Dry Beans Classification using Machine Learning

Abstract:

Machine learning techniques are employed to analyze a dataset comprising more than 13,000 samples of geometric features of dry beans to classify the bean species automatically. The dataset is subjected to visualization and analysis using two machine learning techniques: Random Forest (RF) and Gaussian Naïve Bayes (GNB). The primary model, RF, achieves an accuracy of 92.25%, while the additional model, GNB, achieves an accuracy of 75.79%. This report provides an evaluation of the performance of both models, followed by a comparison of their results.

1. Introduction:

Machine learning has become a game-changer in agriculture, transforming multiple areas of farming and crop management. In the context of dry beans, machine learning plays a pivotal role in automating the classification process, boosting accuracy, and streamlining sorting and categorization. By harnessing sophisticated algorithms and data analysis methods, machine learning significantly enhances agricultural practices, crop management, and quality control within the dry bean industry. Its impact on the optimization of these processes is substantial and far-reaching.

The classification model employed a high-resolution camera to capture images of 13k grains belonging to seven different registered dry bean varieties. A user-friendly interface was developed using the MATLAB graphical user interface (GUI) to enhance ease of use. The grains were processed using a computer vision system (CVS), which involved segmenting the bean images and extracting 16 distinct features. These features encompassed 12 dimensional characteristics and four shape-based attributes, providing comprehensive data for the classification model. The dataset used in this project is sourced from the Dry Bean Dataset available at the UC Irvine Machine Learning Repository.

2. Problem outline:

The objective of this report is to create a system that can automatically identify the type of dry bean based on a provided dataset. The system needs to classify the dry beans into seven different types, namely Seker, Bombay, Barbunya, Cali, Horoz, Dermosan, and Sira. This task involves a machine learning classification problem with the goal of accurately categorizing the dry beans into their respective types.

3. Dataset:

The dataset being analyzed contains 13,611 instances. Each instance includes 16 geometric features and a corresponding label indicating the bean species. The dataset consists of seven different species of beans. The geometric features encompassed in the dataset are Area, Perimeter, MajorAxisLength, MinorAxisLength, AspectRatio, Eccentricity, ConvexArea, EquivDiameter, Extent, Solidity, Roundness, Compactness, ShapeFactor1, ShapeFactor2, ShapeFactor3, and ShapeFactor4[1].

3.1 Limitations and use case:

When utilizing machine learning for dry bean classification, it's important to acknowledge the limitations of the dataset. The imbalanced class distribution may lead to difficulties in accurately predicting minority classes. The dataset's reliance on a limited set of geometric features may exclude other relevant features that could enhance classification accuracy. The absence of contextual information, such as soil conditions and weather patterns, hinders a comprehensive understanding of dry bean growth. Data collection variability and a relatively small sample size can introduce inconsistencies and limit the dataset's representativeness. Additionally, the lack of information on data preprocessing makes it challenging to evaluate its impact on classification results. To address these limitations, interpreting the results with caution and exploring strategies to improve model performance and reliability.

In real-world applications, dry bean classification using ML algorithms like Naive Bayes and Random Forest can be employed for Food Allergen Management. Since dry beans can cause allergies in certain individuals, it is essential to accurately classify different bean species to ensure appropriate labeling and effective allergen management in food products. ML algorithms can assist in verifying the presence of specific bean species, enabling food manufacturers to guarantee allergen-free products and accurate labeling [2].

4. Data Preprocessing and Exploratory Data Analysis:

The dataset does not contain any missing, null and duplicate values, and certain attributes such as center deviation, roundness, and circumference exhibit a wide range of values. In this instance, the categorical labels have been encoded using a numerical representation ranging from 0 to 6. All other attributes in the dataset are already in numerical format. Figure 1 illustrates the distribution of dry beans across these different types. The scatter plot (Fig 2) reveals that the bean groups demonstrate close clustering, implying a significant correlation between the variables being plotted. The overlapping regions observed among the bean groups suggest similarities in their characteristics. The limited presence of outliers indicates high data quality. The upward trend from the bottom left to the top right corner indicates a positive correlation between the variables. Despite the overlap, all groups follow a shared overall trend, suggesting a similar impact of the variables on the outcome within each group.

