Classification of Date Fruits with Principal Component Analysis and Machine Learning Optimization Techniques

Himanshu Dahiya 40234526

Concordia Institute for Information Systems Engineering (CIISE)
Concordia University
Montreal, Canada
dahiyahimanshu27@gmail.com

https://github.com/HimDahiya/INSE-6220-project/blob/main/INSE_6220_DateFruit.ipynb

***Abstract*—Principal Component Analysis (PCA) serves as a technique for reducing dimensionality, specifically adept at handling large datasets. Its capability lies in the transformation of a substantial set of correlated variables into a more compact, uncorrelated form, preserving the majority of the information. This report employs PCA to classify the date fruits.**

Principal Component Analysis stands out as a multivariate method, representing a conventional tool in contemporary data analysis used to simplify intricate datasets. Its foundation lies in leveraging linear correlation to convert original variables into novel, uncorrelated variables. In this study we apply PCA to an assortment of date fruits, aiming to diminish feature dimensions for the categorization of date fruit types. The world cultivates a plethora of fruits, each exhibiting diverse variations. The process of discerning the types of the same fruit based solely on appearance proves to be both time-consuming and labor-intensive. Within this document, three widely recognized classifiers—Naive Bayes, K-Nearest Neighbours, and Decision Tree—are introduced for application on the dataset. The F1-score serves as the metric to evaluate the performance of each model. The study demonstrates promising results in enhancing the efficiency and accuracy of date fruit classification. Additionally, this report incorporates the provision of confusion matrices and Receiver Operating Characteristic (ROC) curves towards the conclusion. In an added section, the most renowned classifiers undergo application to our date set, utilizing an advanced Python Library (PyCaret) to identify the optimal model.

Building on these findings, our research explores further improvements to the classification process by incorporating PCA with three machine learning algorithms: Naive Bayes, K-Nearest Neighbours, and Decision Tree. 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, The findings offer valuable insights into distinguishing features and classifier efficiency. Overall, the study illustrates the practical application of PCA and machine learning in agricultural classification tasks.

***Index Terms*—Principal component analysis, Naive Bayes, K-Nearest Neighbours, and Decision Tree, F1-Score, Dimensionality Reduction, Cross-Validation.**

I. INTRODUCTION

Date fruit is a popular fruit that is consumed worldwide. Different variations of date fruit spreading around the globe possess their unique complexity and distinctive characteristics such as color, taste, shape, and texture. The classification of these fruits can be particularly difficult due to the subtle differences in the aforementioned features that exist between the different species. To solve this issue, automatic date fruit categorization has emerged as a result of the development of machine learning and computer vision. This study proposes a five-category classification scheme for date fruit. The proposed method of using Principal Component Analysis (PCA) to improve the K-Nearest Neighbor (KNN) model for date fruit classification has shown promising results.

Date fruits, a globally cultivated commodity, present a diverse array of varieties with distinct characteristics. Efficient classification of these varieties is crucial for quality control, agricultural management, and market supply. This study explores the synergy of machine learning, specifically Principal Component Analysis (PCA), to streamline the classification process. PCA, a dimensionality reduction technique, is employed to transform complex datasets of date fruit attributes. The study aims to assess the effectiveness of PCA in reducing feature dimensions and subsequently applies three prominent machine learning classifiers—Naive Bayes, K-Nearest Neighbours, and Decision Tree—for accurate date fruit classification.

In our pursuit of enhancing post-harvest efficiency and economic value in date cultivation, we conducted a comprehensive investigation, 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.

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 date 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 data fruit species classification, artificial intelligence, Principal Component Analysis, and machine learning algorithms. Through this exploration, we seek to enhance our understanding of the applicability of advanced machine learning techniques in agricultural contexts and contribute valuable insights to the field.

II. PRINCIPAL COMPONENT ANALYSIS

Numerous datasets in the real world pose a common challenge due to their high dimensionality, which results in processing difficulties, high storage costs, and visualization feasibility issues. To address these challenges, Principal Component Analysis (PCA) and other feature reduction methods have emerged as critical tools. PCA plays a vital role in reducing the complexities associated with large datasets by transforming a vast set of variables into a more manageable and compact form that retains most of the original dataset's information [5] .

