Rice Classification using PCA analysis and Machine Learning

Ilyas Laroussi
Student ID: *40324084*
GitHub link: https://github.com/IyasLa/INSE6220.git

***Abstract*— This project aims to classify rice grains into two distinct categories -Jasmine and Gonen- with the use of Principal Component Analysis (PCA) and machine learning techniques. Features captured from rice grains are often highly dimensional and composed mostly of redundancies and noise, this could harm the performance of classification algorithms. To solve this, we applied PCA to reduce the dimensionality of the original rice feature space into a set of components that account for the most significant variance in the data. Our end results showcase that the integration of PCA not only betters computational efficiency but also boosts classification accuracy. This project highlights the effectiveness of combining statistical data preprocessing with machine learning for product identification and quality control in agriculture.**

Keywords—PCA, Jasmine, Gonen, Machine Learning.

I. INTRODUCTION

Rice has and will always be a staple of food culture all over the world, although it may seem like there's only one type of rice, there's actually a variety of rice types and each of them is used differently, and having diverse qualities, flavors and uses. Two common varieties of rice are Jasmine and Gonen rice. In food production, telling these two types apart is an important task to maintain quality control, this task could be done manually but it could prove to be extremely slow and inconsistent. Machine learning offers a way to automate the classification process and make it faster and more reliable. However, the data used to tell between these two types is mostly redundant and has many dimensions, which may make it hard to automate efficiently. To deal with this, Principal Component Analysis (PCA) can be used. PCA helps reduce the number of features while keeping the most important information, which can help machine learning models perform better.

In this project, PCA is used along with different machine learning models to classify rice samples as either Jasmine or Gonen. The goal is to see if using PCA can improve accuracy and make the models more efficient. This kind of approach could be helpful in agriculture and food industries where quick and reliable classification is needed.

II. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a reducing technique, used to drastically decrease the number of features in a dataset, while maintaining as much of the original information as possible. This is done by recognizing patterns in the data and transforming it into new variables that showcases only the most important difference between samples. PCA is an extremely reliable tool when working with high-dimensional data, simplifying the data and reducing noise, thus improving the performance of machine learning models.

A. Stanndarization

A data matrix X of dimension (n × p) can be subjected to PC A using the four clearly defined phases shown below:

- The first and most crucial stage in ensuring the effectiveness of data analysis is **standardization**, which makes every variable contribute equally to the outcomes. First, the mean vectors (X bar) for every column in the dataset are computed.
  As a p-dimensional overview of our data, the mean vector is expressed as:

$$\bar{X} = \sum_{n}^{i=1} x_i$$

Our data is then converted by subtracting the column mean from each value in our data matrix. Our centered can then be expressed as:

$$Y = HX$$

H: Centering Matrix

- **Covariance Matrix:** This step helps us identify the relationship between variables in our data, and how they vary in regard to each other, it also serves to distinguish variables that are extremely related to the point of redundancy.

The covariance matrix using the following formula:

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

Where T represents the transpose of the matrix.

B. Eigen Values and Eigen vectors:

Eigenvalues indicate the amount of variance carried by each principal component, while eigenvectors represent the directions (axes) of the new feature space. We compute our eigen decomposition using the following:

$$S = A \Lambda A^T$$

C. Principal Components:

The transformed data matrix Z (size n × p) is computed by projecting X onto the principal component directions:

$$Z = YA$$

where A is the matrix of eigenvectors. Each row of Z represents an observation, and each column corresponds to a principal component (PC).

III. Machine Learning Models

A. *Random Forest Classifier*

Random Forest Classifier (RF) is a machine learning algorithm that employs a bunch of decision tress to make the predictions, it is considered to be one of the most versatile and powerful algorithms to do classification tasks, it merges the strengths of multiple models and mitigates overfitting, outdoing most single decision trees, it is highly accurate, robust and provides key insights to data importance.

By creating multiple decision trees and training them on different random sets of our original data.

A. *Logistic Regression Classifier*

Logistic Regression is a statistical model commonly used for binary classification tasks. It models the probability of a binary outcome as a function of input features, using the logistic (sigmoid) function to output a value between 0 and the logistic function is defined as:

$$p = \frac{1}{1 + e^{-z}}$$

B. *Gradient Boosting Classifier*

Gradient boosting classifier (GBC) is a powerful machine learning model, in contrast to basic models that learn from data independently, GBC uses the predictions of multiple weak learns and then groups them to crate one powerful learner. GBC is considered to be highly accurate, interpretable and versatile.

