Principal Component Analysis and Breast Cancer Diagnosis Using Machine Learning

Your Full Name and ID number
GitHub Link: github.com/YourAccount

Abstract—**Principal Component Analysis (PCA) is a dimensionality reduction method that is capable of reducing the dimensionality of large data sets. It can transform a large set of correlated variables into a smaller uncorrelated one while containing most of the information. In this report, PCA is applied on breast cancer diagnosis dataset to classify between benign and malignant tumors. Three different classification algorithms; i.e; logistic regression (LR), k-nearest neighbour (K-NN), and quadratic discriminant analysis (QDA) are applied on original dataset and transformed dataset (after applying PCA) to identify the tumor types. Next each model is tuned with ideal hyperparameters to obtain better performance metrics and the performance of each algorithm is measured using F1 score, confusion matrix and receiver operating characteristic (ROC) curves. The decision boundaries for each model is also shown in order to show the model fitting on the dataset. LR shows the superior performance over all other available machine learning models in PyCaret library. Finally, an experiment is carried out for the interpretation of the model using the explainable AI (artificial intelligence) Shapley values. Extra trees (ET) classifier model is used for this purpose. Overall, the algorithms successfully determines the two classes of breast cancer dataset and F1-score nearly 1 is achieved.**

Index Terms—**Principal component analysis, binary classification, logistic regression, K-nearest neighbour, quadratic discriminant analysis**

I. Introduction

Breast cancer is one of the major causes of death among women around the world compared to the other types of cancer. According to the report of world health organization (WHO), every year about one million women are newly diagnosed with breast cancer, and half of them would die, due to the late detection [1]. Breast Cancer is caused by a mutation in a single cell, which can eventually leads to reckless cell division and can be turned into a malignant tumor for cancer. Malignant tumors spread to the neighboring cells and can attack the other parts of the body. On the other hand, benign tumors cannot expand to other tissues and the expansion is only limited to itself [2]. Early diagnosis of breast cancer is one of the most crucial steps in the follow-up process. In this aspect, machine learning (ML) and data mining methods can help to reduce the number of false positive and false negative decisions in the diagnosis process [3].

In recent years, ML techniques are performing a major role in diagnosis and prognosis of breast cancer by applying classification techniques to identify people with breast cancer, distinguish benign from malignant tumors and to predict prognosis. Clinicians can use accurate classification which can help them prescribing the best treatment plan for their patients. In this report, at first Principle component analysis (PCA) is applied on the breast cancer dataset with the aim to dimentionality reduction. Afterwards, three popular classification algorithms, logistic regression (LR), K-nearest neighbor (K-NN) and Quadratic Discriminant analysis (QDA) is applied on the original dataset and PCA transformed dataset. The purpose is to determine whether a the tumor is a benign or malignant. Finally, with the quest of interpreting classification models explainable AI (artificial intelligence) Shapley values is used. It should be noted that the presented results from the classification algorithms in this report, all of them are represents the results obtained after applying PCA. More clearly, in this report the results are used from the transformed dataset. The classification results of original dataset can be found on the Google Colab notebook.

The rest of the report is organized as follows: Section II describes the PCA methodology, Section III gives an overview of the three classification algorithms, Section IV provides the beast cancer diagnosis dataset description, Section V discusses about PCA results, Section VI provides a extensive analysis of the classification results, Section VII provides a discussion on the explainable AI Shapley values. Finally, in section VIII, the conclusion is drawn.

II. Principal Component Analysis

Most of the real world datasets contains high dimensionality. Processing and storing of these types of datasets are highly expensive and sometimes impossible to visualize. Feature reduction techniques such as PCA helps to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information of the original dataset.

PCA is a technique for reducing the complexity of high-dimensional data while preserving trends and patterns [4]. It accomplishes this goal by condensing the data into fewer dimensions that serve as feature summaries.

A. PCA algorithm

PCA can be applied to a data matrix X with dimension $n \times p$ using the followings steps [5]:

1) **Standardization**: The main goal of this step is to standardize the initial variable so that they all contribute equally to the analysis. At first compute the mean vector

$\bar{x}$ of each column of the data set. The mean vector is a p dimensional vector can be expressed by:

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

The data is standardized by subtracting the mean of each column from each item in the data matrix.The final centered data matrix (Y) can be expressed as follows:

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

where H represents the centering matrix.

2) **Covariance matrix computation**: The aim of this step is to determine the relationship among the variables. Sometimes variables are so closely related that they contain redundant information. In order to identify these correlations, covariance matrix is computed. The $p \times p$ covariance matrix is computed as follows:

$$S = \frac{1}{n-1} Y^T Y. \tag{3}$$

3) **Eigen decomposition**: Using the eigen decomposition the eigenvalues and eigenvectors of S can be computed. Eigenvectors represent the direction of each principle component (PC) whereas eigenvalues represent the variance captured by each PC. Eigen decomposition can be computed using the following equation:

$$S = A \Lambda A^T, \tag{4}$$

where A means the $p \times p$ orthogonal matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues.