Due to its capacity to decrease dimensionality, PCA plays a crucial role in simplifying complex data arrangements. This process entails identifying the principal components that account for the most substantial variability in the data, effectively condensing the information into fewer dimensions. The resulting set of reduced dimensions serves as succinct feature summaries, providing a more effective portrayal of the inherent trends and patterns within the dataset. Essentially, PCA enables a streamlined depiction of intricate data while retaining crucial information, establishing it as an essential technique in the domains of data analysis and machine learning.[4].

A. PCA Algorithm

PCA can be used on any data matrix X with dimensions n × p with the following steps.

1) ***Standardization****:* The primary objective of this stage is to standardize the initial variables to ensure their equal contribution to the analysis. To achieve this, calculate the mean vector barx for each column in the data set. The mean vector is a p-dimensional vector that represents the average value of each variable in the dataset:

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

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 Y = X - Xbar, where X is the original data matrix and Xbar is the mean vector of X:

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

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. Naive Bayes

The Naive Bayes a probabilistic machine learning algorithm, plays a pivotal role in the broader framework of our project. As we delve into the intricate task of categorizing diverse date fruit types, the Naive Bayes classifier emerges as a crucial component in our classification arsenal.

Naive Bayes is used as a classifier to categorize date fruits based on their features. The algorithm can is trained on the features extracted from the date fruit data using PCA and BPSO. Once trained, the model can predict the category of a new date fruit image with high accuracy.

Integration with PCA: Naive Bayes complements the dimensionality reduction capabilities of Principal Component Analysis (PCA). By applying PCA to our date fruit dataset, we aim to capture essential patterns and reduce the dataset's complexity. The subsequent application of Naive Bayes enhances our ability to classify date fruits accurately by leveraging probabilistic models.

In essence, the Naive Bayes classifier, synergistically integrated with PCA, elevates our approach to date fruit classification, offering a probabilistic lens through which we

discern and categorize the diverse varieties within our dataset. The combined strengths of these techniques contribute to the overall success of our machine learning project.

The classifier starts with Bayes' theorem, which expresses the posterior probability of a class given the observed features as a function of the prior probability of the class and the likelihood of the features given the class.

$$P(y|x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n)P(y)}{P(x_1, x_2, ..., x_n)} \quad (6)$$

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.

During the training phase, the model estimates the probabilities from the training data. Then, during the prediction phase, the classifier calculates the posterior probabilities for each class and assigns the instance to the class with the highest probability.

B. K-nearest Neighbors

K-NN (k-nearest neighbors) is a supervised classification algorithm that constructs a model by categorizing samples based on their proximity to the nearest training examples in the feature space. Operating as a lazy learning algorithm, K-NN defers computations until the classification phase, exclusively performing function approximation within local regions. During the training phase, K-NN stores data without undertaking extensive computations, reserving these tasks for subsequent classification. Known for its simplicity, K-NN determines object classification through a majority vote among its neighbors, assigning the object to the most prevalent class among its k nearest neighbors.

K-NN synergistically complements the dimensionality reduction capabilities of Principal Component Analysis (PCA). While PCA transforms the dataset for efficient analysis, K-NN leverages the transformed features to categorize date fruits. The algorithm's lazy learning approach defers computations until the classification phase, aligning well with the preprocessed dataset from PCA. The collaboration between PCA and K-NN showcases the effectiveness of combining dimensionality reduction and proximity-based classification for accurate and streamlined date fruit categorization.

C. Decision Trees

Decision Trees are a widely adopted machine learning algorithm suitable for both classification and regression purposes. Their popularity arises from their adaptability, straightforwardness, and capacity to manage various data formats, encompassing both numerical and categorical data. At the core of Decision Trees is a process of progressively dividing the dataset into subsets, guided by the most impactful attribute at each node. This iterative approach creates a hierarchical, tree-shaped structure, contributing to the algorithm's fundamental mechanism.

