Principal Component Analysis and Multi-Class Classification of Pumpkin Seeds

Gaurav Patyal, 40195250
https://github.com/gpatyal/INSE6220GAURAVPATYAL

Abstract- Principal Component Analysis (PCA) is a statistical method for reducing the dimensionality of mammoth data sets by converting correlated data to uncorrelated data. PCA's basic idea is to reduce the number of variables in a data set while retaining as much information as possible. It achieves this by randomly generating new uncorrelated variables that gradually maximize variance.
I have applied PCA to the Pumpkin Seeds data set in which two types of pumpkin seeds have been taken. Methods used are Logistic Regression, K Neighbors Classifier, Linear Discriminant Analysis, and Extra Trees Classifier algorithms.

Index terms- Principal component analysis(PCA), Logistic regression, Random tree classifier, confusion matrix

I. INTRODUCTION

Healthy food has always been a good choice for the people nowadays. Adding foods that are full of protein, fat, mineral, and carbs are very important. Pumpkin seeds have been the best choice for the people that contains all the nutrients that the body needs. This data consists of two of most crucial quality types of pumpkin seeds like Urgupsivrisi and Cercevelik. These kinds of seeds are grown mostly in Urgup and Karacaoren regions of Turkey.

Several techniques are used for the analysis of the data like PCA, box plot, Scree plot, Pareto Plot, and few Machine learning models are also used to find the correlation between the quality types of pumpkin seeds. Moreover, I have concluded the study.

II. PUMPKIN SEEDS STATISTICS

The dataset was taken from Kaggle. It consists of 7 Columns and 294 Rows. The data includes 294 data that further were classified into 2 seeds type classes. The columns are defined as:

A. Eccentricity-EC
B. Solidity-SL
C. Extent- EX
D. Roundness-RD
E. Aspect Ration-AR
F. Compactness-CP

The following table shows the first 50 dataset following will be a short description of the various features.

EC	SL	EX	RD	AR	CP	class
0.7376	0.9902	0.7453	0.8963	1.4809	0.8207	0
0.8275	0.9916	0.7151	0.844	1.7811	0.7487	0
0.8749	0.9857	0.74	0.7674	2.0651	0.6929	0
0.8123	0.9902	0.7396	0.8486	1.7146	0.7624	0
0.8187	0.985	0.6752	0.8338	1.7413	0.7557	0
0.8215	0.9895	0.7165	0.848	1.7535	0.7522	0
0.7938	0.9929	0.7187	0.8857	1.6443	0.779	0
0.8646	0.9894	0.6736	0.7957	1.9903	0.7067	0
0.8573	0.9886	0.6188	0.793	1.9423	0.7152	0
0.8356	0.9916	0.7443	0.8409	1.8201	0.7394	0
0.8327	0.9917	0.7019	0.844	1.8057	0.7431	0
0.8356	0.9898	0.7457	0.832	1.8205	0.7391	0
0.9011	0.9888	0.6	0.7461	2.3061	0.6566	0
0.8144	0.9913	0.7285	0.8503	1.7231	0.759	0
0.8449	0.9908	0.7377	0.8274	1.8695	0.7304	0
0.838	0.9878	0.7124	0.8357	1.8328	0.7365	0
0.8221	0.99	0.7391	0.8455	1.7565	0.7515	0
0.8429	0.9886	0.6728	0.8307	1.8585	0.7323	0
0.8523	0.9934	0.7692	0.844	1.9117	0.7225	0
0.8439	0.9885	0.7403	0.8331	1.8638	0.7313	0
0.8621	0.9872	0.7469	0.8098	1.9735	0.7104	0
0.8748	0.9904	0.6702	0.7933	2.0643	0.6939	0
0.782	0.9923	0.7237	0.8903	1.6045	0.7884	0
0.8167	0.9923	0.7386	0.8595	1.7328	0.759	0
0.8522	0.9906	0.6526	0.827	1.9115	0.7216	0
0.8176	0.9884	0.6779	0.8497	1.7366	0.7558	0
0.8179	0.9915	0.6861	0.8458	1.738	0.7572	0

