Principal Component Analysis and Classification for Pizza dataset

Golboo Jamshidi
Student ID: 40227331
Gina Cody School of Engineering and Computer Science
Concordia University
Golboojamshidi@gmail.com

Github link:

https://github.com/Golboojamsh idi/INSE6220-Final-project.git

***Abstract*— Principal component analysis, or PCA, is a dimensionally-reduction method often used to reduce the dimensionality of extensive data sets by transforming a large collection of variables into a smaller one that still contains the most critical attribute of the large group. In this study, PCA has been applied to the Pizza data set, including information on the gradient of pizzas and brands of different restaurants considered as a class in this study. This dataset contains 301 observations. First, the "Brands" column in the dataset is converted to "Class", and different brands, which used to be A, B, C, D, E, F, G, H, I, and J, are converted to, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. In the next step, Different classifiers such as Random Forest, Logistic Regression and K-nearest neighbour were applied to the reduced dataset. Different evaluations have been obtained from each of these classifiers to describe the brands of various pizzas.**

A. K-nearest neighbour

Keywords—Principal component analysis, Classification, Regression, K-nearest neighbour,

I .INTRODUCTION

The food industry has undergone many changes in the past two decades, primarily due to the developing and implementation of new technology to meet growing consumer demands for convenience products. However, Current consumer trends bode well for companies able to develop healthy foods that taste good, are suitable for people, and are also good for on - the - go eating.[1]

Pizza, one of the most purchased items in retail food stores, has maintained its market share through the changing nature of the processed foods industry and has even grown in popularity. As a result, pizza is one of the more popular consumer foods. In addition, the trend towards international cuisine and convenience foods has boosted pizza markets in Europe, America, and other continents. As a result, pizza production has been increasing at unprecedented momentum and is expected to increase further in the next decade in response to a growing world population. [1]

The market now offers a wide variety of pizza to suit all palates and meal occasions, with various shapes and flavors and, more recently, health characteristics , and is available all year. This versatility, combined with their acceptance as a healthy and nutritious food, has resulted in their widespread popularity across all population subgroups.[2]

In this report, the Pizza ingredients dataset is first subjected to principal component analysis (PCA) to reduce dimensionality. The original dataset and PCA transformed dataset are then subjected to three popular classification algorithms: logistic regression (LR), K-nearest neighbour (K-NN), and Random Forest (RF). It is worth mentioning that the classification algorithm results presented in this report are the results obtained after using PCA. In machine learning, a classifier is an algorithm that automatically classifies or orders data into one or more of a set of "classes. In this study, classes are different brand of pizza.[3]

Section II introduced PCA as well as step-by-step instructions on how to apply PCA to data. In Section III, various machine learning classification techniques were presented, and the three classification algorithms that will be used in this project were thoroughly described. The dataset was explained in Section IV, and various plots were drawn to analyze the characteristics of the data. PCA was applied to the dataset in Section V, and the results were provided and interpreted. In Section VI, a model was fitted to both the original and transformed data, and various algorithms were adjusted and evaluated based on their ability to predict the data.

II. PRINCIPAL COMPONENT ANALYSIS

A. Description

Principal component analysis (PCA) is a multivariate technique that analyses a data table in which several inter-

correlated quantitative dependent variables describe observations. PCA and other feature reduction techniques help to reduce the dimensionality of large data sets by transforming a large group of variables into a smaller one that retains most of the original dataset's information. This technique helps to visualize the data set more easily and in a cost-effective way.

The main goals of (PCA) consist of:

I. Take the most important information from the data table and summarize it.

II. reduce the size of the data set by retaining only the important information.

III. Simplify the data set's description.

IV. Examine the configuration of the observations and variables.

To achieve these objectives, PCA computes new variables known as principal components, which have been gained as linear combinations of the original variables.[4]

B. *PCA algorithm*

The following steps should be taken to apply the PCA to a dataset.

Standardize the data: The main aim of this process is to standardize the first variables so that they contribute equally to the analysis. To standardize the data, the mean vector of each column using is needed to be calculated from Eq. (1) The centered data is obtained from Eq. (2) Each column of the centered data matrix (Y) has a mean of zero.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (1)$$

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

By subtracting the mean of each column from each item in the data matrix, the data is standardized. The final centered data matrix (Y) can be expressed. H depicts the centering matrix.

Covariance matrix computation: This step aims to determine the relationship between the variables. When variables are closely related, they can contain redundant information. The covariance matrix is used to identify these correlations. The covariance matrix p × p is calculated from Eq. (3)

$$S = \frac{1}{n-1} Y^T Y \qquad (3)$$

Eigenvector & Eigenvalues: The Eigen decomposition can be used to compute S's eigenvalues and eigenvectors. Eigenvectors represent each principal component's (PC) direction, whereas eigenvalues represent the variance captured by each PC. Eigen decomposition is obtained from Eq. (4)

$$S = A \wedge A^T \qquad (4)$$

Principal Component: Finally, the transformed data matrix (Z) of dimension n × p will be computed from the following equation.

$$Z = YA \qquad (5)$$

III. CLASSIFICATION ALGORITHMS BASED ON MACHINE LEARNING

We use classification to categorise new observations or to find the class for them. There are two types of machine learning algorithms: supervised and unsupervised. In general, supervised learning is more commonly used than unsupervised learning, in which we have an input variable called X and an output variable called Y, and we can find the mapping function using the formula Y=f(x). If we have a clear mapping function after finding new inputs or data, the results will be predictable.[5]

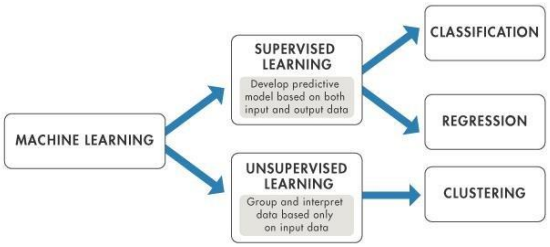

Fig 1: Supervise & Unsupervised learning technique

For instance, by using variables mentioned in the prostate cancer dataset, we can determine a man's prostate cancer status and check his medical condition.

