Classification of Rice seed using Principal component Analysis and Machine Learning Algorithms

Alam Femina Paaul Joseph ID 40225380
GitHub Link: https://github.com/femina-dev/Project_INSE6220

Abstract

Rice is the most broadly consumed staple nourishment for an enormous part of the world's human population, particularly in Asia and Africa. Rice has many genetic varieties, and these varieties are separated from each other due to some of their features. In this report, PCA is applied to the rice seed dataset to classify rice varieties and evaluate the quality of rice seeds. For classification, results made by algorithms such as Light gradient boosting machine (Lightgbm), Naïve Bayes (NB), and K-Nearest Neighbors classifier (KNN), are applied to the original dataset and transformed dataset (after applying PCA) to identify the rice variety based on the features. Performance measurements of three different machine learning algorithms were obtained using rice images and features extracted from the images. Statistic measurements of the confusion matrix, F1 score, and decision boundary of the classification were used to evaluate performance metrics. With the help of these metrics, information about the training and testing success of algorithms has been calculated.

Keywords—Principal Component Analysis, Rice Seed Classification, Light Gradient boosting machine, K-Nearest Neighbors, Gaussian Naive Bayes.

I. INTRODUCTION

Rice cultivation is the main agriculture in many countries. Rice is priced on various parameters in the market. Texture, shape, color, and fracture rate are some of the parameters. The intention of the classification stage is to ensure the purity of the rice seeds from another variety. Machine learning algorithms ensure that a large amount of data is analyzed quickly and reliably. It is important to use such methods in rice production to improve the quality of the final product and to meet food safety criteria in an automated, economical, efficient, and non-destructive way. With the development of information technology, the analysis and the detection of rice seeds are carried out by an automatic computer-aided vision system. In this study, Ipsala, Karacadag, and Arborio which are the three different varieties of rice often grown in turkey were used. The dataset comprises 506 elements with 10 attributes of morphological features that correspond to the classification of rice seeds that distinguish rice varieties.

The dataset used to train the algorithms was collected by extracting the features. Furthermore, these features are seen to be classified using algorithms such as Light gradient boosting machine (Lightgbm), Naïve Bayes (NB), and K-Nearest Neighbors classifier (KNN) from machine learning algorithms. We utilize the Principal Component Analysis (PCA) to extricate the most significant highlights, before utilizing the arrangement. After pre-processing and transforming the data with PCA, we evaluated the transformation using three classification algorithms (Light gradient boosting machine and Naive Bayes, and K-Nearest Neighbours).

The report is further organized as follows: Section II focuses on the technicalities of principal component analysis; Section III introduces the classification algorithms; Section IV reports and analyses the experimental results; Section V explains, and Section VI discusses the classification results, and Section VII covers the I. Explainable AI with Shapley Values and Section VIII sums up this study.

II. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is an exploratory multivariate technique for simplifying complex data sets. It is an orthogonal linear transformation of feature vectors into uncorrelated vectors such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA helps to identify the most significant features and removes uncorrelated features in a dataset. With the number of feature reductions with PCA, the time taken to train the model is significantly reduced and therefore the dataset overcomes overfitting.

PCA Algorithm consists of 4 steps:

1. Standardization or Normalization of data

To analyze the contribution of each variable equally we should standardize the data before performing PCA. This will help to attain biased results at the end of the analysis. Transforming the variables to the same standard can be achieved by subtracting the mean and separating it by the standard deviation for each estimation factor using the formula below. This will ensure that each feature has a mean = 0 and variance =1

$$Z=\frac{n-\mu}{\sigma} \qquad (1)$$

$$Where\ Mean(\mu) = \frac{Sum\ of\ the\ terms}{total\ number\ of\ terms}$$

$$Where\ SD(\sigma) = \sqrt{}\frac{€(x - Mean)^2}{n(total\ number\ of\ terms}$$

The data from the above settings can be represented as a centered data matrix (Y).

$$Y = HX \qquad (2)$$

2. Covariance Matrix computation

To segregate the highly interrelated variables, the covariance matrix can be calculated with the help of the given formula

$$cov(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{x})(Y_i - \bar{y}) \qquad (3)$$

The result is an NxN symmetrical matrix(S)(4) that contains the covariances of all possible datasets and expresses the correlation between two or more features in a multi-dimensional dataset

$$S=\frac{1}{n-1}\ YY^{T} \qquad (4)$$

3. Find the Eigenvectors of the covariance matrix

An eigenvector (V) is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it. The corresponding eigenvalue (λ) is the factor by which the eigenvector is scaled.

Where A is a square covariance matrix, V is a vector, and λ a scalar that satisfies the below condition, and as we know v is a non-zero vector we rearrange and obtain the below equation (6)

$$AV = \lambda V \qquad (5)$$

$$det(A - \lambda I)=0 \qquad (6)$$

$$S = A\Lambda A^{T}, \qquad (7)$$

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

4. Principal Components

It computes the transformed matrix Z which of the size of n×n. The rows of Z represent the observations and the columns of Z represent the PCs. The number of PCs is equal to the dimension of the original data matrix. The equation of Z can be given by:

$$Z = YA \qquad (8)$$

III. MACHINE LEARNING BASED CLASSIFICATION ALGORITHM

Classification models are a method of high importance used in various fields. In class determination, classification models are used to determine which class the data belongs to. The classification model is a model that works by making predictions. The purpose of the classification is to make use of the common characteristics of the data to parse the data in question.

A. Light Gradient Boosting Machine (Lightgbm)

Lightgbm is a gradient-boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with faster training speed and better accuracy and is capable of handling large datasets. Exclusive features (EFB) are bundled to reduce dimensionality ensuring accuracy and efficiency. Gradient-based one-side sampling (GOSS) is used for sampling the dataset. GOSS weights the datasets with larger ingredients higher while calculating the gain. Whereas the smaller gradients are randomly removed, and some are retained to maintain accuracy.

B. K-Nearest Neighbor (K-NN)

K-Nearest Neighbors is a non-parametric lazy learning algorithm that stores all instance corresponding to training data points in n-dimensional space. When unknown discrete data is received, it analyzes the closest k number of instances saved (nearest neighbors) and returns the most common class as the prediction and for real-valued data, it returns the mean of K-nearest neighbors. In the k-NN algorithm, each point is conceptually plotted in a wide-dimensional space, where each axis in space corresponds to a different variable. Given that there is a certain amount of data to Test, the test data is processed one by one with all the available data. The Test data will have many neighbors that are close to itself in terms of all the characteristics measured. For this reason, the closest k-piece data to the test data is selected. As a result, from the selected data is looked at which class has more data and the data tested is said to belong to that class. In our study, the k value was chosen as 1.