0.8064	0.9913	0.7156	0.8761	1.6909	0.7682	0
0.8037	0.9908	0.675	0.8626	1.6806	0.7673	0
0.8166	0.9903	0.7407	0.863	1.7326	0.759	0
0.8189	0.9915	0.6746	0.8636	1.7423	0.7567	0
0.8504	0.9882	0.6923	0.8118	1.9006	0.7237	0
0.8087	0.9874	0.7381	0.8536	1.6999	0.7654	0
0.8095	0.9915	0.7542	0.8716	1.7032	0.7637	0
0.832	0.9917	0.7537	0.8459	1.8024	0.744	0
0.8085	0.9881	0.696	0.844	1.6991	0.7649	0
0.8238	0.9871	0.6765	0.8414	1.7642	0.7516	0
0.7967	0.9905	0.7201	0.8608	1.6545	0.7753	0
0.7709	0.9917	0.7169	0.8894	1.5701	0.7972	0
0.8737	0.9882	0.7354	0.781	2.0556	0.6961	0
0.8702	0.9907	0.7514	0.8047	2.0297	0.701	0
0.8624	0.9898	0.7419	0.8117	1.9751	0.7092	0
0.854	0.9891	0.6624	0.8022	1.9218	0.7202	0
0.8337	0.983	0.6303	0.7986	1.8107	0.7393	0
0.7701	0.9912	0.7266	0.8743	1.5675	0.797	0
0.8752	0.99	0.6644	0.786	2.0674	0.694	0
0.7703	0.9878	0.7015	0.859	1.5681	0.7977	0
0.8003	0.9857	0.7191	0.845	1.6677	0.772	0
0.7611	0.976	0.6867	0.7924	1.5416	0.7943	0
0.771	0.9924	0.7359	0.8754	1.5703	0.7959	0

Fig.1 Dataset for first 50 observation.

Factors	EC	SL	EX	RD	AR	CP
Mean	0.86	0.989	0.68	0.789	2.05	0.054
SD	0.045	0.003	0.064	0.058	0.33	0.703
Max	0.946	0.994	0.778	0.922	3.08	0.829
Min	0.725	0.957	0.501	0.627	1.45	0.567

Fig.2 Summary of each column

III. PRINCIPAL COMPONENT ANAYLSIS

There are enormous datasets in the world which are hard to interpret. Principal Component Analysis (PCA) is a statistical method for reducing the dimensionality of mammoth data sets by converting correlated data to uncorrelated data. PCA's basic idea is to reduce the number of variables in a data set while retaining as much information as possible. It achieves this by randomly generating new uncorrelated variables that gradually maximize variance.

Today, principal component analysis is one of the most widely used multivariate statistical techniques. The first PC displays as much of the dataset's variability as possible. The remaining PCs provide information about the rest of the variation in the dataset.

PCA ALGORITHM

Suppose there is data matrix X, the PCA algorithm is applied to dataset X with the following steps:

Step-1: Center the Data
Compute the centered data matrix $Y = HX$ by subtracting off column means.

Step-2: Covariance Matrix Computation
Here, we will compute the $p \times p$ covariance matrix S of the centered data matrix as given

$$S = \frac{1}{n-1} Y'Y$$

Step-3: Eigen Decomposition
Next step is to compute the eigenvectors and eigenvalues of S with the help of eigen decomposition

$$S = A \wedge A' = \sum_{j=1}^{p} \succ_j a_j a'_j$$

Step-4: Principal Component
Final step is to compute the transformed data matrix $Z = Y A$ of size $n \times p$

$$Z = \left(z'_1, z'_2, \ldots., z'_i, \ldots., z'_p \right) = \begin{pmatrix} z_{11} & \cdots & z_{1p} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{np} \end{pmatrix}$$

IV. CLASSIFICATION ALGORITHM

Classification algorithm is a type of Supervised Learning technique that recognizes the category of new observations based on training data. A classification algorithm in machine learning is like grouping things with similar characteristics. We will use a variety of classification algorithms in this project, including logistic regression, K-nearest Neighbors, random forest, and Decision tree.

- Logistic Regression
  One of the popular machine learning algorithms that is available is logistic regression that defined under supervised learning technique. A categorical dependent variable's output is predicted using logistic regression. As a result, the result must be a categorical or discrete value. It can be Yes or No, 0 or 1, true or

False, and so on, but it does not provide us the exact values as 0 and 1, it presents the probabilistic values that fall between 0 and 1.

- K-nearest Neighbors
  It is considered from one of the simplest machine learning algorithms based on supervised learning technique. K-NN algorithm is used to place the data to the nearest of its category i.e., neighbor category. Mostly used for classification problems but can also be used for regression.

