Principal Component Analysis and Lung Cancer Prediction Using Machine Learning

Karur Mahaboob Danish Basha

Abstract: Principal Component Analysis (PCA) is explanatory technique to learn about data sets, the objective of PCA is to reduce the dimensionality of the data set while retaining as much as possible the variation in data set. Principal Components (PCs) are linear transformations of the original set of variables, and are uncorrelated, ordered so that first few components carry most of the variation in the original data set. In this report, PCA is applied on the Lung Cancer prediction data set, predicting the likelihood of a patient developing lung cancer, identifying the risk factors for lung cancer. Three different classification algorithms i.e., Gaussian Naïve Bayes (Gaussian NB), K Nearest Neighbours (KNN), Decision Trees (DT) are applied on original dataset and transformed dataset (data obtained after applying PCA) to predict the level of cancer. In improvement each model is tuned with ideal hyperparameters to obtain a better performance metrics and performance of each algorithm is measured using F1 score, confusion matrix and Receiver Operating Characteristics (ROC) curves. The decision boundaries for each model are also shown to show the model fitting on the dataset.

Index Terms: Principal Component Analysis, Multi Class Classification, Gaussian Naïve Bayes, K Nearest Neighbours, Decision Trees, Receiver Operating Characteristics.

1. Introduction

Lung Cancer is leading cause of cancer death worldwide, accounting for nearly 1.8 million deaths in recent years. Most lung cancer cases are attribute to smoking, but exposure

to air pollution is also a risk factor. A new study has found that air pollution may be linked to an increased risk of lung cancer, even in non-smokers.

The data represented here is of participants who were divided into two groups i.e., those who lived in areas with high levels of air pollution and those who lived in areas with low levels of air pollution. It has been found that the people in the high-pollution group were more likely to develop lung cancer than those in the low-pollution group. And most importantly the risk was higher in non-smokers than smokers, and the risk increased with age.

As this is an era of Machine Learning technology and its implementation techniques in almost all fields, thus ML techniques are performing a key role in prediction and identification of lung cancer by applying various classification techniques.

algorithms, Gaussian Naïve Bayes (GNB), K Nearest Neighbours (K-NN) and Decision Tree (DT) are applied on the original dataset and PCA transformed dataset. The purpose is to predict the level of cancer considering the respective inputs on different factors. An interpretation on classification models is done in the final step. All the classification model's application and respective results obtained are after applying PCA i.e., in this report the results are used from the transformed dataset. The classification results of dataset are found in the Jupyter Notebook.

The entire report is organized by eight sections. Following are the sections with relevant topics:

Section I: Introduction
Section II: Principal Component Analysis.
Section III: Overview on applied three Machine Learning Classification algorithms.
Section IV: Lung Cancer Dataset Description
Section V: Discussion on obtained PCA results
Section VI: In depth analysis of the

Classification results.
Section VII: Discussion on ROC curves.
Section VIII: Conclusion.

II. Principal Component Analysis

Principal Component Analysis (PCA) is a well-known unsupervised dimensionality reduction technique that constructs relevant features/variables. This relevant feature construction is achieved by linear transforming correlated variables into smaller number of uncorrelated variables. This is done by projecting the original data into the reduced PCA space using the eigenvectors of the covariance/correlation matrix which is Principal Components (PCs).

In precise, PCA is an orthogonal transformation of the data into a series of uncorrelated data living in the reduced PCA space such that first component explains the most variance in the data with subsequent component explaining less.

Why to Use PCA?

- PCA technique is particularly useful in processing data where multi-collinearity exists between the features/variables.
- PCA can be used when the dimensions of the input features are high i.e., lot of variables.
- PCA can be also used for denoising and data compression.

PCA Algorithm:

For the given data matrix X with dimensions **n × p** PCA can be applied with following four main steps.

Step 1: Standardization

The aim of this step is to standardize the range of continues initial variables so that each one of them contributes equally to the analysis.

Firstly, mean vector $\bar{x}$ of each column is computed, this mean vector is of dimension p. Mean vector is represented as

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

In short to compute the centered data matrix **Y = HX** by subtracting off-column means.

Where H represents the centering matrix.

Step 2: Covariance Matrix Computation

The aim of this step is to understand how the variables of the input dataset are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated with redundant information. Computing covariance matrix helps to identify the correlations.