In the project "Principal Component Analysis and Date Fruit Classification Using Machine Learning," Decision Trees emerge as a powerful tool for enhancing the accuracy and interpretability of date fruit classification. The Decision Trees algorithm is particularly well-suited for complementing the feature reduction capabilities of Principal Component Analysis (PCA).

The features derived from PCA play a crucial role in the decision-making process of the tree. Decision Trees leverage the principal components to create decision nodes, ensuring that the classification process is based on the most discriminative features identified by PCA. This integration enhances the interpretability of the classification model.

In the landscape of "Principal Component Analysis and Date Fruit Classification Using Machine Learning," Decision Trees serve as a valuable ally in extracting actionable insights from the transformed dataset. The collaboration between PCA and Decision Trees not only contributes to accurate classification but also facilitates a deeper understanding of the discriminative features driving the categorization of date fruits. The flexibility, interpretability, and ability to handle complex relationships make Decision Trees an integral component of our machine learning project.

IV. DATA SET DESCRIPTION

A plethora of fruits are cultivated globally, each exhibiting diverse varieties. The characteristics determining the fruit type primarily include external appearance features such as color, length, diameter, and shape. The external aspect plays a pivotal role in categorizing fruit types, a process that often requires expertise, proving to be both time-consuming and labor-intensive.

This study aims to categorize various date fruit types—Barhee, Deglet Nour, Sukkary, Rotab Mozafati, Ruthana, Safawi, and Sagai—utilizing three distinct machine learning methods. To achieve this, 898 images representing the seven date fruit types were captured through a computer vision system (CVS). Employing image processing techniques, a total of 34 features, encompassing morphological attributes, shape, and color, were extracted from these images. Initially, models were constructed using logistic regression (LR) and artificial neural network (ANN) methods, yielding performance results of 91.0% and 92.2%, respectively. Subsequently, a stacking model, amalgamating these individual models, enhanced the overall performance to 92.8%. The study concludes that machine learning methods can be effectively employed for the successful classification of date fruit types. The dataset encompasses seven features— PERIMETER, MAJORAXIS, MINORAXIS, ECCENTRICITY, EQDIASQ, SOLIDITY and CONVEXAREA 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 Barhee, Deglet Nour, Sukkary, Rotab Mozafati, Ruthana, Safawi, and Saga. The detailed attributes and labels within the dataset provide a robust foundation for the comprehensive

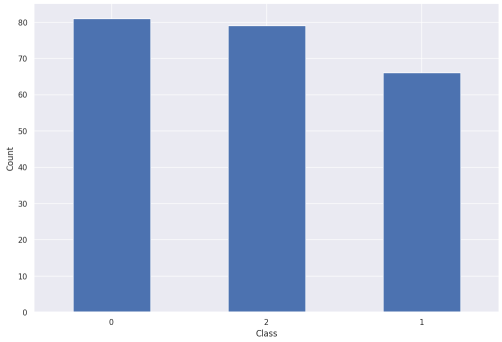

Fig. 1. Count Chart for Class

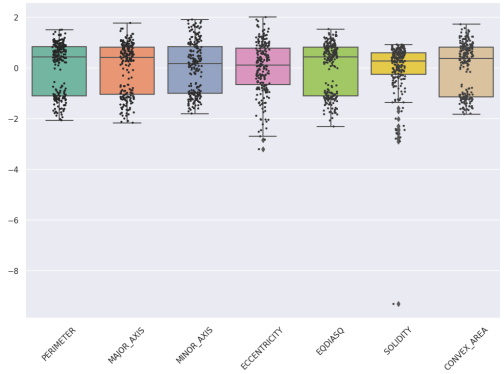

Fig. 2. Box Plot

exploration and analysis of date fruit characteristics in the context of this research project.

Figure 1 depicts a bar chart that shows the count of the instances for Count of classes in dataset.