A. *Naïve Bayes*

The Bayes' Theorem is a straightforward mathematical formula for calculating conditional probabilities. It plays an important role in subjectivist or Bayesian approaches to epistemology, statistics, and inductive logic. Subjectivists, who believe that rational belief is governed by probability laws, rely heavily on conditional probabilities in their theories of evidence and empirical learning models. [6]

Naive Bayes is a probabilistic machine learning algorithm that uses the Bayes Theorem to perform classification tasks. It is primarily used in text classification with a large training dataset. The Nave Bayes Classifier is a simple and effective Classification algorithm that aids in the development of fast machine learning models capable of making quick predictions. It is a probabilistic classifier, which means it predicts based on an object's probability. Spam filtration, sentiment

analysis, and article classification are some popular applications of the Nave Bayes Algorithm.[7]

Bayes' Theorem is central to these endeavors because it simplifies the calculation of conditional probabilities and clarifies important aspects of the subjectivist position. Using the Bayes theorem, we can calculate the likelihood of A occurring given that B has occurred. In this case, B represents the evidence and A represents the hypothesis. The predictors/features are assumed to be independent in this case. That is, the presence of one feature has no effect on the presence of another. As a result, it is referred to as naive. Another assumption made here is that all the predictors have an equal effect on the outcome. [7]

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B = events
$P(A|B)$ = probability of A given B is true
$P(B|A)$ = probability of B given A is true
$P(A), P(B)$ = the independent probabilities of A and B

Fig 2: Naive Bayes Theory Formula

B. *Quadratic Discriminant analisys*

Quadratic Discriminant Analysis (QDA) is a generative model. QDA assumes that each class follow a Gaussian distribution. The class-specific prior is simply the proportion of data points that belong to the class. The class-specific mean vector is the average of the input variables that belong to the class. [8]

For each class of observations, an individual covariance matrix is estimated in QDA. QDA works well when it is known that individual classes have distinguishable covariances. The covariance matrix y must be estimated separately for each class y, where y = 1, 2,…,K. The QDA function is defined as follows:

$$\delta_y(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log_{\pi_k} \quad (4)$$

where x represents the test instance, Σy represents the covariance matrix of class y, μk is the mean vector of class y and πk is the prior probability of class y. The classification rule for QDA is to find the class y which maximizes the quadratic discriminant function. Specifically, the classification rule for QDA can be described as below:

$$G(x) = arg\max_k \delta_k(x) \quad (5)$$

C. *Random Forest Classifier*

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. [9] Random forests are a combination of tree predictors in which each tree is dependent on the values of a random vector sampled independently and with the same distribution for all trees in the forest. [10]
The Random Forest Algorithm is made up of various decision trees, each with the same nodes but different data that results in different leaves. It combines the decisions of multiple decision trees to find an answer that is the average of all these decision trees. It is a supervised learning model; it uses labeled data to "learn" how to classify unlabeled data. The Random Forest Algorithm is used to solve both regression and classification problems, making it a versatile model that is widely used by engineers. [11] Some advantages and disadvantages of this classifier is mentioned below:

Advantages:

- It is a versatile model because it is used for regression and classification problems.
- Prevents overfitting of data.
- Fast to train with test data.

Disadvantages:

- Slow in creating predictions once model is made.
- Outliers and data gaps must be avoided.

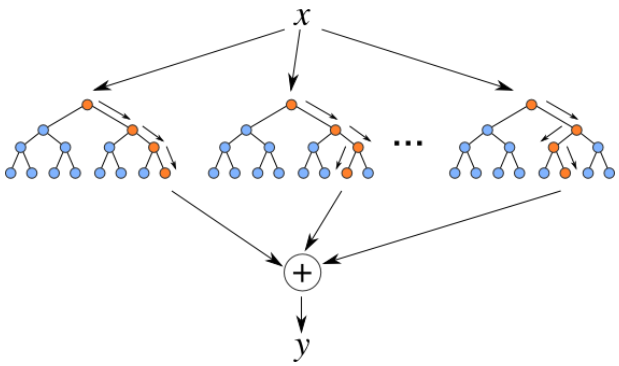

Fig 3: Graphical descriptive algorithm of random forest classifier

IV. DATA SET DESCRIPTION

The dataset which is used in this project is Pizza nutritional analysis component. By looking precisely at the Pizza gradient, there are several components in pizza which make the meal delicious or not. The pizza data set contains measurements for the elements that contribute to the taste of a pizza. These components are as below

mois – The amount of water in the sample per 100 grammes.

prot — The amount of protein in the sample per 100 grammes.

fat — The amount of fat in the sample per 100 grammes.

ash — The amount of ash in the sample per 100 grammes.

sodium — The amount of sodium in 100 grammes of sample.

carb — Carbohydrate content per 100 grammes of sample.

cal — The number of calories in 100 grammes of sample.

another variable in this dataset is Pizza's brands, which represent different providers who contributes to making delicious pizzas. In this study we consider them as classes. these labels used to be A, B, C, D, E, F, G, H, I, J which were changed by 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 respectively.[13]

The dataset proposes seven features for detecting best pizza's brand.

