Unveiling Patterns: Applying Principal Component Analysis on Diabetes Prediction Measures Using Machine Learning

Name: Elham Kazemihojat
ID:4259744
GitHub Link: https://github.com/ElhamKazemihojat/Cousre-6220

Abstract: This report presents a comprehensive analysis of the application of Principal Component Analysis (PCA) to improve the prediction accuracy of diabetes using machine learning techniques. By applying PCA, we have effectively reduced the dimensionality of our dataset, which enhances the efficiency and performance of subsequent classification models. The main focus of this study is to evaluate the effectiveness of three machine learning classifiers—Naïve Bayes, QDA, and ADA Boost—both on the original and the PCA-transformed datasets. Our findings indicate that PCA significantly aids in simplifying the data, thereby facilitating faster and more accurate predictions. For a detailed execution of our analysis, including all codes and visualizations of plots, refer to our work documented in Google Colab. This approach has provided a structured methodology to not only improve model accuracy but also to understand the underlying patterns within the data, essential for early diabetes detection and intervention.

I. Introduction

Diabetes is a significant global health issue characterized by high blood sugar levels, which can lead to severe damage to organs and systems, particularly if left untreated. According to global statistics, the prevalence of diabetes and its mortality rates have been steadily increasing, particularly in lower-middle-income countries. Early detection and treatment are critical in preventing the most severe complications of the disease. Machine learning has emerged as a crucial tool in the early diagnosis and management of diabetes, providing new avenues for predicting the onset of the disease with high accuracy. This report begins with the application of Principal Component Analysis (PCA) on a diabetes dataset to reduce dimensionality and improve the efficiency of subsequent classification algorithms. We then explore the performance of three distinct classifiers(Naïve Bayes, QDA, and ADA Boost) on both the original and PCA-transformed datasets. The effectiveness of these models is assessed to determine the best approach for predicting diabetes presence in individuals.

II. Principal Components Analysis (PCA)

In this study, PCA is utilized to tackle the challenges posed by high-dimensional diabetes data, which can impede the performance of machine learning algorithms due to the curse of dimensionality. PCA assists in simplifying these datasets by reducing the number of dimensions without significant loss of information. The process of PCA includes several key steps:

1. **Standardization:** The data is normalized to ensure that each feature contributes equally to the analysis.
2. **Covariance Matrix Computation:** This matrix helps in understanding the correlation between different features.
3. **Eigenvalue and Eigenvector Calculation:** These are computed from the covariance matrix to identify the directions of maximum variance in the data.
4. **Selection of Principal Components:** Components are selected based on the amount of variance they capture from the data.
5. **Transformation:** The original data is transformed into a new set of dimensions defined by the selected principal components.

This reduction not only simplifies the data but also speeds up the learning process for subsequent machine-learning models. The characteristics of principal components ensure they are orthogonal, reducing redundancy and highlighting the most informative features of the dataset. This makes PCA a crucial step in preprocessing for effective machine learning applications in healthcare diagnostics.

III. PCA algorithm

Here's how we can apply PCA to a dataset:

1. Standardization

standardizing the data to ensure each feature contributes equally. This involves subtracting the mean and scaling to unit variance:

Compute the Mean Vector: Calculate the mean of each feature, ensuring that features with differing scales do not distort the PCA outcome. For diabetes data, features might include glucose concentration, blood pressure, skin thickness, etc.$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Center the Data: Subtract the mean vector from each data point to center the data around the origin.

$$Y = X - 1\bar{x}^T$$

Where X is the original data matrix, 1 is a column vector of ones, and $\bar{x}^T$ is the transpose of the mean vector.

2. Covariance Matrix Computation

After standardizing the data, compute the covariance matrix to identify correlations between features:

Here, YY is the centered data matrix, and SS is the covariance matrix that shows the covariance between each pair of features.

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

3. Eigen Decomposition

The eigenvalues and eigenvectors of the covariance matrix are computed next. Eigenvectors determine the directions of the new feature space, and eigenvalues determine their magnitude (in terms of the variance explained by the new feature space).

Compute Eigenvalues and Eigenvectors: Solve for λ and v where:

$$Sv = \lambda v$$

The eigenvectors v (PC) transform the original data to new dimensions that are uncorrelated, while eigenvalues λ represent the amount of variance captured by each principal component.

4. **Selection of Principal Components**

Choose the number of principal components to keep, typically those that capture significant variance:

Cumulative Variance: Calculate the proportion of variance explained cumulatively by the principal components and choose enough components to exceed a threshold like 85% variance.

Form the Feature Vector: Create a feature vector from the eigenvectors corresponding to the highest eigenvalues.

$$A = [v_1, v_2, \cdots, v_k]$$

Where v_k are the selected eigenvectors.

5. **Transformation into New Feature Space**

Transform the original data matrix X using the selected principal components in A to derive the reduced dataset:

$$Z = YA$$

Where Z is the matrix of transformed data with reduced dimensions, representing the original data in terms of the principal components retained.

IV. Machine Learning Based Classification Algorithm

1. **Naive Bayes (NB):**

Naive Bayes, based on Bayes' Theorem, assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Given dataset features such as age, BMI, insulin levels, and glucose concentrations, it treats these features independently when calculating the probability of diabetes occurrence.

It can quickly predict the class of a dataset. diabetes diagnosis can depend on various independent health indicators, Naive Bayes can be a good fit for making initial assessments in medical predictions.

$$P(C_k|x) = \frac{P(C_k)P(x|C_k)}{P(x)}$$

$P(C_k/x)$ is the probability of class C_k given predictors x, $P(C_k)$ is the prior probability of class Ck, $P(x/C_k)$ is the likelihood which is the probability of predictor given class, and $P(x)$ is the prior probability of predictor.

2. **Ada Boost Classifier (Ada):**

Ada Boost, an ensemble boosting classifier, starts with a weak classifier and iteratively improves it by adjusting the weights of incorrectly classified instances. It focuses on difficult cases in successive training rounds, making it robust for complex classification problems such as predicting diabetes, where combinations of attributes like glucose levels, body mass index, and genetic factors play a role.