IV. Dataset

The dataset used was extracted from Kaggle and contains 12 columns, with 10 columns stating all the features employed to differentiate between the two rice types, 1 column containing the ID and the final column represents the class ‘1’ for Jasmine and ‘0’ for Gonen.

A. *Dataset Before PCA*

This showcases how our data looked before applying PCA method to our main components.

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	EquivDiameter	Extent	Perimeter	Roundness	AspectRation	Class
0	4537	92.229316	64.012769	0.719916	4677	76.004525	0.657536	273.085	0.764510	1.440796	1
1	2872	74.691881	51.400454	0.725553	3015	60.471018	0.713009	208.317	0.831658	1.453137	1
2	3048	76.293164	52.043491	0.731211	3132	62.296341	0.759153	210.012	0.868434	1.465950	1
3	3073	77.033628	51.928487	0.738639	3157	62.551300	0.783529	210.657	0.870203	1.483456	1
4	3693	85.124785	56.374021	0.749282	3802	68.571668	0.769375	230.332	0.874743	1.510000	1
5	2990	77.417073	50.954344	0.752861	3080	61.700780	0.584898	216.930	0.798439	1.519342	1
6	3556	84.323564	55.413061	0.753762	3636	67.287739	0.750211	227.007	0.867148	1.521727	1
7	3788	86.952411	56.444769	0.760664	3866	69.448048	0.800676	235.476	0.858473	1.540487	1
8	2629	74.133114	48.074144	0.761228	2790	57.856260	0.640595	207.325	0.768594	1.542058	1
9	5719	106.721142	68.977700	0.763053	5819	85.332625	0.754983	281.839	0.904748	1.547183	1

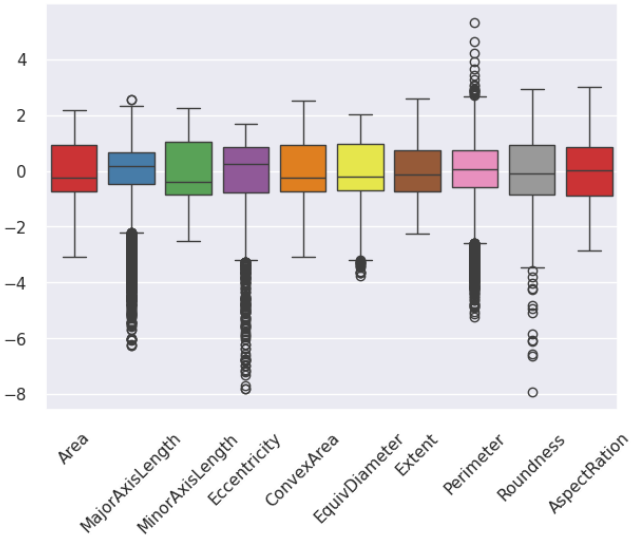

Figure 1: Boxplot

Figure 1 displays the boxplot of our data, it also showcases the features used in our classification, Area, Major axis length, Minor axis length, Eccentricity, Convex area, Equivalent diameter, extent, perimeter, roundness and Aspect Ratio, we can see that most of these features follow an approximate normal distribution, and we can also notice that 5 of these features have outliers, with Major axis length and perimeter have outliers on both sides while the other three only have outliers on the left.

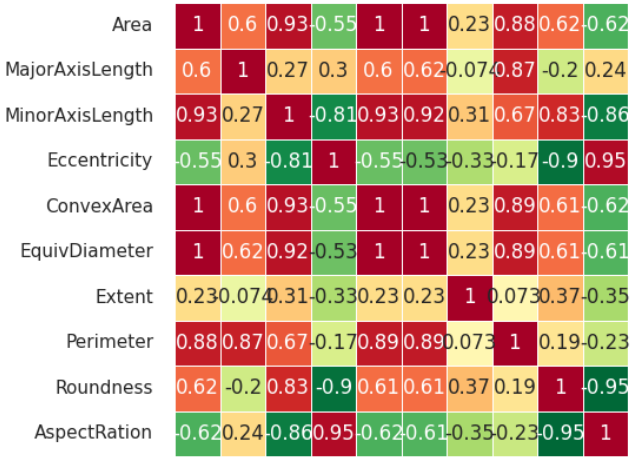

Figure 2: Correlation Matrix

Figure 2 represents our correlation matrix and highlights the relationship between each of our features. We can see that some of our features have strong positive relationships, such as Area, ConvexArea… which may signify potential redundancy, while other features like: Extent display weak correlation, and some features have a negative correlation to each other.