The covariance matrix is of dimension **p × p** symmetric matrix where p is number of dimensions.

The covariance matrix S of

$$S = \frac{1}{n-1} Y'Y \qquad \text{(iv)}$$

Step3: Compute the Eigenvectors and Eigenvalues.

Eigenvectors and eigenvalues are calculated form the covariance matrix to determine the Principal Components (PCs) of the data.

PCs are new variables that are constructed as linear combinations or mixtures of initial variables. These combinations are uncorrelated and most of the information within the initial variables are compressed into the first components.

Using eigen-decomposition eigenvectors and eigen values are computed.

$$S = A\,\Lambda\,A' = \sum_{j=1}^{n} \lambda_j\, a_j a'_j \qquad \text{(ii)}$$

Where,

- A = $(a_1, a_2, \dots, a_j)$ is **p × p** orthogonal matrix $(A'A = I)$ whose columns eigenvectors of S such that $a_j a'_j$ = 1, j = 1, …., p.
- Λ = diag $(\lambda_{1,} \lambda_{2,} \dots., \lambda_{p,})$ is a **p × p** diagonal matrix whose elements are the decreasing order i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{p,}$

Step 4: Principal Components:

The aim of this step is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components. This can be done by multiplying the transpose of the original dataset by the transpose of the feature vector.

Transformed data matrix Z of dimension **n × p** is computed which is represented as

$$\mathbf{Z = YA} \quad \text{(iii)}$$

$$Z = (z'_1, z'_2, \dots, z'_p) = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ \vdots & & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{np} \end{pmatrix}$$

III. Machine Learning Classification Algorithms

A. Gaussian Naïve Bayes (GNB)

In machine learning, naïve bayes classifiers are a family of simple “probabilistic classifiers” based on applying Bayes theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad \text{(v)}$$

Using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{prior \times \text{likelihood}}{evidence}$$

Naïve Bayes classifier is generally a parametric model, which assumes that the presence of particular feature in a class is unrelated to the presence of any other feature.

B. K – nearest neighbour (K-NN)

K-NN is a supervised classification algorithm to train a model by classifying the samples according to the nearest training examples in a feature space. K-NN is called lazy learning algorithm as it approximates the function only locally and defers all computations until classification. 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 neighbours, with the object being assigned to the most common class amongst its k numbers of the nearest neighbours.

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 Euclidian distance of k number of neighbours, takes the K nearest neighbours as per the calculated Euclidian distance. Finally, K-NN assigns the new sample to the class which has the maximum number of samples.

C. Decision Tree Classifier

Decision tree is a non-parametric supervised learning method, which builds classification or regression models in the form of a tree structure. 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. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed. This process is continued on the training set until meeting a termination condition.

The following are motives behind any decision tree algorithm

- Select a best attribute using Attribute Selection Measures(heuristic) to split the records.
- Make that attribute a decision node and breaks the dataset into smaller subsets.
- Starts tree building by repeating this process recursively for each child until one of the conditions will match:
    1. All the tuples belong to the same attribute value.
    2. There are no more remaining attributes.
    3. There are no more instances.

IV Dataset Description

The Lung Cancer prediction dataset is used for this project is taken from the Kaggle. This dataset provides information of different types of features which probes to risk level of lung cancer. The level of lung cancer is categorized into three types i.e., High, Medium, and Low. The dataset projects eight features for the prediction of lung cancer. The features are:

i. Age: The age of the patient

ii. Gender: The gender of the patient

iii. Air Pollution: The level of air pollution exposure of the patient

iv. Alcohol use: The level of alcohol use of the patient

v. Dust Allergy: The level of dust allergy of the patient.

vi. Occupational Hazards: The level of occupational hazards of the patient

vii. Genetic Risk: The level of genetic risk of the patient

viii. Chronic Lung Disease: The level of chronic lung disease of the patient.

It includes thousand entries for each attribute. Finally, it contains a column titled ‘Level’ which is basically the label for the class to predict the level of lung cancer.

The dataset is visualized with Bar Graphs where X-axis consists of ‘Risk Level’ and Y-axis of number of entries. It can be observed from the Figure 1.