4) **Principal components:** It computes the transformed matrix Z that is size of $n \times p$. The rows of Z represents the observations and columns of Z represents the PCs. The number of PCs is equal to the dimension of the original data matrix. The equation of Z can be given by:

$$Z = YA. \tag{5}$$

III. Machine learning-based classification algorithms

A. Logistic Regression (LR)

LR tries to create the best fitting model in order to establish a relationship between the class and features [6]. LR labels the samples as 1 or 0. For example: if the value of the sample is 0.49 or below, it labels the sample as 0. On the other hand, if the value of the sample is 0.5 or above, the LR classifies the sample as 1. This decision is determined by a function, which is called logistic function. The logistic function can be expressed by

$$S(z) = \frac{1}{1+e^{-z}}, \tag{6}$$

here, the output of $S(z)$ is between 0 and 1 and z represents the input to the function. The main advantage of using LR is that it is very easy to implement and can handle the binary classification problem of breast cancer diagnosis such as if the tumor is benign or malignant. However, LR is prone to over fitting in high dimensional data sets. It is also sensitive to outliers that means if any sample is much deviated from the expected range then the classification may provide incorrect results. However, this problem can be overcome with proper feature scaling.

B. K- nearest neighbour (K-NN)

K-NN is a supervised classification algorithm which tries to train a model by classifying the samples according to the nearest training examples in the feature space [7]. K-NN is called lazy learning algorithm as it approximates the function only locally and defers all computations until classification [8]. As a lazy-learner, in the training phase, K-NN only stores the data and performs the computation during the classification process. K-NN is one the simplest classification algorithms, where an object is classified by a majority vote of its neighbors, with the object being assigned to the most common class amongst its k numbers of nearest neighbors.

K-NN uses all labeled training instances as a model of the target function. In order to classify a sample, at first K-NN selects the number of neighbours, calculates the Euclidean distance of k number of neighbours, takes the K nearest neighbors as per the calculated Euclidean distance. Finally, K-NN assigns the new sample to the class which has the maximum number of samples.

C. Quadratic Discriminant Analysis (QDA)

QDA allows for non-linear separation of data [9]. In QDA, an individual covariance matrix is estimated for every class of observations. QDA works well when there is a prior knowledge that individual classes exhibit distinguishable covariances. Mathematically, the covariance matrix Σ_y is required to be estimated separately for each class y, $y = 1, 2, ..., K$. The QDA function is defined as:

$$\delta_y(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) + \log_{\pi_k}, \tag{7}$$

where x represents the test instance, Σ_y represents the covariance matrix of class y, μ_k is the mean vector of class y and π_k 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:

$$G(x) = arg \max_k \delta_k(x). \tag{8}$$

Since the number of QDA parameters is quadratic, QDA should be used with care when the feature space is large.

IV. Data set description

The breast cancer diagnosis dataset that is used for this project is obtained from Kaggle. The dataset provides information for two types of tumor; "Benign" and "Malignant". Benign means the charcterstics of the tumor is not responsible for breast cancer and malignant means the characterstics of tumor

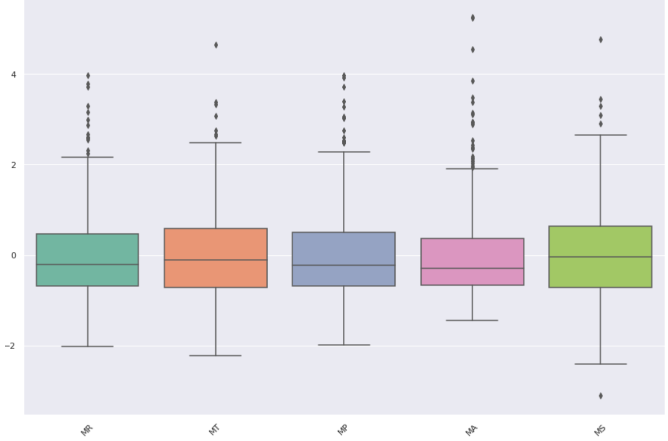


Fig. 1: Box plot

is responsible for breast cancer. The dataset projects 5 features for the detection of breast cancer. The features describe the tumor characteristics. The features are: Mean_radius (MR)", "Mean_texture (MT)", "Mean_perimeter (MP)", "Mean_area (MA)", "Mean_smoothness (MS)". It includes 569 entries for each of these attributes. Finally, it contains a column titled "Diagnosis" which is basically the label for the class to identify the tumor as benign or malignant.

Utilizing the box and whisker plots and their five number summaries on dataset, the distributions, central values and variability of the features were measured. Fig. 1 illustrates the box plot of the features of breast cancer dataset. It can be observed from Fig. 1 that most of the features follow approximately normal distribution. However, outliers exist in all features. All features except MS contain outliers on the left whereas MS contains outliers on both sides.

Fig. 2 shows the correlation matrix for the normalized features of the dataset. The features with large positive numbers are MR, MP, and MA. This evident implies that these three features are highly correlated. Other two features; i.e. MT and MS show less correlation with the other features in the dataset. The pairplot in Fig. 3 supports this observation. The highly correlated features contain higher number of cells with regularly increasing line. On the contrary, MT and MS displays less apparent correlation.

V. PCA RESULTS

PCA is applied on breast cancer dataset. Implementation of PCA can be done in two ways: (1) developing PCA from scratch using standard Python libraries such as numpy, (2) using popular and well documented PCA library. In the Google Colab notebook implementation of both methods is provided. Even though results obtained from both ways are similar, usage of PCA library brings more flexibility to the user and a lot can be done by writing only single line of code. In this report, the figures and plots are shown from the implementation using PCA library.

