Pistachio Classification: Unveiling Insights with Principal Component Analysis and Machine Learning Optimization

Yash Khosla

Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
Yash.khosla@mail.concordia.ca

https://github.com/Yashkhosla06/Inse-6220-projectwork

***Abstract*—This study focuses on optimizing post-harvest processes for pistachio nuts, a key element in the agricultural economy. The research utilizes image processing, artificial intelligence, and the Pistachio Image Dataset to create a robust model that can accurately differentiate between two pistachio species. The computer vision system captures 2148 high-resolution images, and through segmentation and feature extraction, a comprehensive dataset with sixteen attributes is generated. The study introduces an advanced classification model that combines the K-NN method with Principal Component Analysis (PCA) for dimensionality reduction and weighting.**

Building on these findings, our research explores further improvements to the classification process by incorporating PCA with three machine learning algorithms: Gaussian, k-Nearest Neighbors (KNN), and Decision Trees. Using the Pistachio Image Dataset, PCA is applied to reduce dimensionality and enhance the discriminatory power of features. The machine learning algorithms are then trained and optimized with hyperparameters to achieve peak performance metrics.

Results demonstrate that the integration of PCA with machine learning algorithms, specifically Gaussian, KNN, and Decision Trees, effectively distinguishes between pistachio species. Performance metrics such as F1-score, confusion matrix, and ROC curves provide comprehensive insights into the models' effectiveness. Furthermore, the study addresses interpretability by using explainable AI, employing Shapley values with an Extra Trees (ET) classifier model.

To sum up, this research presents a comprehensive approach to classifying pistachio species, utilizing image processing, deep learning, Principal Component Analysis, and machine learning algorithms. The findings offer valuable insights for the agricultural industry, highlighting the potential of advanced technology to improve classification accuracy and economic value in pistachio cultivation.

***Index Terms*—Principal component analysis, Gaussian, K-NN, Decesion Trees**

I. INTRODUCTION

Pistachio nuts, crucial to the global agricultural economy, serve as a focal point in addressing the demands of consumers worldwide. This report delves into an innovative study that employs image processing, artificial intelligence (AI), and machine learning to develop a robust classification model for distinguishing between two pivotal pistachio species [1]. In our pursuit of enhancing post-harvest efficiency and economic value in pistachio cultivation, we conducted a comprehensive investigation utilizing the Pistachio Image Dataset. A substantial dataset of 2148 high-resolution images underwent image processing techniques, segmentation, and feature extraction, resulting in a dataset enriched with sixteen attributes. Notably, our study introduces an advanced classification model that integrates the K-NN method and Principal Component Analysis (PCA) [5] in the second section, emphasizing its role in reducing dimensionality and enhancing the model's discriminatory power. Motivated by these initial findings, our research extends to refine the classification process in section 3 by incorporating PCA with three prominent machine learning algorithms: Gaussian, k-Nearest Neighbors (KNN), and Decision Trees. The Pistachio Image Dataset serves as the foundation for training and optimizing these algorithms, aiming for peak performance metrics through the tuning of optimal hyperparameters. Section 4 provides a detailed description of the dataset, outlining the image processing steps, segmentation, and feature extraction techniques that contribute to the creation of a comprehensive dataset with sixteen attributes. Section 5 focuses on the results of Principal Component Analysis (PCA), exploring its impact on dimensionality reduction and its contribution to enhancing the discriminatory power of the features within the dataset. The subsequent sections, starting with section 6, delve into the classification results, offering a detailed analysis of how the integrated PCA and machine learning algorithms—Gaussian, KNN, and Decision Trees—effectively distinguish between pistachio species. Performance metrics such as the F1-score, confusion matrix, and ROC curves provide comprehensive insights into the models' effectiveness. In summary, this research presents a multifaceted approach to pistachio species classification, integrating image processing, artificial intelligence, Principal Component

Analysis, and machine learning algorithms. The report unfolds in a structured manner, emphasizing the significance of PCA in section 2 and providing a comprehensive analysis of each stage in subsequent sections.