Figure 3: Pair Plot

Finally, the pair plot boosts our results from the correlation matrix, by showcasing the correlation between each of our features.

V. PCA RESULTS

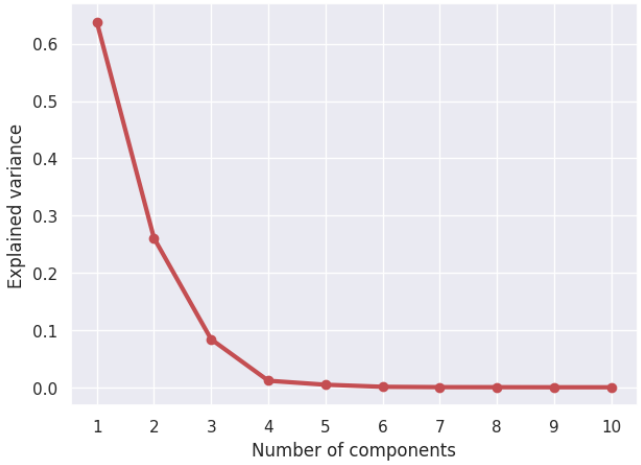

Figure 4: Scree Plot

Figure 4 represents our scree plot, with the x-axis showcasing our components which were previously stated in correlation matrix and box plot, while the y-axis provides the value of explained variance, from our scree plot we can distinguish a very high drop from component 1 with an explained variance of 63.8% and component 2 with an explained variance of 26.1%, this allows us to conclude that 89.9% of all the variance in our data is summarized in these two components, although component 3 may contain some information it is considered to be negligible since the first two components encompass most of our variance, furthermore, as we reach component 4, we notice that the plot plateaus and stabilizes meaning that these components provide negligible information.

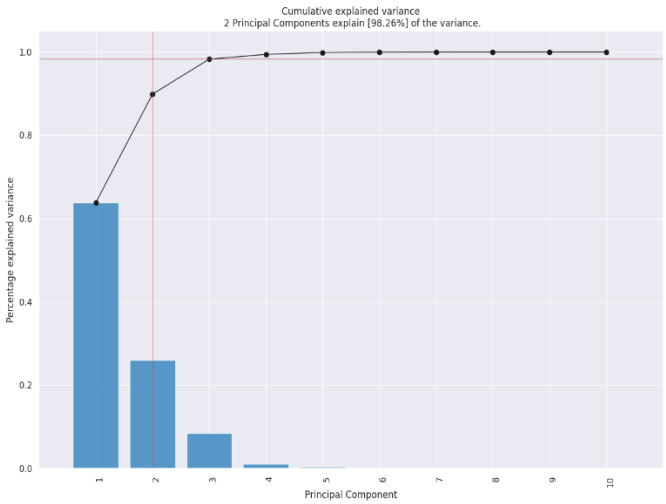

Figure 5: Explained Variance Pareto Chart

Figure 5 is the pareto chart of the contribution of each of our components to the total explained variance, this is further proof supporting our analysis that the first two features represent most of the explained variance in our data.

Figure 6: PC coefficient plot

Figure 6 represents the PC coefficient plot, which showcases how each original feature contributes to our two principal components ‘A1 and A2’. We can notice that MajorAxisLength, Perimeter, Area, and EquivDiameter have strong negative coefficients along the A1 axis which signifies that they contribute significantly, this allows us to conclude that A1 mostly encompasses variation that relates to object size. Looking at our A2 axis we can distinguish a set of features like: Roundness and Extent, have high positive coefficients while previously mentioned features display negative values which means that A2 focuses more on compactness.