Utilizing the box and whisker plots and their eight number summaries on the dataset, the distributions, central values, and variability of the features were measured. Figure 2 illustrates the box plot of the features of the Lung Cancer prediction dataset. It is observed from the dataset that there is variation in each feature and not all are approximately under normal distribution. And an outlier is found above the Age feature. Figure 3 shows the correlation matrix for the normalized features of dataset. The features with large positive numbers are Alcohol Use, Occupational Hazards, Genetic Risk, Chronic Lung Disease, Dust Allergy.

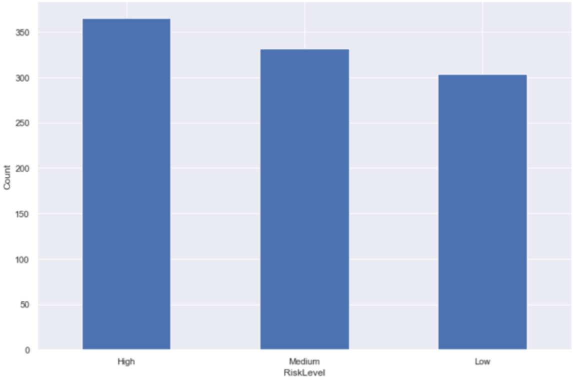


Figure 1: Data Visualization (Bar Graph)

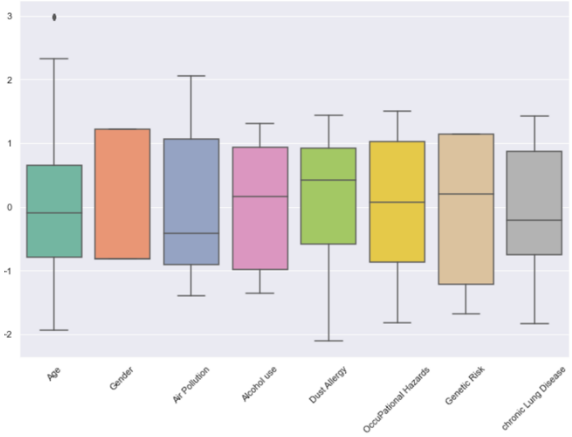


Figure 2: Box Plot

This evident implies that these five features are highly correlated. Other features Age, Gender and

Air Pollution show less correlation with other features in the dataset.

The figure 4 support the observation the highly correlated features contain higher number of cells. On the contrary Age and Gender displays less apparent correlation.

V. PCA Results

PCA is applied on Lung Cancer prediction dataset. PCA is a python package to perform Principal Component Analysis and to create insightful plots. The core of PCA is built on *sklearn* functionality to Find maximum compatibility when combining with other packages. Following are the functionalities implemented using PCA

- Biplot to plot loadings
- Determine the explained variance.
- Extract the best performing features.
- Scatter plot with loadings.
- Outlier detection using Hotelling T2

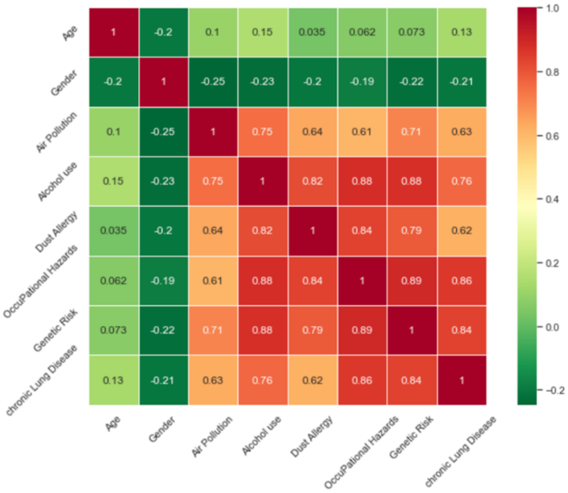


Figure 3: Correlation Matrix

In this project the python PCA library is used which brought more flexibility in all aspects. As a result, the figures and plots which are shown in this report are using PCA library.

By applying PCA steps the feature set of 8 can be reduced to *r* numbers of features where $r < 8$.

The original **n** × **p** dataset is reduced using eigenvector matrix A. Here each column of eigenvector matrix is represented as PC. Each PC captures an amount of data that determines the dimension (*r*). Following obtained matrix is eigenvector matrix which is represented in a graph.