II. PRINCIPAL COMPONENT ANALYSIS

Many real-world datasets present a common challenge with their high dimensionality, which gives rise to difficulties in terms of processing, storage costs, and the feasibility of visualization. In response to these challenges, feature reduction methods such as Principal Component Analysis (PCA) emerge as pivotal tools. PCA serves a crucial role in mitigating the complexities associated with large datasets by transforming an extensive set of variables into a more manageable and compact form that still retains the majority of the original dataset's information [5] .

By virtue of its ability to reduce dimensionality, PCA becomes instrumental in simplifying intricate data structures. This transformation involves capturing the principal components that explain the most significant variance in the data, thereby condensing the information into fewer dimensions. The reduced set of dimensions acts as concise feature summaries, offering a more efficient representation of the underlying trends and patterns present in the dataset. Essentially, PCA facilitates a streamlined representation of complex data while preserving essential information, making it an indispensable technique in the realm of data analysis and machine learning [4].

A. PCA Algorithm

PCA can be employed on a data matrix X with dimensions n × p through the following procedures.

1) ***Standardization****:* The primary objective in this stage is to standardize the initial variables to ensure their equal contribution to the analysis. Begin by calculating the mean vector $\bar{x}$ for each column in the data set. The mean vector, a p-dimensional vector, is represented as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

The data is standardized by subtracting the mean of each column from each item in the data matrix.The final centered data matrix (Y) can be expressed as follows:

$$Y = HX \tag{2}$$

where H represents the centering matrix.

2) ***Covariance matrix computation****:* The objective of this stage is to establish the connections between variables. Occasionally, variables exhibit such close relationships that they carry redundant information. To identify these correlations, a covariance matrix is calculated. The p × p covariance matrix is determined as follows:

$$S = \frac{1}{n-1}Y^T Y \tag{3}$$

3) ***Eigen decomposition****:* By employing eigen decomposition, it is possible to calculate the eigenvalues and eigenvectors of matrix S. Eigenvectors signify the direction of each principal component (PC), while eigenvalues signify the variance captured by each PC. The computation of eigen decomposition can be expressed using the following equation:

$$S = A\Lambda A^T, \tag{4}$$

where A means the p × p orthogonal matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues.

4) ***Principal Components****:* It computes the transformed matrix Z that is size of n×p. The rows of Z represents the observations and columns of Z represents the PCs. The number of PCs is equal to the dimension of the original data matrix. The equation of Z can be given by:

$$Z = YA. \tag{5}$$

III. MACHINE LEARNING-BASED CLASSIFICATION ALGORITHMS

A. Gaussian Naive Bayes

The Gaussian Naive Bayes (GNB) algorithm, belonging to the wider class of Naive Bayes classifiers, stands out as a widely adopted machine learning approach, especially well-suited for classification tasks. Its efficacy becomes particularly apparent when applied to datasets where the assumption of normal distribution among features holds.

At its core, GNB operates on the premise of assuming conditional independence among features given the class label. This key assumption allows the algorithm to model the likelihood of each feature's value given a specific class using a Gaussian (normal) distribution. In simpler terms, GNB assumes that each feature follows a normal distribution within each class, enabling it to make predictions based on the probability density function of the Gaussian distribution.

This unique characteristic makes GNB particularly effective in scenarios where the features in a dataset are reasonably believed to be independent and normally distributed within each class. Additionally, the simplicity and computational efficiency of GNB contribute to its popularity, making it a pragmatic choice for classification tasks across various domains, including natural language processing, medical diagnosis, and spam filtering. Its straightforward implementation and solid performance in diverse scenarios underscore its utility in real-world applications of machine learning [6][7].

