering.

I. INTRODUCTION

Machine learning has grown to become a central tool within many disciplines to assist in inferring meaning from data. One classic application of machine learning is classification of iris species on the basis of given Iris dataset borrows much study in the pattern recognition domain. This paper explores classifying iris species using Decision Tree based on their morphological features.

The Iris dataset contains measurements of the length and width of sepals and petals for three iris species: Setosa, Versicolor, and Virginica. This study seeks to provide a model that will give an accurate prediction of the species, given these measurements.

A Decision Tree is selected for this task because it is simple, interpretable, and very effective with both numerical and categorical data. This paper embeds, as part of this study, data preprocessing stages like cleaning the data and its visualization, which is an important step toward better performance of the model. It is expected that the accuracy and interpretability of this model will be improved by using techniques for feature engineering and feature selection.

These results show that high accuracy in iris species classifier can be obtained with a Decision Tree classifier. It is an indication that such classier will do fine in solving such problems. This research points to the practical application aspects of machine learning techniques and therefore proves a very good ground for further research and development in these areas.

II. MODEL SELECTION

The widely recognized Iris dataset for the project, containing 150 samples of iris flowers, where every iris sample is described by four features: Sepal Length, Sepal Width, Petal Length, and Petal Width. These four features describe the samples into three different species: Iris-setosa, Iris-versicolor, and Iris-virginica. Decision Tree Classifier model has been selected for this task for the following reasons:

1. Interpretability: Compared to other complex models like Neural Networks or SVMs, Decision Trees are quite interpretable. In this model, normally, the decision process would be organized as a tree structure, where each internal node denotes a "decision" on one feature; every branch represents the outcome of that decision and every leaf node holds a class label. Very clear on how the prediction is derived from the model, this can be very helpful for beginners and other stakeholders who need to interpret the results.

2. Feature Handling: Both numeric and categorical features can be given as the input with a decision tree. In the Iris dataset, the numerical features are the length of sepals, widths of sepals, length of petals, and width of petals, and the target feature is categorical and contains species. This makes preprocessing a lot easier since it becomes apparent that, by default, decision trees are able to adjust to such data without the necessity of feature transformations or scaling.

3. Non-linear Relationships: A model decision tree can capture complex non-linear relationships between features with respect to the target variable. This would culminate in those datasets in which the decision boundary between classes may not be linear. For example, the separation among different species of flowers by measurements cannot take a linear feature. In the other way, the capability of partitioning the feature space into regions that best separate the classes makes this work pretty well in non-linear relationship cases.

4. Feature Importance: A decision tree provides a feature importance score on how significantly features contribute to accurately predicting the target value. Hence, the feature

selectivity approach considers only those variables which contribute hugely to the prediction of the target variable, such as major measurements in telling apart Iris species.

5. Performance: The problems of overfitting and underfitting exist in tree-based classifiers. These can be partially alleviated by pruning the trees, reducing their depth, or using ensemble methods, such as the Random Forest, which uses multiple trees and takes an average of the predictions. Actually, for a relatively small and well-prepared dataset like Iris, correctly tuned decision trees can really vie in performance.

6. Scalability: Decision trees are computationally efficient both during training and prediction phases, most especially against more complex models like deep neural networks. This efficiency feature works quite well with small datasets like Iris, where training time is not a big deal, but one needs quick experiments and iteration over parameters.

Finally, through its better model interpretability, better handling of numerical and categorical features, better modelling ability of nonlinear relationships, better assessment of feature importance, and better performance in contemporary advances when well-tuned, the Decision Tree Classifier is justified in this Iris Species Classification task.

III. METHODOLOGY

For the Iris Species dataset, the target variable to be predicted is the species of iris flower based on the given measurements. The target variable typically consists of three classes:

1. Setosa
2. Versicolor
3. Virginica

These classes represent different species of iris flowers. This classification model will predict which species each iris observation belongs to based on its sepal length, sepal width, petal length, and petal width.

A. Data Loading

The process started by loading the Iris dataset from a CSV file to a Pandas Data Frame. The iris dataset contains measures on sepal length, sepal width, petal length, and petal width of three species of iris flowers. It is loaded into a Data Frame to enable efficient manipulation and analysis of the dataset.

B. Data Cleaning

First of all, a number of cleaning steps had to be done to enable the dataset for analysis. This was done by standardizing column names for uniformity and clarity. Specifically, column names were changed to sepal_length, sepal_width, petal_length, petal_width, and class. Column name changes make column names descriptive and without leading or trailing spaces.

Finally, numeric codes were substituted for categorical class labels in the column 'class,' which names the species of Iris flowers. This conversion was done using LabelEncoder, which converts categorical text labels into numerical codes. This is very important because most machine learning algorithms require numeric input for training.

Further, all numeric columns needed to be changed into the float type. The conversion is necessary for mathematical operations and model computations to be accurate.

C. Data Visualisation

The dataset was visualized to bring out insights into the distribution and relationship of features. A scatter matrix was created where all feature pairs' pairwise relationships are shown. Each scatter plot in the matrix is color-coded by species, showing how different species cluster based on their features, which goes a long way in understanding class separability and how features correlate with one another.