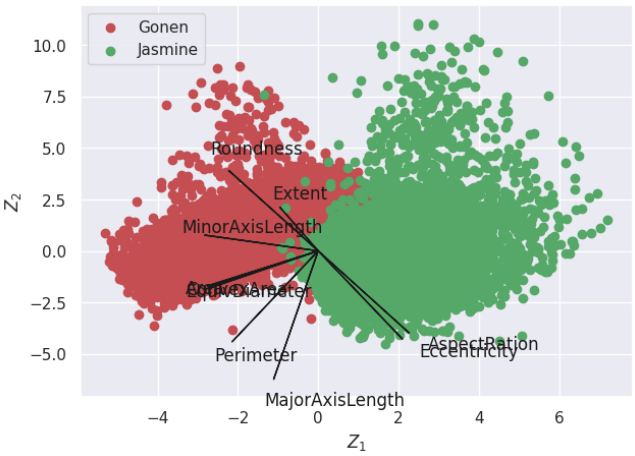

Figure 7: Biplot

Figure 7 shows clearly that our 2 PCS distinguish clearly the two categories of rice (Jasmine and Gonen), with features that correlate to elongation and aspect ratio are considered main factors to distinguish Jasmine, while other features that correlate to roundness and compactness identify Gonen. The directions of each of our lines also interpret how each feature contributes to the separation direction.

Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".

- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

VI. Classification Results

We will be applying classification models to our data, pre PCA and post PCA to analyze the impact of implementing PCA to our dataset. In order to do that, we've imported the **pycaret.classification** library and with a training size of 70% of our original data, we've compared between all available models and picked the top 2 contenders, before PCA we've noticed that the best classifications were: Linear Regression and Random Forest Classifier.

After integrating PCA, we've noticed a shift in classification models ranks, with Gradient Boosting Classifier taking the lead followed by K neighbors classifier, while Random forest was ranked fourth and Logistic regression dropped all the way down to the ninth position, we've opted to use Gradient boosting classifier to showcase the most optimized results, but for the sake of comparison we've kept Random Forest Classifier to compare results pre and post PCA integration.

Using the confusion matrix output by each of our classification models pre and post PCA we compared.

A. Model Comparison Before PCA

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.9890	0.9988	0.9932	0.9880	0.9906	0.9778	0.9778	0.6850
rf	Random Forest Classifier	0.9895	0.9979	0.9925	0.9884	0.9905	0.9788	0.9789	2.1140
qda	Quadratic Discriminant Analysis	0.9892	0.9986	0.9936	0.9867	0.9902	0.9781	0.9782	0.0350
ada	Ada Boost Classifier	0.9890	0.9987	0.9917	0.9883	0.9900	0.9778	0.9778	0.8560
et	Extra Trees Classifier	0.9890	0.9980	0.9914	0.9886	0.9900	0.9778	0.9778	0.4870
xgboost	Extreme Gradient Boosting	0.9890	0.9984	0.9922	0.9878	0.9900	0.9778	0.9778	0.2070
lightgbm	Light Gradient Boosting Machine	0.9889	0.9985	0.9916	0.9883	0.9899	0.9776	0.9776	0.7490
gbc	Gradient Boosting Classifier	0.9888	0.9983	0.9924	0.9873	0.9898	0.9774	0.9775	3.7470
ridge	Ridge Classifier	0.9877	0.9984	0.9971	0.9807	0.9889	0.9751	0.9753	0.0350
lda	Linear Discriminant Analysis	0.9873	0.9988	0.9963	0.9807	0.9885	0.9742	0.9744	0.1830
dt	Decision Tree Classifier	0.9839	0.9839	0.9841	0.9866	0.9853	0.9676	0.9676	0.1120
knn	K Neighbors Classifier	0.9804	0.9942	0.9938	0.9712	0.9824	0.9604	0.9608	0.1100
nb	Naive Bayes	0.9770	0.9983	0.9978	0.9618	0.9794	0.9535	0.9543	0.0340
svm	SVM - Linear Kernel	0.9733	0.9977	0.9892	0.9639	0.9760	0.9459	0.9472	0.0750
dummy	Dummy Classifier	0.5479	0.5000	1.0000	0.5479	0.7079	0.0000	0.0000	0.0280