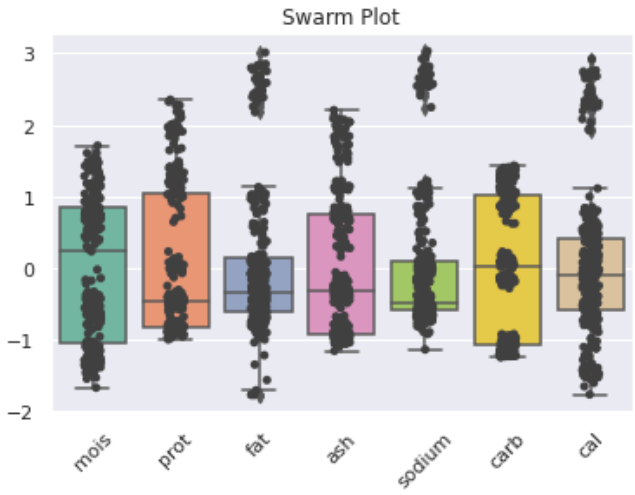

Fig 4: Box & Swarm plot

As it has been shown in Fig 4. The box plot, distribution of data, central data, and quartiles of different features of the Pizza's gradients dataset have been calculated. Carbs and Calories are counted as the features that are normally distributed. However, Sodium, Ash, Protein and Fat and have been skewed to the left and Mois is the only features that have been skewed to the right. It is worth mentioning that All of the components that contribute as a Pizza's gradient have outliner.

	Mois	Prot	Fat	Ash	Sodium	Carb	Cal
Mois	1	0.36	-0.17	0.27	-0.1	-0.59	-0.76
Prot	0.36	1	0.5	0.82	0.43	-0.85	0.07
Fat	-0.17	0.5	1	0.79	0.93	-0.64	0.76
Ash	0.27	0.82	0.79	1	0.81	-0.9	0.33
Sodium	-0.1	0.43	0.93	0.81	1	-0.62	0.67
Carb	-0.59	-0.85	-0.64	-0.9	-0.62	1	-0.023
Cal	-0.76	0.07	0.76	0.33	0.67	-0.023	1

Fig 5: Correlation Matrix

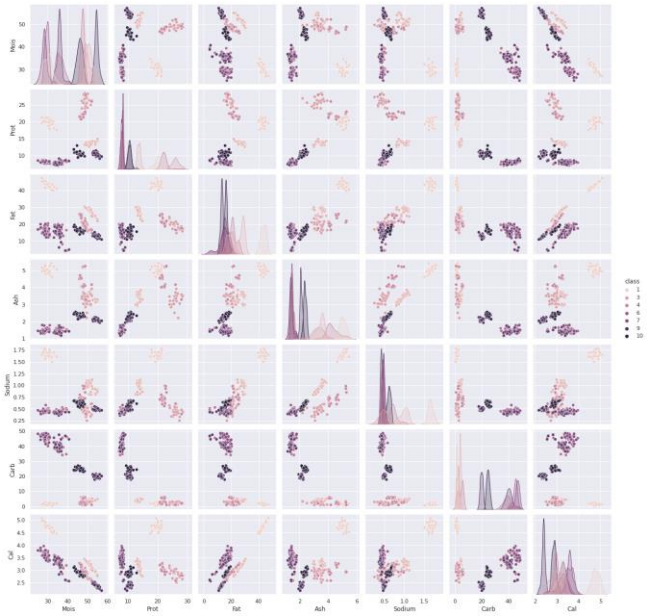

Fig 6: Pair plot

Fig. 5 and Fig 6 are both representing the correlation of the standardized dataset. As it has been shown in Fig. 5, different features of the Pizza's data set have either positive or negative correlations with each other. Sodium and Fat has been known as mostly correlated features, Carb and Ash have been known as the second most correlated features of the dataset. However, this amount contains a negative sign which mean that Carbs and Ash have negative correlation together. Carb is the only features of the dataset that have negative correlation with all the other attributes of the dataset. Cal/Carb and Cal/Prot are know as the least correlated attributes with amount of 0.023 and 0.07 respectively.

Sodium / Ash and Prot / Ash are counted as a third and forth mostly correlated features of pizza's dataset.

addition, by comparing the Pair plot and correlation matrix, it is obvious that, when two features of the data set have a positive correlation, they build a line with a positive angle. However, if the correlation between two features is negative, they will make a line with a negative slope.

The pair plot, which is shown in Fig 6, supported these findings visually. As it has been depicted in Fig 6, Cal and fat have build a line in positive angle which means that they have positive correlation together. On the other hand, Cal and Mois contribute to a line in negative slope which means that they are correlated but negatively.

Based on Fig 6. The attributes which formed scattered form do not have meaningful correlation. For instance,

V. PCA

In this step, the PCA algorithm has been applied to the Pizza dataset. The feature set of seven can be reduced to r features, where r < 7. The eigenvector matrix's columns represent a principal component. This r represents the dimension reduction amount. The related eigenvector is as below:

$$\boldsymbol{\lambda} = \begin{bmatrix} 4.18573434 \\ 2.29811778 \\ 0.415948838 \\ 0.0954925358 \\ 0.0277695834 \\ 0.000338738483 \\ 0.00000955061572 \end{bmatrix} \quad (6)$$

The variance of the data that each principal component can capture is defined as eigenvalues, and it is visually represented by a scree plot and an explained variance plot. The following equation yields the percentage of variance for the jth principal component, according to the course textbook.

$$lj = \frac{\lambda_j}{\sum_j^P \lambda_j} j = 1, \cdots p \quad (7)$$

where λ_j denotes the eigenvalue and the variance of the j-the PC.

Fig 7. And Fig 8. Depict scree plot and pareto plot respectively. Both of these figures show the variation which each principal component contribute to this dataset. Both figures show that the variance of the first is greater than that of the second. Two PCs account for 92.31% of the total variance. In the original dataset, the first PC accounts for 59.59% of the variance and the second PC accounts for 32.72% of the variance. According to the scree plot, the elbow is on the third PC. These two findings suggest that the dimension of the number of features can be reduced to three (r=3).