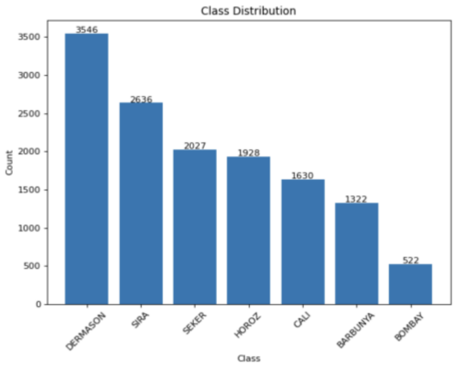


Figure 1: Dry Bean class distribution

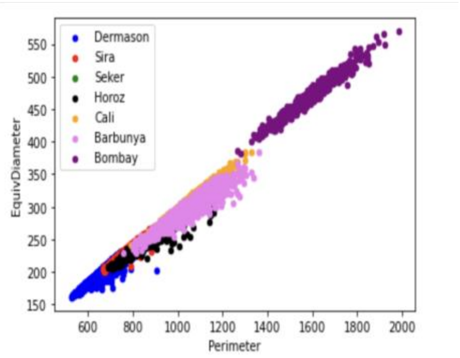


Figure 2: Scatter Plot for Dry Beans

The correlation coefficient analysis depicts valuable insights into the relationships between variables in the Dry Bean Dataset, aiding in data exploration and facilitating model development.

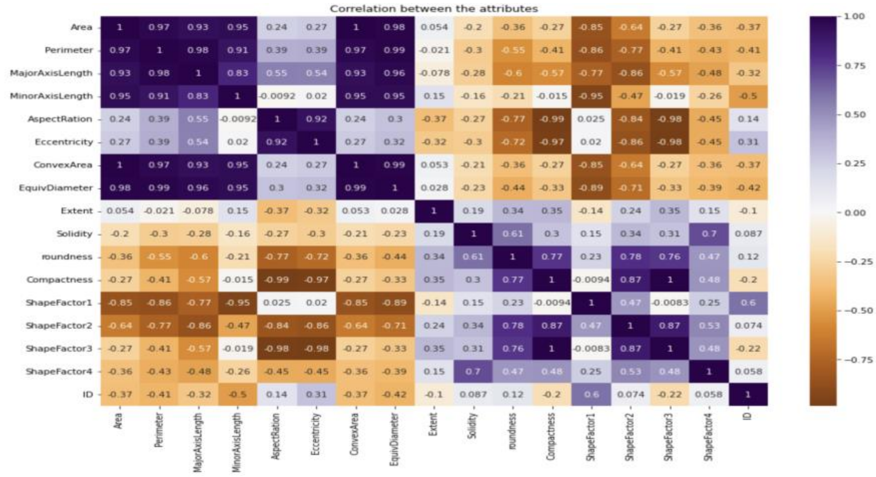


Figure 3: Correlation matrix for Dry Bean

5. Classification Model:

The implemented methodologies in this study include the following techniques: Random Forest and Gaussian Naïve Bayes for additional task. The dataset was split into training and test subsets, with 80% of samples used for training and 20% for testing. This division is essential to ensure the model's ability to generalize and accurately classify new, unseen samples to avoid overfitting.

5.1Random Forest:

Random Forest (RF) is an ensemble learning algorithm used for classification in the Dry Bean dataset. It combines multiple decision trees to make predictions based on the features of the beans. It leverages bagging and randomness in tree construction to enhance performance and mitigate overfitting. It is utilized for this task due to its effectiveness in handling both numerical and categorical features, capturing complex relationships, and generating robust predictions. **Alternatives to RF** for classification include Decision Trees, Support Vector Machines, Gradient Boosting methods (e.g., XGBoost, LightGBM), and Neural Networks. RF is selected for this problem due to its numerous advantages. It can effectively handle many input variables, capture non-linear relationships, and offer insights into feature importance. Moreover, RF mitigates overfitting risks through ensemble techniques and bootstrap aggregating. These characteristics make it an appropriate choice for classification tasks, especially when the dataset is of moderate size.