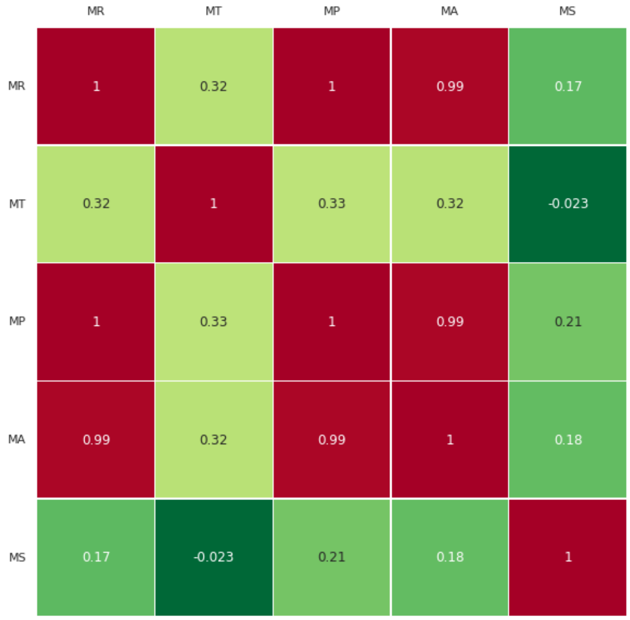


Fig. 2: Correlation matrix

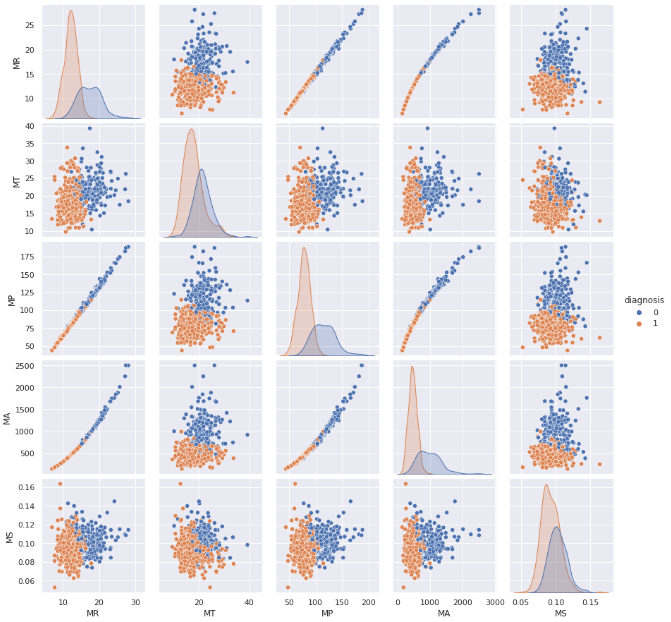


Fig. 3: Pair Plot

By applying the PCA steps, the feature set of 5 can be reduced to r numbers of features where $r < 5$. The original $n \times p$ dataset is reduced using eigenvector matrix A. Each column of the eigenvector matrix A is represented by a PC. Each PC captures an amount of data that determines the dimension (r). The obtained eigenvector matrix (A) for breast cancer dataset is as follows:

$$A = \begin{bmatrix} 0.553 & -0.002 & -0.174 & -0.398 & -0.709 \\ 0.246 & -0.533 & 0.808 & 0.005 & -0.007 \\ 0.555 & 0.024 & -0.144 & -0.418 & 0.703 \\ 0.551 & 0.003 & -0.170 & 0.816 & 0.014 \\ 0.138 & 0.845 & 0.514 & 0.010 & -0.0274 \end{bmatrix}$$

and the corresponding eigenvalues are:

$$\lambda = \begin{bmatrix} 3.177 \\ 1.022 \\ 0.789 \\ 0.016 \\ 0.001 \end{bmatrix}$$

Fig 4 and pareto plot in Fig. 5 demonstrate the scree plot and pareto plot of the PCs. The scree plot and pareto plot display the amount of variance explained by each principal component. The percentage of variance experienced by j-th PC can be evaluated using the following equation:

$$_j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100, j = 1, 2, ..., p, \tag{9}$$

where λ_j represents the eigenvalue and the amount of variance of the j-the PC.

Fig 4 and 5 plot the number of PCs vs the explained variance. It can be observed from both figures that the variance of first two PC's contribute to 83.8% of the amount of variance of the original dataset; i.e. first PC holds 60.4% of variance (l_1= 60.4%) and second PC holds 20.4% of variance (l_2 = 20.4%). The scree plot presents that the elbow is located on the second PC. These two observations imply that the dimension of the featureset can be reduce to two ($r = 2$).

The first principal component Z_1 is given by:

$$Z_1 = 0.553X_1 + 0.246X_2 + 0.555X_3 + 0.551X_4 + 0.138X_5 \tag{10}$$

It can be observed from the first PC that X_1 (MR), X_3 (MP), X_4 (MA) contributes to the most in first PC. However, none of the features have a negligible contribution to the first PC.

The second principal component Z_2 is given by:

$$Z_2 = -0.002X_1 - 0.533X_2 + 0.024X_3 + 0.003X_4 + 0.845X_5 \tag{11}$$

From the second PC Z_2 it can be seen that X_2 (MT) and X_5 (MS) have the highest contribution in the second PC. The contributions for X_1 (MR), X_3 (MP), X_4 (MA) are very small and hence negligible. Therefore, Z_2 can be rewritten as follows:

$$Z_2 = -0.533X_2 + 0.845X_5 \tag{12}$$

Fig. 6 represents the PC coefficient plot. It visually represents the amount of contribution each feature has on the first two PCs. The figure supports the previous calculation of PCs and it can be clearly seen from the figure that MR, MP and MA has the highest contributions in the first PC. On the other hand, MT and MS has the greatest contribution to the second PC. The MT (location=(0.25,-0.62)) has the only negative coefficient and is located bottom left side of the plot and far away from the clusters of features on the right side. MT is located on the positive side of the plot, but the location is far away from the other features.