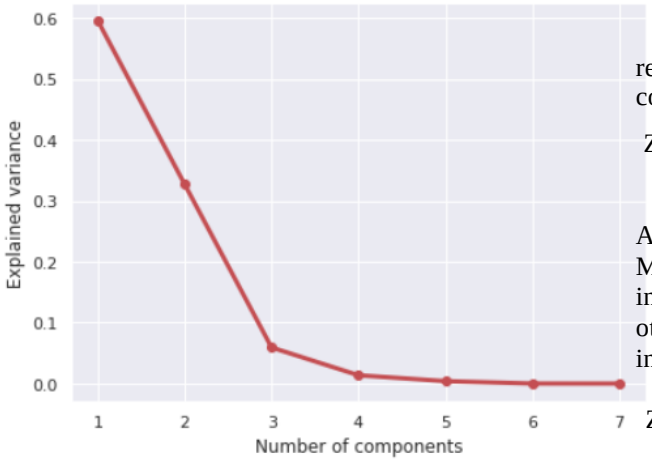

Fig 7: Scree Plot

As it is shown in Fig 8, the elbow is shaped on third principal component which was the aim of this project. The greatest the amount of variation in the first two PCs, a more favorable analysis can be obtained from PCA analysis.

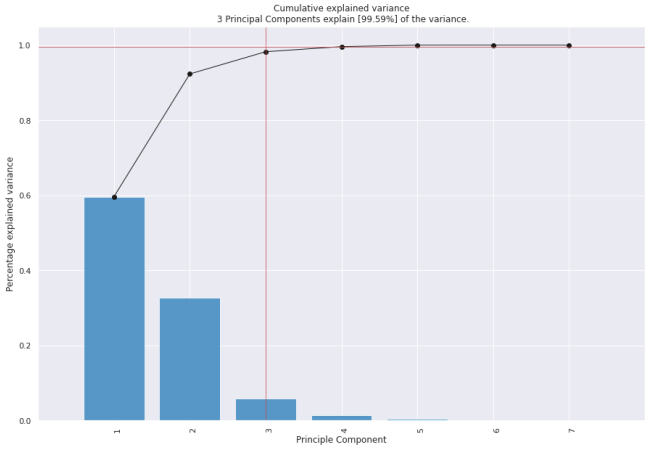

Figure 8 Pareto Plot

```
[ ]  A = out['loadings'].T
     print(A)
```

	PC1	PC2	PC3
Mois	0.064709	-0.628276	-0.421669
Prot	0.378761	-0.269707	0.746027
Fat	0.446666	0.234379	-0.199309
Ash	0.471890	-0.110990	0.056273
Sodium	0.435703	0.201662	-0.455169
Carb	-0.424914	0.320312	0.052237
Cal	0.244487	0.567458	0.113316

Fig 9:Eigenvectors Matrix A

As it is shown in Fig 9. Each column of this matrix represents a PC. The first, second and third principal component (*Z*i) are as below:

$$Z_1 = 0.064709X_1 + 0.378761X_2 + 0.446666X_3 + 0.471890X_4 + 0.435703X_5 - 0.424914X_6 + 0.244487X_7 \quad (8)$$

According to the Eg.8, the least variation contribution is by Mois. Fat, Ash and Sodium have the most variation influence on the first principal component. Carb, on the other hand is the only component which has negative influence on Z_1.

$$Z_2 = -0.628276X_1 - 0.269707X_2 + 0.234379X_3 - 0.110990X_4 + 0.20166X_5 + 0.320312X_6 + 0.567458X_7 \quad (9)$$

From Eq.9 it can be concluded that, more negative contribution has occurred in the second principal component (Mois, Prot, Ash). The least variation contribution is related to Sodium.

$$Z_3 = -0.421669X1 + 0.746027X2 - 0.199309X3 + 0.056273X4 - 0.455169X5 + 0.052237X6 + 0.113316X7 \quad (10)$$

It has been shown that the fourth and sixth features have a minimal impact on PC3, whereas the second feature has a positive effect on the third principal component.
The Fig 10 Which show The PC coefficient plot. It depicts the amount of contribution each feature makes on the first two PCs. The figure validates the previous PC calculation, and it is clear from the figure that, Prot, Ash, Mois and Fat have the most contribution to the first principal component. On the other hand, Sodium is the only pizza's gradient which has the great contribution on second principal component. Inversely, Mois, is the only component which is located in negative area (bottom right) which means that it has negative effect on the first PC.

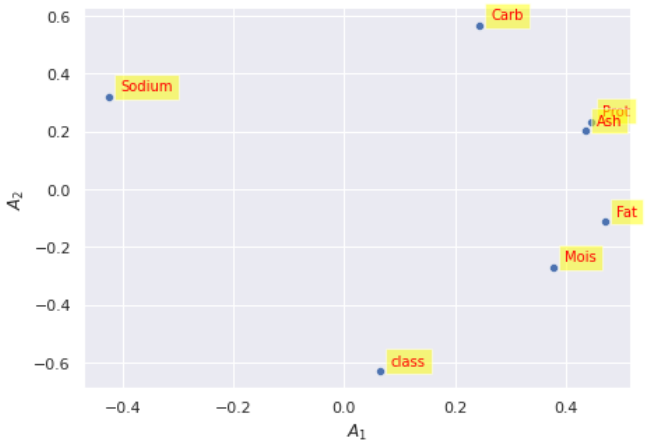

Fig 10. Coefficient plot

The Biplot which is depicted in Fig 11 is another way to show the contribution of different pizzas ingredient in two first principal component. As it shown in Fig 11. Each vector represents each pizza's component. Axis of the biplot show the PC1 and PC2. And the colorful dots represent each observation; angles of each vector with axis represent the amount of contribution each pizza's gradients have had on these two principal components. If the direction the vector is same as the axis, this component has positive contribution to the related PC. As it shown in Fig.11, Ash, Sodium and Fat have small angle with PC1 in the same direction which means that these three components have the most positive contribution to the first PC.Prot on the other hand is counted for the second biggest contribution to the first PC again in the positive direction. Mois, for instance has the lowest angle with second principal component in negative direction which match to our observation in the previous plot.

