Machine Learning - Dry Bean classification

Abstract

In the context of dry bean classification, there are many different types and sizes of beans, resulting in a challenging task. Traditional bean classification methods, such as manual inspection and colorimetric analysis, are time-consuming and labor-intensive. Machine learning models can be used to automate the bean classification process and improve accuracy. In this study, we used two machine learning models to classify dry beans: multi-layer perceptron (MLP) and decision trees (DT). We trained both models on a dataset of dry beans with known labels. The MLP model achieved an accuracy of 93% on the test set, while the decision tree model achieved an accuracy of 89%. This study shows that both MLP and decision trees can be used to effectively classify dry beans. MLP achieved higher accuracy than decision trees. This could be because MLP is a more powerful algorithm than decision trees. This study could lead to new automated bean classification systems that are faster, more accurate, and more cost-effective than traditional methods.

1. Introduction

The classification of dry beans based on their physical characteristics is a vital task in the agricultural industry. Accurate and automated classification methods can streamline operations, improve quality control, and ensure consistent labeling. In this context, the application of machine learning algorithms, such as Multi-Layer Perceptron (MLPs) and Decision Tree, has proven effective in solving the dry bean classification problem.

The Multi-Layer Perceptron (MLP) is a versatile neural network architecture that learns complex patterns by adjusting weights and biases between interconnected nodes. It excels in tasks such as classification and regression, with the ability to capture intricate relationships in data. On the other hand, Decision Trees are hierarchical structures that recursively split data based on features, forming interpretable decision paths. They are suitable for visualizing and understanding decision-making processes. While MLPs offer flexibility and nonlinearity, Decision Trees provide transparency and robustness. Both algorithms contribute to the diverse landscape of machine learning techniques, empowering researchers to solve a wide range of classification and regression problems effectively.

2. Dataset

The dry bean classification **problem**, as described in the paper by (Murat Koklu, 2020), aims to classify different varieties of dry beans based on their features. Dry bean size, shape, color, texture, and texture are among the physical characteristics included in this study. The goal is to build a model that classifies beans accurately by analyzing these features.

Agricultural applications can benefit from the ability to accurately classify dry beans in the **context** of this problem because dry beans are widely consumed worldwide. The **dataset** consists of 13,611 dry bean samples, each with a known label. A total of 16 features, 12 dimensions and 4 shape forms are included, which contains geometric properties (such as area, perimeter, length, and width), color-related features (mean, standard deviation, and correlation of RGB channels), and texture-related features (contrast, energy, homogeneity, and correlation calculated from the gray-level co-occurrence matrix).

The **labels** are for 7 different varieties of dry beans (Turkish Standards Institution, 2009): Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira.

Additionally in this dataset, Exploratory Data Analysis (EDA), followed by model prediction and comparison is done. **Data cleaning** is not required since the dataset doesn't contain any null or duplicate values.

This dataset can be **applied** in agriculture monitoring and quality control. Some **limitations** of the dataset include limited variability (only seven labels), beans of different varieties may have similar sizes and shapes, and the dataset may have an imbalanced class distribution. Because the dataset is based on turkeys, it may introduce bias and limit the generalizability of the models.

3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) in the dry bean classification problem involves examining and understanding the dataset to gain insights into its structure, distribution, and relationships between variables.

As many algorithms work with numerical data rather than categorical data, LabelEncoder converts target labels into integer values (0 to n_classes-1). In Fig 1, by creating these scatter plots, it becomes possible to observe the distribution and relationship between the 'Perimeter' and 'EquivDiameter' features for each dry bean variety. It helps in identifying any patterns or differences in these features among different varieties.