C. Naive Bayes (NB)

In machine learning, naïve Bayes is a “probabilistic classifier” based on applying Bayes theorem with strong(naïve) independence assumptions between the features. The process of training NB can be carried out efficiently in a supervised learning environment, depending on the structure of the probability model. NB stands out as the most accurate classifier that can be used in situations where there is no adherence between a particular feature present in the system and other features. In NB the learning module uses this model to predict the classification of a new sample by creating an estimated model of existing features. Using Bayesian probability terminology, the above equation can be rewritten as

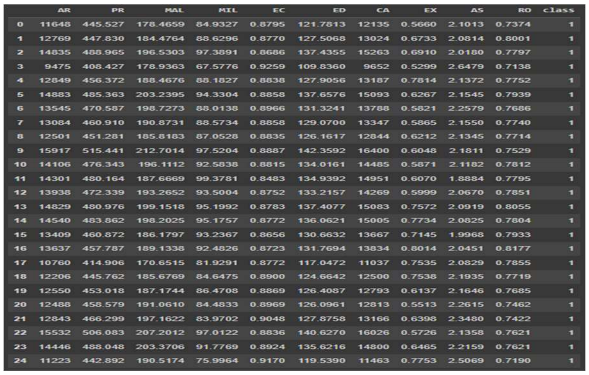
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad (9)$$

$$\text{Posterior} = \frac{prior\ x\ likelihood}{evidence} \qquad (10)$$

IV. EXPERIMENTAL RESULTS OF CLASSIFICATION

D. Data Classification

The data used in this analysis is Rice Seed Data. It has been taken from Kaggle. It has a total of 506 observations. It also includes a class that defines the type of rice variety class 1 “Ipsala”, class 2 “Karacadag”, and class 3 “Arborio”.



	AR	PR	MAL	MIL	EC	ED	CA	EX	AS	RO	class
0	11648	445.527	178.4659	84.9327	0.8795	121.7813	12135	0.5660	2.1013	0.7374	1
1	12769	447.830	184.4764	88.6296	0.8770	127.5068	13024	0.6733	2.0814	0.8001	1
2	14835	488.965	196.5303	97.3891	0.8686	137.4355	15263	0.6910	2.0180	0.7797	1
3	9475	408.427	178.9363	67.5776	0.9259	109.8360	9652	0.5299	2.6479	0.7138	1
4	12849	456.372	188.4676	88.1827	0.8838	127.9056	13187	0.7814	2.1372	0.7752	1
5	14883	485.363	203.2395	94.3304	0.8858	137.6576	15093	0.6267	2.1545	0.7939	1
6	13545	470.587	198.7273	88.0138	0.8966	131.3241	13788	0.5821	2.2579	0.7686	1
7	13084	460.910	190.8731	88.5734	0.8858	129.0700	13347	0.5865	2.1550	0.7740	1
8	12501	451.281	185.8183	87.0528	0.8835	126.1617	12844	0.6212	2.1345	0.7714	1
9	15917	515.441	212.7014	97.5204	0.8887	142.3592	16400	0.6048	2.1811	0.7529	1
10	14106	476.343	196.1112	92.5838	0.8815	134.0161	14485	0.5871	2.1182	0.7812	1
11	14301	480.164	187.6669	99.3781	0.8483	134.9392	14951	0.6070	1.8884	0.7795	1
12	13938	472.339	193.2652	93.5004	0.8752	133.2157	14269	0.5999	2.0670	0.7851	1
13	14829	480.976	199.1518	95.1992	0.8783	137.4077	15083	0.7572	2.0919	0.8055	1
14	14540	483.862	198.2025	95.1757	0.8772	136.0621	15005	0.7734	2.0825	0.7804	1
15	13409	460.872	186.1797	93.2367	0.8656	130.6632	13667	0.7145	1.9968	0.7933	1
16	13637	457.787	189.1338	92.4826	0.8723	131.7694	13834	0.8014	2.0451	0.8177	1
17	10760	414.906	170.6515	81.9291	0.8772	117.0472	11037	0.7535	2.0829	0.7855	1
18	12206	445.762	185.6769	84.6475	0.8900	124.6642	12500	0.7538	2.1935	0.7719	1
19	12550	453.018	187.1744	86.4708	0.8869	126.4087	12793	0.6137	2.1646	0.7685	1
20	12488	458.579	191.0610	84.4833	0.8969	126.0961	12813	0.5513	2.2615	0.7462	1
21	12843	466.299	197.1622	83.9702	0.9048	127.8758	13166	0.6398	2.3480	0.7422	1
22	15532	506.083	207.2012	97.0122	0.8836	140.6270	16026	0.5726	2.1358	0.7621	1
23	14446	488.048	203.3706	91.7769	0.8924	135.6216	14800	0.6465	2.2159	0.7621	1
24	11223	442.892	190.5174	75.9964	0.9170	119.5390	11463	0.7753	2.5069	0.7190	1

Figure 1 Data set

Figure (1) depicts the first 25 rows of the rice seed data. From the given data we can define the two types of variables

- Dependent Variable
- Independent Variable

The dependent variable is the class, and the independent variable (morphological features) is the Area (AR), Major axis length (MAL), Extent(EX), Minor axis length(MIL), Eccentricity(EC), Perimeter(PR), Roundness(RO), Convex Area(CA), Equivalent diameter(ED), and Aspect Ratio(AR).

E. Box Plot

A box plot is a graphical representation of information dependent on the base, first quartile, middle, third quartile, and greatest. Figure 2 illustrates the box plot with the morphological features of the rice dataset. It is observed from that figure the features such as Area (AR), Perimeter (PR), Major Axis Length (MAL), Equivalent Diameter (ED), Convex Area (CA), and Minor Axis Length (MIL) are positive/right skewed whereas the parameter Eccentricity(EC) is negatively/left skewed. It is evident that feature Extend (EX), Aspect Ratio (AS), and Roundness (RO) follow an approximately normal distribution. However, there seem to be no outliers among the features in the given dataset. Therefore, all the data points from the dataset are used for further analysis.

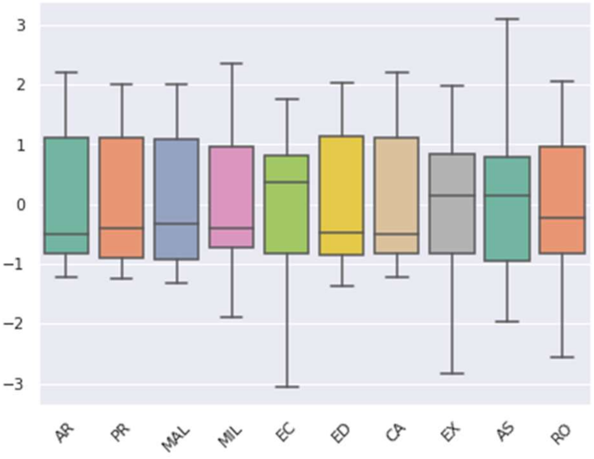
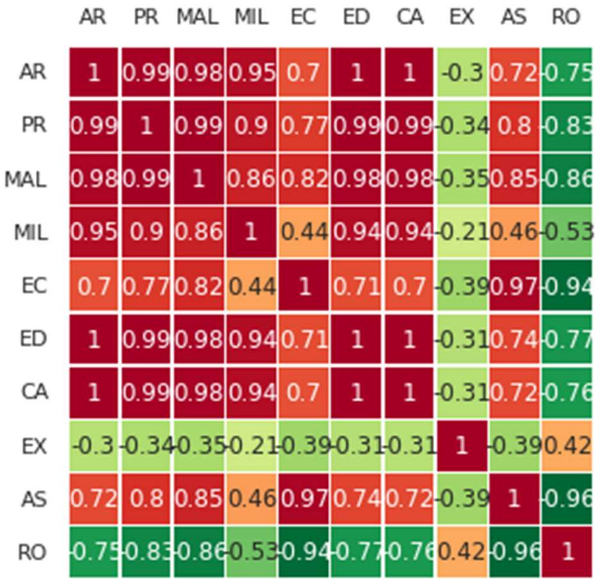