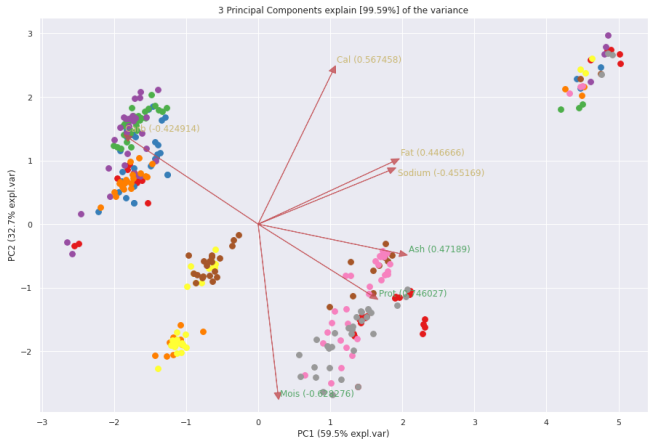

Figure 11 Biplot

VI. BEST CLASSIFICATION MODEL

In this part of the project, three different classifiers have been applied to Pizza gradients dataset and their results have been obtained. To investigate the effects of PCA on Pizza's dataset, The classification algorithms are applied to the original dataset. in addition to a PCA-applied dataset with three PCAs components. Python's PyCaret library is used for classification. The original dataset is divided into train and test sets with 70% and 30%, respectively.
There are many factors to evaluate and analyse the performance of each model, such as accuracy, precision, recall, AUC, ROC, confusion matrix, and so on. Precision measures the fraction of positive predictions, and recall measures the fraction of positives detected for each class.[14]
The F1-score combines a classifier's precision and recall into a single metric by taking their harmonic mean. The function of F_1 score can be obtained from equation below.

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \qquad (11)$$

A performance comparison can be created with PyCaret. Compare all available classification algorithms on the target dataset and select the best model with the highest accuracy.
By observing Figure 12. It can be concluded that the best model of classification before applying PCA was, Light Gradient Boosting Machine, Random Forest Classifier and Extra three classifier. However, by comparing the figure 12 table with Figure 13, the best model Classifier after reduction is converted to Naive Bayes, Extra tree Classifier and Quadratic discriminant analysis.
It is clear that by reducing the dimension of the original dataset, the performance (accuracy) of the classifiers has decreased. Light Gradient Boosting Machine (4%), Random Forest Classifier (2%). Furthermore, the best model for reduced dimension data (Nave Bayes) has 2% lower accuracy than the best model for original data (Light Gradient Boosting Machine).

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9044	0.9918	0.9100	0.9126	0.8956	0.8933	0.8978	0.181
rf	Random Forest Classifier	0.8991	0.9958	0.9050	0.9047	0.8854	0.8876	0.8939	0.321
et	Extra Trees Classifier	0.8936	0.9922	0.9000	0.9028	0.8837	0.8814	0.8863	0.161
nb	Naive Bayes	0.8827	0.9903	0.8900	0.8934	0.8755	0.8693	0.8733	0.031
lr	Logistic Regression	0.8822	0.9862	0.8900	0.8919	0.8748	0.8686	0.8723	0.911
lda	Linear Discriminant Analysis	0.8775	0.9879	0.8850	0.8750	0.8619	0.8634	0.8696	0.018
knn	K Neighbors Classifier	0.8725	0.9811	0.8800	0.8762	0.8591	0.8579	0.8638	0.034
gbc	Gradient Boosting Classifier	0.8512	0.9862	0.8600	0.8509	0.8386	0.8342	0.8416	0.765
dt	Decision Tree Classifier	0.8345	0.9079	0.8450	0.8229	0.8126	0.8156	0.8235	0.030
qda	Quadratic Discriminant Analysis	0.8137	0.9667	0.8217	0.7748	0.7729	0.7924	0.8071	0.041
ridge	Ridge Classifier	0.6386	0.0000	0.6200	0.5327	0.5606	0.5971	0.6245	0.018
svm	SVM - Linear Kernel	0.5740	0.0000	0.5650	0.4610	0.4815	0.5241	0.5658	0.029
ada	Ada Boost Classifier	0.4632	0.8137	0.4550	0.3665	0.3807	0.4001	0.4968	0.200
dummy	Dummy Classifier	0.1064	0.5000	0.1000	0.0113	0.0205	0.0000	0.0000	0.014

Figure 12: Best model before PCA

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
nb	Naive Bayes	0.8883	0.9887	0.8950	0.8970	0.8737	0.8755	0.8823	0.018
et	Extra Trees Classifier	0.8883	0.9924	0.8950	0.9075	0.8807	0.8755	0.8801	0.179
qda	Quadratic Discriminant Analysis	0.8877	0.9892	0.8967	0.9000	0.8805	0.8749	0.8791	0.019
rf	Random Forest Classifier	0.8775	0.9908	0.8850	0.8871	0.8680	0.8635	0.8684	0.217
lda	Linear Discriminant Analysis	0.8772	0.9879	0.8850	0.8897	0.8698	0.8631	0.8670	0.016
dt	Decision Tree Classifier	0.8725	0.9292	0.8750	0.8944	0.8661	0.8580	0.8629	0.017
knn	K Neighbors Classifier	0.8670	0.9859	0.8750	0.8729	0.8569	0.8516	0.8563	0.025
lightgbm	Light Gradient Boosting Machine	0.8617	0.9887	0.8700	0.8699	0.8480	0.8459	0.8521	0.109
gbc	Gradient Boosting Classifier	0.8567	0.9775	0.8650	0.8893	0.8508	0.8404	0.8463	0.951
lr	Logistic Regression	0.8196	0.9807	0.8300	0.8021	0.7987	0.7990	0.8060	0.048
svm	SVM - Linear Kernel	0.7284	0.0000	0.7167	0.6621	0.6679	0.6968	0.7171	0.022
ridge	Ridge Classifier	0.5219	0.0000	0.5100	0.3582	0.4046	0.4666	0.5029	0.013
ada	Ada Boost Classifier	0.4731	0.8380	0.4550	0.3712	0.3854	0.4074	0.4803	0.166
dummy	Dummy Classifier	0.1064	0.5000	0.1000	0.0113	0.0205	0.0000	0.0000	0.016