B. Model Comparison After PCA

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9881	0.9979	0.9911	0.9872	0.9891	0.9759	0.9759	3.2140
knn	K Neighbors Classifier	0.9875	0.9943	0.9913	0.9861	0.9887	0.9748	0.9748	0.1250
ada	Ada Boost Classifier	0.9873	0.9984	0.9900	0.9869	0.9884	0.9743	0.9743	0.4510
rf	Random Forest Classifier	0.9871	0.9968	0.9908	0.9858	0.9883	0.9740	0.9740	1.0220
xgboost	Extreme Gradient Boosting	0.9867	0.9981	0.9903	0.9856	0.9879	0.9732	0.9732	0.1410
qda	Quadratic Discriminant Analysis	0.9866	0.9980	0.9964	0.9795	0.9879	0.9728	0.9730	0.0680
et	Extra Trees Classifier	0.9863	0.9974	0.9894	0.9856	0.9875	0.9722	0.9723	0.4180
lightgbm	Light Gradient Boosting Machine	0.9863	0.9983	0.9898	0.9852	0.9875	0.9722	0.9722	0.4900
lr	Logistic Regression	0.9858	0.9980	0.9916	0.9827	0.9871	0.9713	0.9713	0.0720
svm	SVM - Linear Kernel	0.9852	0.9981	0.9897	0.9834	0.9865	0.9700	0.9700	0.0750
ridge	Ridge Classifier	0.9833	0.9979	0.9918	0.9781	0.9849	0.9663	0.9664	0.0930
lda	Linear Discriminant Analysis	0.9833	0.9979	0.9918	0.9781	0.9849	0.9663	0.9664	0.0570
dt	Decision Tree Classifier	0.9816	0.9814	0.9840	0.9826	0.9833	0.9629	0.9629	0.0720
nb	Naive Bayes	0.9776	0.9970	0.9966	0.9639	0.9800	0.9546	0.9553	0.0560
dummy	Dummy Classifier	0.5491	0.5000	1.0000	0.5491	0.7089	0.0000	0.0000	0.0610

C. Before PCA

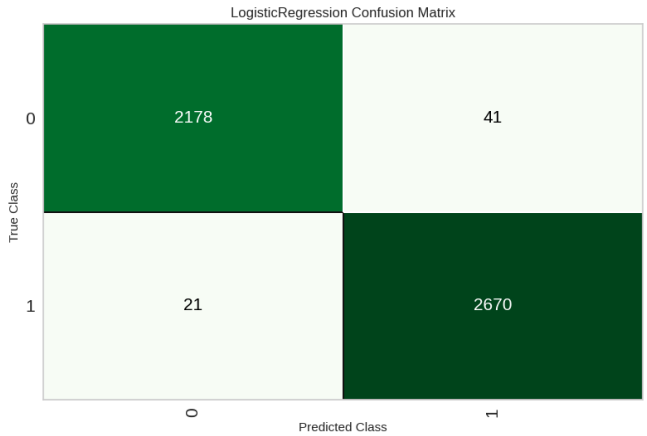

Figure 8: Confusion Matrix LR Pre-PCA

From the confusion matrix provided by logistic regression we can see that 2178 Gonen rice grains were correctly predicted as Gonen and 2670 Jasmine rice grains were correctly identified. While 41 Jasmine grains were falsely identified as Gonen, and 21 Gonen grains were identified as Jasmine.

This means that our classification has an accuracy score of: 98.9%.