Figure 2 Box Plot

F. Covariance Matrix

The covariance matrix is a type of matrix that is used to represent the covariance between the equivalent relating components of an arbitrary vector. It is a square matrix where diagonal elements represent variance, and the off-diagonal elements represent covariance. Covariance shows the control of the direct connection between factors. The covariance matrix in figure 3 shows that the Area (AR) is highly correlated with all other variables. The features with large positive numbers are AR, PR, MAL, MIL, EC, ED, CA, and AS. This evidence implies that these 8 features are highly/positively correlated with all other variables. Whereas EX and RO show less correlation or are negatively correlated with all other features in the dataset.



	AR	PR	MAL	MIL	EC	ED	CA	EX	AS	RO
AR	1	0.99	0.98	0.95	0.7	1	1	-0.3	0.72	-0.75
PR	0.99	1	0.99	0.9	0.77	0.99	0.99	-0.34	0.8	-0.83
MAL	0.98	0.99	1	0.86	0.82	0.98	0.98	-0.35	0.85	-0.86
MIL	0.95	0.9	0.86	1	0.44	0.94	0.94	-0.21	0.46	-0.53
EC	0.7	0.77	0.82	0.44	1	0.71	0.7	-0.39	0.97	-0.94
ED	1	0.99	0.98	0.94	0.71	1	1	-0.31	0.74	-0.77
CA	1	0.99	0.98	0.94	0.7	1	1	-0.31	0.72	-0.76
EX	-0.3	-0.34	-0.35	-0.21	-0.39	-0.31	-0.31	1	-0.39	0.42
AS	0.72	0.8	0.85	0.46	0.97	0.74	0.72	-0.39	1	-0.96
RO	-0.75	-0.83	-0.86	-0.53	-0.94	-0.77	-0.76	0.42	-0.96	1

Figure 3 Covariance Matrix

G. Scatter plot

A "pair plot" is otherwise called a scatterplot, in which one variable in similar information is coordinated with another variable. The histogram on the diagonal permits us to see the appropriation of a single variable while the scatter plots on the upper and lower triangles show the relationship between two factors. For Example, the leftmost plot in the second row shows the scatter plot of Perimeter (PR) versus Area (AR).

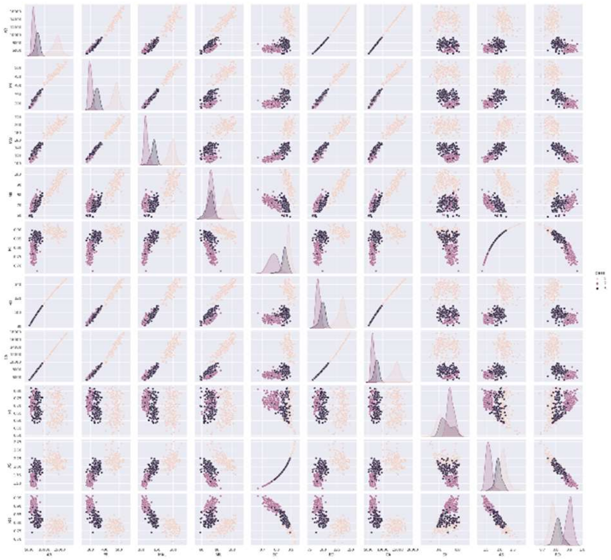


Figure 4 Pair plot/scatterplot

The pair plot in Fig. 4 supports this observation. The highly correlated features contain a higher number of cells with regularly increasing lines. On the contrary, AS, RO, and EC, RO display less apparent correlation.RO and EX are negatively correlated to all the other feature sets.

V. PCA RESULTS

Principal Component Analysis (PCA) is one of the famous techniques for dimension reduction, feature extraction, and data visualization. In general, PCA is characterized by a change of a high-dimensional vector space into a low-dimensional space. 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.

A. PCA for Dimensionality reduction

The purpose of PCA is to reduce the features of the data. There are 10 attributes/features in this data, and after applying PCA to the dataset the number of attributes has been reduced to 3 which is depicted in the Pareto diagram Fig 5. The first two components are considered to reduce the dimensionality since they contribute most of the variance. Therefore, the optimal component number for this dataset is chosen as 2 and we can reduce the complexity of the data by keeping the first two components. These two observations imply that the dimension of the feature set can be reduced to two (r = 2) 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 the Rice seed dataset is as follows

$$\mathbf{A}=\begin{bmatrix} 0.344 & -0.218 & -0.052 & -0.099 & 0.047 & -0.485 & 0.267 & -0.415 & -0.212 & 0.542 \\ 0.352 & -0.121 & -0.007 & 0.083 & 0.084 & 0.051 & -0.453 & 0.639 & -0.290 & 0.381 \\ 0.354 & -0.048 & 0.041 & -0.073 & 0.259 & 0.018 & -0.553 & -0.339 & 0.613 & -0.007 \\ 0.299 & -0.455 & -0.193 & 0.081 & -0.276 & 0.580 & 0.381 & 0.048 & 0.285 & 0.127 \\ 0.299 & 0.408 & 0.269 & -0.508 & -0.638 & 0.018 & -0.037 & 0.032 & 0.034 & 0.036 \\ 0.346 & -0.200 & -0.041 & -0.078 & 0.017 & 0.230 & -0.189 & -0.339 & -0.601 & -0.515 \\ 0.344 & -0.214 & -0.052 & -0.047 & 0.013 & -0.536 & 0.239 & 0.399 & 0.223 & -0.524 \\ -1.431 & -4.516 & 8.796 & 3.798 & -1.474 & -8.468 & 1.174 & 6.279 & -3.417 & 8.808 \\ 0.306 & 0.388 & 0.266 & -0.165 & 0.625 & 0.281 & 0.417 & 0.100 & -0.024 & 0.015 \\ -0.314 & -0.346 & -0.190 & -0.821 & 0.217 & 0.060 & -0.048 & 0.118 & -0.002 & 0.040 \end{bmatrix}$$