A heatmap of Pearson's correlation coefficients was also plotted to see the linear relationships among numeric features. This heatmap describes the strength and direction of correlations, helping in feature selection and interdependencies between them.

D. Data Preprocessing

Feature scaling was applied to all the features so that each of them would have a mean of zero and a standard deviation of one. Feature scaling standardizes the features to have a mean of zero and a standard deviation of one. This is based on the fact that many machine learning algorithms often operate better when features are in similar scales, hence eliminating dominance of some by others due to their scale.

E. Feature Engineering

Feature engineering involved the creation of new features that would probably improve the model's performance. For this case, it was the new feature 'petal_area' created by multiplying petal length by petal width. This new feature may capture additional information to help improve distinction between species by the model.

F. Data Splitting

In order to assess the model, the dataset had to be divided into a training and a testing set. A training set is used to build the model, and a test set is used to provide an estimate of how well it generalizes to unseen data. The 80-20 split used 80% of the data for training and 20% for testing.

G. Model Training and Evaluation

A Decision Tree Classifier was chosen for training because it is interpretable and very effective in classification tasks. The model was then trained on the training set to learn patterns and relationships between features and their corresponding class labels.

The predictions for the test set were then used to evaluate the trained model. Performance metrics, including accuracy, precision, recall, and F1 score, were computed to evaluate the model's effectiveness. Accuracy measures the proportion of correct predictions. Precision measures the proportion of true positive predictions out of all positive predictions, and recall measures the proportion of actual positives correctly identified. The F1 score is a balanced measure of both precision and recall.

These metrics give insight into how accurate the model will be in classifying Iris species and are critical to understanding its real-world working.

H. Metrics Used to Evaluate the Model's Perfromance

1. **Accuracy**: Measures the ratio of correctly predicted instances to the total instances. Formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision**: Measures the ratio of true positive instances to the sum of true positive and false positive instances. Formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Recall**: Measures the ratio of true positive instances to the sum of true positive and false negative instances. Formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. **F1 Score**: The harmonic mean of precision and recall. Formula:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

IV. COMPARISON WITH REGRESSION MODEL

The two fundamental types of predictive modeling tasks are classification and regression, which differ in methodologies and objectives.

When choosing between classification and regression models for analysis, the nature of the target variable should be a critical consideration. For example, algorithms designed for classification perform well when the target variable consists of discrete categories or classes. In the Iris dataset, for instance, flowers are categorized into one of three species: 'Iris-setosa', 'Iris-versicolor', or 'Iris-virginica', with each flower being assigned to exactly one class. The discrete nature of the target variable makes classification the most suitable approach, directly addressing the need to assign data to predefined classes.

In contrast, regression models predict continuous numeric values. They are evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score, all of which gauge prediction accuracy by how close the predicted values are to the actual numeric values. Although these metrics are useful for assessing the accuracy of continuous predictions, they do not measure the ability to categorize data into distinct classes. Therefore, these regression metrics are more relevant to tasks involving numeric predictions rather than categorical classifications.

Classification models work by identifying decision boundaries or rules that effectively separate different classes based on input features. For example, in the Iris dataset, a Decision Tree Classifier generates decision rules that distinguish between species based on sepal and petal measurements. The primary goal is to map each data point to the correct predefined category, which is crucial when sorting items into specific groups, such as identifying flower species.

Conversely, regression models aim to fit a continuous function to the data to minimize the difference between predicted and actual values. For instance, Linear Regression would be used to estimate petal length based on other characteristics. This method is suitable for applications requiring numeric predictions rather than categorical classifications.

In real-world scenarios, classification models excel when data points need to be assigned to distinct categories. For example, the Iris dataset, which involves classifying iris flowers into different species, is inherently suited for classification techniques. These models provide performance metrics like accuracy, precision, and recall, which are essential for evaluating how well the model distinguishes between classes. This ability is critical for tasks that require categorization rather than numeric prediction, making classification a preferable choice over regression in such contexts.

V. CONCLUSION

This report shows successful implementation of a Decision Tree Classifier on the Iris dataset, containing 150 iris flower samples classified into three species. This Decision Tree has been chosen since it is interpretable and works with numeric and categorical features; moreover, it is nonlinear. Model performance was measured in terms of accuracy, precision, recall, and the F1 score, which proved that the classifier is effective at performing classification tasks. Comparing it with regression, it was evident that the former was more appropriate for classification into different classes. The analysis definitely proves the plus of a Decision Tree Classifier in tasks with discrete outcomes—clear and actionable predictions. The findings validate the efficiency of the Decision Tree and provide a robust approach in which students and practitioners can learn about and apply machine learning models in practical scenarios.

REFERENCES

Ella Siman, Product Marketing Manager and Zanini, A. (2024) *How to train your own AI model: Step-by-step guide*, *Bright Data*. Available at: https://brightdata.com/blog/web-data/train-an-ai-model (Accessed: 14 July 2024).

AI & Insights (2023) *A tutorial on building and training an AI model from scratch*, *Medium*. Available at: https://medium.com/muthoni-wanyoike/a-tutorial-on-building-and-training-an-ai-model-from-scratch-c3adb9dc1912 (Accessed: 14 July 2024).