Figure 13: Best model after PCA

As a result, these three algorithms are used for evaluation throughout the rest of the experiment. Furthermore, these three algorithms are used to train, tune, and evaluate the original and transformed datasets. In both experiments, three classification algorithms which were applied to the original dataset and the other three classifications, which were applied to the reduced dataset are available on google colab. However, this study aims to explain the best three classifiers and their results on dataset after applying PCA.

Tuning hyperparameters is an important part of improving a model's performance. PyCaret hyperparameter tuning consists of three steps: create a model, tune it, and evaluate its performance. First, a classification model for each algorithm is created. Then, The tuned model() function is then used to optimize the model's hyperparameters. This function tunes the model using effective hyperparameters on a predefined search space and scores it using stratified K-fold cross validation. PyCaret applies 10 fold stratified K-fold validation to the three algorithms by default. Based on result, which was obtained from PCA, the best classifier algorithm for Pizza data set is Naive Bayes classifier. It worth mentioning that Naive Bayes is a straightforward but elegant classification algorithm which means that some common methods fail to improve the result of the Naive Bayes case. For instance, one of the first methods that come to mind is tuning the model's hyper-parameters. However, the Naive Bayes classifier has a minimal parameter set. Furthermore, depending on the implementation, sometimes the number of classes is the only parameter which we have no control over in practice**.** So, more than hyper-parameter tuning is needed to improve Naive Bayes classifier accuracy. Like all machine learning algorithms, we can boost the Naive Bayes classifier by applying simple techniques to the dataset, like data preprocessing and feature selection or removing the correlated features.[15] here are some solutions to improve Naïve Bayes classification performance:

1. Remove correlated features
2. Using logarithm probabilities
3. Eliminating zero observation problem
4. Handling continuous variable
5. Handle text data
6. Retrain model
7. Parallelize probability calculation
8. Using small dataset

Fig 14, Fig 15, Fig 16, demonstrate the model's decision boundaries on the transformed dataset. A decision boundary is a hyperplane that divides data points into distinct classes, and the algorithm switches between them. The figures' x-axis corresponds to the first PC, while the y-axis corresponds to the second PC. The square-shaped dots represent classes observations.

By comparing figures below together, it can conclude that Naïve bayes classifier has the best function among two other classifiers. Since the number of colored dots which is not related to its place boundary is fewer among two other figures.

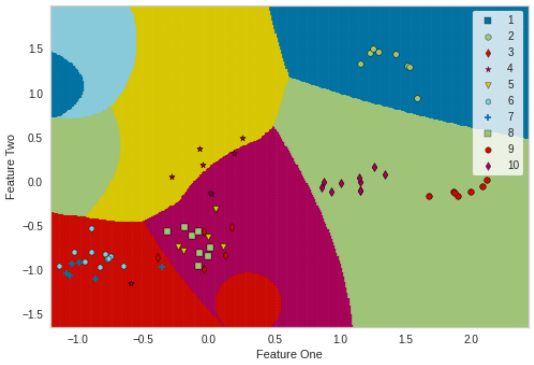

Figure 14: Decision boundary Naive bayes

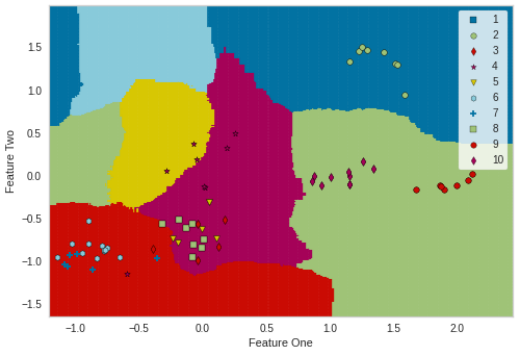

Figure 15:decision boundary-Extra trees

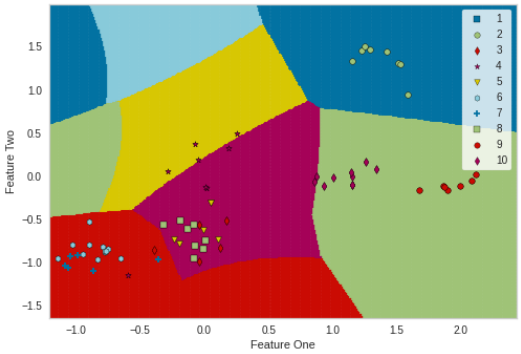

Figure 16: decision boundary Quadratic discriminant analysis

The confusion matrix is another tool for visualizing the classifier's performance by displaying the True Positives (TP), False Positives (FP), and True Negatives (TN) for each class. The confusion matrix is defined as the matrix that contains the proportion of predicted versus actual class instances. It depicts correct and incorrect predictions with count values and breaks them down by class. The confusion matrix tables for the three algorithms that were applied to the transformed dataset are shown in Fig. 17, Fig 18, Fig 19. The original dataset's confusion matrices can be found in the Google Colab notebook which is not the objective of this study. The horizontal axis represents class prediction, while the vertical axis represents true label.

Fig 17, correspond to the naïve bayes algorithm, which was applied to reduced data, it it has been shown in fig 17, there is one missed label in class 4 which was predicted to be in class 3 and there are 2 miss labeling have happened in class 4 which were predicted to be in class 8. The number of mislabeling in Naïve bayes classifier which were predicted wrongly are 12.