Ada Boost can enhance the performance of simple models through its ensemble method, especially in cases where the interaction of several factors leads to diabetes, improving the accuracy over what individual weak classifiers could achieve alone.

$$F(x) = \sum_{t=1}^{T} \alpha_t \, h_t$$

$F(x)$ is the final strong classifier, $h_t(x)$ is a weak classifier's prediction, and α_t is the weights assigned to the weak classifiers.

3. **Quadratic Discriminant Analysis (QDA):**

QDA provides a statistical method for modeling and separating two or more classes of objects or events by a second-degree equation. This method models the probability distributions of predictors conditionally to each class of the dependent variable, which allows for more flexible class separation. In the diabetes dataset, where physiological measurements might follow different distributions in diabetic versus non-diabetic individuals, QDA can be very effective.

In QDA the features associated with each class (diabetic vs. non-diabetic) show different levels of variance and do not share a common covariance structure, allowing for a more nuanced understanding of the data.

$$\delta_k(x) = -\frac{1}{2} In\,|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) + In\,P(C_k)$$

Where $\delta_k(x)$ is the discriminant function for class k, x is the feature vector, μ_k and Σ_k are the mean and covariance of the class k, and $P(C_k)$ is the prior probability of class k.

V. Dataset Description:

The diabetes dataset, sourced from the UCI Machine Learning Repository, is utilized to explore the structure and relationships among predictor variables for diabetes in a given population. This dataset consists of 768 entries from female patients of Pima Indian heritage, structured into several key attributes crucial for diabetes analysis.

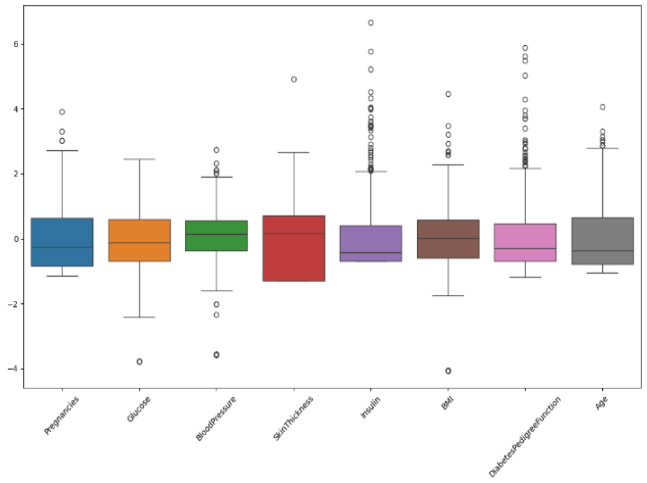
1. **Pregnancies:** Number of times pregnant.
2. **Glucose:** Plasma glucose concentration 2 hours post an oral glucose tolerance test.
3. **Blood Pressure:** Diastolic blood pressure (mm Hg).
4. **Skin Thickness:** Triceps skin fold thickness (mm).
5. **Insulin:** 2-hour serum insulin (mu U/ml).
6. **BMI:** Body mass index (weight in kg/(height in m)^2).
7. **Diabetes Pedigree Function:** A function scoring likelihood of diabetes based on family history.
8. **Age:** Age (years).
9. **Outcome:** classifies individuals as diabetic (1) or not (0)

Each of these attributes provides insights into clinical and physiological factors important for understanding diabetes risk. The dataset has been pre-processed for consistency, with imputed missing values ensuring comprehensive data integrity. The features are numeric or integer types, prepared for analysis.

For our PCA, the 'Outcome' attribute, has been excluded. This decision is based on the PCA objective to explore and understand the relationships among predictor variables independently of the diabetes status. Including the 'Outcome' might bias the PCA by forcing components to align with the outcome distribution rather than revealing natural groupings and variations among the predictors. By excluding it, we focus solely on the multidimensional structure of the predictors, allowing for unbiased identification of patterns that exist in the data without the influence of the outcome variable.

Utilizing this dataset, our study aims to apply various machine learning classification techniques to effectively predict diabetes onset, providing valuable insights that could assist healthcare professionals in early diagnosis and preventive care planning. The understanding derived from the dataset through exploratory data analysis will further guide the selection and tuning of appropriate predictive models, underpinning our methodological approach detailed in the subsequent sections of this report.

Figure 1-Boxplot



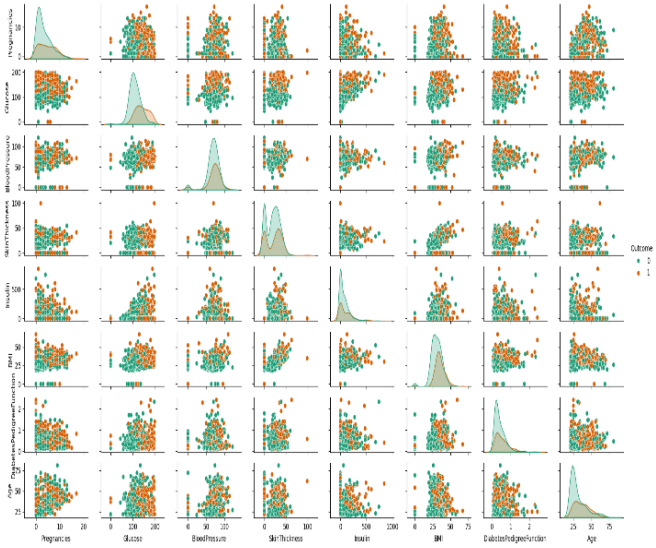
The boxplot (Figure 1) for the diabetes dataset indicates moderate variability across most physiological measures. Notably, 'Glucose' and 'BMI' show a wider spread of values, reflecting their significant role in diabetes prediction. Outliers in several categories suggest that there are individual cases with significantly different measurements, potentially indicative of unique medical profiles.

