INSE 6220 project

Kasra Farrokhi (40168728)

***Abstract*—Diabetes is a chronic condition that occurs when the body is not able to use sugar as energy. In order to use glucose (sugar) for energy, our bodies require a hormone called insulin. Diabetes occurs when the body is not able to use and store sugar for energy because the pancreas because the pancreas: does not make insulin, makes too little insulin, and cannot respond to the insulin that is made. In this paper, according to the data we have, we compare three classification methods to see if we can find any correlation between diabetes in people according to their symptoms or not. At first, we use PCA which is a linear dimensionality reduction technique. Then, we use Decision Tree Classifier, Gaussian Naive Bayes classifier, and K-Nearest-Neighbor (KNN) classifier.**

I. Introduction

Diabetes is a disease in which your body either can't produce insulin or can't properly use the insulin it produces. Insulin is a hormone produced by your pancreas. Insulin's role is to regulate the amount of glucose (sugar) in the blood. There are multiple symptoms for diagnosing diabetes but most prevalent ones are number of pregnancies, glucose rate, blood pressure, skin thickness, insulin, BMI, and diabetes pedigree function. In the following, we will delve into each symptom.

A. Number of Pregnancies in Diabetes

In the United States, about 1% to 2% of pregnant women have type 1 or type 2 diabetes and about 6% to 9% of pregnant women develop gestational diabetes. Diabetes during pregnancy has increased in recent years. Recent studies found that from 2000 to 2010, the percentage of pregnant women with gestational diabetes increased 56% and the percentage of women with type 1 or type 2 diabetes before pregnancy increased 37%.

Diabetes in pregnancy varies by race and ethnicity. Asian and Hispanic women have higher rates of gestational diabetes and black and Hispanic women have higher rates of type 1 or type 2 diabetes during pregnancy.

B. Glucose Rate

One of the top priorities for people living with diabetes is managing blood sugar. This means ensuring that your blood sugar levels are within the targets that you and your healthcare team have set. The Canadian Diabetes Association clinical practice guidelines have recommended targets for people with type 2 diabetes. There are different target ranges for blood sugar levels before you eat (this is called a "fasting state") and after you eat a meal.

C. Blood Pressure in Diabetes

High blood pressure (hypertension) can lead to many complications of diabetes, including diabetic eye disease and kidney disease, or make them worse. Most people with diabetes will eventually have high blood pressure, along with other heart and circulation problems.

D. Skin Thickness

Diabetes can affect many parts of your body, including your skin. When diabetes affects the skin, it's often a sign that your blood sugar (glucose) levels are too high. This could mean that: You have undiagnosed diabetes, or pre-diabetes or your treatment for diabetes needs to be adjusted

E. Insulin

In type 2 diabetes, there are primarily two interrelated problems at work. Your pancreas does not produce enough insulin — a hormone that regulates the movement of sugar into your cells — and cells respond poorly to insulin and take in less sugar.

F. BMI

Any increase in BMI above normal weight levels is associated with an increased risk of being diagnosed as having complications of diabetes mellitus. For men, the increased risk of these complications occurred at higher BMI levels than in women. Ocular complications occurred at higher BMI levels than other complication types in both men and women.

G. Diabetes Pedigree Function

A particularly interesting attribute used in the study was the Diabetes Pedigree Function. It provides some data on diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient.

In this paper, we Try to present an algorithm to predict the diabetes based on symptoms. In the remainder of this paper, section II provides an overview of Principal Component Analysis, or PCA algorithm which is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets. Section III introduces applied ML algorithms for diabetes prediction, section IV describes the dataset, and at last section V, VI and VII present the result of the PCA, classification methods results and conclusion, respectively.

II. Principal Component Analysis

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used 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 in the large set.

Reducing the number of variables of a dataset naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make

analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible. In the following we explained step by step explanation of PCA:

A. Get your data

Separate your data set into Y and X. Y will be the validation set and X will be the training set. In simple terms, we will use X for our study and use Y to check whether our study is correct.

B. Give your data a structure

Take the 2 dimensional matrix of independent variables X. Rows represent data items and columns represent features. The number of columns is the number of dimensions.

For each column, subtract the mean of that column from each entry. (This ensures that each column has a mean of zero.)

C. Standardize your data

Given the columns of X, are features with higher variance more important than features with lower variance, or is the importance of features independent of the variance? (In this case, importance means how well that feature predicts Y.)