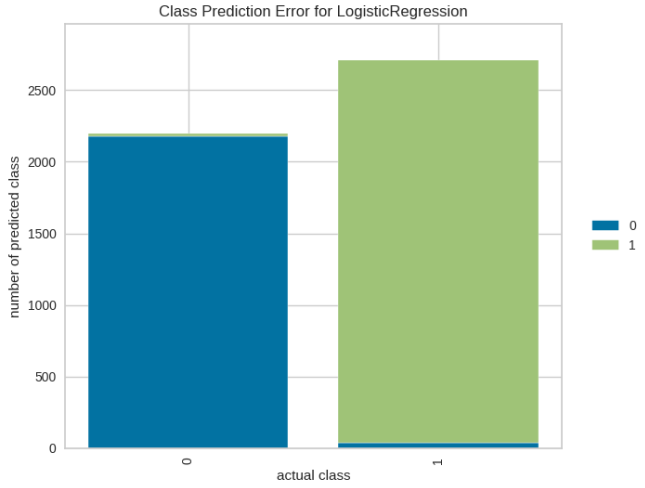

Figure 9: Prediction Error LR post-PCA

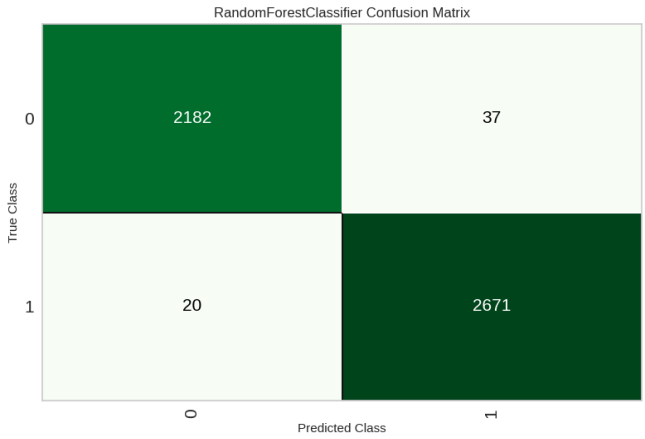

Figure 10: Confusion Matrix RF pre-PCA

Using Random Forest Classifier we can extract from the following confusion matrix that 2182 Gonen Rice grains were correctly predicted and 2671 Jasmine grains were correctly identified while only 37 Jasmine grains were falsely identified as Gonen grains and 20 Gonen grains were falsely identified as Jasmine grains. This totals to an accuracy score of 98.94%.

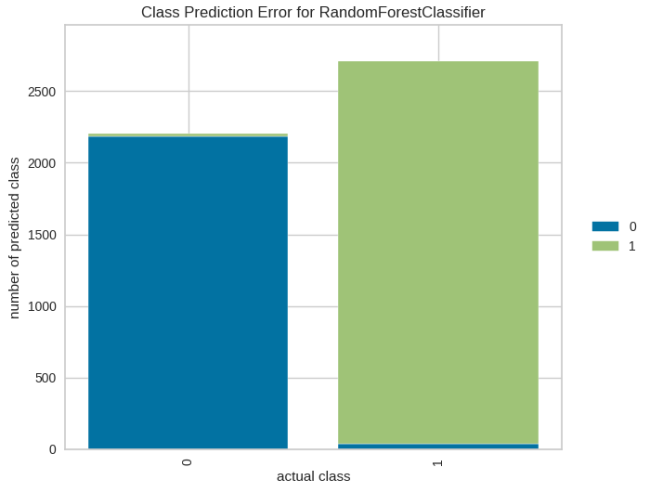

Figure 11: Prediction Error RF Post-PCA

D. After PCA

After applying PCA to our dataset, we've got the following results

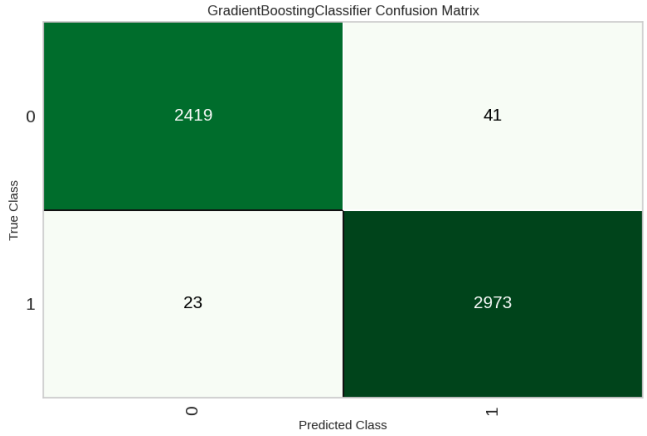

Figure 12: Confusion Matrix GB Post-PCA

After the implementation of PCA, as previously mentioned, Gradient Boosting was on top of the models list, and from the confusion matrix we can conclude that our model identifies much more grains correctly.