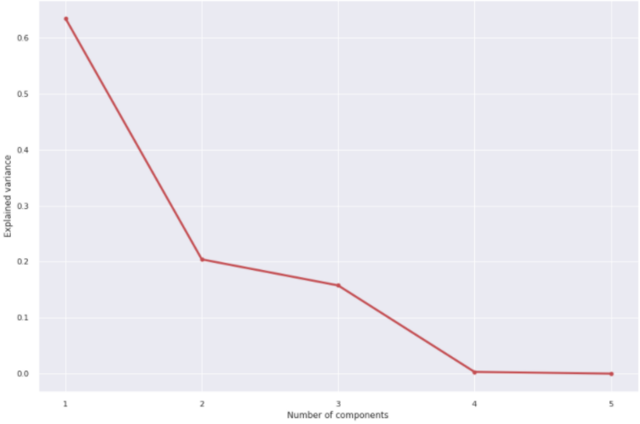


Fig. 4: Scree Plot

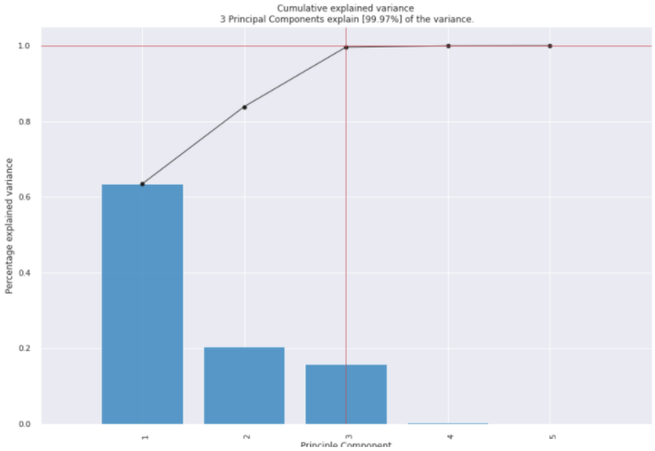


Fig. 5: Pareto Plot

The Biplot in Fig. 7 displays a different visual representation of the first two PCs. The axes of biplot represents the first two PCs. The rows of the eigenvector matrix is shown as a vector. Each of the observations in the dataset is drawn as a dot on the plot. The vectors for features namely, MR, MP and MA show very small angle with the first PC and very large angle with the second PC. This evident supports that the analysis of the PC coefficient plot of Fig. 6. It implies that these three features have a large contribution to the first PC and very small contribution to the second PC. On the other hand, the vectors for (MT) and (MS) shows the opposite phenomenon. They create a larger angle with the first PC and smaller angle with the second PC. It means that they are more related to the second PC rather than the first one. Furthermore, the vectors which follow the same direction are positively correlated with each other. For instance, MR, MP and MA are facing in the same direction.