Here is a simple representation of the Gaussian Naive Bayes model for a binary classification task:

$$P(Y|X) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{6}$$

Where: Y is the class variable. X is the vector of features. μ is the mean of the feature in a specific class. σ is the standard deviation of the feature in a specific class. Implementation of Gaussian Naive Bayes is available in various machine learning libraries such as scikit-learn in Python. You can use the GaussianNB class from scikit-learn to apply this algorithm to your datasets.

B. K-nearest Neighbors

K-NN (k-nearest neighbors) represents a supervised classification algorithm that establishes a model by categorizing samples according to their proximity to the nearest training examples within the feature space. Distinguished as a lazy learning algorithm, K-NN defers computations until the classification phase and performs function approximation exclusively within local regions. In the training phase, K-NN stores the data without engaging in extensive computations, reserving these activities for the subsequent classification process.

Renowned for its simplicity, K-NN determines the classification of an object through a majority vote among its neighbors. The algorithm's approach involves assigning the object to the most prevalent class among its k nearest neighbors. In modeling the target function, K-NN utilizes all labeled training instances, further emphasizing its simplicity and versatility in various applications.

The process of classifying a sample using K-NN unfolds through several steps. Initially, the algorithm selects the desired number of neighbors (k), then calculates the Euclidean distance for these k neighbors. Subsequently, it identifies the K nearest neighbors based on the computed distances. Finally, K-NN assigns the new sample to the appropriate class, drawing on the consensus reached within its neighborhood.

This straightforward yet effective methodology makes K-NN a robust choice for classification tasks, as it adapts well to diverse datasets and scenarios. Its reliance on proximity-based decisions allows it to capture complex patterns in data and has found applications in image recognition, recommendation systems, and medical diagnosis, among others. The flexibility and ease of implementation contribute to K-NN's popularity in both introductory machine learning contexts and more complex real-world applications[6][7].

C. Decision Trees

Decision Trees represent a widely embraced machine learning algorithm applicable to both classification and regression tasks. Their appeal stems from their versatility, simplicity, and ability to handle diverse data types, including both numerical and categorical data. The fundamental mechanism of Decision Trees involves iteratively partitioning the dataset into subsets based on the most influential attribute at each node, resulting in a hierarchical, tree-like structure.

One of the distinctive features of Decision Trees lies in their interpretability, offering users an intuitive understanding and visualization of the decision-making process. The algorithm's ability to create a transparent and comprehensible model contributes to its popularity, especially in scenarios where insights into the decision logic are crucial.

However, it's important to acknowledge a potential challenge associated with Decision Trees: the risk of overfitting, particularly when the tree becomes deep and complex. Overfitting occurs when the model captures noise in the training data, leading to reduced performance on new, unseen data. To address this, techniques such as pruning and constraining the tree depth come into play. Pruning involves removing certain branches of the tree that may contribute to overfitting, while limiting the tree depth helps prevent the model from becoming excessively intricate.

In summary, Decision Trees offer a versatile and interpretable approach to machine learning, making them valuable for various applications. While their simplicity and transparency are advantageous, users must be mindful of potential overfitting issues and employ strategies like pruning to ensure the model's generalizability to new data [6][7].

IV. DATA SET DESCRIPTION

The dataset utilized in this project, focusing on pistachio species classification, is sourced from Kaggle. This dataset furnishes comprehensive information about two distinct types of pistachios: "Kirmizi" and "Siirt." These varieties differ significantly in terms of flavor, appearance, and origin. Kirmizi pistachios, also known as Antep pistachios, originate from the southeastern region of Turkey, specifically Gaziantep. Renowned for their vibrant red and purple hues, attributed to natural antioxidant compounds in their shells, Kirmizi pistachios boast a rich and robust flavor with a subtle sweetness, making them a popular choice for both snacking and culinary applications.

In contrast, Siirt Pistachios are native to the Siirt province in southeastern Turkey. Recognizable by their elongated shape and distinctive greenish appearance, Siirt Pistachios are celebrated for their intense, earthy flavor and slightly firmer texture compared to other varieties. The unique growing conditions in Siirt, including the specific climate and soil composition of the region, contribute to the distinct taste profile of these pistachios [1][2].