The principal Component can be calculated using the formula

$$Z = XA \tag{11}$$

The column values of the data are the principal components. The first principal component Z1 is given by

Z1 = 0.034 (Area) + 0.35 (Perimeter) + 0.35 (Major Axis Length) + 0.29 (Minor Axis Length) + 0.29 (Eccentricity) + 0.34 (Equivalent diameter) + 0.34 (Convex Area) - 1.43 (Extent) + 0.306 (Aspect ratio) - 0.31 (Roundness)

$$Z_1=0.034X1+035X2+0.35X3+0.29X4+0.29X5+0.34X6+0.34X7-1.43X8+0.30X9-0.31X10 \tag{12}$$

Z_2 = -0.218 (Area) -0.121 (Perimeter) -0.048 (Major Axis Length) -0.455 (Minor Axis Length) + 0.408(Eccentricity) -0.200 (Equivalent diameter) -0.214 (Convex Area) – 4.156 (Extent) + 0.388 (Aspect ratio) - 0.346 (Roundness)

$$Z_2= - 0.21X_1 - 0.12X_2 - 0.04X_3 - 0.45X_4 + 0.40X_5 -0.20X_6 -0.21X_7 -4.1X_8 +0.3X_9 -0.3X_{10} \tag{13}$$

As our goal is to extract important information from the data matrix. Based on these values, which are also presented in variance/pareto plot figure 5, we can infer that the first two components account for more than 91% of the variance in the rice seed data set. Thus, the minimum dimension to represent our data is d = 2, and we can omit the last components for our classification phase, which will be described in the next section. Figure 6 shows the scatter plot of PC2 Coefficients vs PC1 Coefficients. It visually represents the amount of contribution each feature has on the first two PCs. From this figure we can see that Extent and Roundness variables are plotted on the left side, Aspect ratio and Eccentricity

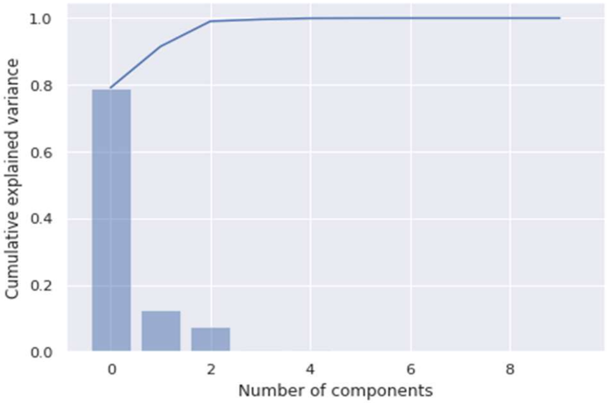


Figure 5 Pareto chart, variance chart

are plotted on the top right side and all other variables are plotted on the bottom right

- For PC1 Eccentricity, Aspect ratio, Minor Axis length, Perimeter, Equivalent diameter, Area, Major Axis length, and Convex Area have positive co-efficient but Roundness and Extend have negative coefficients
- For PC2 Eccentricity and Aspect ratio have positive coefficients but other variables such as Area, MAL, MIL, Perimeter, Convex Area, Equivalent diameter, Roundness, and Extend have negative co-efficient.
- The variables Eccentricity, aspect ratio, and Equivalent diameter and Convex Area have similar correlation
- Extend and Aspect Ratio contributes more to PC2 than other variables.
- Major Axis Length (MAL) contributes more to PC1 than other variables.

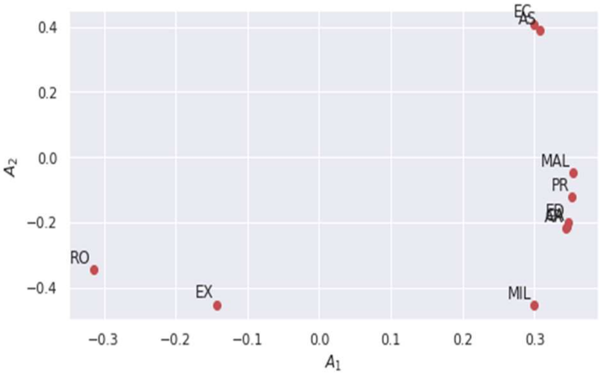


Figure 6 PC coefficient plot

B. Biplot

The biplot in Fig. 7 helps visualize both the principal component coefficients for each variable and the principal component scores for each observation in a single plot. The axes in the biplot represent the principal components (columns of A), and the observed variables (rows of A) are represented as vectors. Each observation (row of Z) is represented as a point in the biplot. The color of the point gives the rice class associated with it. From the biplot, we

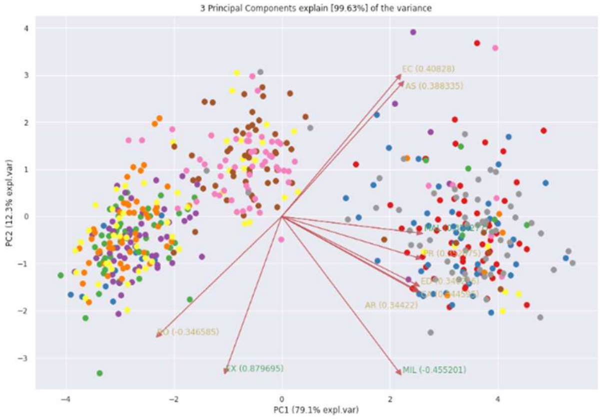


Figure 7 Bi-plot

can see that the first principal component has 2 negative coefficients for the feature Roundness and Extend and 8 positive coefficients for the other features. That corresponds to 2 vectors directed into the left half of the plot, and 8 vectors directed into the right half of the plot, respectively. The second principal component, represented by the vertical axis, has 2 positive coefficients for the variable EC and AS and 5 negative coefficients for the features MIL, AR, PR, ED, MAL. That corresponds to vectors directed into the top and bottom halves of the plot, respectively. This indicates that this component distinguishes between observations that have high values for the first set of variables and low for the second, and observations that have the opposite.

The vectors for features namely, MAL, PR show very small angle with the first PC and very large angle with the second PC. This evident supports that the analysis of the PC coefficient plot of Fig. 6. It implies that these two features have a large contribution to the first PC and very small contribution to the second PC. On the other hand, the vectors for (EC, AS MIL and EX) shows the opposite phenomenon. They create a larger angle with the first PC and smaller angle with the second PC. It means that they are more related to the second PC rather than the first one. Furthermore, the vectors which follow the same direction are positively correlated with each other. For instance, MAL, PR, ED, CA, and AR are facing in the same direction.