Fig 1: Scatter plot

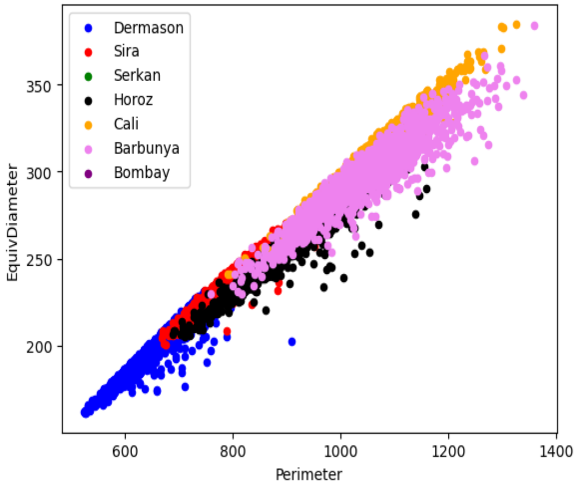
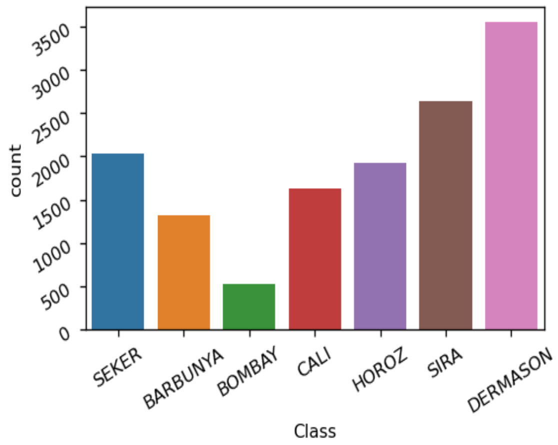


Fig 2: Bar graph



The resulting plot in Fig 2, allows us to observe the relative frequencies of each bean variety and identify potential class imbalances or disparities, which is crucial for determining whether the dataset is balanced or skewed, as it can impact the model's performance in classification tasks.

A heatmap and cluster mapping using correlation is done to understand the relationship between variables and identify patterns or clusters within the data. To standardize the features, we must divide the dataset into training and testing sets, scaling them to unit variance.

4. Model Creation

4.1 Multi-Layer Perceptron (MLP)

The multilayer perceptron (MLP) is an artificial neural network used for supervised learning. It consists of layers that are interconnected, and each performs a small mathematical operation. It is a feedforward neural network, meaning information flows forward. In this study, MLP is used to classify dry bean varieties according to their characteristics based on patterns and relationships found in the data. Input features can be mapped to higher-dimensional spaces with MLP's multiple layers and non-linear activation functions, allowing them to be effectively separated into classes. Each perceptron acts as a linear classifier and creates a hyperplane and the mathematical equation of the hyperplane is:

$$x_n w=0$$

Here x_n is the input feature for data instance n and w is weight vector which include bias (w+b). The output of each perceptron in MLP can be calculated as:

$$z = \sum_{i=1}^{n} x_i w_i + b$$

Where, z is the weighted sum of the inputs and biases, w is the weights associated with each input feature, x is the input features, and b is the bias term.
MLP addresses the **classification task** in the dry bean problem by learning the underlying patterns and relationships in the dataset to classify different varieties of dry beans based on their features. The input layer of neurons receives the input data, and the output layer of neurons produces the output data. The hidden layers of neurons are responsible for learning the relationship between input and output data.

- **Input layer:** 16 parameters from bean data (X1–X16) were used as input parameters.
- **Hidden Layers:** After analyzing the study results, the optimal hidden layer structure was found to be 6.
- **Output Layer**: 7 types of dried beans Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira are numbered O1-O7.

MLPs are trained using backpropagation algorithms. Backpropagation is an iterative algorithm that adjusts the weights of the neurons in the network. MLPs can solve a variety of problems, including classification, regression, and forecasting, so the error between the predicted output and the actual output is minimized. They are successfully used in a variety of applications, including image classification, speech recognition, and natural language processing.
Alternatives to MLP include other neural network architectures like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), as well as traditional machine learning algorithms such as Support Vector Machines (SVMs) or Random Forests. **MLP is chosen** for its ability to learn complex patterns in the data and handle non-linear relationships. It is well-suited for image and pattern recognition tasks, which aligns with the dry bean classification problem. MLP can be sensitive to the choice of hyperparameters, and it may require a large amount of training data to generalize well. It can also be prone to **overfitting** if not properly regularized.