- Random Forest
  Another popular machine learning algorithm that comes under supervised learning technique. To refine the performance of the model and answer a complex problem, multiple classifiers are combined, and this process is known as ensemble learning. There are number of decision tress that results a class prediction and the class that got majority of votes will be our final output. The more the number of trees in the forests, the higher the accuracy.

- Decision Tree
  It can be used to answer regression and classification problems, but it is preferred for classification problems. In this feature of the dataset can be represented by internal nodes, decision rules are represented by branches, and every node represents the outcome. It is capable of thinking as much like human's ability to think while deciding which makes it easier to understand.

V. DATASET DESCRIPTION

This dataset was comprised from two regions from turkey i.e., Urgup and Karacaoren. Measurements were made possible after using gray and binary forms of threshold techniques. Pie Chart mentioned below explains the classes in the pumpkin.

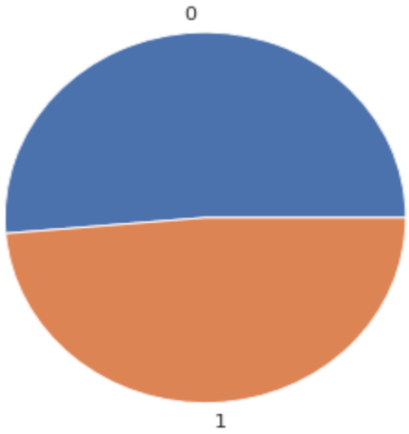

Fig.3 Pie Chart

Figure 4 shows a box plot that is having all the observations in the form of lines and box. The central line in the box indicates the median value whereas the lines that are extending outside the box represents range. Outliers can be seen as the points lying outside the lines of the box pot.

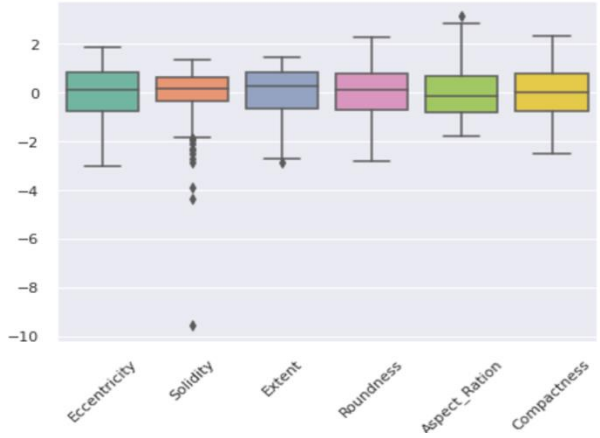

Fig.4 Box Plot

Figure 5 shows the swarm plot that represents all the observations which is also considered to be an extended version of box plot. As you can see there are few outliers as well.

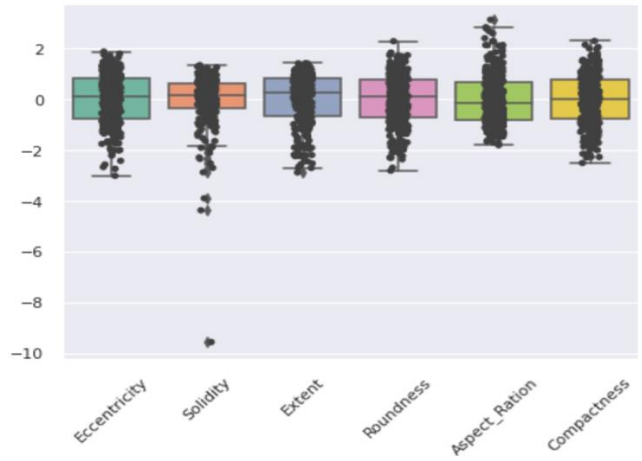

Fig.5 Swarm Plot

In figure 6, the pair plot is shown. The easiest way to show the relationships between each variable. It is basically a module which gives the relationships in a dataset.