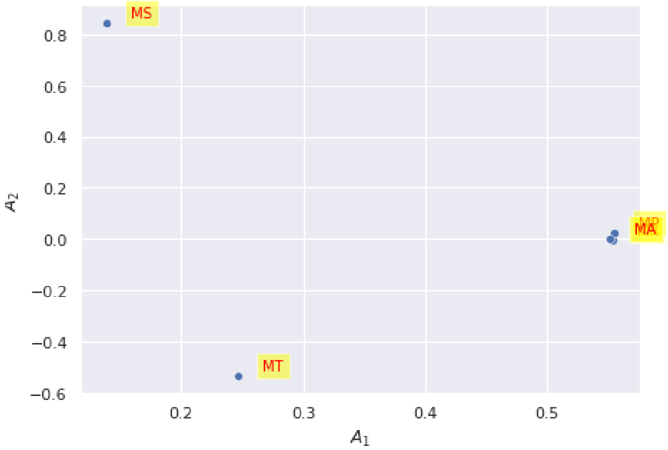


Fig. 6: PC coeficient plot

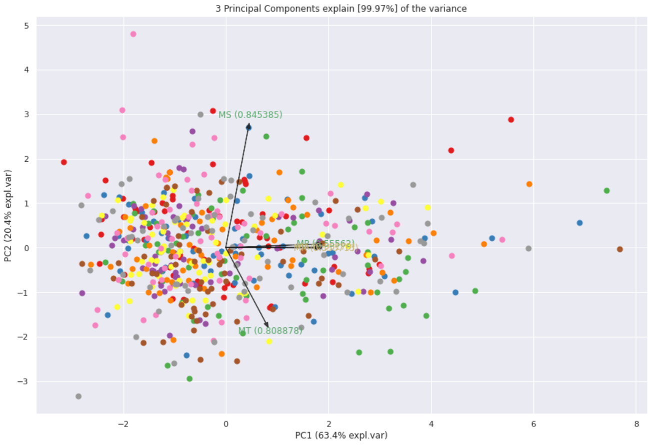


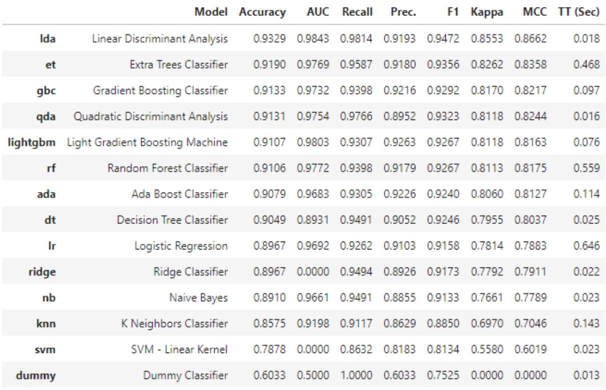
Fig. 7: Biplot

VI. Classification results

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

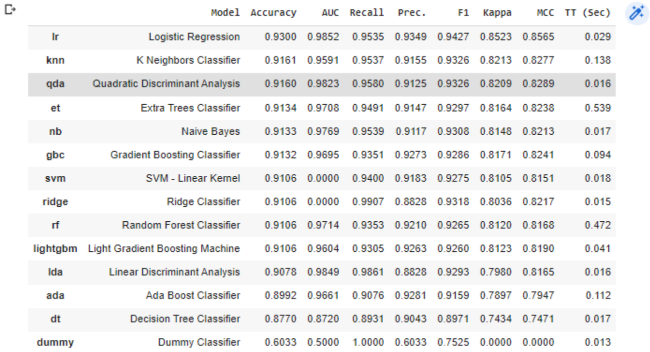
Using PyCaret, it is possible to create a performance comparison table among all available classification algorithms on the target dataset and find the best model with the highest accuracy. It can be observed from the Fig. 8 that, before applying PCA, the best three classification models with the highest accuracies on breast cancer dataset are Linear Discriminant Analysis (LDA), extra trees classifier (ET), and gradient boosting classifier (gbc).

On the other hand, Fig. 9 demonstrates the comparison among the classification models after applying PCA. Here, it can be seen that the best three models which give the highest



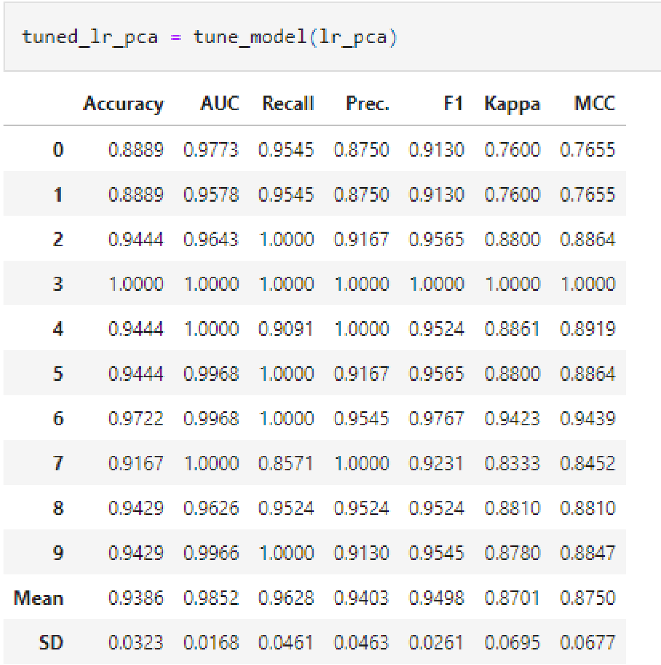
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lda	Linear Discriminant Analysis	0.9329	0.9843	0.9814	0.9193	0.9472	0.8553	0.8662	0.018
et	Extra Trees Classifier	0.9190	0.9769	0.9587	0.9180	0.9356	0.8262	0.8358	0.468
gbc	Gradient Boosting Classifier	0.9133	0.9732	0.9398	0.9216	0.9292	0.8170	0.8217	0.097
qda	Quadratic Discriminant Analysis	0.9131	0.9754	0.9766	0.8952	0.9323	0.8118	0.8244	0.016
lightgbm	Light Gradient Boosting Machine	0.9107	0.9803	0.9307	0.9263	0.9267	0.8118	0.8163	0.076
rf	Random Forest Classifier	0.9106	0.9772	0.9398	0.9179	0.9267	0.8113	0.8175	0.559
ada	Ada Boost Classifier	0.9079	0.9683	0.9305	0.9226	0.9240	0.8060	0.8127	0.114
dt	Decision Tree Classifier	0.9049	0.8931	0.9491	0.9052	0.9246	0.7955	0.8037	0.025
lr	Logistic Regression	0.8967	0.9692	0.9262	0.9103	0.9158	0.7814	0.7883	0.646
ridge	Ridge Classifier	0.8967	0.0000	0.9494	0.8926	0.9173	0.7792	0.7911	0.022
nb	Naive Bayes	0.8910	0.9661	0.9491	0.8855	0.9133	0.7661	0.7789	0.023
knn	K Neighbors Classifier	0.8575	0.9198	0.9117	0.8629	0.8850	0.6970	0.7046	0.143
svm	SVM - Linear Kernel	0.7878	0.0000	0.8632	0.8183	0.8134	0.5580	0.6019	0.023
dummy	Dummy Classifier	0.6033	0.5000	1.0000	0.6033	0.7525	0.0000	0.0000	0.013

Fig. 8: Comparison among classification models before applying PCA



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.9300	0.9852	0.9535	0.9349	0.9427	0.8523	0.8565	0.029
knn	K Neighbors Classifier	0.9161	0.9591	0.9537	0.9155	0.9326	0.8213	0.8277	0.138
qda	Quadratic Discriminant Analysis	0.9160	0.9823	0.9580	0.9125	0.9326	0.8209	0.8289	0.016
et	Extra Trees Classifier	0.9134	0.9708	0.9491	0.9147	0.9297	0.8164	0.8238	0.539
nb	Naive Bayes	0.9133	0.9769	0.9539	0.9117	0.9308	0.8148	0.8213	0.017
gbc	Gradient Boosting Classifier	0.9132	0.9695	0.9351	0.9273	0.9286	0.8171	0.8241	0.094
svm	SVM - Linear Kernel	0.9106	0.0000	0.9400	0.9183	0.9275	0.8105	0.8151	0.018
ridge	Ridge Classifier	0.9106	0.0000	0.9907	0.8828	0.9318	0.8036	0.8217	0.015
rf	Random Forest Classifier	0.9106	0.9714	0.9353	0.9210	0.9265	0.8120	0.8168	0.472
lightgbm	Light Gradient Boosting Machine	0.9106	0.9604	0.9305	0.9263	0.9260	0.8123	0.8190	0.041
lda	Linear Discriminant Analysis	0.9078	0.9849	0.9861	0.8828	0.9293	0.7980	0.8165	0.016
ada	Ada Boost Classifier	0.8992	0.9661	0.9076	0.9281	0.9159	0.7897	0.7947	0.112
dt	Decision Tree Classifier	0.8770	0.8720	0.8931	0.9043	0.8971	0.7434	0.7471	0.017
dummy	Dummy Classifier	0.6033	0.5000	1.0000	0.6033	0.7525	0.0000	0.0000	0.013