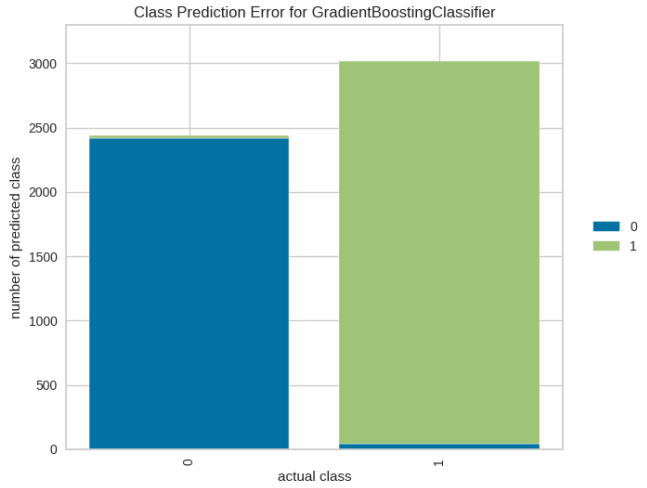

Figure 13: Prediction Error GB post-PCA

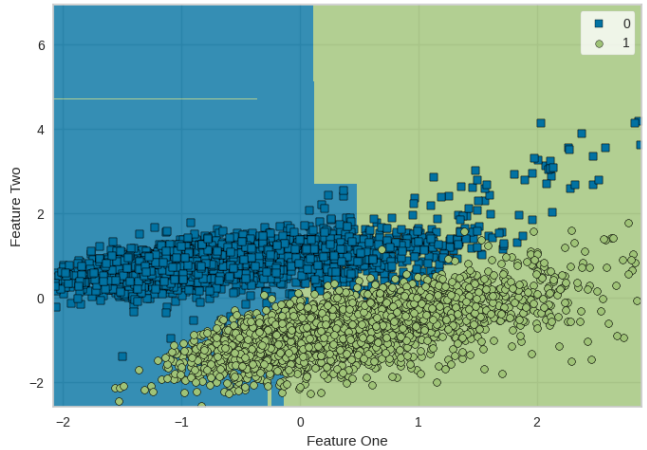

Figure 14: Decision Boundary GB post-PCA

This figure describes how the model relied on each of the features to separate between the two rice types (Jasmine and Gonen) using the our 2 principal components, and where t classifies each grain whenever it predicts the type of rice.

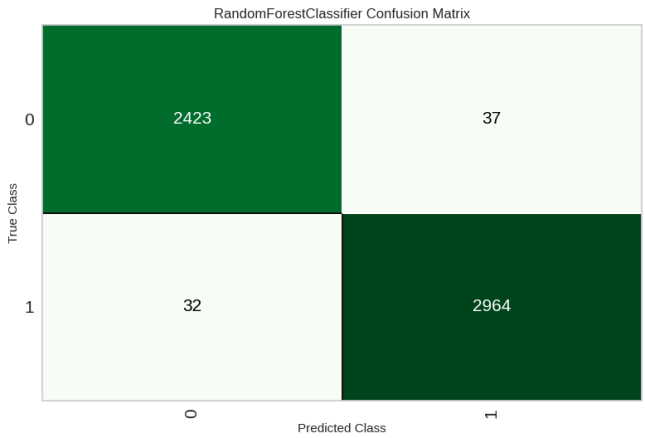

Figure 15: Confusion Matrix RF Post-PCA

We've opted to use RF for the sake of comparison between before and after PCA implementation although it didn't rank in the top two models to consider after PCA, and we can also notice improvements to our prediction model.