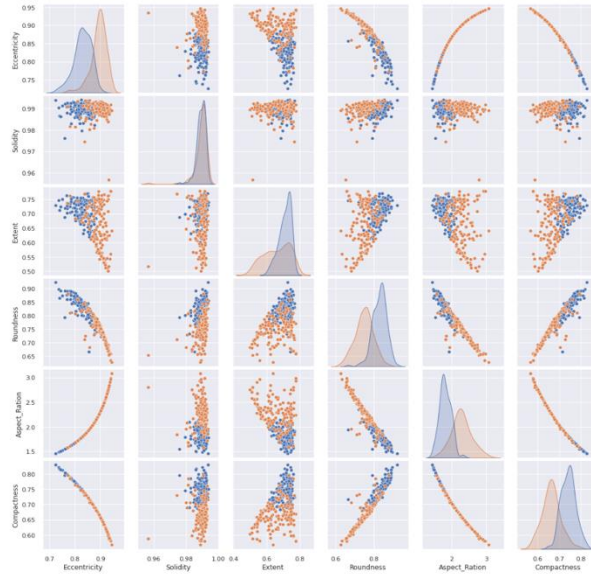

Fig.6 Pair Plot

The Covariance matrix tells us about differences between two random variables. It is used to compute the covariance between each column of a data matrix. Also known as dispersion matrix.

Because the variables are corelated, the diagonal value is 1. Covariance is high between Eccentricity and aspect ratio, extent and roundness, compactness, and extent. Whereas the aspect ratio is negatively correlated with most of the other variables. Covariance matrix calculation can also be expressed as

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})^T$$

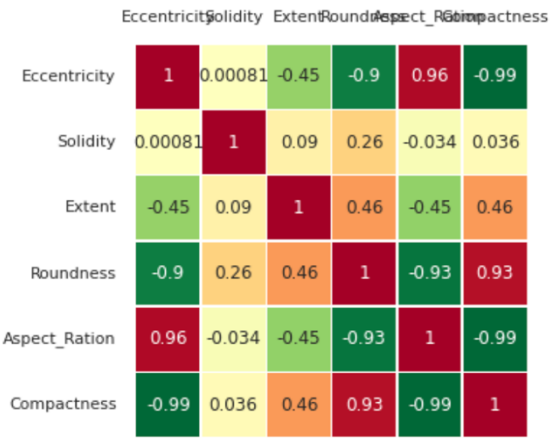

Fig.7 Covariance Matrix

VI. PCA RESULT

Following, PCA was applied to the pumpkin seed dataset which was explained in the Section I & II. As the steps performed in section II reduced the data set from seven feature (p=6) to *r* features with *r* < 6. By using eigenvector matrix(A), the original *n* x *p* data set was decreased.

Here is the eigenvectors matrix[A] =

$$\begin{bmatrix} 4.765 & -1.337 & 1.271 & -6.076 & 4.871 & -3.637 \\ -5.815 & -9.641 & -1.370 & 2.017 & 8.646 & 9.168 \\ -2.796 & -1.175 & 9.528 & 4.933 & -4.050 & 1.667 \\ -4.720 & -1.428 & -1.524 & -7.351 & -4.389 & 2.092 \\ 4.820 & -9.516 & 1.299 & 1.057 & -7.307 & -4.433 \\ -4.859 & 9.624 & -1.303 & 1.961 & 1.689 & -8.188 \end{bmatrix}$$

The eigenvalues are represented as:

$$\lambda = \begin{bmatrix} 4.1389 \\ 1.0463 \\ 0.7326 \\ 0.0662 \\ 0.0357 \\ 0.0004 \end{bmatrix}$$

Eigenvalues can be defined as the variations that are present in each PC captures in the data. To depict the level of variance that each PC accounts for, a scree plot and a pareto plot are used. The following formula can be used to calculate the variance percentage of j^{th} value:

$$l_j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100\%, \quad for\ j = 1, \ldots, p$$

where the j^{th} element in the PC is represented by λ_j.

Figure 8 shows the scatter plot for the observations.

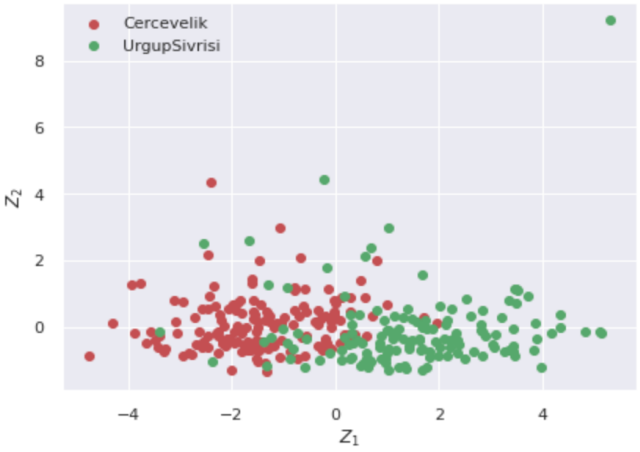

Fig.8 Scatter Plot

Figure 9 can be defined as Scree Plot that shows in which Y-axis shows the explained variance and X-axis shows the number of components.