If the importance of features is independent of the variance of the features, then divide each observation in a column by that column's standard deviation. Call the centered and standardized matrix Z.

D. Get Covariance of Z

Take the matrix Z, transpose it and multiply the transposed matrix by Z.

$$Convariance(Z) = Z^T Z$$

The resulting matrix is the covariance matrix of Z, up to a constant.

E. Calculate Eigen Vectors and Eigen Values

Calculate the eigenvectors and their corresponding eigenvalues of $Z^T Z$.

The eigendecomposition of $Z^T Z$ is where we decompose $Z^T Z$ into

$$PDP^{-1}$$

where P is the matrix of eigenvectors D is the diagonal matrix with eigenvalues on the diagonal and values of zero everywhere else.

The eigenvalues on the diagonal of D will be associated with the corresponding column in P that is, the first element of D is λ and the corresponding eigenvector is the first column of P. This holds for all elements in D and their corresponding eigenvectors in P. We will always be able to calculate PDP^{-1} in this fashion.

F. Sort the Eigenvectors

Take the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_p$ and sort them from largest to smallest. In doing so, sort the eigenvectors in P accordingly. (For example, if λ_3 is the largest eigenvalue, then take the third column of P and place it in the first column position.)

Call this sorted matrix of eigenvectors P^*. The columns of P^* are the same as the columns of P in a different order. Note that these eigenvectors are independent of one another.

G. Calculate the new features

Calculate

$$Z^* = ZP^*$$

This new matrix, Z^*, is a centered/standardized version of X but now each observation is a combination of the original variables, where the weights are determined by the eigenvector. As a bonus, because our eigenvectors in P^* are independent of one another, each column of Z^* is also independent of one another.

H. Drop unimportant features from the new set

We need to determine which features from the new set we wish to keep for further study.

Side note: As each eigenvalue is roughly the importance of its corresponding eigenvector, the proportion of variance explained is the sum of the eigenvalues of the features you kept divided by the sum of the eigenvalues of all features.

There are three common methods to do this:

Method1: Arbitrarily select how many dimensions we want to keep

Method2: Calculate the proportion of variance for each feature, pick a threshold, and add features until you hit that threshold

Method3: Calculate the proportion of variance for each feature, sort features by proportion of variance and plot the cumulative proportion of variance explained as you keep more features. One can pick how many features to include by identifying the point where adding a new feature has a significant drop in variance explained relative to the previous feature, and choosing features up until that point.

III. Classification Algorithms

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. In this report, three classification algorithms have been applied: A generative classifier known as Naive Bayes, a lazy learner algorithm know as K-Nearest-Neighbors, and Decision Tree Classifier.

A. Naive Bayes Classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem

states the following relationship, given class variable y and dependent feature vector x_1 through x_n:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \quad (1)$$

Using the naive conditional independence assumption that

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y) \quad (2)$$

for all i, this relationship is simplified to

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \quad (3)$$

Since $P(x_1, ..., x_n)$ is constant given the input, we can use the following classification rule:

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \quad (4)$$

$$\hat{y} = argmax_y P(y)\prod_{i=1}^{n} P(x_i|y) \quad (5)$$

B. K_Nearest Neighbours

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. Notice in the image above that most of the time, similar data points are close to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph. There are other ways of calculating distance, and one way might be preferable depending on the problem we are solving. However, the straight-line distance (also called the Euclidean distance) is a popular and familiar choice.

Algorithm 1 K_nearest Neighbours algorithm

1) Load the data
2) Initialize K to your chosen number of neighbors
3) For each example in the data
4) Calculate the distance between the query example and the current example from the data.
5) Add the distance and the index of the example to an ordered collection
6) Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
7) Pick the first K entries from the sorted collection
8) Get the labels of the selected K entries
9) If regression, return the mean of the K labels
10) If classification, return the mode of the K labels

C. Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Every decision tree includes a root node, some branches, and leaf nodes. The internal nodes present within the tree describe the various test cases. Decision Trees can be used to solve both classification and regression problems. The algorithm can be thought of as a graphical tree-like structure that uses various tuned parameters to predict the results. The decision trees apply a top-down approach to the dataset that is fed during training.