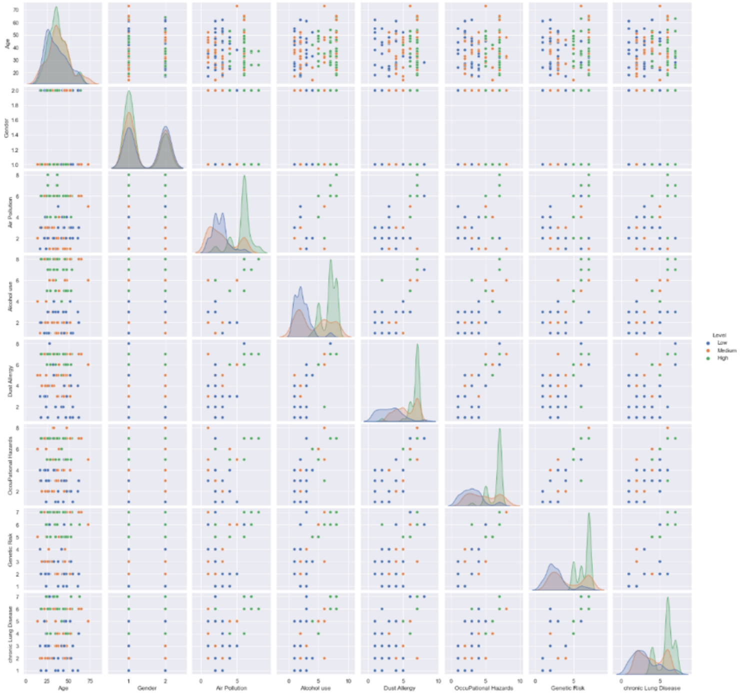


Figure 4: Pair Plot

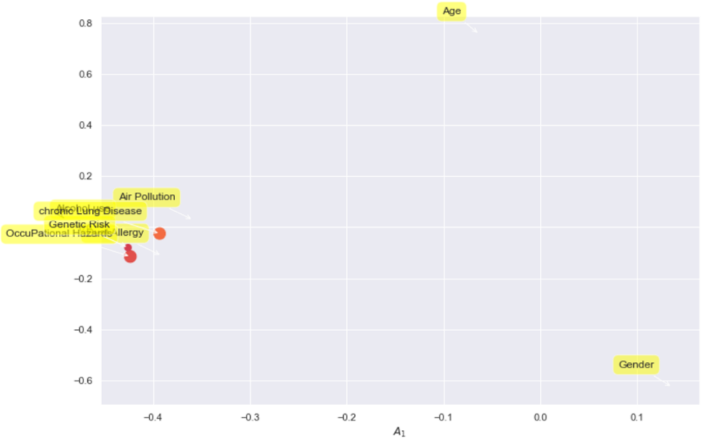


Figure 5: Eigenvector Matrix

Eigenvector matrix is 8 × 8 dimension and to confirm whether the obtained matrix is eigenvector matrix or not a small check is preformed i.e., calculating '*Sum of Squares of Each Row*' produced the result '1'.

Eigen Vector Matrix A is following

$$\begin{bmatrix} -0.0634 & 0.7572 & 0.6367 & -0.0045 & 0.0978 & -0.0457 & 0.0608 & -0.0417 \\ 0.1358 & -0.6267 & 0.7555 & -0.1311 & 0.0000 & -0.0016 & 0.0017 & 0.0005 \\ -0.3585 & 0.0287 & -0.0615 & -0.8493 & -0.2633 & -0.1829 & -0.0056 & -0.2062 \\ -0.4253 & -0.0173 & 0.0711 & -0.0840 & 0.1478 & 0.6509 & -0.5147 & 0.3099 \\ -0.3911 & -0.1114 & -0.0404 & -0.0234 & 0.7107 & -0.4831 & 0.0939 & 0.2915 \\ -0.4234 & -0.1148 & 0.0522 & 0.3555 & 0.0704 & -0.0670 & -0.2335 & -0.7838 \\ -0.4257 & -0.0799 & 0.0197 & 0.1138 & -0.0944 & 0.3816 & 0.8026 & 0.0143 \\ -0.3930 & -0.0250 & 0.1001 & 0.3381 & -0.6165 & -0.3961 & -0.1528 & 0.4000] \end{bmatrix}$$

And respective corresponding eigenvalues are:

$$\lambda = \begin{bmatrix} 4.9414 \\ 1.1421 \\ 0.7937 \\ 0.4554 \\ 0.3801 \\ 0.1366 \\ 0.1063 \\ 0.0521 \end{bmatrix}$$

The principal components from the above matrix are generated like

$$Z_1 = -0.0634X_1 + 0.7572X_2 + 0.6367X_3 - 0.0045X_4 + 0.0978X_5 - 0.0457X_6 - 0.0608X_7 - 0.0417X_1. \quad \text{(vi)}$$

Similarly, the other principal components $Z_2, .., Z_7$ will be generated with respect to the Eigen Vector matrix values.