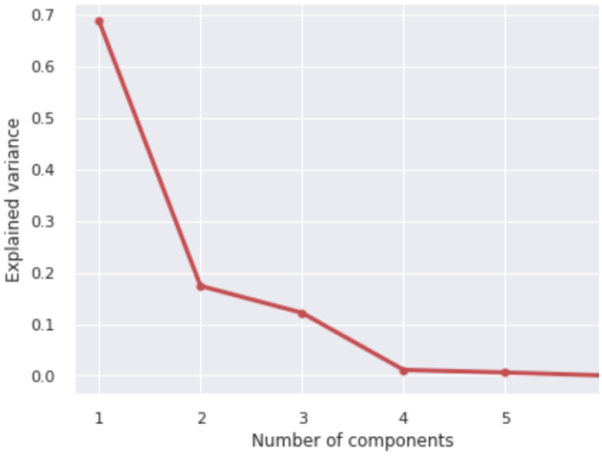

Fig.9 Scree Plot

From the figure 10 and 11, we can say that the first three PCs are responsible for over **99.39%**. In pareto plot, Y-axis is represented by the explained variance percentage bar.

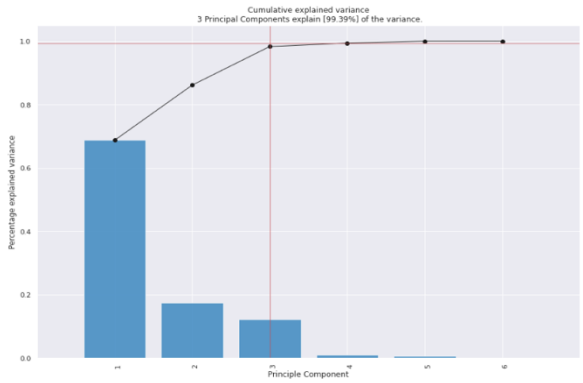

Fig.10 Pareto Plot

With the use of eigenvector matrix, we can easily write the Z_1, Z_2, and Z_3value.

The first principal component (Z_1) is

$Z_1 = 4.765\ X_1 - 5.815\ X_2$-$2.796X_3 - 4.720X_4$+$4.820X_5 - 4.859X_6$

When we look at first PC(Z_1), we can easily say that X_1 and X_5 have the highest contribution. And every component has good contributions to the first PC.

The second principal component (Z_2) is

$Z_2 = -1.337\ X_1 - 9.641\ X_2$-$1.175X_3 - 1.428X_4$-$9.516X_5 + 9.624X_6$

Again, when we look at the second PC (Z_2), we can say that X_6 has the highest contribution in the PC. Other components like $X_1, X_2, X_3, X_4 and\ X_5$ are smaller but can be considered.

The third principal component (Z_3) is

$Z_3 = 1.271\ X_1 - 1.370\ X_2$+$9.528X_3 - 1.524X_4$+$1.299X_5 + -1.303X_6$

Finally, in the third principal component (Z_3), we can say that X_1, X_3, and X_5 contribute to the highest contribution in the PC. Other components have also contributed in the PC.