The Decision Tree algorithm uses a data structure called a tree to predict the outcome of a particular problem. Since the decision tree follows a supervised approach, the algorithm is fed with a collection of pre-processed data. This data is used to train the algorithm.

Decision trees follow a top-down approach meaning that the root node of the tree is always at the top of the structure while the outcomes are represented by the tree leaves. Decision trees are built using a heuristic called recursive partitioning (commonly referred to as Divide and Conquer). Each node following the root node is split into several nodes.

The key idea is to use a decision tree to partition the data space into dense regions and sparse regions. The splitting of a binary tree can either be binary or multi way. The algorithm keeps on splitting the tree until the data is sufficiently homogeneous. At the end of the training, a decision tree is returned that can be used to make optimal categorized predictions.

An important term in the development of this algorithm is Entropy. It can be considered as the measure of uncertainty of a given dataset and its value describes the degree of randomness of a particular node. Such a situation occurs when the margin of difference for a result is very low and the model thereby doesn't have confidence in the accuracy of the prediction.

IV. Dataset Description

Diabetes dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The datasets consists of several medical predictor variables and one target variable. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

The distributions, central values and variability of the dataset's features where examined using box and whisker plots and their five number summaries. As seen in 1. As you can see some features like Glucose and BMI are normally distributed and some features are not.

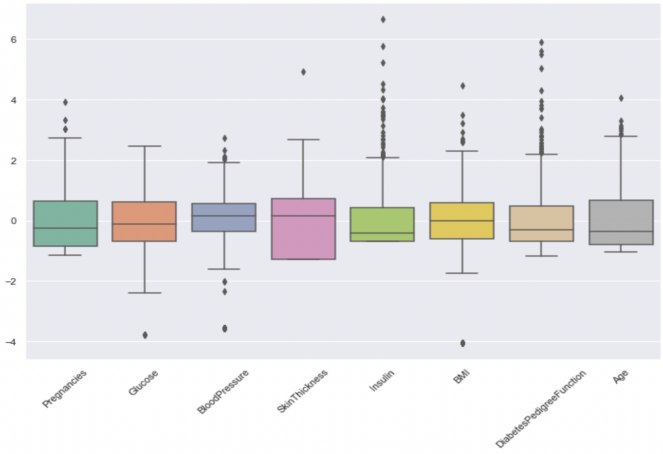

Fig. 1. Box and whisker plot of centered features

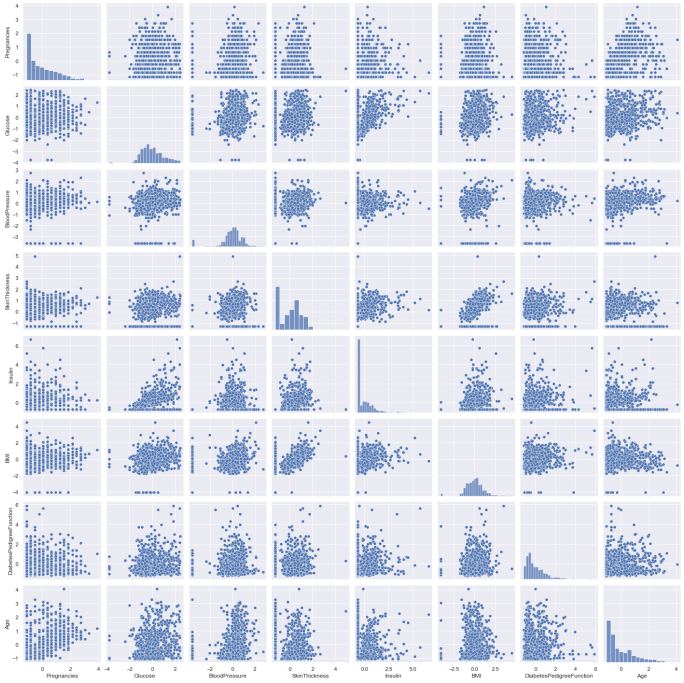

Fig. 3. Pair Plot

The correlation matrix of the normalized features is shown in 2. It is evident from all numbers in the matrix that all the features are not related to each other because the numbers of correlation matrix are close to zero. But as you can see some correlations like age and pregnancies are more correlated to each other. This observation is more supported by the pair plot. The neutral correlations are shown by scattered points.