C. Scree Plot

A scree plot is a line of the eigen values of factors or principal components in an analysis. A scree test is a procedure for finding statistically significant factors or components using a scree plot. A scree plot displays the eigen values in a downward curve, ordering the eigen values from largest to smallest as shown in the below figure. In the figure below Components, 1 and 2 has higher variance and the graph decreased at an exponential rate resulting least value for components 4,5,6,7,8,9 and 10. The first PC holds 79.2% of the variance (l1=0.79), the second PC holds 12.3% of the variance (l2 = 0.123%) and the third PC holds 8.1% (l3=.081%). The scree plot also presents that the elbow is located on the second PC. These two observations imply that the dimension of the feature set can be reduced to two (r = 2) as L1 + L2 contributes

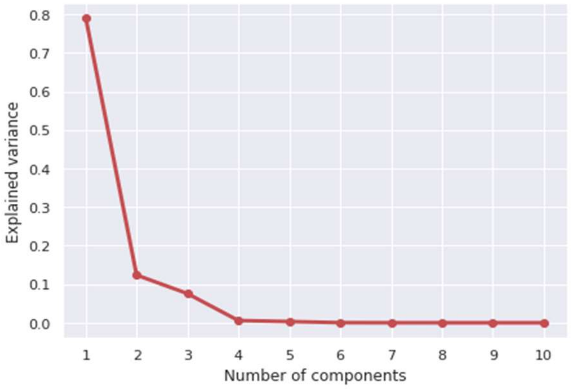


Figure 8 Scree Plot

91% and the above two factors are retained as significant for further analysis.

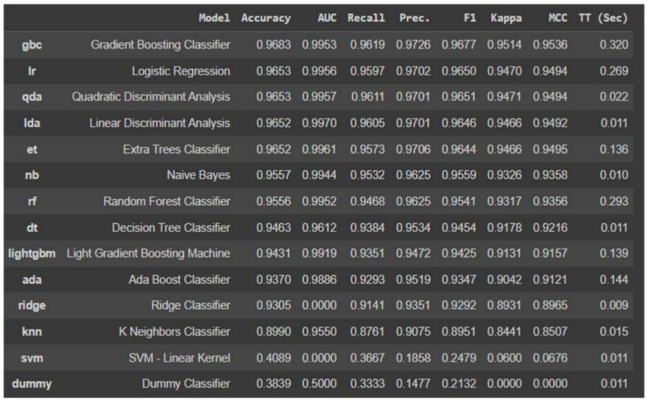
VI. CLASSIFICATION RESULTS

In this section, the performance of three classification algorithms on the Rice seed dataset is discussed. To observe the effects of PCA on the Rice seed dataset, the classification algorithms are applied to the original dataset as well as the PCA-applied dataset with three PCA components. The classification is performed using the PyCaret library of Python. The original dataset is split into train and test sets with a proportion of 70% and 30%, respectively. For the sake of reproducibility, the session id is set to 123.

Using pycaret a performance comparison table among all available classification algorithms on the target dataset was created, and the best model with the highest accuracy was identified. It is clearly illustrated in figure 9 that before applying PCA, the best three classification models with the highest accuracies are Gradient Boosting Classifier (GBC), Logistic Regression (LR), and Quadratic Discriminant Analysis (QDA).

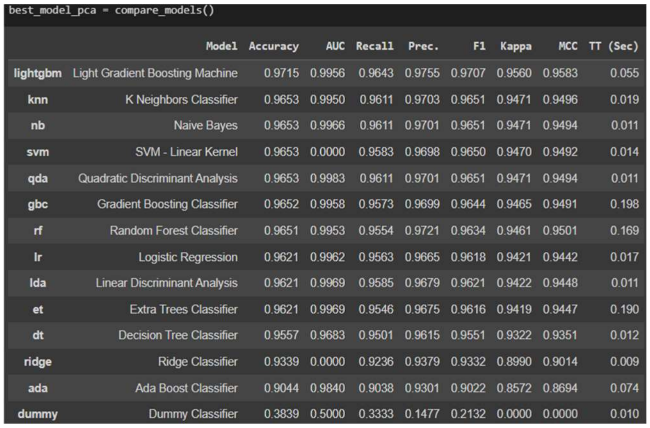
However, figure 10 demonstrates the comparison among the classification models after applying PCA, which states that the best three classifiers are Light Gradient boosting Machine (LightGBM), K-Nearest Neighbour classifier(K-NN), and Naïve Bayes Classifier (NB). Hence these three algorithms are taken for further analysis. The original and transformed dataset are trained, tuned, and evaluated using these three algorithms. Both experiments (classification algorithms applied on the original dataset and transformed dataset) can be found in the Google Collab notebook. However, in this report, we only focus on the results obtained after applying PCA (transformed dataset). Hyperparameter tuning plays a vital role in improving the performance of a model. Hyperparameter tuning with PyCaret involves three steps; create a model, tune, and evaluate its performance. Firstly, a classification model for each algorithm is produced. Further tune model() function is used for tuning the model with ideal hyper parameters. Thus, effective hyperparameters on a pre-defined search space for the model are tuned and scored using stratified K-fold cross-validation. In this is study we have applied 10-fold stratified K-fold validation on the three algorithms.

LightGBR Model is tuned with n_estimators and learning_rate as a strategy for achieving higher accuracy. n_estimators control the number of decision trees while learning_rate is the step size parameter of the gradient



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9683	0.9953	0.9619	0.9726	0.9677	0.9514	0.9536	0.320
lr	Logistic Regression	0.9653	0.9956	0.9597	0.9702	0.9650	0.9470	0.9494	0.269
qda	Quadratic Discriminant Analysis	0.9653	0.9957	0.9611	0.9701	0.9651	0.9471	0.9494	0.022
lda	Linear Discriminant Analysis	0.9652	0.9970	0.9605	0.9701	0.9646	0.9466	0.9492	0.011
et	Extra Trees Classifier	0.9652	0.9961	0.9573	0.9706	0.9644	0.9466	0.9495	0.136
nb	Naive Bayes	0.9557	0.9944	0.9532	0.9625	0.9559	0.9326	0.9358	0.010
rf	Random Forest Classifier	0.9556	0.9952	0.9468	0.9625	0.9541	0.9317	0.9356	0.293
dt	Decision Tree Classifier	0.9463	0.9612	0.9384	0.9534	0.9454	0.9178	0.9216	0.011
lightgbm	Light Gradient Boosting Machine	0.9431	0.9919	0.9351	0.9472	0.9425	0.9131	0.9157	0.139
ada	Ada Boost Classifier	0.9370	0.9886	0.9293	0.9519	0.9347	0.9042	0.9121	0.144
ridge	Ridge Classifier	0.9305	0.0000	0.9141	0.9351	0.9292	0.8931	0.8965	0.009
knn	K Neighbors Classifier	0.8990	0.9550	0.8761	0.9075	0.8951	0.8441	0.8507	0.015
svm	SVM - Linear Kernel	0.4089	0.0000	0.3667	0.1858	0.2479	0.0600	0.0676	0.011
dummy	Dummy Classifier	0.3839	0.5000	0.3333	0.1477	0.2132	0.0000	0.0000	0.011