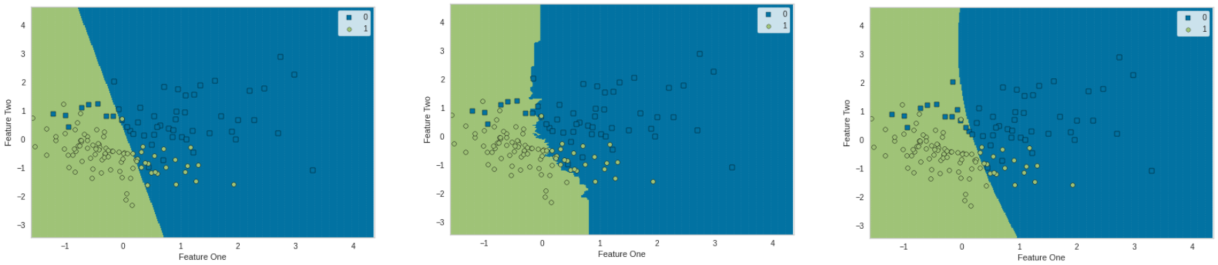
Fig. 9: Comparison among classification models after applying PCA



```
tuned_lr_pca = tune_model(lr_pca)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8889	0.9773	0.9545	0.8750	0.9130	0.7600	0.7655
1	0.8889	0.9578	0.9545	0.8750	0.9130	0.7600	0.7655
2	0.9444	0.9643	1.0000	0.9167	0.9565	0.8800	0.8864
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	0.9444	1.0000	0.9091	1.0000	0.9524	0.8861	0.8919
5	0.9444	0.9968	1.0000	0.9167	0.9565	0.8800	0.8864
6	0.9722	0.9968	1.0000	0.9545	0.9767	0.9423	0.9439
7	0.9167	1.0000	0.8571	1.0000	0.9231	0.8333	0.8452
8	0.9429	0.9626	0.9524	0.9524	0.9524	0.8810	0.8810
9	0.9429	0.9966	1.0000	0.9130	0.9545	0.8780	0.8847
Mean	0.9386	0.9852	0.9628	0.9403	0.9498	0.8701	0.8750
SD	0.0323	0.0168	0.0461	0.0463	0.0261	0.0695	0.0677

Fig. 10: LR metrics score after hyperparameter tuning



(a) Logistic Regression (LR).

(b) K-nearest Neighbour (KNN)

(c) Quadratic Discriminant Analysis (QDA)

Fig. 11: Decision Boundaries of the three algorithms applied on transformed dataset

accuracy on the transformed dataset are LR, K-NN, and QDA. Hence, these three algorithms are taken for the evaluation purposes during the rest of the experiment. The original and transformed dataset are trained, tuned and evaluated using these three algorithms. Both experiments (classification algorithms applied on original dataset and transformed dataset) can be found in Google Colab notebook. However, in this report, we only focus on the results obtained after applying PCA (transformed dataset).

In order to improve the performance of a model, hyperparameters tuning plays an important role. Hyperparameter tuning with PyCaret involves three steps; create model, tune it and evaluate its performance. At first, classification model per algorithm is produced. Then tune_model() function is used for tuning the model with ideal hyperparameters. This function automatically tunes the model with effective hyperparameters on a pre-defined search space and scores it using stratified K-fold cross validation. By default, PyCaret applies 10 fold stratified K-fold validation on the three algorithms. Moreover, LR model is tuned with L2 penalty, a regularization technique to prevent overfitting problem. For K-NN, the number of k-nearest members are tuned and the reg_param for regularization is tuned for QDA. It can be observed from the Fig. 10 that tuned LR model metrics are better than the base model metrics (before hyperparameter tuning).

Fig. 11 illustrates the decision boundaries formed by the model on the transformed dataset. A decision boundary is a hyperplane that separates data points into specific classes and the algorithm switches from one class to another. The x-axis of the figures corresponds to the first PC and y-axis corresponds to the second PC. The square shaped dots represent the observations for class 0 (benign) and class 1 is represented by the round shaped dots. The figure displays the differences among the decision boundaries that is formed by the algorithms. It is clearly visible from the figure that LR has the best decision boundary than K-NN and QDA. The decision boundary of LR can separate the data instances of both classes more accurately.

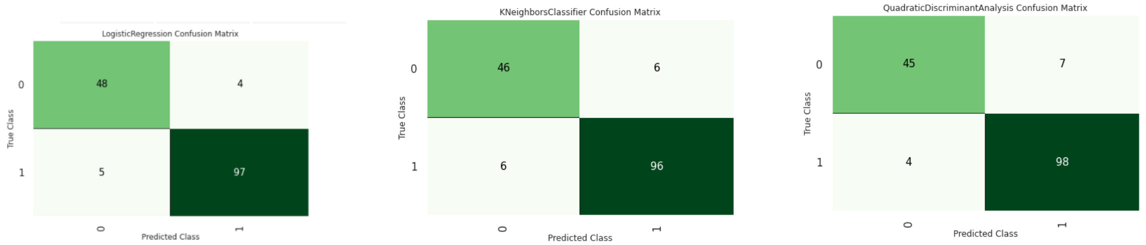
Since the breast cancer dataset is a binary classification problem, precision and recall can evaluate the performance of each class individually. Precision and recall are two measurements which together are used to evaluate the performance of classification. Precision is defined as the fraction of relevant instances among all retrieved instances, whereas recall, represents the fraction of retrieved instances among all relevant instances [10]. The obtained results from precision and recall is presented using the confusion matrices Fig. 12. The confusion matrix is defined as the matrix providing the mix of predicted vs. actual class instances [11]. It illustrates correct and incorrect predictions with count values and breaks down for each class. The Fig. 12 shows the confusion matrix tables for the three algorithms which were applied on transformed dataset. The confusion matrices for the original dataset can be found in the Google Colab notebook. In the figure, the horizontal axis represents the class prediction and vertical axis represents the true label. First of all, the comparison model on Fig. 9 shows that LR outperforms all other compared models including K-NN and QDA. This observation is also supported by the confusion matrices of Fig.12. LR misclassified the lowest numbers of instances. For example, LR misclassfied 4 instances from class 0 (Benign) as class 1 (malignant) and 5 instances of class 1 (malignant) are misclassified as class 0 (benign). On the other hand, K-NN misclassified 6 instances of class 0 (benign) and 6 instances of class 1 (malignant) whereas QDA misclassified 7 instances of class 0 (benign) and 4 instances of class 1 (malignant).