Figure 2-Heatmap

The heatmap (Figure 2) visualizes correlations among different variables related to diabetes. The strongest positive correlation is between 'Age' and 'Pregnancies', suggesting that older individuals tend to have had more pregnancies. 'SkinThickness' and 'Insulin' also show a moderately positive correlation, possibly reflecting physiological interactions. The majority of other variables exhibit weak to moderate correlations, indicating relatively independent relationships with each other within this dataset.

This pair plot (Figure 3) visualizes the relationships between different variables in the dataset, distinguished by the 'Outcome', respectively. Variables such as 'Glucose', 'BMI', and 'Insulin' show distinct distributions between the two outcomes, suggesting they may be significant in predicting diabetes. There are noticeable clusters and trends: for example, higher glucose levels are more prevalent in the diabetic outcome. Some variables show overlapping distributions, indicating a more complex relationship with the diabetes outcome.

Figure 3- Pairplot



VI. PCA RESULTS

After a thorough examination of the dataset, the PCA method was applied to distill the essence of the data. Utilizing the PCA function from the scikit-learn library, the analysis was streamlined, facilitating the transformation of the dataset into a new matrix of principal components ordered by their variance contribution.

The PCA approach enabled the identification of the PCs that capture the majority of information inherent in the dataset, providing a restructured form of data with principal components ranked by the magnitude of their information content. This transformation allowed for the truncation of the dataset, by discarding the components with minimal information, thus simplifying the feature space without substantially compromising the predictive integrity of the model. The analysis revealed that the leading components contained the most valuable insights for predicting diabetes, underscoring their importance in the subsequent modeling process.

The eigenvectors of our data are:

$$\begin{bmatrix} 0.128 & 0.393 & 0.36 & 0.44 & 0.435 & 0.452 & 0.271 & 0.198 \\ 0.594 & 0.174 & 0.184 & -0.332 & -0.251 & -0.101 & -0.122 & 0.621 \\ -0.013 & 0.468 & -0.535 & -0.238 & 0.337 & -0.362 & 0.433 & 0.075 \\ 0.081 & -0.404 & 0.056 & 0.038 & -0.35 & 0.054 & 0.834 & 0.071 \\ -0.476 & 0.466 & 0.328 & -0.488 & -0.347 & 0.253 & 0.120 & -0.109 \\ 0.194 & 0.094 & -0.634 & 0.01 & -0.271 & 0.685 & -0.086 & -0.033 \\ -0.589 & -0.06 & -0.192 & 0.282 & -0.132 & -0.035 & -0.086 & 0.712 \\ 0.118 & 0.45 & -0.011 & 0.566 & -0.549 & -0.342 & -0.008 & -0.212 \end{bmatrix}$$

and the eigenvalues are:

$$\begin{bmatrix} 2.097 \\ 1.733 \\ 1.031 \\ 0.877 \\ 0.763 \\ 0.684 \\ 0.42 \\ 0.405 \end{bmatrix}$$

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:

$$j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100, j = 1,2, \dots, p,$$

where λ_j represents the eigenvalue and the amount of variance of the *j-the* PC. These two Figures, plot the number of PCs vs the explained variance. It can be observed from both figures that the variance of first 5 PC's contribute to 81% of the amount of variance of the original dataset. The first 4 principal component are given by:

Z_1=0.128×Pregnancies+0.393×Glucose+0.360×BloodPressure+0.440×SkinThickness+0.435×Insulin+0.452×BMI+0.271×DiabetesPedigreeFunction+0.198×Age

Z_1 captures a general overview of metabolic indicators where Glucose, SkinThickness, Insulin, and BMI have high coefficients, suggesting this component might represent an overall metabolic syndrome component that includes major factors influencing diabetes.

Z_2=0.594×Pregnancies+0.174×Glucose+0.184×BloodPressure−0.332×SkinThickness−0.251×Insulin−0.101×BMI−0.122×DiabetesPedigreeFunction+0.621×Age

Z_2 significantly weights Age and Pregnancies, with a negative influence from SkinThickness and Insulin. This component might be capturing life-stage or age-related changes, particularly those related to older age and reproductive history.

Z_3=−0.013×Pregnancies+0.468×Glucose−0.535×BloodPressure−0.238×SkinThickness+0.337×Insulin−0.362×BMI+0.433×DiabetesPedigreeFunction+0.075×Age

Z_3 contrasts BloodPressure against Glucose and DiabetesPedigreeFunction. This component may represent specific pathological or physiological contrasts between cardiovascular and glycemic health indicators.

Z_4=0.081×Pregnancies−0.404×Glucose+0.056×BloodPressure+0.038×SkinThickness−0.350×Insulin+0.054×BMI+0.834×DiabetesPedigreeFunction+0.071×Age

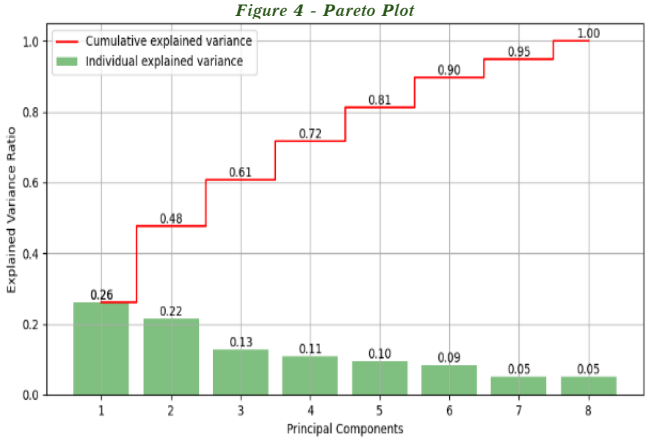
Z_4 is heavily influenced by DiabetesPedigreeFunction, suggesting that this component might capture genetic or familial risk factors associated with diabetes, contrasted particularly against Glucose and Insulin.