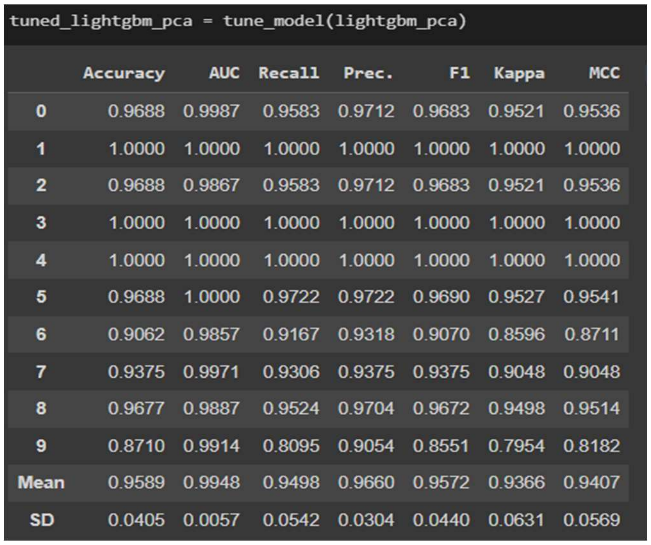
Figure 9 Comparison among classification models before applying PCA



```
best_model_pca = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9715	0.9956	0.9643	0.9755	0.9707	0.9560	0.9583	0.055
knn	K Neighbors Classifier	0.9653	0.9950	0.9611	0.9703	0.9651	0.9471	0.9496	0.019
nb	Naive Bayes	0.9653	0.9966	0.9611	0.9701	0.9651	0.9471	0.9494	0.011
svm	SVM - Linear Kernel	0.9653	0.0000	0.9583	0.9698	0.9650	0.9470	0.9492	0.014
qda	Quadratic Discriminant Analysis	0.9653	0.9983	0.9611	0.9701	0.9651	0.9471	0.9494	0.011
gbc	Gradient Boosting Classifier	0.9652	0.9958	0.9573	0.9699	0.9644	0.9465	0.9491	0.198
rf	Random Forest Classifier	0.9651	0.9953	0.9554	0.9721	0.9634	0.9461	0.9501	0.169
lr	Logistic Regression	0.9621	0.9962	0.9563	0.9665	0.9618	0.9421	0.9442	0.017
lda	Linear Discriminant Analysis	0.9621	0.9969	0.9585	0.9679	0.9621	0.9422	0.9448	0.011
et	Extra Trees Classifier	0.9621	0.9969	0.9546	0.9675	0.9616	0.9419	0.9447	0.190
dt	Decision Tree Classifier	0.9557	0.9683	0.9501	0.9615	0.9551	0.9322	0.9351	0.012
ridge	Ridge Classifier	0.9339	0.0000	0.9236	0.9379	0.9332	0.8990	0.9014	0.009
ada	Ada Boost Classifier	0.9044	0.9840	0.9038	0.9301	0.9022	0.8572	0.8694	0.074
dummy	Dummy Classifier	0.3839	0.5000	0.3333	0.1477	0.2132	0.0000	0.0000	0.010

Figure 10 Comparison among classification models after applying PCA



```
tuned_lightgbm_pca = tune_model(lightgbm_pca)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9688	0.9987	0.9583	0.9712	0.9683	0.9521	0.9536
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.9688	0.9867	0.9583	0.9712	0.9683	0.9521	0.9536
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.9688	1.0000	0.9722	0.9722	0.9690	0.9527	0.9541
6	0.9062	0.9857	0.9167	0.9318	0.9070	0.8596	0.8711
7	0.9375	0.9971	0.9306	0.9375	0.9375	0.9048	0.9048
8	0.9677	0.9887	0.9524	0.9704	0.9672	0.9498	0.9514
9	0.8710	0.9914	0.8095	0.9054	0.8551	0.7954	0.8182
Mean	0.9589	0.9948	0.9498	0.9660	0.9572	0.9366	0.9407
SD	0.0405	0.0057	0.0542	0.0304	0.0440	0.0631	0.0569

Figure 11 LightGBM metrics score after hyperparameter tuning

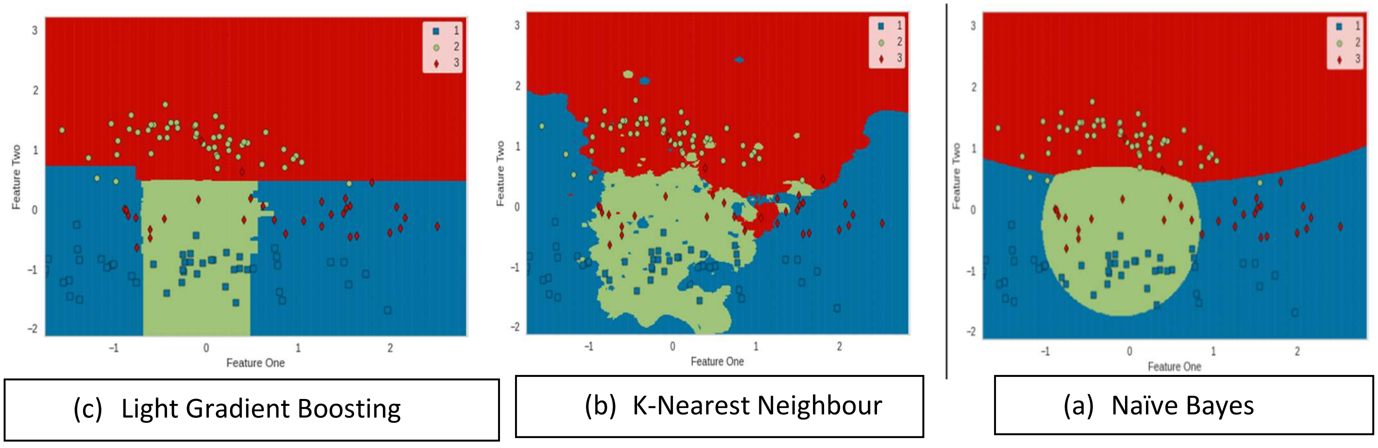
descent. Gradient boosted ensembles have a learning_rate parameter that controls the learning speed. Other parameters that affect overfitting are max_depth, and num_leaves, the higher max_depth, the more levels the tree has, which makes it more complex and prone to overfit. Regularization parameters such as lambda_l1 reg_lambda=0.1 are tuned to control overfitting.

The k-NN algorithm classifies new data points based on the similarity index which is usually a distance metric. It uses a majority vote when classifying the new data. For K-NN, the number of knearest members is tuned. Three hyper parameters namely n-neighbors (decide the best k based on the values computed earlier), weights (Check whether adding weights to the data points is beneficial to the model or not), and metric (The distance metric to be used will calculate the similarity) these are used to achieve higher accuracy.