Figure 11

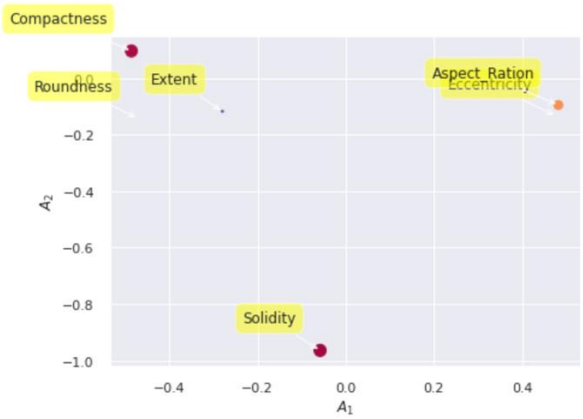

Fig.11 PC coefficient plot

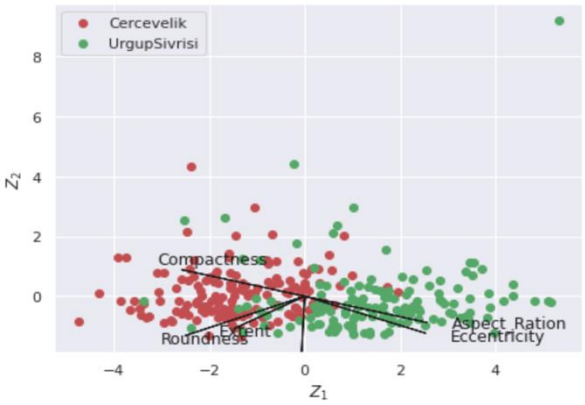

Fig.12 Biplot

Figure 12 shows Biplot which represents two types of pumpkin seeds i.e., Cercevelik and UrgupSivrisi. The Cercevelik is represented by red color on the left side of the plot. On the other hand, UrgupSivrisi is shown by green color on the right side of the plot. From this figure we can see that Aspect ratio and Eccentricity has the positive relation with the first principal component as shown above in Eq.Z_1.

VII. CLASSIFICATION RESULT

We have applied classification algorithm on pumpkin seeds with the help of Pycaret library and shap. To train and test the dataset we have used common classification algorithm like K-Neighbors classifier, Decision tree classifier, random forest classifier, and logistic regression.

To start off, we have split the dataset for 70-30%. After this we have used an inbuild method in pycaret known as Compare model that is used to compare all the models and the best model is returned.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.8857	0.9390	0.8264	0.9296	0.8682	0.7682	0.7799	0.466
ridge	Ridge Classifier	0.8751	0.0000	0.8014	0.9290	0.8480	0.7446	0.7619	0.013
qda	Quadratic Discriminant Analysis	0.8749	0.9315	0.8250	0.9178	0.8556	0.7452	0.7621	0.014
lda	Linear Discriminant Analysis	0.8749	0.9386	0.8264	0.9164	0.8620	0.7466	0.7580	0.013
lightgbm	Light Gradient Boosting Machine	0.8743	0.9186	0.8486	0.8890	0.8620	0.7459	0.7543	0.062
et	Extra Trees Classifier	0.8696	0.9568	0.8264	0.8952	0.8542	0.7361	0.7437	0.453
nb	Naive Bayes	0.8646	0.9295	0.8611	0.8752	0.8606	0.7273	0.7374	0.013
knn	K Neighbors Classifier	0.8640	0.9232	0.7681	0.9344	0.8311	0.7223	0.7422	0.113
lr	Logistic Regression	0.8594	0.9295	0.8139	0.8899	0.8402	0.7146	0.7279	0.386
dt	Decision Tree Classifier	0.8579	0.8568	0.8625	0.8558	0.8545	0.7145	0.7215	0.014
gbc	Gradient Boosting Classifier	0.8535	0.9272	0.8361	0.8716	0.8392	0.7039	0.7220	0.078
ada	Ada Boost Classifier	0.8424	0.9295	0.8250	0.8574	0.8286	0.6813	0.6967	0.092
svm	SVM - Linear Kernel	0.7629	0.0000	0.8056	0.8003	0.7469	0.5253	0.5747	0.012
dummy	Dummy Classifier	0.5243	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.011