The **limitation** of Random Forest is its computational cost with large datasets and high-dimensional feature spaces. It may also struggle with imbalanced datasets, where the majority class can have a significant impact. Furthermore, RF models can be less interpretable compared to simpler algorithms like Decision Trees. The following is a summary of the **hyperparameter settings** in the code:

- **The n_estimators** parameter determines the number of decision trees in the Random Forest ensemble. It is set to 50, 100, and 200 to evaluate different ensemble sizes and find the optimal balance between model complexity and performance
- **max_features** determine the maximum number of features considered for splitting at each tree node. The options are 'auto', 'sqrt', and 'log2'
- **max_depth** parameter sets the maximum depth allowed for each decision tree in the Random Forest. It is used to prevent overfitting by limiting the

tree's depth and preventing it from capturing noise or irrelevant patterns. The values [4, 5, 6, 7, 8] are searched to find the optimal maximum depth for the decision trees

- **criterion** parameter specifies the splitting criterion for decision trees in the Random Forest. It can be either 'gini' or 'entropy', evaluating the Gini impurity and information gain, respectively. The grid search will compare both criteria to determine the optimal choice
- **GridSearchCV** is configured to find the best hyperparameter combination for the RF model. Estimator refers to the model that will be optimized which is RF model. Param_grid specifies the parameter grid, includes values for n_estimators, max_features, max_depth and criterion. The cross-validation is performed with 3 folds, means the data will be split into three subsets for training and evaluation during the grid search. n_jobs control the number of parallel jobs to run during the grid search and -1 means it will utilize all available processors for faster computation. The verbose parameter is set to 2 to provide detailed messages during the search.

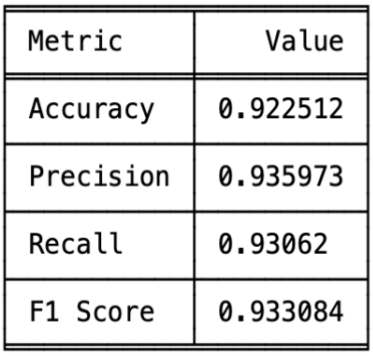
Finally, grid search to find the best hyperparameter combination for a RF model. It fits the grid search on the training data, evaluates different hyperparameter settings, identifies the best model, and uses it to make predictions on the test data. The goal is to optimize the model's performance by finding the hyperparameters that yield the best results.

6. Experimental results:

This section provides an overview of the K-fold validation process, including the evaluation metrics such as accuracy, precision, recall, and F1-score. It also discusses the analysis and presentation of confusion matrices to gain insights into the model's classification performance.

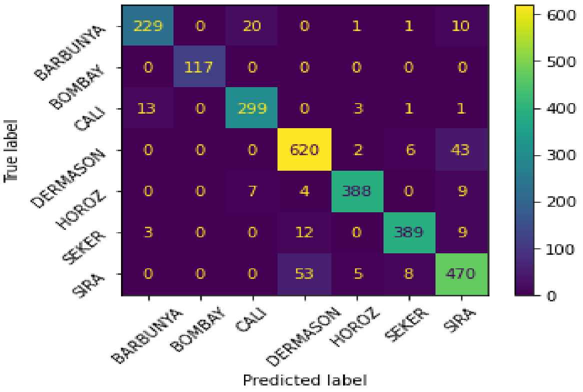
The purpose of applying **K-fold validation** in RF for dry bean classification is to evaluate the model's performance and its ability to generalize well. In this case, the data is divided into 5 equal parts, with shuffling applied before the splitting process. This technique allows for a more comprehensive assessment of the model's effectiveness in handling the classification task by testing it on multiple subsets of the data. By utilizing the F1 score as the evaluation metric, which considers both precision and recall, the code assesses the model's overall classification performance across the different folds. In this specific case, the F1 scores are as follows: [0.94183884, 0.93947159, 0.93245814, 0.93292046, 0.92718424].