Z_5=−0.476×Pregnancies+0.466×Glucose+0.328×BloodPressure−0.488×SkinThickness−0.347×Insulin+0.253×BMI+0.120×DiabetesPedigreeFunction−0.109×Age

Z_5 highlights a nuanced interaction between metabolic and reproductive health, characterized by positive weights on Glucose and BloodPressure, and negative weights on Pregnancies, SkinThickness, and Insulin. This component may reflect unique physiological interplays that are particularly relevant in studies focusing on the impact of pregnancy and metabolic factors on diabetes.

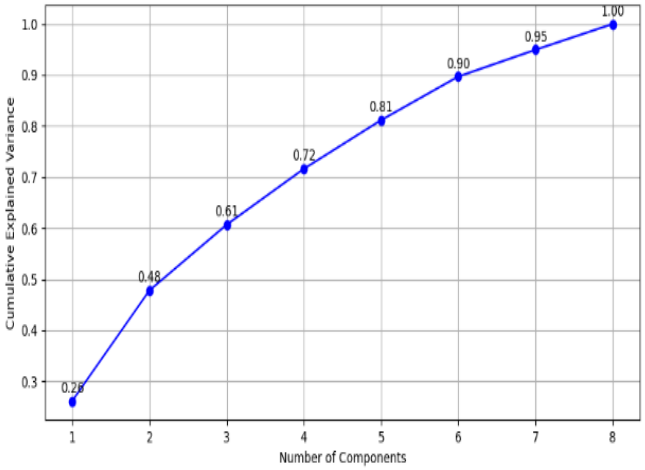
the Pareto plot is essential for visualizing the contribution of each principal component to the total variance. This plot will help identify the key components that capture the most significant information, allowing for a more focused analysis on the critical dimensions of the data. By showing the cumulative percentage of variance explained by the components, the Pareto plot guides the decision on how many components should be retained to achieve a balance between complexity and information retention.
Scree and pareto plots are pivotal in PCA for visualizing the importance of each component. They guide the selection of the number of components to retain, which enhances computational efficiency and clarifies data interpretation. The scree plot is particularly useful for identifying the 'elbow' where the variance starts to level off, while the pareto plot is beneficial for ensuring that a substantial amount of variance is retained, typically adhering to the 80/20 rule. Both plots are instrumental in simplifying complex datasets, leading to improved visualization and more manageable models

Figure 4 - Pareto Plot



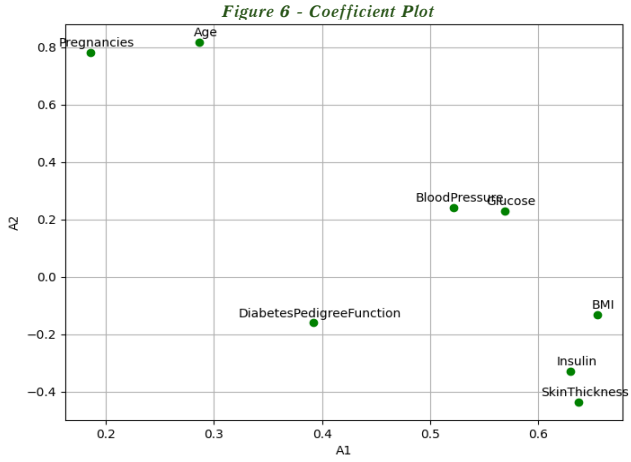
The Pareto plot (Figure 4) reveals that the cumulative explained variance reaches approximately 80% by the fifth component. This observation supports the application of the Pareto principle to reduce dimensionality while still capturing the bulk of the dataset's variability, thereby streamlining the data with minimal loss of explanatory power.

Figure 5 - Scree Plot



The scree plot(Figure 5) indicates that the first principal component captures the majority of the variance within the dataset, showcasing a pronounced reduction in the variance explained by subsequent components. The leveling off of the line after the fourth component suggests that the inclusion of additional components offers diminishing returns in terms of explained variance. This pattern aids in determining the optimal number of components to retain for effective data compression without significant loss of information.

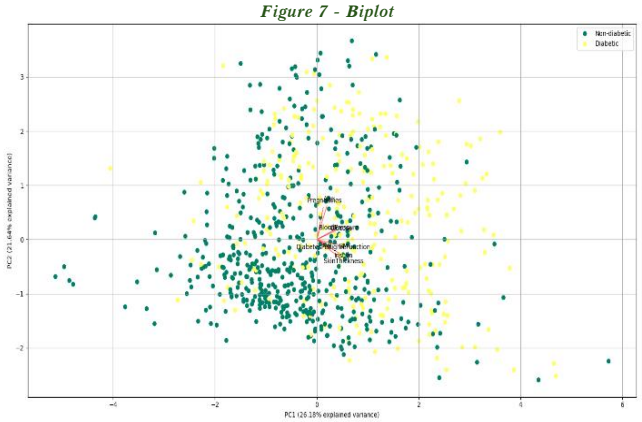
Figure 6 - Coefficient Plot



The Coefficient plot shows that 'Glucose', 'BMI', 'Blood Pressure', and 'Insulin' have the highest coefficients on the first

principal component, indicating they contribute most to the variance, which may reflect their importance in the underlying structure of the data.

Figure 7 - Biplot



In biplot(Figure 7), the axes represent the first two PCs of the dataset. PC1, depicted along the horizontal axis, appears to capture a significant proportion of the variance within the data, as indicated by the longer vectors for features like 'Glucose' and 'BMI', which project primarily in the direction of PC1. This suggests that these features are important contributors to the variance explained by PC1. PC2, on the vertical axis, seems less influential, with shorter vectors like 'BloodPressure' and 'Pregnancies' pointing towards it, implying a lesser but still notable contribution to the data's variability.

The distribution of dots, potentially representing individual records, is more dispersed along PC1 than PC2, which corroborates the higher explained variance of PC1.