4.1.1 Hyper-parameter setting

A randomized search CV is used to optimize the hyper-parameters:

- **hidden_layer_sizes**: This parameter determines the architecture of the neural network by specifying the number of neurons in each hidden layer. The values are randomly selected integers between 5 and 8, inclusive. Here, the value is set to 6.
- **activation**: The activation function determines the non-linearity of the neurons. The parameter value is chosen from a list of activation function options here tanh is used. 'tanh': Hyperbolic tangent function (f(x) = tanh(x)).
- **solver**: The solver parameter determines the optimization algorithm used to update the weights. The parameter value is chosen from a list of solver options, here 'lbfgs' is used. 'lbfgs': Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm.
- **learning_rate**: This parameter determines how the learning rate changes during training. The parameter value is chosen as 'constant' in this model.
  'constant': The learning rate remains constant throughout training
- **random_state**: This parameter sets the random seed for reproducibility. The values are randomly selected integers between 2 and 8, inclusive. Here, the optimal value is 4.

4.1.2 Model Evaluation

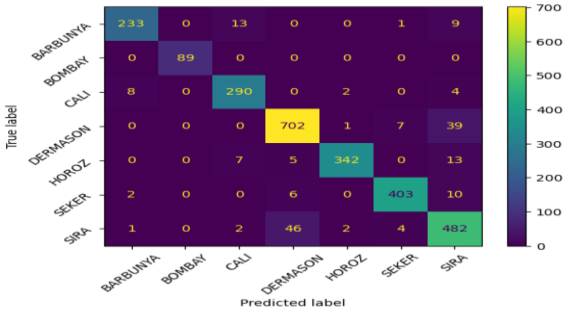
In this report, the evaluation is done using accuracy, precision, recall and F1 score, respectively. The MLP model evaluation is done using performance metrics, confusion matrices, classification report and cross-validation, the modes used in this model.

Performance metrics measure the model's accuracy and ability to correctly classify instances across different classes and provide valuable insights into its performance on a per-class basis and overall performance. **Confusion matrix** compares the values of the labelled class against the predicted class thus it is suitable for multiclass evaluation. It helps to identify the number of test inputs that are correctly classified and misclassified. The **classification report** offers a concise overview of the model's performance for individual classes and helps in identifying classes where the model excels or requires improvement. **Cross-validation** is employed to assess the model's performance on multiple subsets of the data. It helps in estimating the model's generalization ability and provides a more robust evaluation by mitigating the impact of specific train-test splits. Here, for this analysis the value of k is taken as 5. From the classification report in Fig 3, we can see that for Bombay class the model performed well with all metrics values as 1, whereas for class Sira the performed worse with 0.90,0.86,0,0.88 for precision, recall and F1, respectively. For the rest of the classes a good score was obtained for precision, recall and F1 as observed in Fig 3. The K-Fold cross-validation was performed with a k value of 5 and an output obtained with an F1 score of 0.98, 0.97, 0.96, 0.98 and 0.99 on different folds.

Fig 3: Classification Report

Bean Type	Precision	Recall	F1	Support
BARBUNYA	0.97	0.92	0.95	256
BOMBAY	1.00	1.00	1.00	89
CALI	0.94	0.96	0.93	304
DERMASON	0.90	0.96	0.93	749
HOROZ	0.97	0.95	0.96	367
SEKER	0.98	0.96	0.97	421
SIRA	0.90	0.86	0.88	537

Fig 4: Confusion matrix



As shown in Fig 4, the value along the diagonal is correctly predicted and the others are misclassified. Here, Dermason has the highest number of correctly predicted classes i.e., 702 correct predictions, the percentage of correctly predicted value is highest for Bombay while Sira has the lowest percentage. Some misclassifications, however, can be justified by the similarity between features of different varieties of beans. To resolve this issue, it can be addressed by adding more external descriptions and applying a balance to the dataset so that beans can be distinguished better. It is possible to overcome this issue by tuning hyperparameters and selecting features more carefully. In this analysis, I have decided to move forward with the MLP rather than the Decision tree.

Comparing these values with the study conducted by (Murat Koklu,2020), the accuracy, precision, recall and F1 scores have increased since the hyperparameter was optimized.