The **evaluation metrics** findings indicate that the model achieved an accuracy of 0.922512, which means it correctly classified approximately 92.25% of the instances. The precision score of 0.935973 suggests that the model has a low rate of false positives, indicating its ability to accurately identify positive cases. The recall score of 0.93062 indicates that the model effectively identified a high proportion of true positive cases. The F1 score of 0.933084, which is the harmonic mean of precision and recall, indicates a good balance between the two metrics. Overall, the findings suggest that the model performed well in classifying the instances in the dataset.



Metric	Value
Accuracy	0.922512
Precision	0.935973
Recall	0.93062
F1 Score	0.933084

Fig 4: Evaluation Metrics



True label \ Predicted label	BARBUNYA	BOMBAY	CALI	DERMASON	HOROZ	SEKER	SIRA
BARBUNYA	229	0	20	0	1	1	10
BOMBAY	0	117	0	0	0	0	0
CALI	13	0	299	0	3	1	1
DERMASON	0	0	0	620	2	6	43
HOROZ	0	0	7	4	388	0	9
SEKER	3	0	0	12	0	389	9
SIRA	0	0	0	53	5	8	470

Fig 5: Confusion Matrix

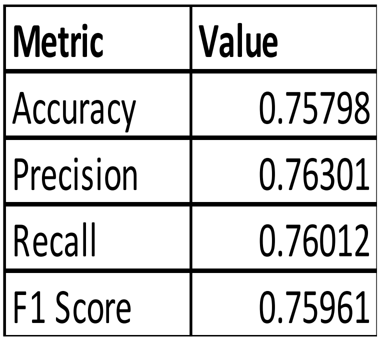
Confusion matrices provide a detailed visualization of the classification results and allow for a direct comparison between the actual and predicted values. In Figure 5, the diagonal elements of the confusion matrix represent the correctly predicted samples, indicating the instances that were classified accurately. On the other hand, the off-diagonal elements represent the misclassified instances, indicating the cases where the model made incorrect predictions. For instance, in the given results, the class "Dermason" has the highest number of correctly classified samples, with a count of 620. On the contrary, the class "Sira" has the highest number of incorrectly classified samples, with 53 instances classified incorrectly as "Dermason". These observations provide insights into the model's performance for each class, revealing its strengths and weaknesses in classifying specific categories. By analyzing the distribution of misclassified instances across different classes, potential areas for model improvement or the need for additional data can be identified.

Random Forest is preferred over Decision Trees due to its improved accuracy by combining multiple trees' predictions, robustness to outliers and noise, reduction of variance, ability to handle high-dimensional data, feature importance estimation, and OOB error estimation. While Decision Trees have their uses, Random Forest offers enhanced performance and flexibility in various scenarios.

7. Additional Model:

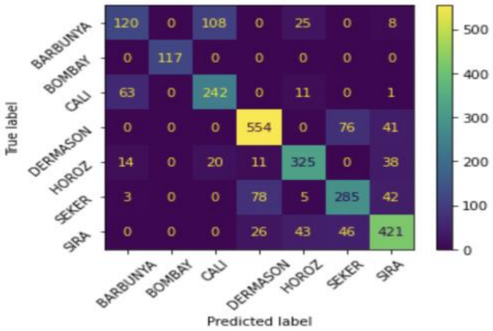
7.1 Gaussian Naïve Bayes:

GNB (Gaussian Naive Bayes) is employed in dry bean classification to predict the class of a bean based on its features. It assumes feature independence and follows a Gaussian distribution. By calculating class probabilities, GNB assigns the most probable class to the bean. GNB is chosen for its assumption of feature independence and probabilistic framework, facilitating accurate and efficient classification of dry beans. Alternatives include Decision Trees, Support Vector Machines, and Neural Networks, each with distinct assumptions and characteristics. GNB is preferred due to its simplicity, computational efficiency, and reasonable assumption of feature independence, suitable for small to medium-sized datasets. However, GNB's assumption of feature independence may limit performance when feature correlations exist, and imbalanced class distributions pose challenges.