The vectors representing each feature's loading on the PCs indicate their correlation: vectors pointing in the same direction suggest positive correlation, while those in opposite directions indicate negative correlation. For example, 'BloodPressure' and 'Pregnancies' appear to be positively correlated, contributing similarly to PC2. The clustering of the dots might reveal patterns or groupings inherent to the dataset that are relevant for further analysis.

Table 1-Comparison among classification models before applying

	Model		Accuracy	AUC	Recall	Precision	F1	Kappa	MCC	TT(Sec)
1	dummy	Dummy	0.6537	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0010
2	lr	Logistic Regression	0.7403	0.8002	0.6250	0.6250	0.6250	0.4263	0.4263	0.0200
3	dt	Decision Tree	0.7056	0.7043	0.7000	0.5600	0.6222	0.3859	0.3924	0.0075
4	lda	Linear Discriminant Analysis	0.7316	0.7973	0.6125	0.6125	0.6125	0.4072	0.4072	0.0102
5	qda	Quadratic Discriminant Analysis	0.7662	0.7984	0.6875	0.6548	0.6707	0.4897	0.4900	0.0051
6	et	Extra Trees Classifier	0.7143	0.7872	0.5750	0.5897	0.5823	0.3652	0.3653	0.0570
7	gbc	Gradient Boosting Classifier	0.7489	0.7954	0.6625	0.6310	0.6463	0.4519	0.4522	0.0972
8	rf	Random Forest Classifier	0.7532	0.7948	0.6500	0.6420	0.6460	0.4566	0.4566	0.0846
9	ada	Ada Boost Classifier	0.7446	0.7552	0.6250	0.6329	0.6289	0.4342	0.4342	0.0547
10	ridge	Ridge Classifier	0.7316	0.7007	0.6000	0.6154	0.6076	0.4037	0.4038	0.0061
11	nb	Naive Bayes	0.7446	0.7905	0.6625	0.6235	0.6424	0.4441	0.4445	0.0005
12	lightgbm	Light Gradient Boosting Machine	0.7229	0.7895	0.6500	0.5909	0.6190	0.4021	0.4033	0.1013
13	knn	K-Neighbors Classifier	0.6883	0.7162	0.5625	0.5488	0.5556	0.3156	0.3157	0.0045
14	svm	SVM-Linear Kernel	0.7446	0.8025	0.6250	0.6329	0.6289	0.4342	0.4342	6.3720

Table 2-Comparison among classification models after applying

		Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC	TT(Sec)
1	dummy	Dummy	0.6537	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	lr	Logistic Regression	0.7143	0.7611	0.4875	0.6094	0.5417	0.3378	0.3423	0.0020
3	dt	Decision Tree	0.6797	0.6580	0.5875	0.5341	0.5595	0.3087	0.3096	0.0009
4	lda	Linear Discriminant Analysis	0.7186	0.7608	0.4875	0.6190	0.5455	0.3458	0.3510	0.0010
5	qda	Quadratic Discriminant Analysis	0.7273	0.7559	0.5000	0.6349	0.5594	0.3660	0.3714	0.0000
6	et	Extra Trees Classifier	0.6840	0.6848	0.4875	0.5493	0.5166	0.2831	0.2842	0.0592
7	gbc	Gradient Boosting Classifier	0.7100	0.7354	0.4875	0.6000	0.5379	0.3299	0.3336	0.0747
8	rf	Random Forest Classifier	0.6753	0.6871	0.5125	0.5325	0.5223	0.2765	0.2766	0.0840
9	ada	Ada Boost Classifier	0.7359	0.7537	0.4625	0.6727	0.5481	0.3705	0.3835	0.0396
10	ridge	Ridge Classifier	0.7229	0.6676	0.4875	0.6290	0.5493	0.3539	0.3599	0.0000
11	nb	Naive Bayes	0.7359	0.7565	0.5000	0.6557	0.5674	0.3823	0.3896	0.0010
12	lightgbm	Light Gradient Boosting Machine	0.6797	0.6828	0.4625	0.5441	0.5000	0.2666	0.2685	0.0295
13	knn	K-Neighbors Classifier	0.6840	0.7093	0.4875	0.5493	0.5166	0.2831	0.2842	0.0020
14	svm	SVM-Linear Kernel	0.7186	0.7612	0.4750	0.6230	0.5390	0.3418	0.3483	3.8976

VII. Classification Results:

In this part, we have a comprehensive analysis of the outcomes of applying various classification algorithms. The dataset is partitioned into a training set and a test set with a split of 70% for training and 30% for testing.

This analysis includes evaluating the performance of each algorithm based on standard metrics such as accuracy, area under the receiver operating characteristic curve (AUC), recall, precision, F1 score, Kappa statistic, and Matthews correlation coefficient (MCC). We compares these metrics before and after the application of PCA. The time taken for the model to train or predict (TT) is also a crucial factor in assessing the models' efficiency.

Accuracy and AUC: These are general measures of model performance. Post-PCA, most models show a slight decrease in accuracy and AUC. However, the SVM-Linear Kernel's accuracy remains the same, and its AUC slightly increases.

Recall, Precision, and F1 Score: These are critical metrics for the balance between detecting the positive class and the relevance of the prediction. In the post-PCA application, a noticeable drop in recall is observed across all models, which might suggest a reduced ability to identify true positives. However, precision and F1 scores appear more mixed, with some models experiencing a slight increase.

Kappa and MCC: These statistics measure the agreement of the classification with the true outcomes, taking into account the possibility of the agreement occurring by chance. After applying PCA, a general trend of decrease in these metrics is observed, indicating potential overfitting or loss of predictive quality in the reduced dimensionality space.

Time Taken (TT): The efficiency of models in terms of computation time generally improves after applying PCA, as shown by reduced TT values. The SVM-Linear Kernel, however, shows a significant increase in TT, suggesting a more complex model fitting despite the reduced feature space.

As shown in Table 1, the top three classification models prior to applying PCA, based on their high accuracies are Quadratic Discriminant Analysis (QDA), Random Forest Classifier(RF) and Gradient Boosting Classifier(GBC).