Figure 2 represents a Box and Whisker Plot for the seven features we used to classify the date fruits. The Box Plot gives a visual representation of the mean, range, and outliers for all the variables in the data set. From the Box Plot it can be seen that all the variables have outliers, indicating that the data set has a wide range for all of its variables. However, it is noteworthy that outliers are present in all features

Figure 3 depicts the covariance matrix for the 7 variables in the data set. Covariance matrix is used to determine the variance between all the possible variables in the data set. A high value indicates a positive relation between the variables whereas a low value indicates an inverse relation between the variables. Based on the covariance matrix, similar relationships can be derived for all possible pairs of variables in the data set.

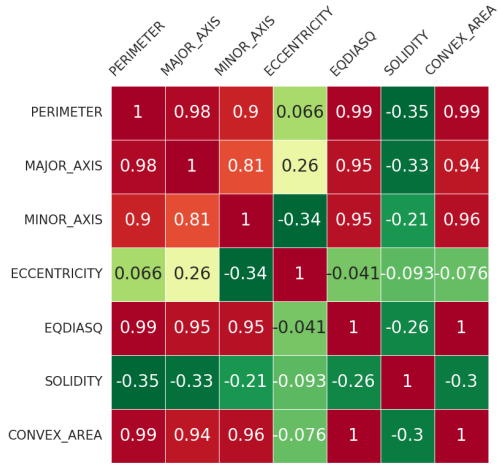

Fig. 3. Covarience Matrix

To affirm this observation, Fig. 4 presents a Pairplot, reinforcing the initial correlation analysis. Features demonstrating higher correlation are characterized by a greater number of cells exhibiting a consistently increasing line. In contrast, other features display less evident correlation, as depicted by fewer occurrences of a consistently ascending line. This thorough examination of feature relationships and correlations offers valuable insights into the inherent structure of the date fruit dataset. It enriches our comprehension of its characteristics and lays the foundation for additional analysis in the realm of date fruit species classification.

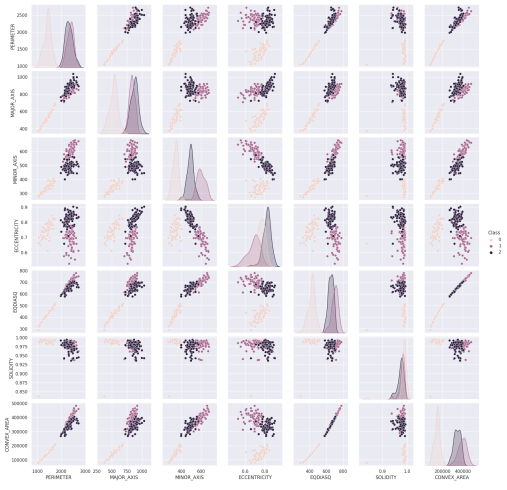

Fig. 4. Pair Plot

V. PCA RESULTS

The Date Fruit dataset underwent PCA application, revealing a noteworthy outcome: the algorithm effectively diminished the dataset's dimensions. The original n x p matrix was transformed into a more compact matrix by reducing the number of columns to r, where r p. This reduction was facilitated by employing an eigen vector matrix denoted as A. Complementing the eigen vector matrix was an associated eigenvalue matrix.

PCA can be implemented in two ways: (1) creating PCA from scratch using common Python libraries like numpy, and

(2) utilizing established and extensively documented PCA libraries. Both methods are implemented in the Google Colab notebook. While the results obtained from both approaches are comparable, employing the PCA library offers greater flexibility to users, allowing them to achieve significant tasks with just a single line of code. Throughout this report, figures and plots are presented based on the implementation using the PCA library. 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 date fruit dataset is as follows: $A =$