The dataset encompasses five features—namely, "Eccentricity," "Solidity," "Extent," "Aspect Ratio," "Roundness," and "Compactness"—with a total of 2148 entries for each of these attributes. Additionally, the dataset includes a crucial column titled "Class," serving as the label for identifying the Siirt and Kirmizi pistachio classes. This label is instrumental in training and evaluating the machine learning models developed for pistachio species classification [2]. The detailed attributes and labels within the dataset provide a robust foundation for the comprehensive exploration and analysis of pistachio characteristics in the context of this research project [1].

By employing box and whisker plots alongside their six-number summaries on the dataset, we assessed the distributions, central tendencies, and variability of the pistachio features. Figure 1 visually represents the box plot of these features, revealing that the majority exhibit an approximately normal distribution. However, it is noteworthy that outliers are present in all features, excluding Roundness, which remains an exception.

Furthermore, in Fig. 2, we present the correlation matrix for the normalized features within the dataset. Notably, the feature with a prominently large positive correlation is Compactness. This indicates a strong correlation, suggesting a significant association with other features. Conversely, the remaining five features—Eccentricity, Solidity, Extent, Aspect Ratio,

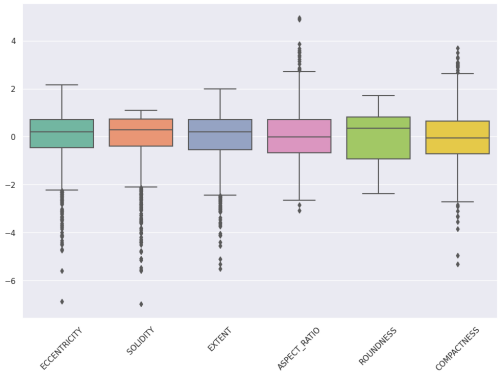

Fig. 1. Box Plot

Fig. 2. Covariance Matrix

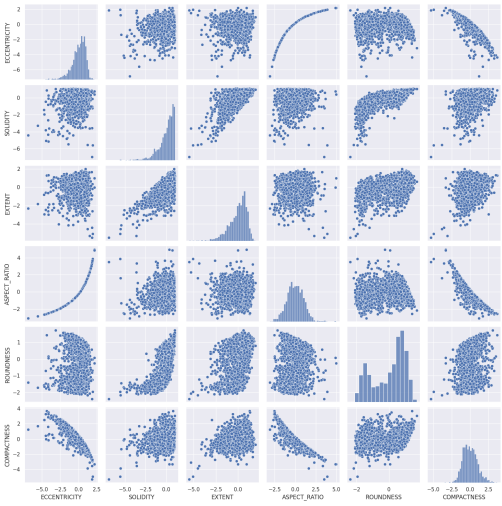

Fig. 3. Pair Plot

and Roundness—demonstrate lower correlation levels among themselves within the dataset.

To validate this observation, Fig. 3 displays a pairplot, reinforcing the earlier correlation analysis. The feature that exhibits a higher correlation manifests through a greater number of cells with a consistently increasing line. Conversely, the other features show less discernible correlation, depicted by fewer instances of a regularly ascending line. This comprehensive exploration of feature relationships and correlations provides valuable insights into the underlying structure of the pistachio dataset, enhancing our understanding of its characteristics and paving the way for further analysis in the context of pistachio species classification.