On the other hand, Fig 19 is correlation matrix of Quadratic discriminant analysis which were applied to data set after PCA. Fig 19 depicted that class number 6 has the greatest number of predictions among all other correlation matrix. As it is shown in fig 19, there are 11 observations exist in class 6 which were labeled as label 7. The second most mislabeling belong to class number 7 which this algorithm model predicted to be in class number 7. On the other hand, this model has some small incorrect labeling. For instance, class number 4 and 7 just have one incorrect label which were predicted to be in class 6 and 8 respectively.

The total number of incorrect labeling in Quadratic discriminant analysis which were applied to PCA data set was 20 which is huge number.

By comparing confusion matrix of Naïve Bayes algorithm and Quadratic discriminant analysis we could conclude that Naïve bayes classification still gained the best model of classification among the other classifiers for the mentioned dataset.

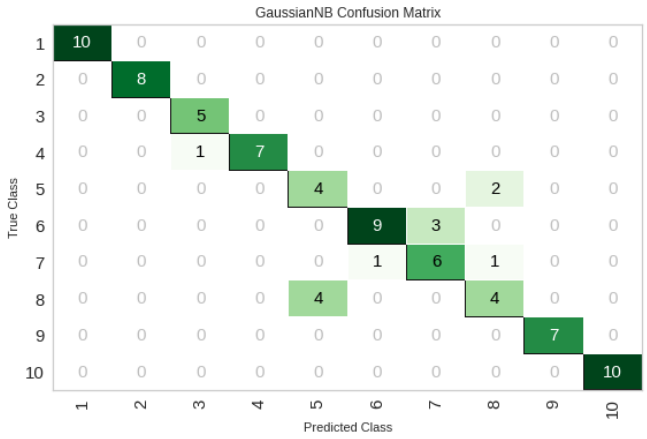

Fig 17: Confusion matrix-NB

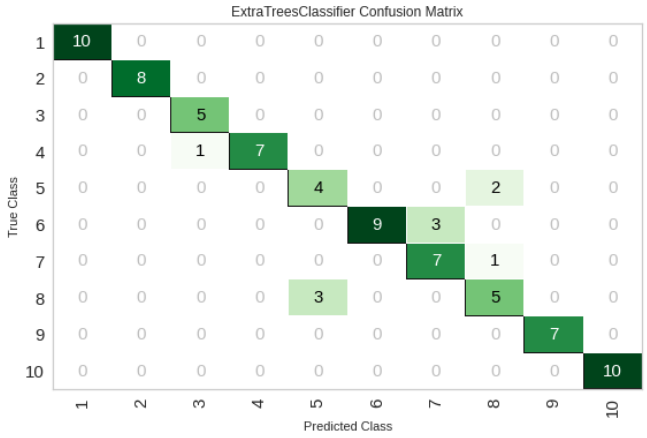

Fig 18:Confusion Matrix-Extra trees

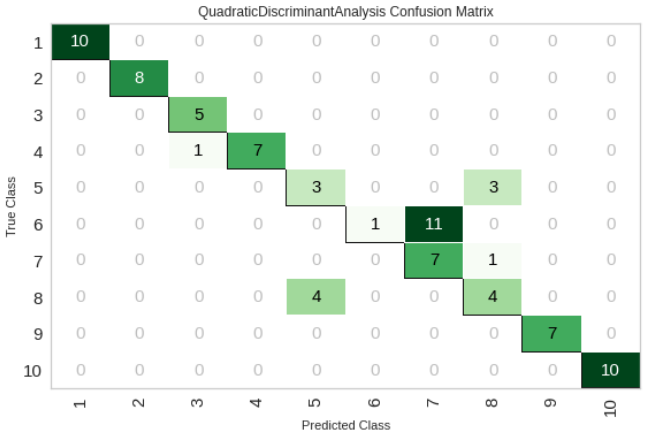

Fig 19:Confusion Matrix-Quadratic discriminant analysis

The final evaluation can capture the functionality of different classifiers in this study is ROC. The curve depicts two parameters: the True Positive Rate and the False Positive Rate. These are the primary components of constructing a confusion matrix. As a result, the ROC curve and confusion matrix are inextricably linked and can be viewed as different visual representations of the same measurement. Fig 20 depicted the mentioned curve for our best model “Naïve Bayes”. It plots the false positive rate (x-axis) versus the true positive rate (y-axis) for a variety of candidate threshold values ranging from 0.0 to 1.0.

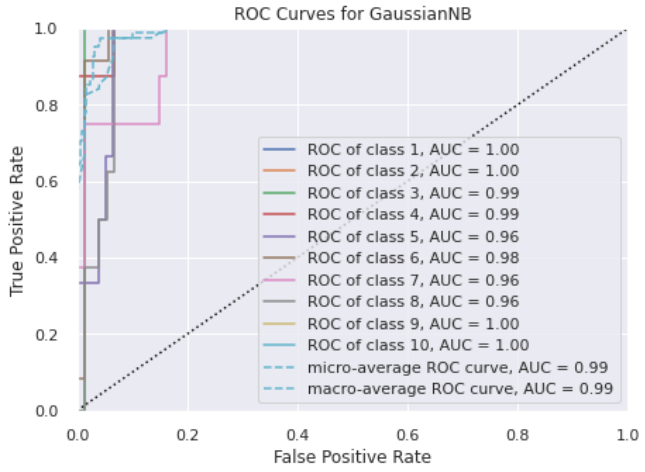

Figure 20: ROC-NB

As it shown in the figure 20, the ROC curve for the Gaussian NB model has the highest accuracy. It also displays a graph of the macro and micro average curves. The ROC curve and AUC values show that LR is the best at predicting both classes, with an accuracy of 99%. This result is consistent with the confusion matrix results. As a result, the three algorithms are capable of correctly classifying the benign and malignant classes.