Conversely, Table 2 reveals the classification models' performance after PCA was applied. The three models demonstrating the highest accuracy and F1 score on the PCA-transformed dataset are the Naive Bayes(NB), Ada Boost Classifier(Ada) and Quadratic Discriminant Analysis(QDA). So these models are selected for further evaluation in the ongoing experiment. The datasets, both original and transformed, are trained, tuned, and evaluated with these algorithms.

Decision boundary plots visually illustrate where a classification model divides the input space, clarifying how different classes are distinguished. These plots are critical for understanding the behavior of the model and its sensitivity to the input features.

The decision plots(Figure 8) for the three classifiers—Naive Bayes, Ada Boost, and Quadratic Discriminant Analysis—reveal distinct strategies in data separation. Naive Bayes assumes a simple linear relationship, which might underfit complex datasets. The Ada Boost's series of decision stumps exhibit a piecewise linear approach that captures more complexity but may overfit with excessive boundaries. Quadratic Discriminant Analysis, with its curved decision surfaces, suggests a more flexible model that adapts to data distribution, potentially offering a balance between bias and variance.

Figure 8 - decision plots

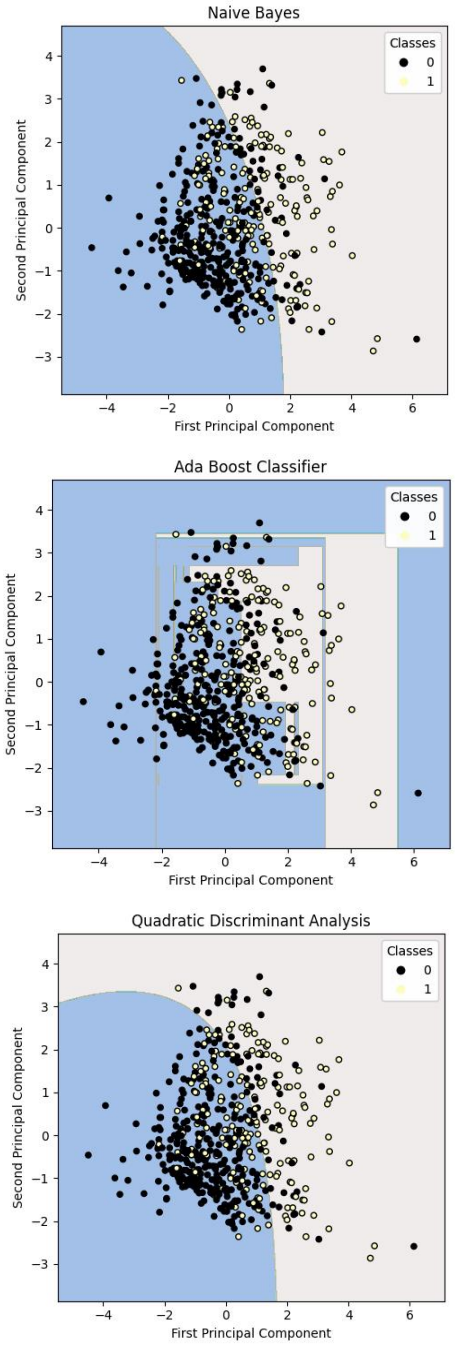
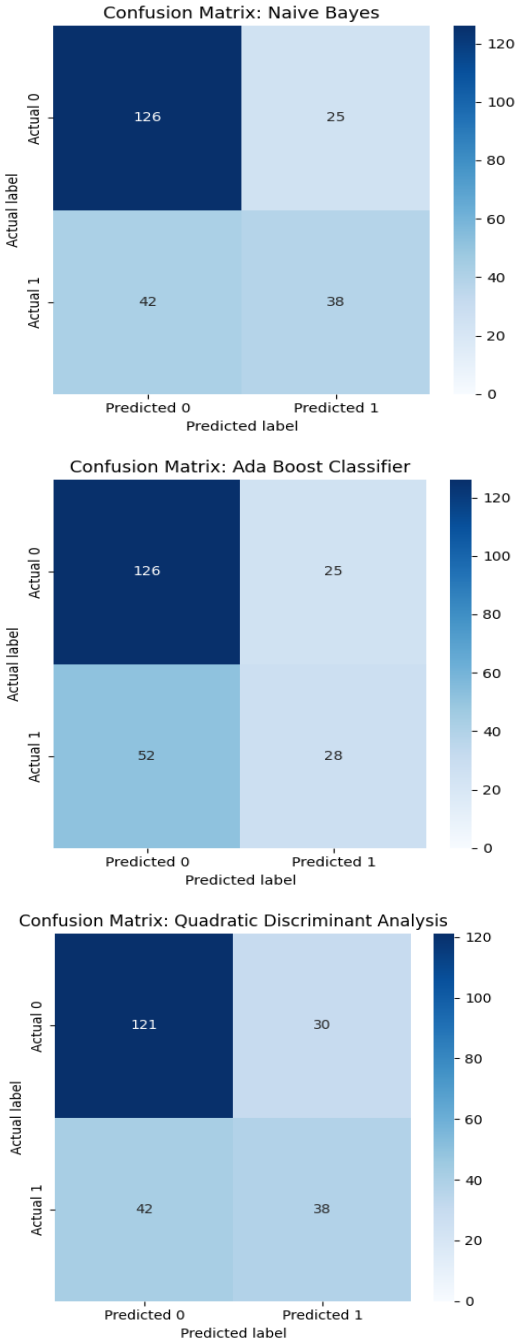


Figure 9 - Confusion Matrixes



So if the dataset requires capturing complex patterns and the features have significant interactions, Quadratic Discriminant Analysis seems to be the best choice. It offers a more sophisticated boundary that could potentially lead to higher classification accuracy. However, this conclusion is drawn purely from visual inspection and in next steps we decide it.
Confusion matrices provide a comprehensive snapshot of a classifier's performance, showing the exact number of correct and incorrect predictions. They enable the calculation of various performance metrics, which are essential for evaluating the trade-offs between sensitivity and specificity.