$$\begin{pmatrix} -4.04 & 8.30 & 5.53 & 5.652 & .... & -4.868 \\ -1.38 & 4.98 & -7.03 & -3.73 & .... & -8.14 \\ -9.22 & -2.42 & 4.45 & -6.86 & .... & -1.95 \\ 4.78 & 5.73 & -9.44 & 1.60 & .... & -3.05 \\ -1.11 & 4.39 & -1.90 & -6.21 & .... & 3.23 \\ 3.98 & -3.90 & -7.31 & -4.26 & .... & -9.52 \\ -9.99 & -3.87 & -1.65 & 1.61 & .... & -4.80 \\ -5.45 & 3.76 & -3.22 & -1.47 & .... & -8.19 \end{pmatrix}$$

and the corresponding eigen values are:

$$\lambda = \begin{bmatrix} 4.903 \\ 1.249 \\ 8.544 \\ 1.193 \\ 7.748 \\ 3.559 \\ 1.612 \end{bmatrix}$$

Fig. 5 represents the scree/elbow plot and in Fig. 6 pareto plot of the PCs. The Scree plot is instrumental in determining which principal components are worth retaining for subsequent analysis. This plot visually represents the eigenvalues of the covariance matrix. Examining the plot reveals a noticeable elbow at the fourth data point. A parallel observation is evident in the Pareto chart shown in Figure 6, which serves the purpose of calculating the percentage contribution of each eigenvalue relative to the total sum of all eigenvalues. component. The percentage of variance experienced by j-th PC can be evaluated using the following equation:

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

The first PC contributes to 69.7% of variance the second component 17.8% third, fourth, fifth and sixth contributing 12.2%, 0.2%, 0.1%, 0.1% 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:

$$\begin{aligned} Z_1 =& 4.04X_1 - 1.38X_2 - 9.22X_3 \\ &+ 4.78X_4 - 1.11X_5 - 3.98X_6 + 9.99X_7 - 5.45X_8 \end{aligned} \tag{8}$$

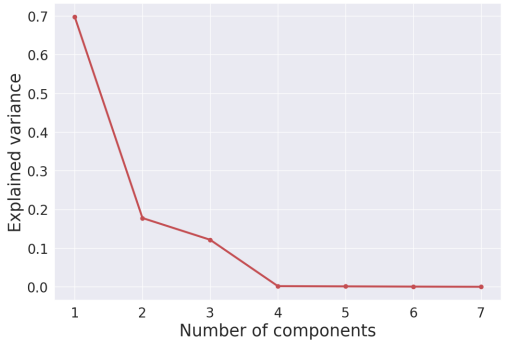

Fig. 5. Scree Plot

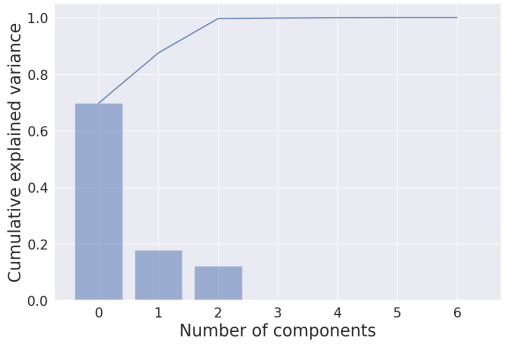

Fig. 6. Pareto Plot

From (8), it can be observed that X1, X3, X4, X7 have significant contributions in the equation for the first principal component.The second principal component Z2 is given by:

$$\begin{aligned} Z_2 =& 8.30X_1 + 4.98X_2 2.42X_3 \\ &+ 5.73X_4 + 4.39X_5 - 3.90X_6 - 3.87X_7 + 3.76X_8 \end{aligned} \tag{9}$$

Using a PC coefficient plot, figure 7 depicts each variable's contribution to the first two PCs. The plot is a visual representation of the contribution of each variable/feature on the first two principal components. After visually inspecting the plot, it can be concluded that the results from the coefficient plot align with the equations of z1 and z2 derived earlier from equation 7 and 8. From the plot it can be seen that X1 , X7 , X4 , and X3 are responsible for the major contributions towards the first principal component.

The Biplot in Fig. 8 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. The two axis of the biplot are the representation of the first and second principal component respectively. The biplot is a combined representation of the entire set of data points and the corresponding eigen vectors for each of the variables.