Naïve Bayes is a powerful algorithm for predictive modeling under supervised learning algorithms that has higher accuracy and speed. Gaussian Naive Bayes is a variant of Naive Bayes used in this study which supports continuous values and has an assumption that each class is normally distributed. The variance smoothing parameter (var _smoothing) specifies the portion of the largest variance of all features to be added to variances for the stability of calculation. Ideally, Gaussian Naive Bayes assumes that features follow the normal distribution, however, to address this we can perform hyperparameter tuning by power transformation on each feature to make it normally distributed, thereby improving the accuracy. One way to analyze the performance of models is to use the evaluate_model () function which displays a user interface for all the available plots for a given model. It internally uses the plot_model () function.

However, this study covers the Decision boundary, confusion matrix, F1 score, and ROC AUC curve to evaluate the performance of the model. Fig. 12 illustrates the decision boundaries formed by the model on the transformed dataset. While training a classifier on the rice seed dataset, using a specific classification algorithm, a set of hyper-planes, called Decision Boundary are defined, that separates the data points into specific classes, where the algorithm switches from one class to another. The model can predict a value for any possible combination of inputs in our feature space. The x-axis of the figures corresponds to the first PC and the y-axis corresponds to the second PC. The blue square-shaped dots represent class 1 (Ipsala), the green circle dots represent class 2 (Karacadag), red diamond-shaped dots represent class 3 (Arborio). The figure displays the decision boundaries (the line between regions) of three algorithms. In the LightGBM model, the decision boundary appears to be linear. Naïve Bayes is a slightly nonlinear classifier, and a parabolic curve having a smooth curve boundary line. The decision surface for K-NN is complex and non-smooth. Among the three algorithms LightGBM performed well with a clear linear line that separates Class A, B and C. Some of the point from each class is present in other regions because in linear model, its difficult to get the exact boundary line separating the three classes.

A confusion matrix shows the combination of the actual and predicted class. It is a good measure of whether models can account for overlap in class properties and understand which class are most easily confused. The obtained values from precision and recall are presented using the confusion matrices. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. The Fig. 12 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 Collab notebook. LR misclassified the lowest numbers of instances. Overall LightGBM misclassified, 2 instances of class 3 are misclassified as class 1 and class 2. Whereas Naïve Bayes and KNN classifier misclassified 1 instance of class 2 as class 3 and 2 instances of class 3 as class 2. Another measurement of the performance evaluation is F1-score. The F1-score combines the precision (positive predicted value) and



(c) Light Gradient Boosting

(b) K-Nearest Neighbour

(a) Naïve Bayes

Figure 12 Decision boundary of three algorithms applied on the transformed dataset

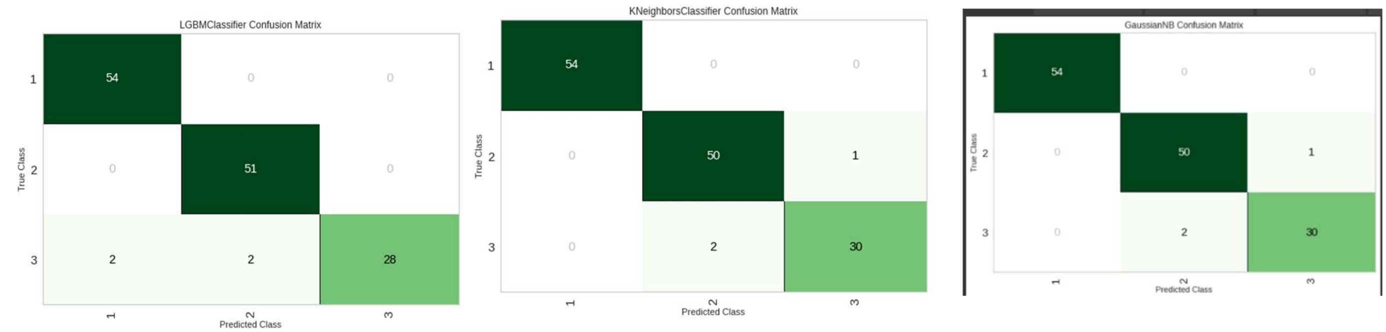


Figure 13 Confusion Matrix of three algorithms applied on the transformed dataset

recall(sensitivity) of a classifier into a single metric by taking their harmonic mean [7]. The F1-score helps to determine the better classifier. The function of the F1-score can be defined as below:

$$\textbf{F1 score} = 2 \times \left(\frac{precision \times recall}{precision + recall}\right) \qquad (14)$$

It can be observed from Fig. 9 and Fig. 10 that the F1-score of LightGBM, K-NN, and Naïve Bayes has improved significantly after applying PCA. This observation implies the fact that dimension reduction weakens the dependencies among the features. F1- score of LightGBM and K-NN enhanced more after the model is tuned with its ideal hyperparameters. (FPR). All these observations are evidence of the benefits of applying PCA and hyperparameter tuning**.** As the final analysis step, the receiver operating Characteristics (ROC) curve for the LR algorithm is shown in Fig. 14. An ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True positive rate (TPR) and False Positive rate (FPR).

$$\text{TRP} = \left(\frac{TP}{TP + FN}\right) \qquad \text{FRP} = \left(\frac{FP}{FP + TN}\right)$$

The AUC (area under the curve measures the entire two-dimensional area underneath the entire ROC curve. Fig 14 plots the false positive rate (x-axis) versus the true positive rate (y-axis) for a few different candidate threshold macros and micro average curves. The Roc curve for the LightGBM classifier with an area under the curve (AUC) is 0.99. The roc curve for the Naive Bayes is shown in Fig.14 and the AUC is 0.98. The ROC curve and AUC values shows that LightGBM is the best classifier at predicting class 1 and 2 with 99% accuracy and class 3 is predicted 98% accurately by K-NN and Naïve Bayes classifier. From the two roc curves, the Logistic Regression performs better than the Naive Bayes. Therefore, it is observed that the three algorithms are capable of successfully classifying the 3 classes of rice variety.

VII. Explainable AI with Shapley Values

It is becoming increasingly important to interpret and explain individual model predictions to decision-makers, and end-users. A common form of model explanations is based on feature attributes. Feature importance helps to estimate the contribution of each feature in the prediction process. Hence to get an overview of the most important features on the PCs, we use the SHAP values by importing the open source "shap" library of python. Therefore, for the shap analysis, I have chosen the fourth best model of the transformed dataset that is "Random Forest (RT)". Like other models, at first, an RT model is created and tuned with ideal hyperparameters. Then the tuned model is passed to the shap library for producing the interpretation plots. In this case, each PC acts as a player in the coalition. The summary plot of SHAP values is shown in fig 16. Using a typical feature importance bar chart, we use a density scatter plot of SHAP values for each feature to identify how much impact each feature has on the model output for individuals in the validation dataset. Features are sorted by the sum of the SHAP value magnitude across samples.