Figure 6 demonstrates the Screen plot of PCs whereas Figure 7 demonstrates the Pareto plot of PCs where the features variances are represented in high frequency to. low frequency. Both Screen and Pareto plots 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.

$$\text{j} = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100, \text{ j} = 1, 2, \ldots, \text{p.} \quad \text{(vii)}$$

where $\lambda_j$represents the eigenvalue and the amount of variance of the j^{th} PC.

Figures 6 and 7 plot the number of PCs vs explained variance. It can be observed from both figures that the variance of first three PCs contribute to 85.9% of the amount of variance of the original dataset, i.e., first PC holds 61.7% of variance (l1= 61.7%), second PC holds 14.3% of variance (l2= 14.3%) and the third PC holds 9.9% of variance (l3= 9.9%). The screen plot presents that elbow located on the second PC.

The Biplot in Figure 9 display the different visual representation of PCs. The axes of biplot represents the first two PCs. Each of the observations in the dataset is drawn as a dot on the plot.

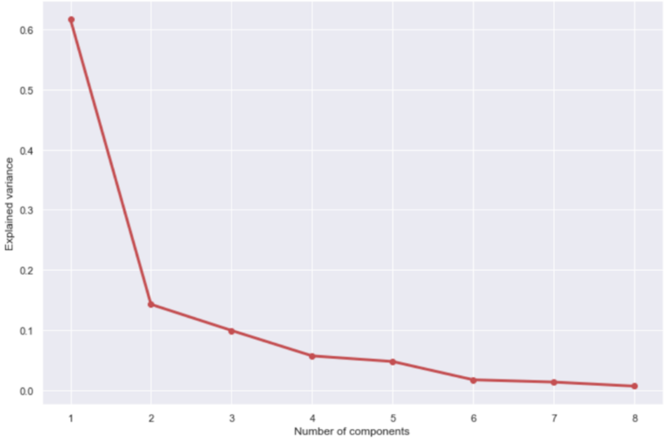


Figure 6: Screen Plot

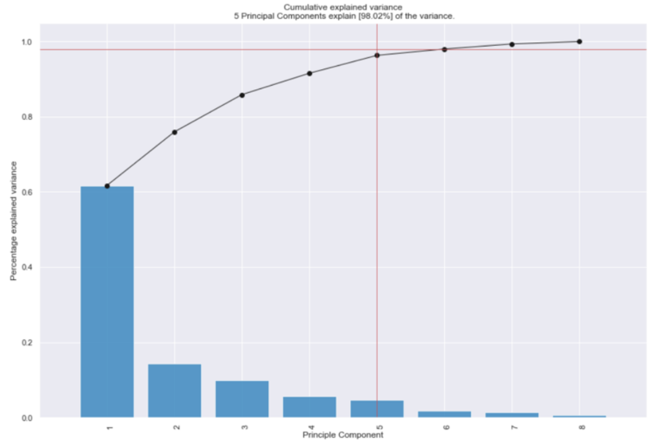


Figure 7: Pareto Chart

The vectors for features namely Air Pollution, Alcohol use, Dust Allergy, Occupational Hazards Genetic Risk and Chronic Lung Disease shows very small angle with first PC and large angle with second PC. This shows that these five features have large contribution to the first PC and very small contribution to the second PC. Figure 8 supports this evidence. The vectors Age and Gender shows opposite phenomenon. As they create a less angle with second PC and very large with first PC, this supports that they are very much related to second PC and less related to the first PC. By observing the figure 9 it shows that the vecto rs Air Pollution, Alcohol use, Dust Allergy, Occupational Hazards Genetic Risk and Chronic Lung Disease are in same direction which infers that they are positively correlated whereas the Age and Gender are in opposite direction clears that they are negatively correlated.