Fig.13 Compare model

Figure 14 will show the best model i.e., Random Forest classifier.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=-1, oob_score=False, random_state=123, verbose=0,
                       warm_start=False)
```

Fig.15 Best Model

Following that will be confusion matrix [Fig.15] of the random forest classifier that explains performance of the algorithm. It is very simple to understand, and correct predictions can be made by analyzing the diagonals and incorrect predictions can be made through off diagonals.

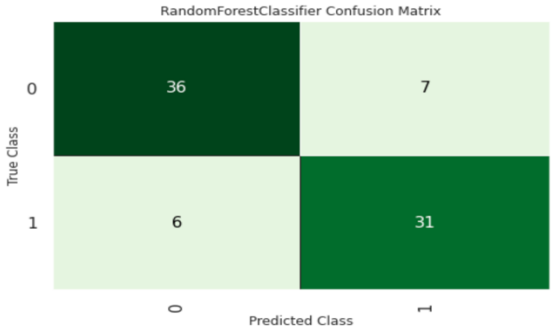

Fig.16 Confusion matrix

Different models are generated for KNN, random forest, logistic regression, and decision tree matrix. After which tuning will be done for the algorithms that will help the model's performance without disturbing the variation.

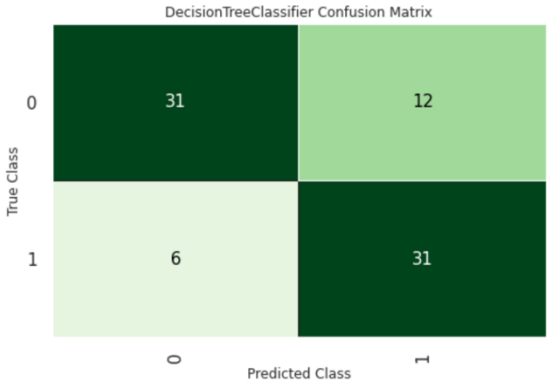

Fig.17 Confusion matrix for Decision tree classifier

Figure 17 explains about algorithm classification. Dark green diagonal boxes tell us that how many sample were correctly classified and other off diagonal tells us the samples that were wrongly classified.

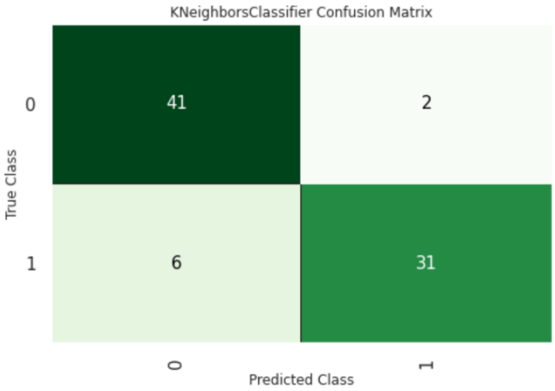

Fig.18 Confusion matrix for K-Neighbor Classifier

Figure 18 gives an idea about the prediction done by the algorithm. Algorithm classified 41 samples correctly for class 0. And 31 samples for the class 1.

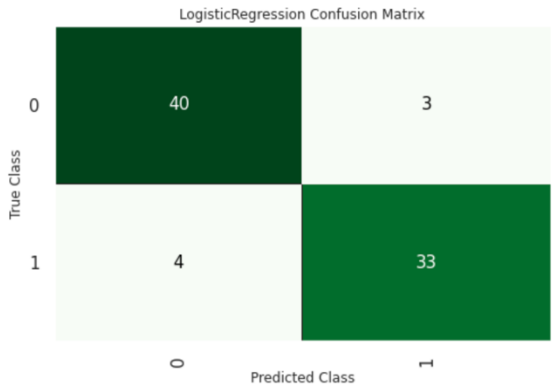

Fig.19 Confusion matrix for logistic regression

Figure 19 shows that the algorithm classified 40 samples correctly for class 0 and 33 samples correctly for class 1. On the other hand, 3 samples were misclassified for class 0 and 4 samples for class 1.

Moreover, after using PCA the best model that we got was logistic regression. After tuning and testing we got the below accuracy table. As we can the after using the PCA, best model for the dataset has been changed i.e., logistic regression.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8643	0.9317	0.8486	0.8816	0.8558	0.7257	0.7369	0.265
ridge	Ridge Classifier	0.8591	0.0000	0.8139	0.8952	0.8409	0.7138	0.7291	0.010
lda	Linear Discriminant Analysis	0.8591	0.9328	0.8139	0.8952	0.8409	0.7138	0.7291	0.011
knn	K Neighbors Classifier	0.8585	0.9230	0.8139	0.8975	0.8411	0.7129	0.7288	0.112
nb	Naive Bayes	0.8538	0.9227	0.8375	0.8827	0.8467	0.7057	0.7230	0.012
rf	Random Forest Classifier	0.8538	0.9304	0.8028	0.8899	0.8343	0.7035	0.7174	0.454
qda	Quadratic Discriminant Analysis	0.8535	0.9275	0.8028	0.9052	0.8330	0.7030	0.7258	0.013
gbc	Gradient Boosting Classifier	0.8482	0.9081	0.8125	0.8787	0.8282	0.6919	0.7112	0.059
lightgbm	Light Gradient Boosting Machine	0.8482	0.9265	0.8139	0.8833	0.8322	0.6928	0.7137	0.050
ada	Ada Boost Classifier	0.8430	0.9038	0.8264	0.8576	0.8343	0.6837	0.6933	0.074