VI. CLASSIFICATION RESULT

This section of the report provides details on four distinct machine learning algorithms applied to the dataset. The pri-

Fig. 7. Coefficient Plot

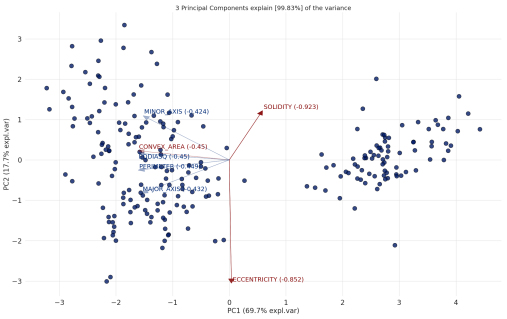

Fig. 8. BiPlot with Eigen Vector

best_model = compare_models()

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	1.0000	0.9973	1.0000	1.0000	1.0000	1.0000	1.0000	
et	Extra Trees Classifier	0.9804	0.9987	0.9804	0.9817	0.9805	0.9703	0.9709	
dt	Decision Tree Classifier	0.9742	0.9812	0.9742	0.9765	0.9742	0.9609	0.9620	
ada	Ada Boost Classifier	0.9742	0.9867	0.9742	0.9765	0.9742	0.9609	0.9620	
gbc	Gradient Boosting Classifier	0.9742	0.9966	0.9742	0.9765	0.9742	0.9609	0.9620	
qda	Quadratic Discriminant Analysis	0.9738	0.9952	0.9738	0.9808	0.9739	0.9609	0.9642	
lda	Linear Discriminant Analysis	0.9738	1.0000	0.9738	0.9804	0.9736	0.9606	0.9640	
xgboost	Extreme Gradient Boosting	0.9738	0.9938	0.9738	0.9748	0.9732	0.9604	0.9614	
nb	Naive Bayes	0.9683	0.9900	0.9683	0.9745	0.9668	0.9514	0.9552	
rf	Random Forest Classifier	0.9679	0.9987	0.9679	0.9712	0.9679	0.9515	0.9531	
ridge	Ridge Classifier	0.9675	0.0000	0.9675	0.9756	0.9676	0.9515	0.9553	
lightgbm	Light Gradient Boosting Machine	0.9675	0.9957	0.9675	0.9735	0.9671	0.9509	0.9540	
knn	K Neighbors Classifier	0.8096	0.9305	0.8096	0.8104	0.7937	0.7093	0.7275	
dummy	Dummy Classifier	0.3604	0.5000	0.3604	0.1304	0.1914	0.0000	0.0000	
svm	SVM - Linear Kernel	0.3354	0.0000	0.3354	0.1149	0.1705	0.0000	0.0000	

Fig. 9. ML Algorithms before applying PCA.