Another measurement of the performance evaluation is F1-score. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean [10]. It is a great metric to compare the results among the classifiers. For example: classifier A has a higher recall, and classifier B has higher precision. In this situation, the F1-score helps to determine the better classifier. The function of F1-score can be defined as below:

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}. \tag{13}$$

It can be observed from Fig. 8 and Fig. 9 that the F1-score of LR, K-NN, and QDA has improved significantly after applying PCA. This observation implies to the fact that dimension reduction weakens the dependencies among the features. F1-score of LR and K-NN enhanced more after the model is tuned with its ideal hyperparameters. All these observations are evidence of benefits of applying PCA and hyperparameters tuning.

As final analysis step, The receiver operating characterstic (ROC) curve for LR algorithm are shown in Fig. 13. A



(a) Logistic Regression (LR).

(b) K-nearest Neighbour (KNN)

(c) Quadratic Discriminant Analysis (QDA)

Fig. 12: Confusion matrices of the three classification algorithms applied on transformed dataset

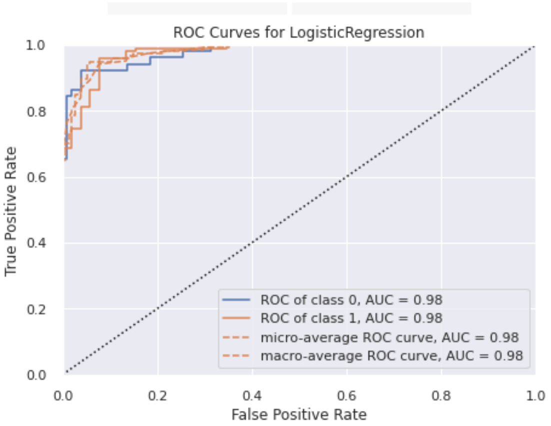


Fig. 13: ROC curve LR

ROC curve is a graph that demonstrates the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate, False Positive Rate. These parameters are the main components of building confusion matrix. Hence, ROC curve and confusion matrix are closely related and can be considered as different visual representation of same measurement. ROC curves for K-NN and QDA can be found in the Google Colab notebook. The ROC curve of LR in Fig. 13 reflects the results of confusion matrix. It plots the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0. It also shows the graph of macro and micro average curve. The ROC curve and AUC values shows that LR is the best at predicting both classes and can predict 98% accurately. This result mirrors the results obtained from the confusion matrix. Therefore, it is observed that the three algorithms are capable of successfully classifying the benign and malignant class.

VII. Explainable AI with Shapley Values

Model interpretability is an important metric in the context of ML. There are several ways of enhancing the explainability of a model and feature importance is one of them. Feature importance helps to estimate the contribution of each feature in the prediction process. Hence in order to get an overview of the most important features on the PCs, we use the SHAP values by importing the open source "shap" library of Python. SHAP (Shapley Additive Explanations) is a method that was first introduced by Lundberg and Lee [12] in 2016. SHAP explains individual predictions based on the optimal Shapley values using the concept of game theory. Specifically, SHAP can explain the prediction of an instance x by computing the contribution of each feature to the prediction [13]. Borrowing the concept of game theory, each feature of a dataset act as players in a coalition. A player can be an individual feature value, e.g. for tabular data. or a group of feature values. Shapley values describes how to adequately distribute the prediction among the featureset.

The shap library of Python is still in its development stage and it only supports tree-based models (i.e; decision tree, random forest, extra trees classifier) for binary classification problem. Since, breast cancer diagnosis dataset is a binary classification problem, shap analysis cannot be performed on LR, KNN, and QDA. Therefore, for the shap analysis, I have chosen the fourth best model of the transformed dataset that is "Extra trees classifier (ET)". Similar to other models, atfirst, an ET model is created and tuned with ideal hyperparameters. Then the tuned model is passed to the shap library for producing the interpretation plots. For our case, each PC acts as a player in the coalition.

Fig. 14 displays the summary plot of SHAP values. The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a PC and an instance. The y-axis represents the PCs and Shapley values are positioned on the x-axis. More specifically, component_1 represents the first PC, component_2 represents the second PC and component_3 represents the third PC. We can observe some jittered overlapping points in the direction of y-axis which indicates the distribution of the Shapley values per PC. All the PC's are ordered according to their importance. This observation supports the pareto plot and scree plot which indicates that the first PC holds the most feature variance. The red color indicates high PC value and blue color indicates low PC value. To interpret the summary plot, we can say that a low level of PC value has a high and positive impact on the breast cancer diagnosis. Similarly, a high level of PC value has a low and negative impact on the breast cancer diagnosis. More clearly, PCs are negatively correlated with the target variable.