The confusion matrices of the three classifiers demonstrate varying levels of performance. Naive Bayes and Quadratic Discriminant Analysis both show a balanced number of true positives and false negatives but differ in false positives. Ada Boost shows a tendency to minimize false positives at the expense of an increased number of false negatives, indicating a conservative prediction strategy.

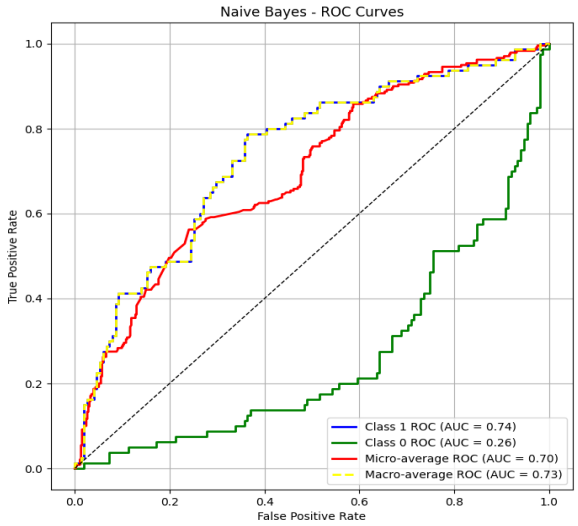
The matrices suggest that Quadratic Discriminant Analysis might offer a better balance in terms of the overall error rate, but the context of the problem should dictate which errors are more acceptable, influencing the choice of the classifier.
In our evaluation, we give special attention to the F1-score as a metric for performance assessment. The F1-score is particularly effective for comparative analysis of classifiers as it encapsulates both precision and recall into a singular figure through their harmonic mean. This metric is especially useful when there is a need to balance the trade-off between recall and precision, such as when one classifier demonstrates higher recall while another exhibits greater precision. The F1-score is mathematically represented as:

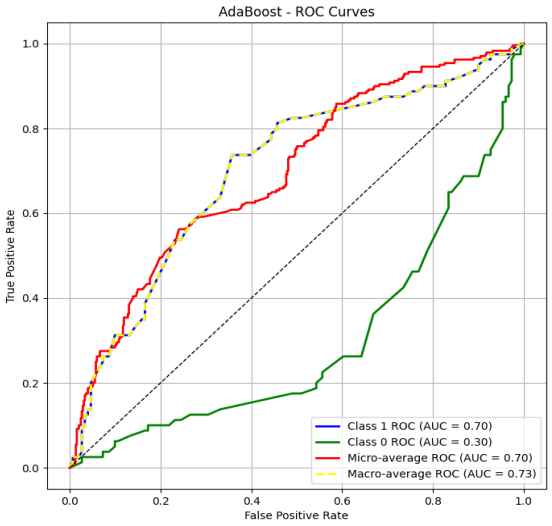
$$F1 - score = 2 \times \frac{presision \times recall}{presision \times recall}$$

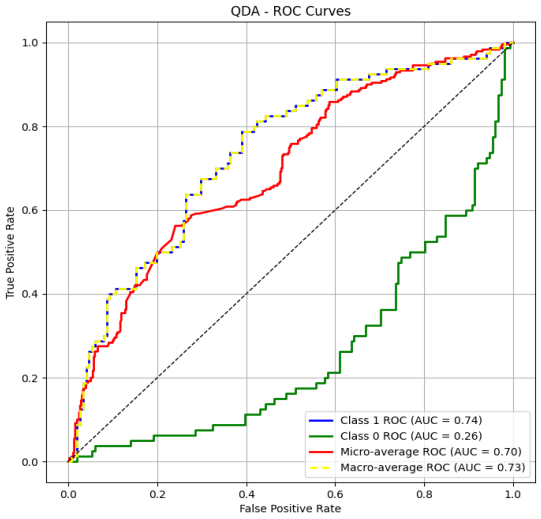
Quadratic Discriminant Analysis (QDA) maintains the highest F1 score both before and after applying PCA, indicating its robustness and effectiveness for dataset. Both Naive Bayes and Ada Boost Classifier see improvements in their F1 scores after

PCA is applied, which suggests that they benefit from the reduced complexity of the data. However, QDA outperforms them in terms of F1 score in both scenarios.

Figure 10 - ROC Curve's







The ROC curve is a powerful tool for visualizing and selecting classifiers based on their performance with respect to false positives and true positives at various thresholds. AUC values close to 1.0 suggest excellent model performance, while AUC values around 0.5 suggest no better performance than random guessing. The AUC values here suggest that all models provide reasonable predictions, but there is room for improvement.

The three classifiers show similar macro-average and micro-average AUC values around 0.70-0.73, indicating moderate classification effectiveness across both classes.

The plots for all three models show that each model has strengths and weaknesses, but none dramatically outperform the others in terms of AUC values.

Given the above, further investigation into model tuning, feature selection, or exploring different modeling techniques might be needed to improve the predictive performance for this dataset.

VIII. EXPLAINABLE AI WITH SHAPLEY VALUES:

Explainable AI (XAI) aims to demystify the decisions made by machine learning models, making them understandable to humans. A key technique in XAI is the use of Shapley values, based on cooperative game theory, which attributes the contribution of each feature in a model to its overall prediction. This method offers a fair and equitable way to assess feature importance by considering all possible combinations of features.

The SHAP library in Python provides tools to compute and visualize Shapley values, facilitating the interpretation of machine learning models across a variety of algorithms from tree-based models to deep neural networks. It offers several methods:

- ✓ Tree SHAP for tree-based models.
- ✓ Kernel SHAP for any model using a regression-based approach.
- ✓ Deep SHAP for deep learning models.