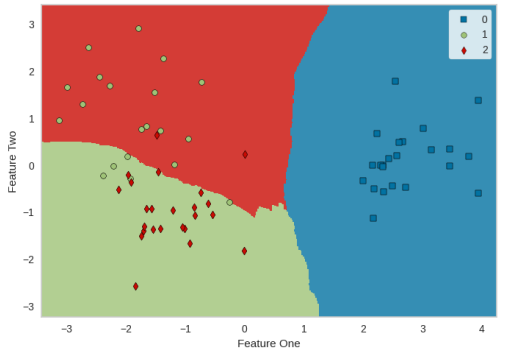

Fig. 10. Decision Boundary for KNN

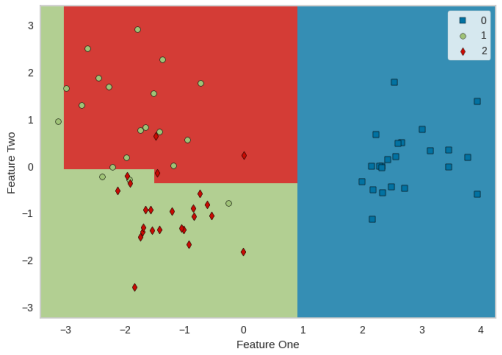

Fig. 11. Decision Boundary for Decision Tree

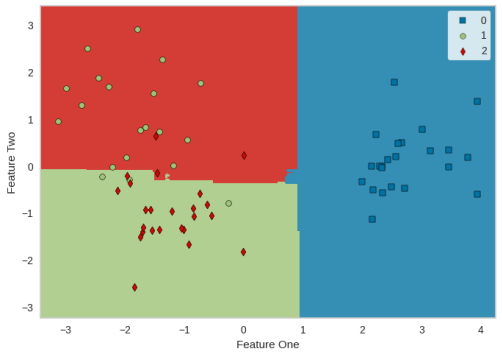

Fig. 12. Decision Boundary for RF

mary aim of these algorithms is to assess the impact of PCA on the dataset. To better comprehend the correlation between machine learning algorithms and PCA, the entire process was conducted twice—once on the original dataset and once on the dataset after applying PCA.

Figure 9 illustrates that the optimal machine learning algorithms for the dataset differ before and after PCA application. To delve deeper into the disparities among algorithms, we will also scrutinize the Logistic Regression algorithm. These algorithms play a crucial role in fine-tuning the analyzed dataset, contributing to model optimization, thereby enhancing overall performance and accuracy.

An effective method to gauge the accuracy and efficacy of a model involves decision boundary plots. A decision boundary serves as a line or surface that segregates different categories within a dataset, primarily used with binomial datasets. The axes of decision boundary plots represent the first and second principal components, providing a visual representation of the model's performance.

A. k-Nearest Neighbors (KNN)

1) 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.