V. PCA RESULTS

PCA is applied on Pistachio data set. Implementation of PCA can be done in two ways: (1) developing PCA from scratch using standard Python libraries such as numpy, (2) using popular and well documented PCA library. In the Google Colab notebook implementation of both methods is provided. Even though results obtained from both ways are similar, usage of PCA library brings more flexibility to the user and a lot can be done by writing only single line of code. In this report, the figures and plots are shown from the implementation using PCA library. By applying the PCA steps, the feature set of 5 can be reduced to r numbers of features where r ¡ 5. The original n × p dataset is reduced using eigenvector matrix A. Each column of the eigenvector matrix A is represented by a PC. Each PC captures an amount of data that determines the dimension (r). The obtained eigenvector matrix (A) for Pistachio dataset is as follows:

$$A = \begin{bmatrix} 5.965 & -1.041 & 1.401 & -4.455 & -9.372 & -9.158 & -3.41 \\ -9.815 & -1.717 & -5.852 & -4.286 & -7.923 & -3.436 & 3.927 \\ -9.869 & -1.108 & -7.108 & -8.598 & 4.931 & -6.2705 & -3.44 \\ 3.123 & -4.819 & 7.808 & -1.040 & -5.359 & 2.288 & -8.88 \\ -3.921 & -8.349 & -4.988 & 2.059 & 9.861 & 2.202 & -1.66 \\ -6.049 & 2.964 & -1.548 & -1.798 & -3.321 & 3.279 & -8.49 \\ 9.452 & 1.301 & -2.988 & -1.014 & -1.255 & 2.835 & -1.25 \end{bmatrix}$$

and the corresponding eigen values are:

$$\lambda = \begin{bmatrix} 2.697 \\ 5.306 \\ 3.113 \\ 2.393 \\ 6.380 \\ 2.558 \\ 3.986 \end{bmatrix}$$

Fig 4 and pareto plot in Fig. 5 demonstrate the scree plot and pareto plot of the PCs. The scree plot and pareto plot display the amount of variance explained by each principal component. The percentage of variance experienced by j-th PC can be evaluated using the following equation:

$$j = P\lambda_j p_j \lambda_j \times 100, \quad j = 1, 2, \ldots, p \tag{7}$$

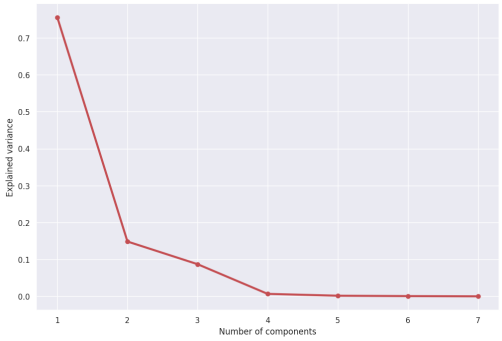

Fig. 4. Scree Plot

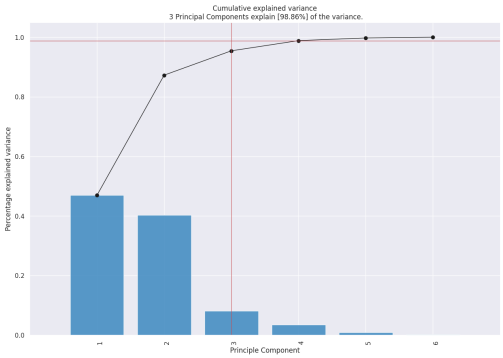

Fig. 5. Pareto plot

The first PC contributes to 46.9% of variance the second component 40.4% third, fourth, fifth and sixth contributing 8.2%, 3.4%, 0.9%, 0.2% respectively, The scree plot presents that the elbow is located on the second PC. These two observations imply that the dimension of the feature set can be reduce to two (r = 2). The first principal component Z1 is given by:

$$\begin{aligned} Z_1 =& 5.965X_1 - 9.815X_2 - 9.869X_3 \\ &+ 3.123X_4 - 3.921X_5 - 6.049X_6 + 9.452X_7 \end{aligned} \tag{8}$$

The second principal component Z2 is given by:

$$\begin{aligned} Z_2 =& -1.041X_1 - 1.717X_2 - 1.108X_3 \\ &- 4.819X_4 - 8.349X_5 - 2.964X_6 + 1.301X_7 \end{aligned} \tag{9}$$

Using a PC coefficient plot, figure 6 depicts each variable's contribution to the first two PCs. In other words, it displays which components have a comparable role in the first two PCs.