Fig. 2. Feature Covariance Matrix

V. PCA Results

Then, PCA was applied to the diabetes dataset exactly like the presented algorithm in section II. Now, our number of features is r which is less than 8 ($r < 8$). The original $n \times p$ dataset is reduced using the eigenvalue matrix (A). Each column of the eigenvalue matrix represent a PC. The amount of information each PC captures in the data determines the reduced dimensions (r). The eigenvalue matrix obtained by applying PCA to the diabetes dataset is:

$$A = \begin{bmatrix} 0.12 & 0.59 & -0.013 & 0.08 & -0.47 & 0.19 & -0.58 & 0.12 \\ 0.39 & 0.17 & 0.47 & -0.40 & 0.47 & 0.094 & -0.06 & 0.45 \\ 0.36 & 0.18 & -0.53 & 0.056 & 0.33 & -0.63 & -0.19 & -0.011 \\ 0.44 & -0.33 & -0.24 & 0.038 & -0.49 & 0.001 & 0.28 & 0.56 \\ 0.43 & -0.25 & 0.34 & -0.35 & -0.35 & -0.27 & -0.13 & -0.55 \\ 0.45 & -0.10 & -0.36 & 0.05 & 0.25 & 0.68 & -0.035 & -0.34 \\ 0.27 & -0.12 & 0.43 & 0.83 & 0.12 & -0.08 & -0.09 & -0.008 \\ 0.20 & 0.62 & 0.07 & 0.07 & -0.11 & -0.03 & 0.71 & -0.21 \end{bmatrix}$$

The Scree plot and Pareto plot are used to visually represent the amount of variance that each PC accounts for. The percentage of variance accounted for by the j^{th} PC is computed using the following equation:

$$l_j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100\%, for j = 1, ..., p \tag{6}$$

where λ_j is the amount of variance and eigenvalue of the j^{th} PC. For diabetes dataset, λ_js and l_js are represented by

the matrices below:

$$\lambda = \begin{bmatrix} 2.09711056 \\ 1.73346726 \\ 1.03097228 \\ 0.87667054 \\ 0.76333832 \\ 0.68351839 \\ 0.42036353 \\ 0.40498938 \end{bmatrix} \tag{7}$$

$$l = \begin{bmatrix} 0.26179749 \\ 0.21640127 \\ 0.12870373 \\ 0.10944113 \\ 0.09529305 \\ 0.08532855 \\ 0.05247702 \\ 0.05055776 \end{bmatrix} \tag{8}$$

Fig.4 and Fig.5 plot the number of principal components versus the explained variance. The Scree plot shows the elbow is located at the third PC. Therefore, based on the explained variance of the first three PCs and the Scree plot, the dimension of the data can be reduced to three $(r = 3)$.