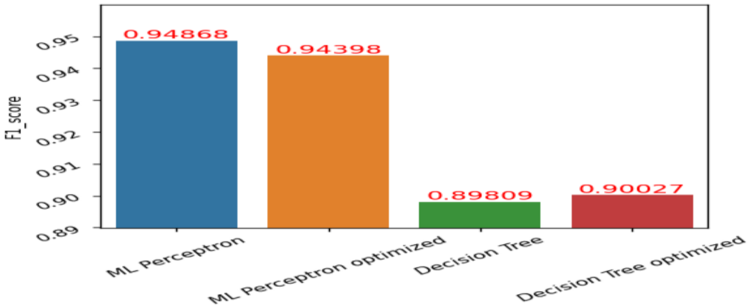
4.2 Extra Task: Decision Tree

A decision tree is a machine learning algorithm used for regression and classification. It creates a hierarchical structure of nodes and branches based on the features of the data. The tree starts with a root node and recursively splits the data at each internal node based on the feature that provides the most information gain or Gini impurity reduction. The splitting process continues until the leaves of the tree are pure or meet certain stopping criteria.

The decision tree model in machine learning is developed using a top-down, recursive approach known as the ID3 (Iterative Dichotomiser 3) algorithm. This algorithm selects the best feature to split the data based on information gain or Gini impurity. Once the decision tree is developed, it can be evaluated using various metrics such as accuracy, precision, recall, and F1-score. Additionally, decision tree performance can be visualized using a confusion matrix. To optimize decision tree's performance, hyperparameters can be tuned through random search. However, decision trees are prone to overfitting, especially if they grow deep and complex. Overfitting occurs when the tree captures noise or specific patterns that are present in the training data but do not generalize well to unseen data.

- **Performance Comparison**:

Fig 5: Comparison of MLP and DT



MLP: The MLP model achieved an accuracy of 0.9331, precision of 0.9473, recall of 0.9411, and F1-score of 0.9439.
Decision Tree: The decision tree model achieved an accuracy of 0.8931, precision of 0.9064, recall of 0.8967, and F1-score of 0.9003.

Comparing the performance of the additional MLP model with the primary decision tree model, we observe that both models achieve similar accuracy and precision. However, the MLP model slightly outperforms the decision tree in terms of recall and F1-score. This indicates that the MLP model captures a slightly better balance between identifying positive instances (recall) and overall classification accuracy (F1-score). However, we are at the disadvantage of less discrimination between bean seeds' features and shape. This may be considered as one of the reasons whereas multiple reasons such as class imbalance, weightage, etc., can also affect MLP performance.

- **Justification for the Method Choice**: The MLP model was chosen as an additional model because of its ability to capture complex patterns and relationships in the data. The dry bean classification problem may have intricate features that cannot be effectively captured by a simple decision tree. By using the MLP, we aim to leverage its non-linear learning capabilities to improve the classification accuracy.

The decision tree and multi-layer perceptron (MLP) are two machine learning models used for the dry bean classification problem. The decision tree creates a hierarchical structure based on features, while the MLP consists of interconnected layers of neurons. The decision tree is interpretable but may struggle with complex relationships, while the MLP is more flexible but less interpretable. The MLP was chosen as an additional model to capture complex patterns in the data. In terms of performance, the MLP slightly outperforms the decision tree in terms of recall and F1-score, indicating its ability to identify positive instances and maintain overall accuracy.

5. Conclusion

In conclusion, the dry bean classification problem was addressed using two machine learning models, Multi-layer Perceptron (MLP) and Decision Tree (DT). Evaluation of the models revealed that the MLP model achieved a higher accuracy compared to the DT model, indicating its superior performance in accurately classifying different types of dry beans. The MLP model's ability to capture complex patterns and relationships through interconnected layers of neurons contributed to its higher accuracy. While the DT model demonstrated respectable performance and offered interpretability, the MLP model outperformed it in terms of accuracy. Thus, the MLP model emerges as a preferred choice for achieving higher accuracy in the classification of dry beans.

Reference

- Murat Koklu, 2020, Multiclass classification of dry beans, www.sciencedirect.com/science/article/abs/pi19311573

- Dry bean dataset, https://archive-beta.ics.uci.edu/dataset/602/dry+bean+dataset

- Tuning the hyperparameter, https://scikit-learn.org/stable/modules/grid_search.html

- Decision tree classifier, https://scikit-learn.org/stable/modules/

- StandardScaler https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html