The Biplot in Fig. 7 displays a different visual representation of the first two PCs. The axes of biplot represents the first two PCs. The rows of the eigenvector matrix is shown as a vector. Each of the observations in the dataset is drawn as a dot on the plot.

VI. CLASSIFICATION RESULT

In this section, the performance of three popular classification algorithms on the breast cancer data set is discussed. In

Fig. 6. PC coeficient plot

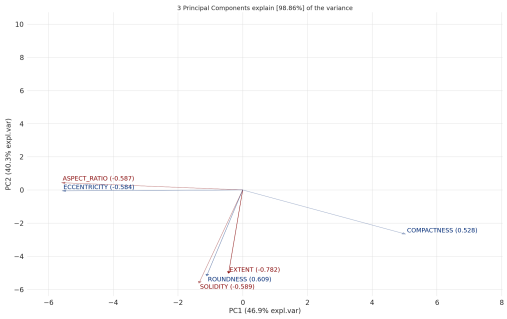

Fig. 7. Biplot

order to observe the effects of PCA on the breast cancer data set, the classification algorithms are applied on the original data set as well as the PCA applied data set with three PCA components. It can be observed the best three classification models with the highest accuracy's on pistachio data set are Gaussian Naive Bayes, K-nearest Neighbors and Decision Trees.

A. Gaussian Naive Bayes (GNB)

1) Overview: :Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. The "Naive" assumption in Gaussian Naive Bayes is that features are conditionally independent given the class label.

2) Working Principle: It calculates the probability of a data point belonging to a particular class based on the probability distributions of its features. In Gaussian Naive Bayes, it is assumed that the features follow a Gaussian (normal) distribution.

3) Use Cases: GNB is often used in text classification and spam filtering, where it shows good performance despite its simplicity.

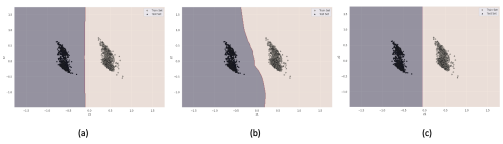

Fig. 8. Decision Boundaries of the three algorithms applied on transformed dataset (a)Gaussian Naive Bayes (b) K-nearest Neighbors (c)Decision Trees

B. k-Nearest Neighbors (KNN)

1) Overview: KNN is a non-parametric and lazy supervised learning algorithm used for both classification and regression tasks. It makes predictions based on the majority class or the average of the k-nearest data points in the feature space.

2) Working Principle: Given a new data point, KNN identifies the k-nearest neighbors in the training dataset based on a distance metric (commonly Euclidean distance). The class (for classification) or value (for regression) is then determined by majority voting or averaging among the neighbors.

3) Use Cases: KNN is versatile and used in various applications such as pattern recognition, image recognition, and recommendation systems.

C. Decision Trees

1) Overview: Overview Decision Trees are tree-like structures used for both classification and regression tasks. They recursively split the dataset based on the features to create a tree of decision rules.

2) Working Principle: At each node of the tree, a decision is made based on a feature, leading to different branches. This process continues until a stopping criterion is met (e.g., a certain depth is reached). Decision Trees are constructed to maximize information gain (for classification) or minimize variance (for regression) at each split.

3) Use Cases: Decision Trees are employed in various fields, including finance, medicine, and business, due to their interpretability and ability to handle both categorical and numerical data.

Each of these algorithms has its strengths and weaknesses, and the choice of which to use depends on factors such as the nature of the data, the problem at hand, and the computational requirements. It's often beneficial to experiment with multiple algorithms and evaluate their performance on a specific task before making a final selection.