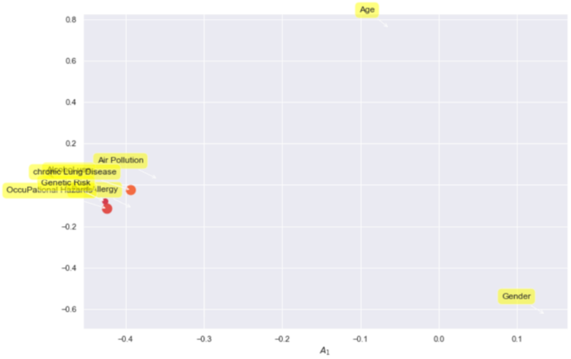


Figure 8: PC Coefficient Plot

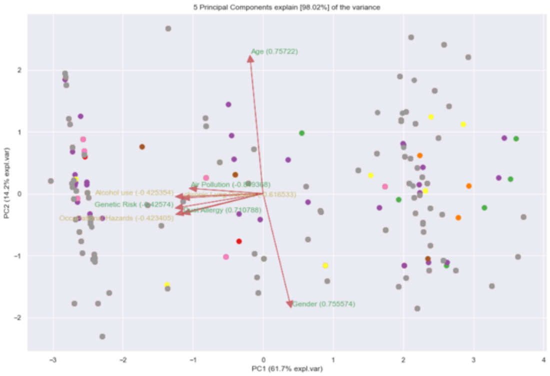


Figure 9: Biplot

Control charts for the PCs where each PCs control chart have their respective UCL, CL and LCL lines are drawn, and it has been found that out of control points are not found from the following two control charts.

VI. In Depth Analysis of Classification Results

Performance of three popular ML classification algorithms on Lung Cancer Prediction dataset is analysed and discussed here. In order to observe the effects of PCA on the lung cancer dataset, the classification algorithms are applied on PCA applied dataset.

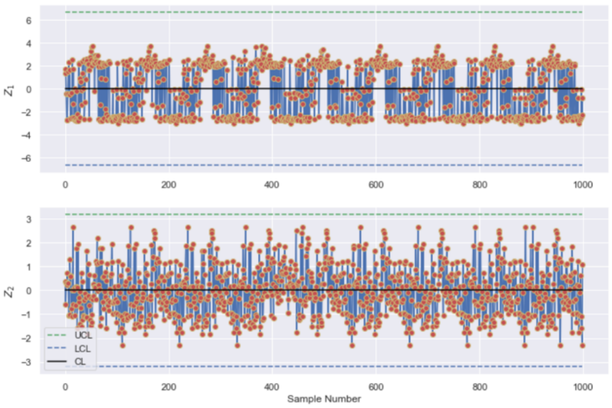


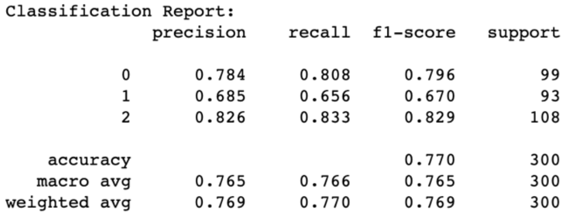
Figure 10: Control Charts

Training dataset size is set to 70% and the rest 30% is set to the testing process. All the applied process can be found in Jupyter notebook. This report focuses on results obtained after applying PCA. The obtained confusion matrices are on 'Predictable Feature' of the dataset named 'Level' where it consists of three classifications Low, Medium, and High which are numbered to 0, 1 and 2 respectively. Following is the representation of classifications of 'Level' class

Low – 0
Medium – 1
High - 2

In order to improve the performance of models, hyperparameters tuning plays an important role. Hyperparameters tuned with KNN and DT models.

Gaussian Naïve Bayes (GNB) is applied for this PCA dataset the score generated for this is 77%, classification report and Naïve Bayes Confusion matrix are seen in following Figure 11 and 12 respectively. The scatter plot also been drawn where the Tarin dataset and Test dataset are plotted in between the PCs.



```
Classification Report:
              precision    recall  f1-score   support

           0      0.784     0.808     0.796        99
           1      0.685     0.656     0.670        93
           2      0.826     0.833     0.829       108

    accuracy                          0.770       300
   macro avg      0.765     0.766     0.765       300
weighted avg      0.769     0.770     0.769       300
```

Figure 11: GNB Classification Report

Whereas for KNN model tuning with hyperparameters i.e., grid with 'k' neighbours and then followed three steps they are finding the best 'k' and then getting the best 'k', finally applying that best 'k'. Score for this KNN obtained is 97.6%. The classification report, Confusion Matrix and Scatter plot of train and test dataset are obtained which are seen in the following Figures 14, 15 and 16 respectively.