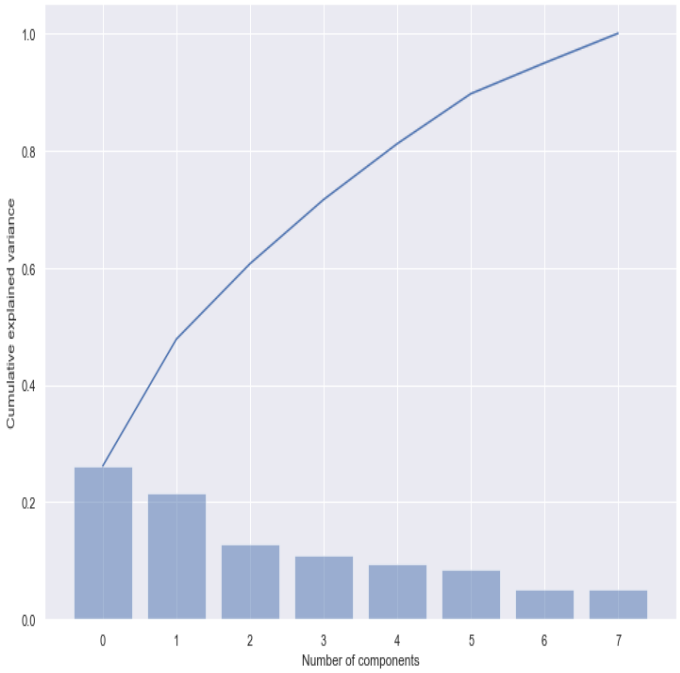

Fig. 5. Pareto Plot

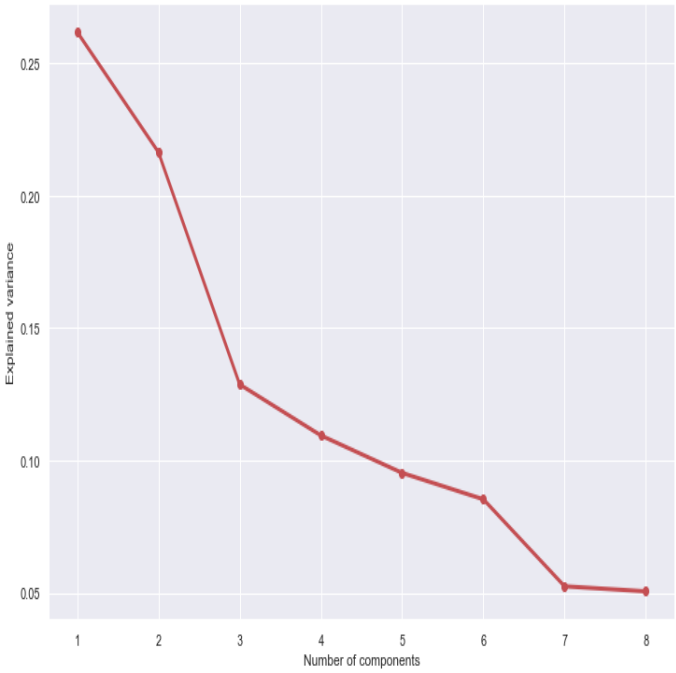

Fig. 4. Scree Plot

The PC coefficient plot in Fig. 6 visually represent the contribution has on the first three PCs. The plot is a visual representation of the coefficients. As you can see, Pregnancies and Age features are located at the left side of the plot far from other features at the right corner of the plot. Also, all the features have positive coefficients for Z_1 and Insulin and Skin Thickness have negative coefficients in Z_2. (For the convenience, we plot only two components for the PC coefficient plot).