et	Extra Trees Classifier	0.8427	0.9178	0.7694	0.8989	0.8177	0.6803	0.6970	0.413
svm	SVM - Linear Kernel	0.8424	0.0000	0.8278	0.8589	0.8346	0.6831	0.6947	0.011
dt	Decision Tree Classifier	0.8003	0.7974	0.7792	0.8191	0.7805	0.5969	0.6192	0.013
dummy	Dummy Classifier	0.5243	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.011

Fig.20 Best model using PCA

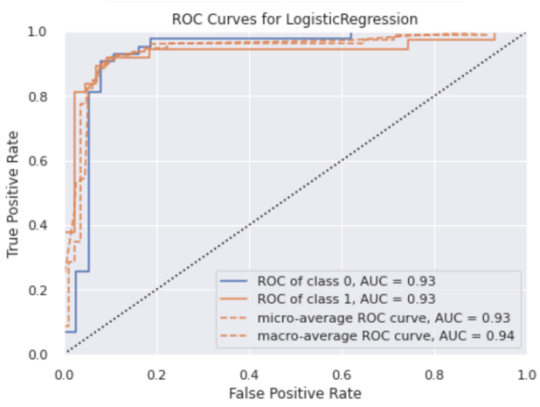

Fig. 21 AUC for Logistic regression

Figure 21 gives an explanation about the classifier's ability to differentiate between the classes. Model would be good at predicting if the AUC is higher.

VIII. EXPLAINED AI WITH SHAPLEY VALUES

In this section, we will use the Shapley values to make decisions about the dataset. This will tell us that which feature add more value to the model outcome and up to which extend it can impact the model. SHAP stands for Shapley Additive exPlanations. Introduced by Lundberg and Lee(2017). Motive of this was to individually predict any machine learning model while explaining the same.

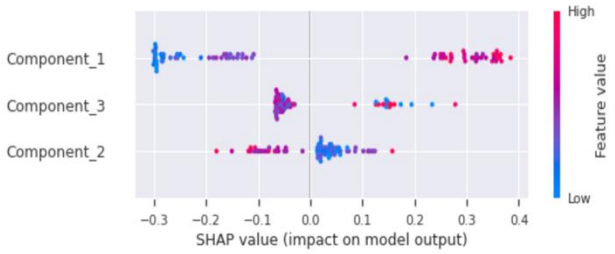

Fig.22

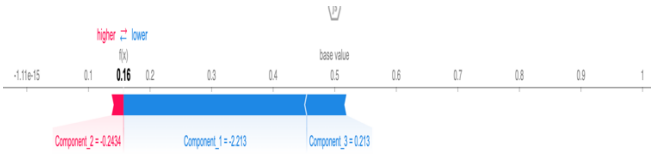

Fig.23

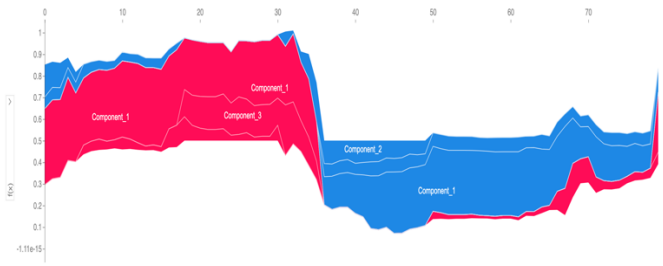

Fig.24

IX. Conclusion

To recapitulate, in this project we have applied Principal Component Analysis and classification algorithm from machine learning on the Pumpkin Seeds dataset which has 2 types of the seeds (Urgupsivrisi and Cercevelik). As we have seen that after using PCA, we got around 90% of the variation in the first two principal component. Moreover, we have used few machine learning algorithms on our dataset and found out that Random Forest Classifier was considered the best model. After which we applied PCA to this and observed that logistic regression was the best mode. At the end we applied SHAP values to get an idea of the feature which is most influencing the data.

REFERENCES

1. https://www.kaggle.com/datasets/muratkokludataset/pumpkin-seeds-dataset
2. https://www.javatpoint.com/machine-learning
3. https://monkeylearn.com/blog/classification-algorithms/
4. https://datascienceplus.com/understanding-the-covariance-matrix/
5. https://medium.com/analytics-vidhya/pairplot-visualization-16325cd725e6
6. https://www.projectpro.io/article/7-types-of-classification-algorithms-in-machine-learning/435
7. https://builtin.com/data-science/step-step-explanation-principal-component-analysis
8. https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202
9. https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c
10. https://c3.ai/glossary/data-science/shapley-values/
11. https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30#:~:text=In%20a%20nutshell%2C%20SHAP%20values,answer%20the%20%E2%80%9Chow%20much%E2%80%9D.
12. Ben Hamza, A. (2022). Advanced Statistical Approaches to Quality. Concordia Institute for Information Systems Engineering