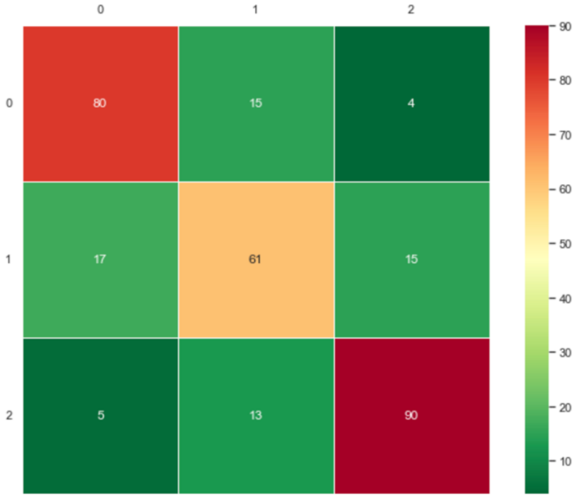


Figure 12: GNB Confusion Matrix

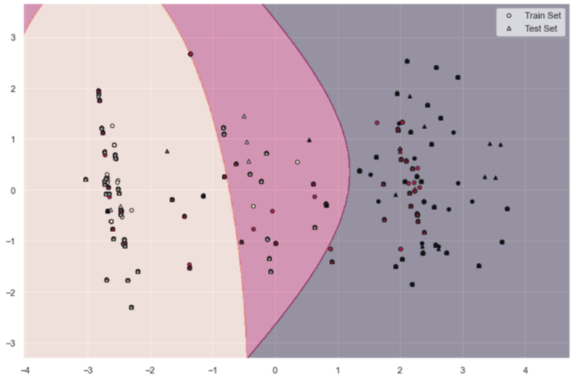
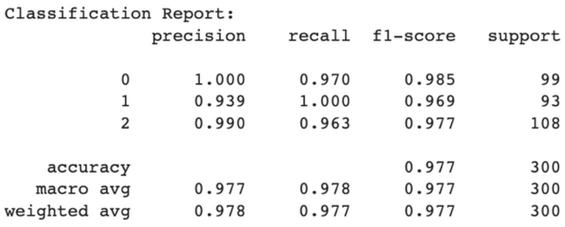


Figure 13: GNB Scatter Plot



```
Classification Report:
              precision    recall  f1-score   support

           0      1.000     0.970     0.985        99
           1      0.939     1.000     0.969        93
           2      0.990     0.963     0.977       108

    accuracy                          0.977       300
   macro avg      0.977     0.978     0.977       300
weighted avg      0.978     0.977     0.977       300
```

Figure 14: KNN Classification Report

Finally, the Decision Tree (DT) has been applied on the PCA applied dataset with hyperparameter 'depth' provided the depths with powers of 2 i.e., 2, 4, 8, 16, 32, 64. Again here three stages of process are followed they are finding the best depth, Get the best tree depth and then applying best depth.