D. Decision Boundaries of the three algorithms

Figure 8 illustrates the decision boundaries formed by the model on the transformed dataset. A decision boundary is a hyperplane that separates data points into specific classes and the algorithm switches from one class to another. The x-axis of the figures corresponds to the first PC and y-axis corresponds to the second PC. The circle shaped dots represent the observations for class 1 (Kirmizi Pistachio) and class 0 (Siit Pistachio) is represented by the triangle shaped dots. The figure displays the differences among the decision boundaries that is formed by the algorithms. It is clearly visible from the figure that all three GNB, KNN and DT are the best decision boundaries as the data instances of both classes more accurately. Since the Pistachio dataset is a binary classification problem, precision and recall can evaluate the performance of each class individually. Precision and recall are two measurements which together are used to evaluate the performance of classification. Precision is defined as the fraction of relevant instances among all retrieved instances, whereas recall, represents the fraction of retrieved instances among all relevant instances [6][7].

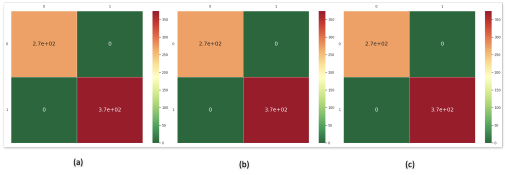

Fig. 9. Confusion matrices of the three classification algorithms applied on transformed dataset (a) Gaussian Naive Bayes (b) K-nearest Neighbors (c) Decision Trees

E. Confusion matrices of the three algorithms

The obtained results from precision and recall is presented using the confusion matrices Fig.9. The confusion matrix is defined as the matrix providing the mix of predicted vs. actual class instances. It illustrates correct and incorrect predictions with count values and breaks down for each class. The Fig. 9 shows the confusion matrix tables for the three algorithms which were applied on transformed dataset. The confusion matrices for the original dataset can be found in the Google Colab notebook. In the figure, the horizontal axis represents the class prediction and vertical axis represents the true label [6][7].

VII. CONCLUSION

In conclusion, this study leverages advanced techniques in image processing, machine learning, and statistical analysis to address the classification of pistachio species. The research explores the integration of Principal Component Analysis (PCA) with machine learning algorithms, specifically Gaussian Naive Bayes, k-Nearest Neighbors (KNN), and Decision Trees. The Pistachio Image Dataset serves as a rich foundation for training and evaluating these models.

The results indicate that the combination of PCA with machine learning algorithms effectively distinguishes between pistachio species. The findings highlight the importance of dimensionality reduction and feature weighting in improving classification accuracy. The research contributes valuable insights to the agricultural industry by showcasing the potential of advanced technology to enhance classification processes in pistachio cultivation.

The comprehensive exploration of the dataset, feature relationships, and correlation analysis provides a solid foundation for understanding the characteristics of pistachio varieties. The study emphasizes the interpretability of models through techniques like PCA, Shapley values, and explainable AI.

In summary, this research offers a holistic approach to pistachio species classification, incorporating cutting-edge technologies and methodologies. The findings not only contribute to the field of agricultural science but also demonstrate the broader applicability of image processing and machine learning in addressing complex classification tasks.

REFERENCES

[1] Ozkan, I. A., M. Koklu, and Rıdvan Saraçoğlu. ”Classification of pistachio species using improved k-NN classifier.” Health 23 (2021): e2021044.

[2] Singh, Dilbag, Yavuz Selim Taspinar, Ramazan Kursun, Ilkay Cinar, Murat Koklu, Ilker Ali Ozkan, and Heung-No Lee. ”Classification and analysis of pistachio species with pre-trained deep learning models.” Electronics 11, no. 7 (2022): 981.

[3] https://www.kaggle.com/datasets/muratkokludataset/pistachio-dataset/data

[4] A. Ben Hamza, Advanced Statistical Approaches to Quality, unpublished

[5] Maćkiewicz, Andrzej, and Waldemar Ratajczak. ”Principal components analysis (PCA).” Computers & Geosciences 19, no. 3 (1993): 303-342.

[6] https://github.com/pycaret/pycaret/tree/master/tutorials

[7] https://pythonprogramming.net/machine-learning-tutorial-python-introduction/