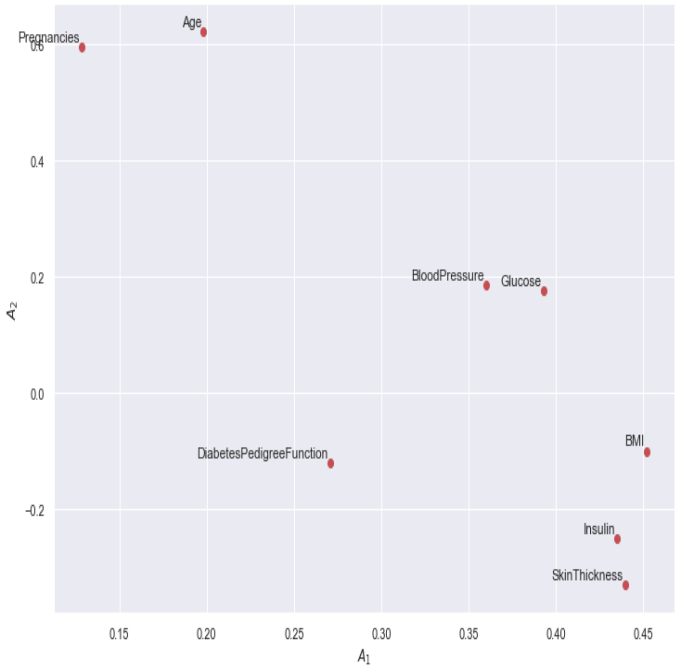

Fig. 6. PC Coefficient Plot

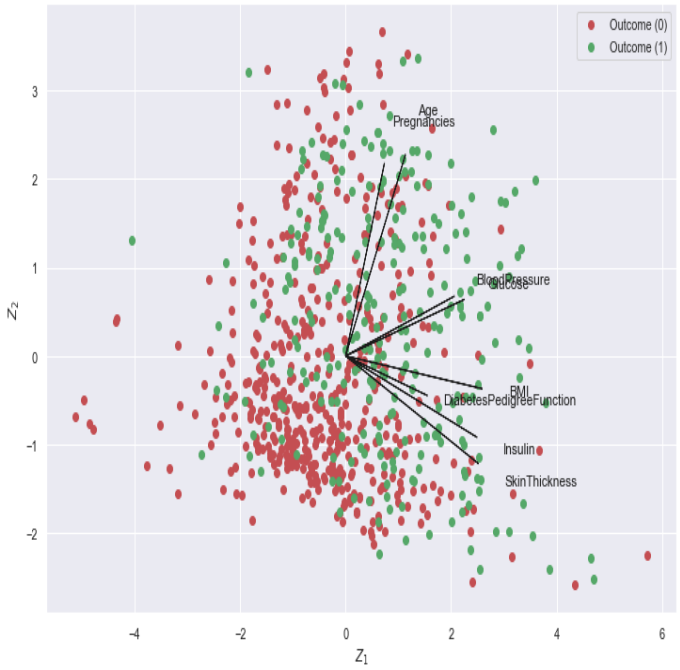

Fig. 7. Bi Plot

The Bi Plot in 7 represents an alternative view to the effect each feature has on the first two PCs. The Bi Plot axes represent the two principal components (Columns of A). Whereas the rows of each eigenvector matrix are represented as vectors. Each observation is represented as a point in the biplot. The angles between the vectors and the axes represent the contribution of the variables. The acute angles between the area vector and the horizontal axis and the perimeter vector and the horizontal axis reaffirm that they contribute the most to the first PC. The biplot also reflects that the first PC has a positive coefficient for all variables. In addition, the vectors that point in the same direction are positively correlated. For instance, the Pregnancies and Age features, point in the same direction. The indicates that these two features has a positive correlation. As you can see, Glucose and Blood Pressure have the same situation.

Biplot also shows the location of each data point and the class distribution on the reduced dimentional plane. The location of the observations on the plot indicate the score each observation has for the three PCs. The red points belongs to negative diabetes and the green ones are for positive diabetes. As you can see, the points belong to negative diabetes have the least scores for both the first and second PCs.

VI. Classification Results

In this section, the performance of three common classification algorithms are evaluated on the Diabetes dataset. The models are trained and tested on the original dataset, on the transformed dataset(i.e.PCs) and on the first two PCs of the dataset. A total of 9 experiments have been conducted. To begin, all three datasets where split into a training and a test set with a 70-30 split. Then, a grid search with k-fold cross validation were conducted on the training sets to determine the ideal hyper parameter values for each algorithm. Specifically, the number of k nearest neighbours was tuned for K-NN and the $depth$ of the DT was tuned. The splitting criterion for the DT was kept as the Gini impurity index instead of the information gain based entropy criterion. Note, there is no hyper-parameter to tune for NB since the parameters are estimated using maximum liklihood. Since the Diabetes dataset is a binary class classification algorithm, the categorical distribution was used to build the prior distribution. Further, since the dataset has continuous features, the Gaussian liklihood was selected to build the liklihood distribution.

Also, K-fold cross validation was selected for the hyperparameter tuning because the Diabetes dataset is considered a small dataset.

Using these ideal hyper parameter values, the models were retrained on the entire training sets, and evaluated on the test sets. Three versions of each mode; were trained. The test sets were used to compare the performance of the different algorithm and to examine the impact PCA has on performance. At last, all models were implemented with the help of scikit-learn.

In this report, we use precision and recall to evaluate the outcome of three represented classification algorithms. Precision measure what fraction of predictions are actually positive, whereas measures the fraction of positives that where actually detected for each class. The results for precision and recall are visually represented using confusion matrices and Receiver Operation Characteristic (ROC) curves. By definition a confusion matrix C is such that $C_{i,j}$ is equal to the number of observations known to be in group i and predicted to be in group j. Thus in binary classification, the count of true negatives is $C0,0$, false negatives is $C1,0$, true positives is $C1,1$ and false positives is $C0,1$. ROC curves plot the false positive rate versus the true positive rate for various threshold values.

The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision. In this case, the F1-scores for both the classifiers can be used to determine which one produces better results.

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{9}$$

The Macro averaged F1-score was selected for this study since the diabetes dataset has balanced classes. The Macro averaged F1-score computes the average of all F1-scores $(\sum_c = 1^C \frac{F_1(C)}{C})$. The macro averaged F1-score for each algorithm and dataset are shown in Fig. 8. From the plot you can see that the performance of all three datasets were close to each other. Therefore, we can conclude that feature reduction technique with PCA algorithm does not make any difference from the dataset and it is possible to make the results even better.