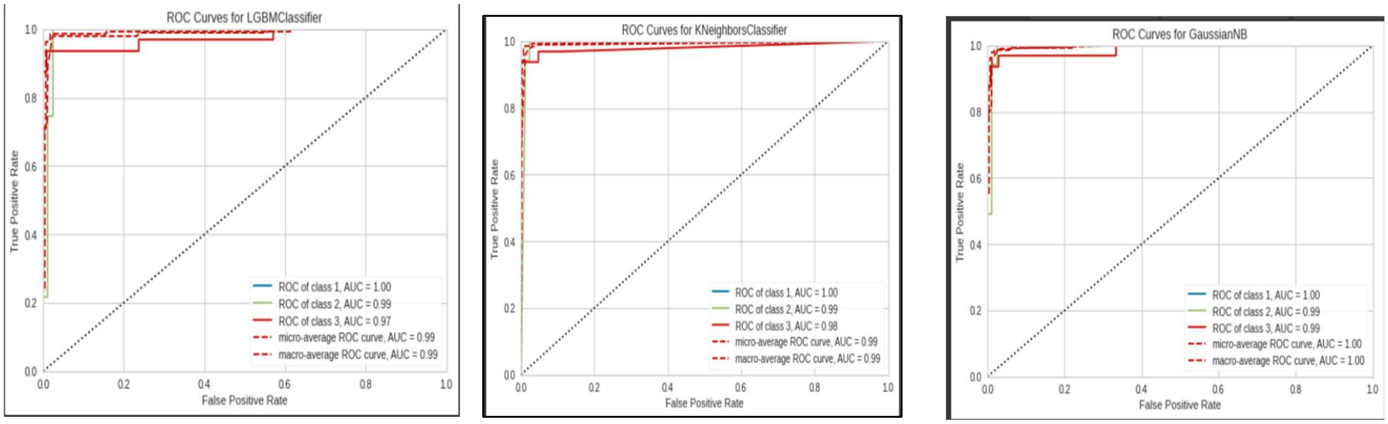


Figure 14 ROC/AUC of three respective algorithms

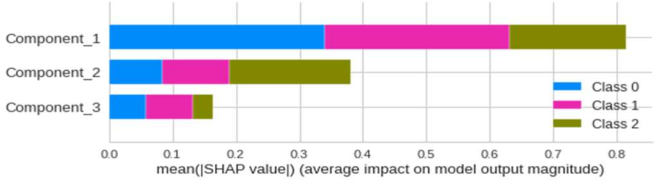


Figure 16 Summary plot

The y-axis represents the PCs and Shapley values are positioned on the x-axis. More specifically, component_1 represents the first PC, component_2 represents the second PC and component_3 represents the third PC. All the PCs are ordered according to their importance. This observation supports the Pareto plot and scree plot which indicates that the first PC holds the most feature variance. To interpret the summary plot, Component 1 /PC1 is the most important feature followed by PC2 and PC3. Besides seeing the overall trend of feature impact, we can call the force_plot (Figure 17) method to visualize how features contribute to individual predictions. In this case, PC1 and PC2 have a positive impact on the prediction while PC3 has a negative impact on the prediction. This plot displays the features each contributing to pushing the model output from the base value.

It is the mean prediction of the test set. Here, the base value is 0.3. In the plot, the bold value 0.01 is the model's score for this observation. Higher scores lead the model to predict 1 and lower scores lead the model to predict 0. The blue color on the first PC indicates that it is pushing the prediction to be lower. However, this plot is only output for this observation. It does not describe the predicted output of the entire model. Fig.16 displays the combined force plot of all PCs. This plot is a combination of all individual force plots with 90-degree rotation and is stacked horizontally. In this plot, the y-axis is the x-axis of the individual force plot. There are 135 data points in the transformed test set, hence x-axis has 135 observations. This combined force plot shows the influence of each PC on the current prediction. Values in the blue color are considered to have a positive influence on the prediction whereas values in the red color have a negative influence on the prediction. From the figure18, the observations ranging from 85 to 125 all the components show a negative influence on prediction.

VIII. CONCLUSION

To conclude, PCA and three popular classification algorithms are applied to the Rice seed dataset which holds information on morphological features to classify the rice varieties. We applied PCA on the original dataset and found 91% of the variance in the first two PC components and 95% with the first three PCs. Hence, the feature set is reduced to 2 from 10. Extensive experiments were conducted on the first two PCs and different plots are generated to validate the obtained results from different perspectives. We analyzed the impact of PCA on different classifiers. To move forward, three classification

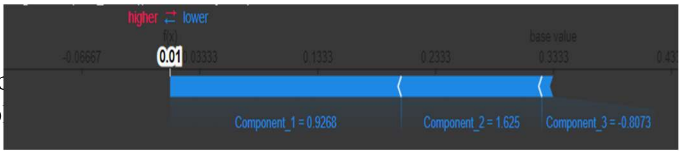


Figure 17 Force plot for a single observation

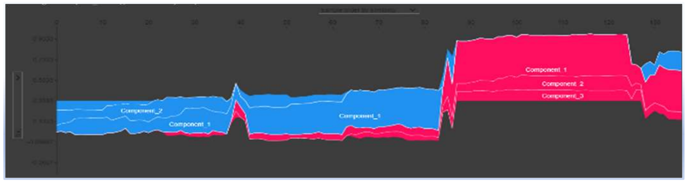


Figure 18 Combined Force plot for a single observation

algorithms, LightGBM, K-NN, Naïve bayes, QDA were applied on original dataset as well as transformed dataset with first three components. Each algorithm was tuned with the ideal hyperparameter settings and performance evaluation was conducted by comparing confusion matrices, ROC curves, decision boundary and F1 scores. The hyper parametric tuning performance metrics score of each algorithms has improved significantly. The gradient boosting classifier (GBC), Logistic regression and Quadratic discrimination analysis (QDA), performed the best on the original dataset, however after applying PCA LightGBM, K-NN and Naïve bayes performed the highest and showed the best performance metrics. Finally, to increase the interpretability of the model, several interpretation plots are produced using explainable AI Shapley values. To summarize, all three algorithms were able tp successfully classify the Rice variety.

REFERENCES

[1]. https://www.muratkoklu.com/datasets/

[2].https://www.kaggle.com/code/mushfirat/rice-classification-99-2-accuracy/data?select=Rice_Image_Dataset

[3].https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643

[4].https://stackabuse.com/implementing-pca-in-python-with-scikit-learn/

[5].https://www.hindawi.com/journals/js/2020/7041310/

[6].https://www.researchgate.net/publication/343655657_Analysing_Rice_Seed_Quality_Using_Machine_Learning_Algorithms

[7]. C. Goutte and E. Gaussier, "A probabilistic interpretation of precision,
recall and f-score, with implication for evaluation," in
European conference on information retrieval. Springer,
2005, pp. 345–359.