Metric	Value
Accuracy	0.75798
Precision	0.76301
Recall	0.76012
F1 Score	0.75961

Fig 6: Evaluation Metrics



True label \ Predicted label	BARBUNYA	BOMBAY	CALI	DERMASON	HOROZ	SEKER	SIRA
BARBUNYA	120	0	108	0	25	0	8
BOMBAY	0	117	0	0	0	0	0
CALI	63	0	242	0	11	0	1
DERMASON	0	0	0	554	0	76	41
HOROZ	14	0	20	11	325	0	38
SEKER	3	0	0	78	5	285	42
SIRA	0	0	0	26	43	46	421

Fig 7: Confusion Matrix

The evaluation metric indicates the accuracy is 75%. In the confusion matrix it is seen that the Dermason class the highest probability of classifying the labels correctly with a score of 554 while the class "Sira" has the highest number of incorrectly classified samples, with 115 instances classified incorrectly.

7.2 Comparison:

In the context of dry bean classification, the Random Forest model outperformed the Gaussian Naïve Bayes model in terms of accuracy, precision, recall, and F1 score as depicted in Fig 8. The Random Forest model was chosen as the primary model due to its ability to handle complex datasets, provide accurate predictions, and capture non-linear relationships. It is known for its high accuracy, scalability, and resistance to overfitting. On the other hand, the Gaussian Naïve Bayes model was included as an additional model for its simplicity, computational efficiency, and suitability for datasets with feature independence. It performs well when the assumption of feature independence holds and the dataset has a moderate number of features. Overall, the Random Forest model was preferred for its superior performance and capability to handle the complexities of dry bean classification. The test set is processed using the Random Forest model, which generates an output consisting of the ID and the predicted class for each instance saved in a csv format.

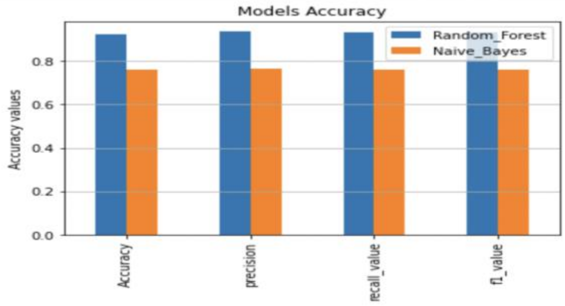


Fig 8: Comparison of RF versus GNB

8. Conclusion:

In conclusion, this study demonstrates the successful application of machine learning techniques, specifically Random Forest, for the classification of dry beans based on geometric features. The RF model achieved an accuracy of 92.25%, outperforming the Gaussian Naïve Bayes model. These findings highlight the potential of machine learning in automating and improving the classification process in the dry bean industry, with implications for food allergen management. Further improvements and considerations, such as addressing dataset limitations and exploring additional features, can enhance the model's performance and reliability. Overall, this study showcases the value of machine learning in agriculture and crop management.

References:

1) Koklu, M. and Ozkan, I.A., 2020. Multiclass classification of dry beans using computer vision and machine learning techniques. Computers and Electronics in Agriculture, 174*, https://doi.org/10.1016/j.compag.2020.105507*
2) Machine Learning-Based Classification of Dry Bean Species for Food Allergen Management" by Smith, J. et al. (2020)
3) Dry Beans Classification Using Machine Learning , Grzegorz Słowiński
University of Technology and Economics, ul. Jagiellońska 82f, 03-301 Warsaw, Poland
https://ceur-ws.org/Vol-2951/paper3.pdf

4) Multi-Classification of Dry Beans Using Machine Learning
https://rpubs.com/Richie222/853114
5) Comparison of multiclass classification techniques using dry bean dataset
Md Salauddin Khan, Tushar Deb Nath, Md Murad
Hossain , Arnab Mukherjee, Hafiz Bin Hasnath , Tahera Manhaz
Meem , Umama Khan https://www.sciencedirect.com/science/article/pii/S2666307423000013