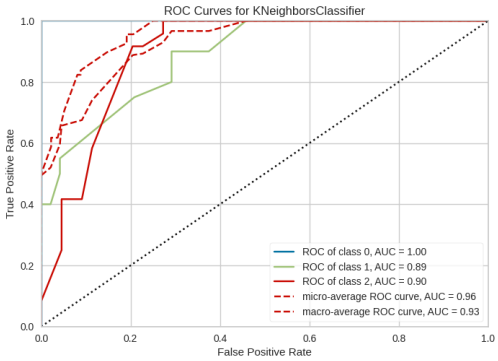

Fig. 13. ROC for KNN

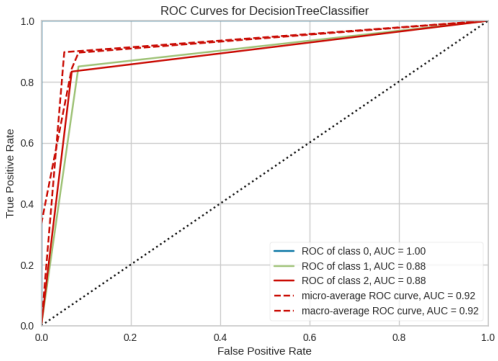

Fig. 14. ROC for Decision Tree

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

B. 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.

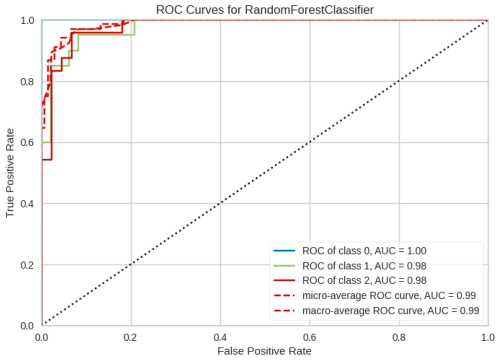

Fig. 15. ROC for RF

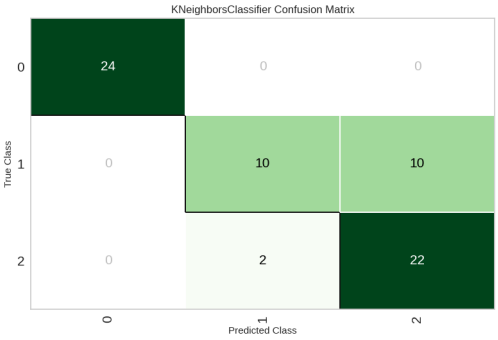

Fig. 16. Confusion Matrix for KNN

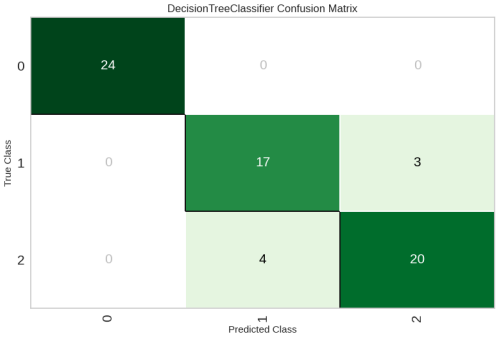

Fig. 17. Confusion Matrix for Confusion Matrix

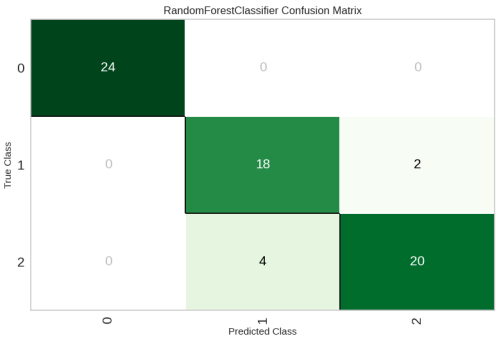

Fig. 18. Confusion Matrix for RF

C. 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 (Barhee dates) and class 0 (Deglet Nour,) 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 date fruit 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].

D. Confusion matrices of the three algorithms

The obtained results from precision and recall is presented using the confusion matrices Fig.16, Fig.17, Fig.18 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.16, Fig.17, Fig.18 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

To summarize, this investigation employs sophisticated methods in image processing, machine learning, and statistical analysis to tackle the classification of date species. The study delves into the integration of Principal Component Analysis (PCA) with machine learning algorithms, specifically Gaussian Naive Bayes, k-Nearest Neighbors (KNN), and Decision Trees, utilizing the Date Fruit Image Dataset for training and evaluation.

The outcomes demonstrate that the amalgamation of PCA with machine learning algorithms effectively distinguishes between date species. The results underscore the significance of dimensionality reduction and feature weighting in enhancing classification accuracy. This research provides valuable contributions to the agricultural sector by showcasing how advanced technology can elevate classification processes in dates cultivation.

The thorough exploration of the dataset, scrutiny of feature relationships, and correlation analysis lay a robust foundation for understanding the characteristics of date varieties. The study places emphasis on the interpretability of models through techniques like PCA.

In conclusion, this research presents a comprehensive approach to date fruit species classification, incorporating state-of-the-art technologies and methodologies. The findings not only enrich the field of agricultural science but also demonstrate the broader applicability of image processing and machine learning in addressing intricate classification tasks.

REFERENCES

[1] Santi Kumari Behera, Amiya Kumar Rath, Abhijeet Mahapatra & Prabira Kumar Sethy, "Identification, classification & grading of fruits using machine learning & computer intelligence: a review", Journal of Ambient Intelligence and Humanized Computing, 2020

[2] A. Al-Nuaimi, A. Al-Jumaily, and A. Al-Jumaily, "Fruit classification using traditional machine learning and deep learning techniques: A comparative study", in Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications (AMLTA), 2020

[3] https://www.kaggle.com/datasets/muratkokludataset/date-fruit-datasets/data

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

[5] A. Al-Nuaimi, A. Al-Jumaily, and A. Al-Jumaily, "Machine Learning–Based Detection and Sorting of Multiple Fruits Using RGB-D Images", Food Analytical Methods, 2021

[6] Selva Prabhakaran, "How Naive Bayes Algorithm Works? (with example and full code)", Machine Learning Plus, 2023 1

[7] "Naive Bayes for Machine Learning", Machine Learning Mastery, 2023 https://pythonprogramming.net/machine-learning-tutorial-python-introduction/