Fig. 15 represents the force plot for a single observation. It should be noted that this plot can only be made for one

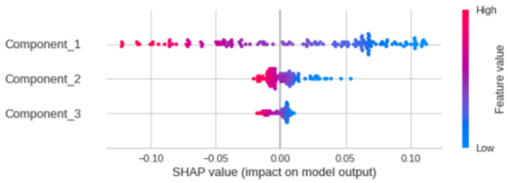


Fig. 14: Summary plot

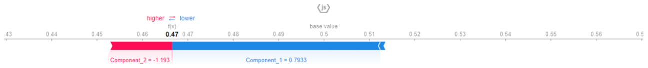


Fig. 15: Force plot for a single observation

observation. For this particular example, the 32th observation is chosen. This plot displays the features each contributing to push the model output from the base value. The base value means the value that would be predicted if no features are known for the current output [12]. In other words, it is the mean prediction of the test set. Here, the base value is 0.5. In the plot , the bold 0.47 is the model's score for this observation. Higher scores lead the model to predict 1 and lower scores lead the model to predict 0. The red color indicates that the second PC pushes the model for higher prediction and the blue color on the first PC indicates that it is pushing the prediction to be lower. This particular observation classified as class 0 (benign) because it is pushed mostly towards left. However, this plot is only a output for this observation. It does not describe the predicted output of the entire model.

Fig.16 displays the combined force plot of all PCs. This plot is a combination of all individual force plots with 90 degree rotation and are stacked horizontally. In this plot, y-axis is the x-axis of the individual force plot. There are 154 data points in the transformed test set, hence x-axis has 154 observations. This combined force plot shows the influence of each PC on the current prediction. Values in the blue colour considered to have a positive influence on the prediction whereas values in the red colour have a negative influence on the prediction.

VIII. Conclusion

In conclusion, PCA and three popular classification algorithms are applied on breast cancer dataset. The breast cancer dataset holds information on attributes of tumors to identify them as benign or malignant. At first, PCA is applied on the original dataset. The first two PC's apprehends 83% variance of the data. Hence, the featureset is reduced to 2 from 5. Extensive experiments are conducted on the first two PC's and different plots are generated to validate the obtained results from different perspectives. To move forward, three classification algorithms, LR, K-NN and QDA, are applied on the original dataset as well as transformed dataset with first three components. Each algorithm is tuned with the ideal hyperparameter settings and performance evaluation is conducted by comparing confusion matrices, ROC curves and F1-scores. It is observed that after hyperparameter tuning performance metrics score of each algorithm has improved significantly. The LDA, ET and GBC algorithms performed the best on the original dataset. Interestingly, after applying PCA, LR, K-NN and QDA performed the highest and showed the best performance metrics. Finally, in order to increase the interpretability of the model, several interpretation plots are produced using explainable AI shapley values. To summarize, all three algorithms can successfully determine the tumor types for breast cancer diagnosis.

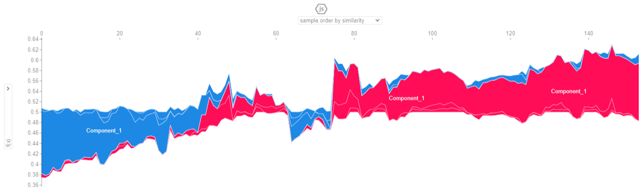


Fig. 16: Combined force plot

References

[1] L. A. Aaltonen, R. Salovaara, P. Kristo, F. Canzian, A. Hemminki, P. Peltomäki, R. B. Chadwick, H. Kääriäinen, M. Eskelinen, H. Järvinen *et al.*, "Incidence of hereditary nonpolyposis colorectal cancer and the feasibility of molecular screening for the disease," *New England Journal of Medicine*, vol. 338, no. 21, pp. 1481–1487, 1998.

[2] S. Chakraborty, "Bayesian binary kernel probit model for microarray based cancer classification and gene selection," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4198–4209, 2009.

[3] M. Karabatak and M. C. Ince, "An expert system for detection of breast cancer based on association rules and neural network," *Expert systems with Applications*, vol. 36, no. 2, pp. 3465–3469, 2009.

[4] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[5] A. B. Hamza, *Advanced Statistical Approaches to Quality*. Unpublished.

[6] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[7] W. Lee, S. J. Stolfo, and K. W. Mok, "Mining in a data-flow environment: Experience in network intrusion detection," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 114–124.

[8] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[9] S. Bose, A. Pal, R. SahaRay, and J. Nayak, "Generalized quadratic discriminant analysis," *Pattern Recognition*, vol. 48, no. 8, pp. 2676–2684, 2015.

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

[11] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass confusion matrix reduction method and its application on net promoter score classification problem," *Technologies*, vol. 9, no. 4, p. 81, 2021.

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

[13] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book