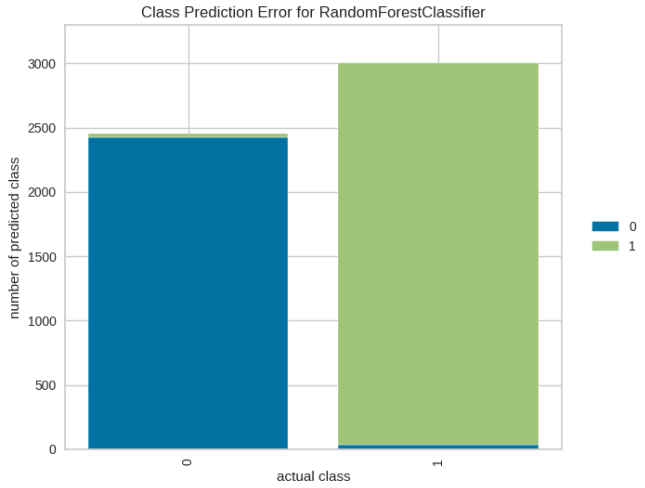

Figure 16: Prediction Error RF Post-PCA

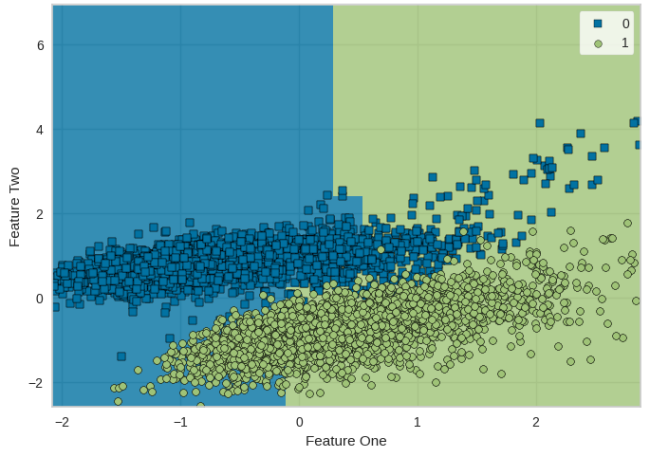

Figure 17: Decision Boundary RF Post-PCA

E. Remarks

The decision boundary plot illustrates how the Random Forest classifier distinguishes between the two rice varieties using the first two principal components obtained via PCA (represented here as Feature One and Feature Two). Each point on the plot corresponds to a sample from the dataset, with class 0 samples represented by blue squares and class 1 samples by green circles. The shaded regions indicate the areas where the classifier predicts each class, with blue denoting class 0 and green denoting class 1.
The visualization demonstrates that the PCA transformation effectively captured the variance necessary to differentiate the two rice varieties, enabling a clear separation in the transformed feature space. Most samples from each class fall within their respective prediction regions, indicating high classification accuracy.

F. SHAP Summary Plot

Using RF classifier, the following SHAP summary was plotted:

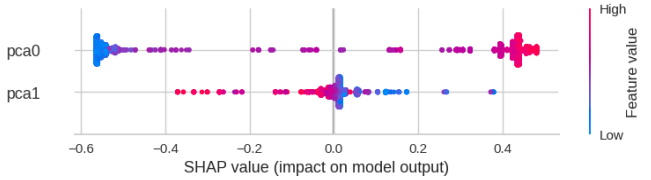

Figure 18: SHAP Summary Plot

From this SHAP summary plot we can notice how each of our principal components (PCA0, PCA1) contribute to identifying and predicting the type of grain analyzed, where we can see that PCA0 has the most notable impact on the output of our classification model, where high values of PCA0 (pink and red) lead to class 1 Jasmine and lower values tend to lead to class 0 Gonen. While PCA1 our second component plays only a supporting role to PCA0 with its values mainly clustered around 0.

Figure 19: SHAP values contributions on all predictions

This SHAP value summary plot visualizes the contribution of the first principal component (PCA0) to the predictions made by a Random Forest classifier trained to distinguish between two types of rice. The X-axis represents individual samples sorted by model output, while the Y-axis shows the SHAP value, which quantifies the impact of PCA0 on the prediction for each sample. Each line traces the effect of PCA0 across all samples. The color gradient represents the actual value of PCA0 for each sample—red indicating higher values and blue indicating lower ones. The plot reveals that PCA0 plays a dominant role in the model's decision-making, with high values (red) pushing predictions toward one rice class and

low values (blue) toward the other. The sharp transition in the middle of the plot indicates that PCA0 effectively separates the two classes, and the model relies heavily on this single component. This demonstrates that the dimensionality reduction via PCA was successful in capturing key discriminatory features between the rice varieties.

VII. Conclusion

In conclusion, through this project we've been able to evaluate the importance of PCA in reducing noise and redundancy in datasets, and how implementing this method greatly helps classification models in providing better output in less time, and this through the observation of components where sometimes only a small number of components represents a very high percentage of variance in the data, in our case 2 components represented 89.9%. Which leads us to also noticing how PCA sometimes changes the classification models, as logistic regression was in the lead along random forest before PCA but we've seen that Gradient Boosting was more precise after PCA. Finally using our SHAP values we've gained more insight on how each of our 2 components plays a role in predicting, where we noticed that PCA0 was the main component and PCA1 was a supporting one.

References

[1] I. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer, 2002.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] S. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. doi:10.1002/wics.101

[3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. doi:10.1023/A:1010933404324

[4] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[6] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] T. K. Ghosh, "Application of PCA and machine learning in agriculture," *International Journal of Computer Applications*, vol. 178, no. 39, pp. 7–10, 2019.

[8] A. W. M. S. Arulampalam et al., "Image-based rice grain classification using PCA and SVM," *International Conference on Signal and Image Processing*, 2012. doi:10.1109/ICSIP.2012.6418775 "