Benefits of Using SHAP:

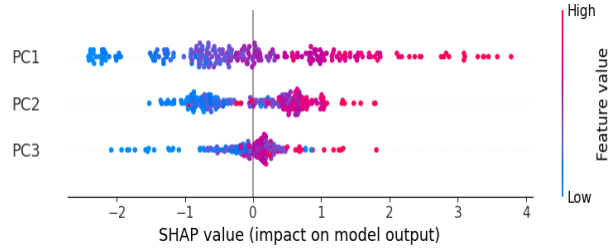
- ✓ Consistency and Accuracy: SHAP values accurately reflect the impact of features on predictions.
- ✓ Flexibility: It works with many types of models.
- ✓ Rich Interpretations: It offers detailed visualizations of feature impacts.

SHAP enhances transparency and trust in AI applications, critical in sectors like healthcare and finance where understanding model rationale is essential. By employing Shapley values, the SHAP library helps clarify the inner workings of complex models, supporting responsible AI practices.

Between our three classification methods,SHAP library can not be performed on None of mentioned classification models so according to Table 2 the best tee-based model for performing SHAP is Gradient Boosting Classifier, With an accuracy of 0.7100, this model is fully supported by SHAP's TreeExplainer.

A summary plot (Figure 11)visualizes SHAP values to show the impact of features on a machine learning model's output. It combines feature importance with effects, where the y-axis lists features by importance and the x-axis displays SHAP values indicating the impact magnitude. In our plot, three principal components (PC1, PC2, PC3) are analyzed. PC1 shows the broadest spread of SHAP values, suggesting significant influence on diabetes predictions with both positive and negative impacts. PC2 and PC3 exhibit lesser impacts, indicating they may capture less critical information. Colors from blue to red represent low to high feature values, showing how feature magnitudes affect model predictions. PC1's influence, demonstrated by its wide range of SHAP values, underscores its role in capturing essential predictive features for diabetes. This analysis helps in understanding which components are most vital for accurate predictions and guides further model refinement.

Figure 11- Summary Plot



A force plot (Figure 12) visualizes how individual features influence a model's prediction for a specific observation. It starts from a base value, typically the mean prediction if no features were known, and shows how each feature's value shifts the prediction away from this base.in this Force plot I chose observation index 23. Base Value ,Represents the average outcome prediction.Red Color, Indicates features pushing the prediction higher towards a positive outcome. Blue Color, Suggests features pushing the prediction lower, towards a negative outcome.Outcome Prediction, The balance of red and blue arrows shows the net effect of all features on the model's prediction. The predominance of either color dictates whether the prediction leans more towards a positive or negative outcome.

This plot provides detailed insights into the model's decision-making process for individual predictions, highlighting the contributing factors and their directional impacts. This is particularly useful for clinical settings, where understanding the basis for predictions can guide treatment decisions.

Figure 12- Force Plot

The Combined Force Plot (Figure 13) provides a comprehensive visualization of the cumulative impact of features on a model's predictions across a dataset.

Figure 13 - Combine Force Plot

In this plot, observations are aligned along the x-axis, and the y-axis shows normalized cumulative SHAP values, using blue to indicate positive influences and red for negative influences on the predictions. Blue areas suggest observations where features increase the likelihood of the outcome, whereas red areas indicate a decrease.

This visualization aids in understanding the variability of feature impacts, highlighting observations with particularly strong influences either positively or negatively. It serves as a crucial tool for identifying influential data points and understanding the overall behavior of the model, facilitating targeted analyses and model refinements.

IX. Conclusion:

The comprehensive analysis conducted in this report has highlighted the substantial benefits of PCA with advanced machine learning classifiers for enhancing the prediction accuracy of diabetes onset. The implementation of PCA successfully reduced the dimensionality of the dataset, which mitigated the curse of dimensionality and thus improved the operational efficiency of subsequent machine learning models. Our empirical findings indicate that PCA not only facilitates a more streamlined data processing workflow but also significantly enhances the predictive capabilities of the classifiers used.

Among the classifiers evaluated—Naïve Bayes, Quadratic Discriminant Analysis (QDA), and ADA Boost—QDA consistently demonstrated superior performance across various metrics, including accuracy, precision, and F1 score, particularly when applied to the PCA-transformed data. This suggests that QDA is particularly adept at handling the transformed feature space where principal components encapsulate the most informative variance of the data. The robustness of QDA in this context may be attributed to its ability to model the differentiated variances and covariances within the data, capturing nuanced patterns that are crucial for accurate diabetes prediction.

Furthermore, the study revealed that while PCA generally improves classifier performance, the extent of improvement varies among different classifiers. For instance, Naïve Bayes and ADA Boost showed varied results in terms of sensitivity and specificity, indicating that the choice of classifier should consider the specific characteristics of the data and the clinical objectives. Additionally, the analysis underscores the importance of considering trade-offs between model complexity and interpretability, particularly in a healthcare setting where decision-making relies heavily on understanding model predictions.

In conclusion, this research not only demonstrates the effectiveness of PCA in enhancing the predictive accuracy of machine learning models for diabetes but also provides a detailed comparative analysis of classifiers, offering insights into their respective strengths and limitations in handling PCA-transformed data. The findings encourage continued exploration into the integration of dimensionality reduction techniques and advanced classifiers to refine predictive models further, aiming for optimal performance in clinical applications.

References:

- Advanced Statistical Approaches to Quality Textbook – Concordia University
- UCI Machine Learning Repository. (n.d.). Pima Indians Diabetes Database. Retrieved from https://archive.ics.uci.edu/ml/datasets/diabetes
- Scikit-Learn Developers. (2021). Decomposing signals in components (matrix factorization problems). Retrieved from https://scikit-learn.org/stable/modules/decomposition.html#pca
- Jolliffe, I. T. (2002). Principal Component Analysis. Springer.
- Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal, 15, 104-116.
  World Health Organization. (2020). Diabetes. Retrieved from https://www.who.int/news-room/fact-sheets/detail/diabetes
- Centers for Disease Control and Prevention. (2021). National Diabetes Statistics Report. Retrieved from https://www.cdc.gov/diabetes/data/statistics-report/index.html