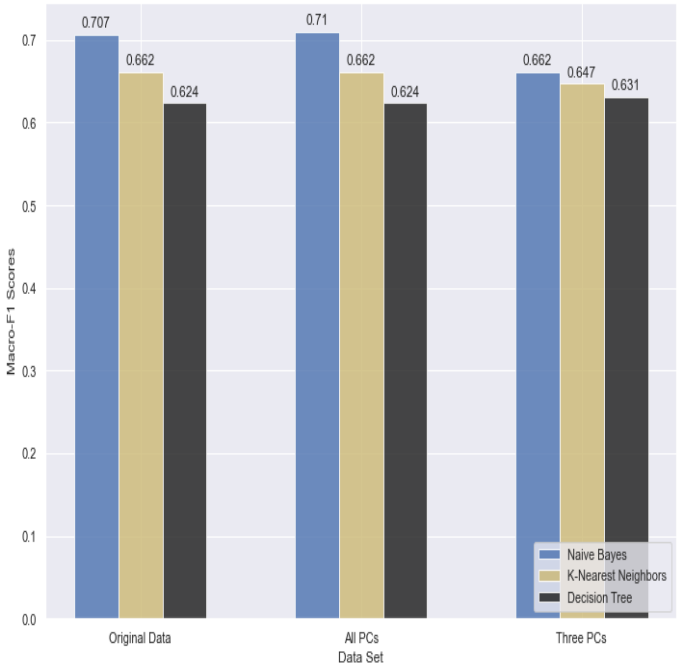

Fig. 8. Micro average F1-score on the test set for all three datasets

Fig.9, Fig.10, and Fig.11 illustrates the decision boundaries formed by the models on the three PCs datasets. The plots visually illustrates the NB algorithm superior performance compared to the K-NN and DT algorithms.

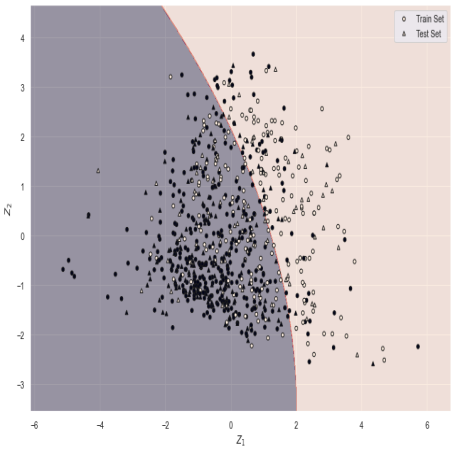

Fig. 9. Naive Bayes

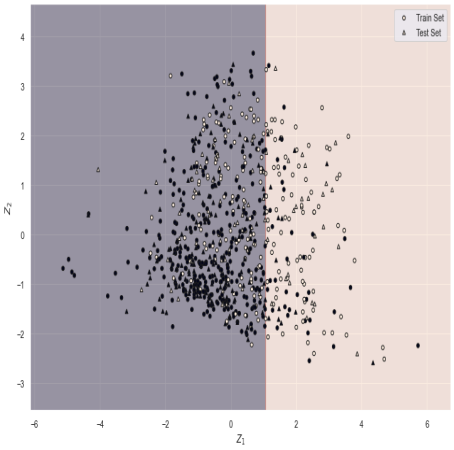

Fig. 10. K-Nearest Neighbours

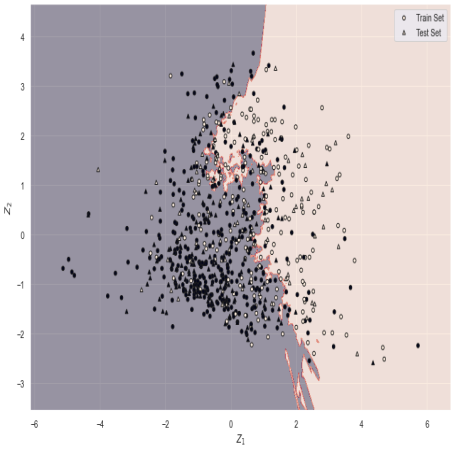

Fig. 11. Decision Tree

Next, the confusion matrices were generated to determine where the algorithms have difficulty discriminating between classes. The computation matrix horizontal axis represent the models' predictions and the vertical axis represents the true label. For the sake of brevity, only the confusion matrices from the original dataset are shown. As you can see in Fig. 12, Fig. 13, and Fig. 14 accuracy of Gaussian NB classifier is 76%, for the Decision Tree accuracy is 73%, and finally, the accuracy of K-Nearest neighbour is 72%.