VII. EXPLAINABLE AI WITH SHAPELY VALUE

In the context of ML, model interpretability is an important metric. There are various ways to improve a model's interpretability, and one of them is feature importance. The importance of each feature in the prediction process aids in estimating its contribution. As a result, we use the SHAP values to get an overview of the most important features on the PCs.[16] SHAP uses game theory to explain individual predictions based on optimal Shapley values. SHAP can, for example, explain the prediction of an instance x by calculating the contribution of each feature to the prediction.[17] by Using a game theory concept, each feature of a dataset acts as a player in a coalition. A player can be a single feature value, such as in tabular data, or a group of feature values. Shapley values describe how to distribute the prediction evenly across the feature set. Python's shap library is still in development, and it only supports tree-based models (e.g., decision tree, random forest, extra trees classifier). In this study, the best tree-based model after applying PCA was Extra trees classifier which were considered as a second-best model by examining in the previous section.

Fig 21 shows a summary plot of SHAP values the summary plot combines the importance of features with the effects of features.

A Shapley value for a PC and an instance is represented by each point on the summary plot. The PCs are represented on the y-axis, and the Shapley values are on the x-axis. More specifically, component 1 denotes the first PC, component 2 the second PC, and component 3 the third PC. We can see some jittered overlapping points in the y-axis direction, indicating the distribution of Shapley values per PC. All the PCs are ranked in order of importance.

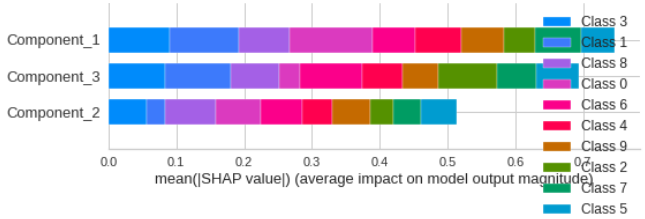

Figure 21:shaply value

This observation fortifies the Pareto and scree plots, which show that the first PC has the most feature variance. The red color represents a high PC value, while the blue color represents a low PC value. As it is obvious, first and second component contribute to the most variation of this dataset.

The 32nd observation is selected for this example.as it is shown in Fig 22. This plot depicts the features that each contribute to pushing the model output away from the baseline value. The base value is the value that would be predicted if no features for the current output were known.

In this case, the starting point is 0.1054. The bold 0.01 in the plot represents the model's score for this observation. Higher scores cause the model to predict 1, while lower scores cause it to predict 0. The red color indicates that the second PC is pushing the model for higher prediction, while the blue color on the first PC indicates that it is pushing for lower prediction.

Figure 22: Forced plot

The combined force plot of all PCs is displayed. This plot is a combination of all individual force plots that have been rotated 90 degrees and stacked horizontally. The y-axis in this plot corresponds to the x-axis in the individual force plot. Because the transformed test set contains 154 data points, the x-axis contains 154 observations.

This combined force plot depicts each PC's impact on the current prediction. Values in blue are thought to have a positive influence on the prediction, whereas values in red are thought to have a negative influence on the prediction.

VIII. CONNCLUSION

In order to reduce the dimension and classify the Pizza Micronutrients dataset, PCA and three machine learning (ML) algorithms (Naïve Bayes, Extratrees classifier, and Quadratic discriminant) were used. The Pizza dataset included ten classes (various pizza brands) and seven features (various component). Using the PCA algorithm, it was discovered that the first three eigenvalues contain nearly 99% of the data information; thus, the data dimension was reduced to three components. Several visualizations and

plots were created to explain and define the nature of the dataset (such as swarm plots and pair plots, among others) and the PCA results. (Explained variance plot, Biplot, and so on) Following that, 90% of the data was used. For the transformed data, the Gaussian Nave Bayes algorithm was also tested which was counted as the best model after applying PCA. All algorithms' decision boundaries were drawn, and their performance was evaluated and compared to other algorithms using F1-scores, Confusion Matrices, and AUC's. To summarize, all ML algorithms have adequate results for classifying data, creating models, and predicting unknown data. The results also revealed that applying PCA to the original dataset reduced the performance of the three algorithms.

[1] Lemki, S.A. and Ferris, D.A. "Production of sourdough frozen pizza and fresh foccacia using MIVAC spices and herbs" (2001).

[2] Sharma, J.L. and Caralli, S. *A Dictionary of Food and Nutrition*, CBS Publishing House Limited, New York, NY. (2006)

[3] Du, C. J., & Sun, D. W. (2005). Comparison of three methods for classification of pizza topping using different colour space transformations. *Journal of food engineering*, *68*(3), 277-287.

[4] Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433-459.

[5] Alpaydin, Ethem. Introduction to machine learning. MIT Press, 2020.

[6] Bayes' Theorem. https://plato.stanford.edu/entries/bayes-theorem/#1. Stanford Encyclopedia of Philosophy.(2003)

[7] Navie Bayes Classifier.https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c(2022).

[8] Quadraticdiscriminantanalysis.https://towardsdatascience.com/quadratic-discriminant-analysis (2022)

[9] Machine Learning - Logistic Regression, march 2019, https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm.

[10] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32..

[11] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[12] Schott, Madison. "Random Forest Algorithm for Machine Learning." Medium, 27 Feb. 2020, medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb..

[13] Principal Component Analysis - Pizza Dataset - dataset by sdhilip, https://data.world/sdhilip/pizza-datasets.(2022)

[14] Murphy, K. P. Machine learning: a probabilistic perspective. MIT press. (2012).

[15] How to improve naïve bayes classifier performance. https://www.baeldung.com/cs/naive-bayes-classification performance.(2022)

[16] Use Python to interpret & explain models (preview) - Azure Machine Learning. https://learn.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability-aml (2022)

[17] Molnar, Interpretable Machine Learning, 2nd ed., (2022).

[18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Advances in neural information processing systems, vol. 30,(2017).