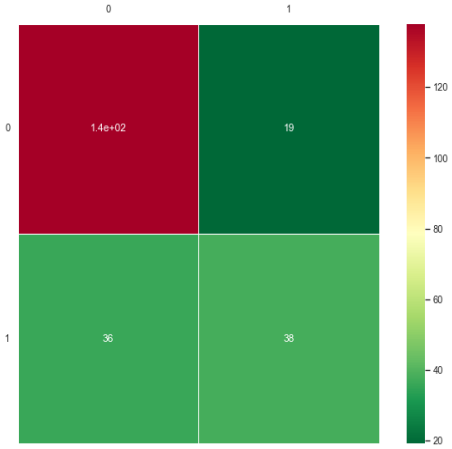

Fig. 12. Naive Bayes Confusion Matrix

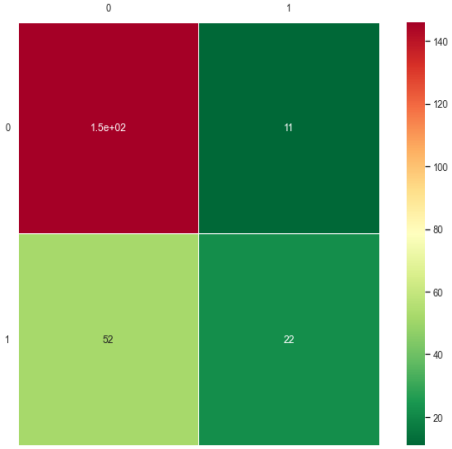

Fig. 13. K-Nearest Neighbours Confusion Matrix

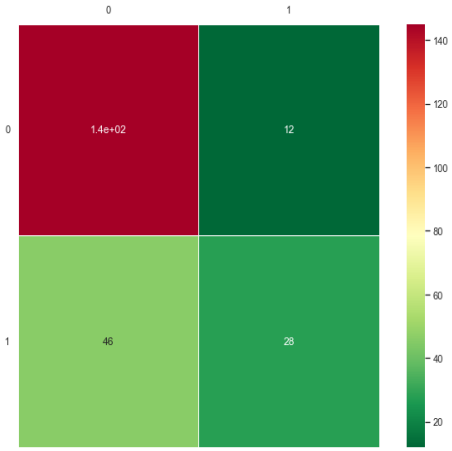

Fig. 14. Decision Tree Confusion Matrix

VII. Conclusion

As you can see in the previous section, we applied PCA algorithm with three classification algorithms on Diabetes dataset. At first, we applied PCA algorithm and found out that

we can reduce our features to three. Then, we applied Naive base, K-NN, and DT algorithms and realized that with PCA algorithm our accuracy does not fall and also Naive Bayes classification is the best classification method for classification of the Diabetes dataset.

For all three datasets, we found F1-score, accuracy, precision, and recall but we did not mention all these materials in here. If you like to know the details of implementation of this algorithm you can. find it in Jupyter Notebook file presented besides this report.

REFERENCES

[1] https://www.aboutkidshealth.ca/Article?contentid=1717&language=English&hub=diabetes

[2] https://www.diabetescarecommunity.ca/healthy-living-landing/diabetes-management/managing-blood-sugar/?gclid=CjwKCAjw3cSSBhBGEiwAVIl0Zxatgwssoc2SnVc_2F1LZWKQt6WySfWRAe8-Uf3qMZjkIUndZhMS5RoCvGYQAvD_BwE

[3] https://www.sciencedirect.com/science/article/abs/pii/S0190962287700723

[4] https://pubmed.ncbi.nlm.nih.gov/25580754/#:~:text=Conclusions%3A%20Any%20increase%20in%20BMI,BMI%20levels%20than%20in%20women

[5] https://machinelearningmastery.com/case-study-predicting-the-onset-of-diabetes-within-five-years-part-1-of-3/#:~:text=A%20particularly%20interesting%20attribute%20used,those%20relatives%20to%20the%20patient.

[6] https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[7] https://www.javatpoint.com/classification-algorithm-in-machine-learning#:~:text=The%20Classification%20algorithm%20is%20a,number%20of%20classes%20or%20groups.

[8] https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[9] https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f