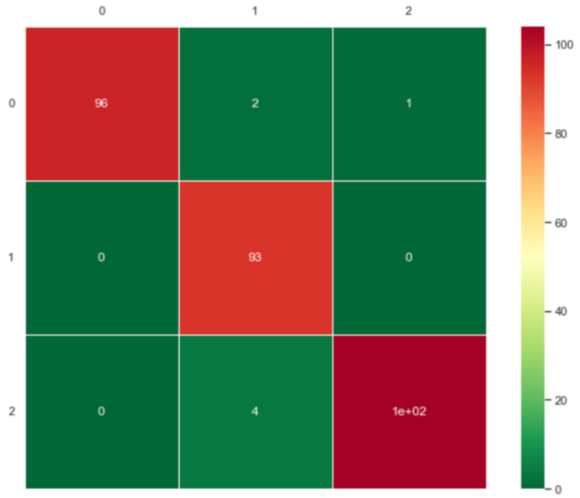


Figure 15: KNN Confusion Matrix

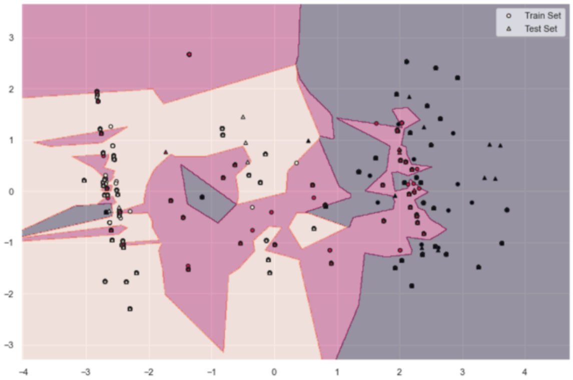
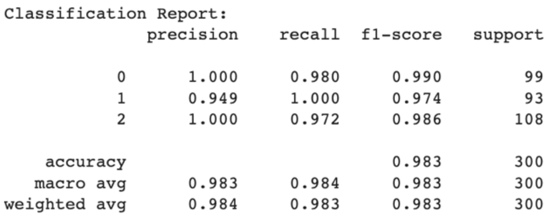


Figure 16: KNN Scatter Plot

The score generated after applying this DT is 98.3%. The classification report of DT is in following figure 17.



```
Classification Report:
              precision    recall  f1-score   support

           0      1.000     0.980     0.990        99
           1      0.949     1.000     0.974        93
           2      1.000     0.972     0.986       108

    accuracy                          0.983       300
   macro avg      0.983     0.984     0.983       300
weighted avg      0.984     0.983     0.983       300
```

Figure 17: DT Classification Report.

DT confusion matrix and Scatter plot are seen in figures 18 and 19 respectively.

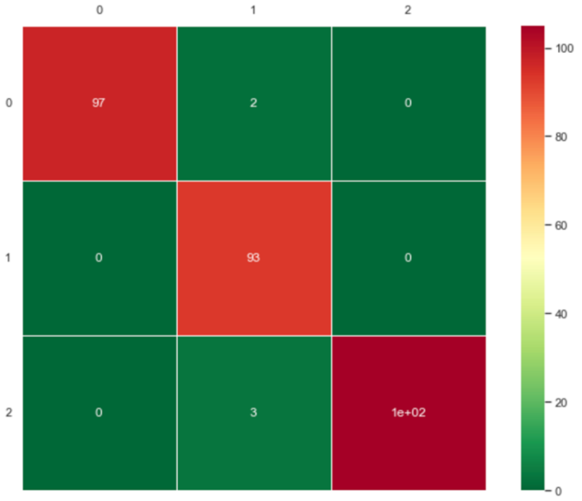


Figure 18: DT Confusion Matrix.

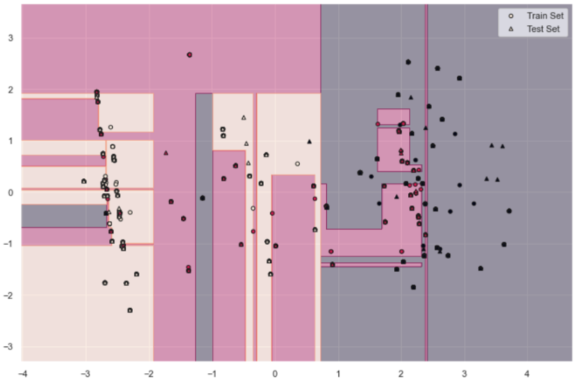


Figure 19: DT Scatter plot

VII. Discussion on ROC Curves

ROC curve is a graph that demonstrates the performance of classification model at all classification thresholds. The curve plots two parameters: True Positive, False Positive Rate. These parameters are the main components of building the 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 the applied models GNB, KNN, and DT are in the following figure 20

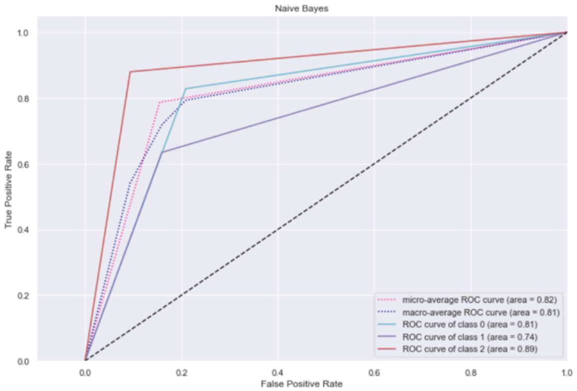


Figure 20 (i): GNB ROC Curve

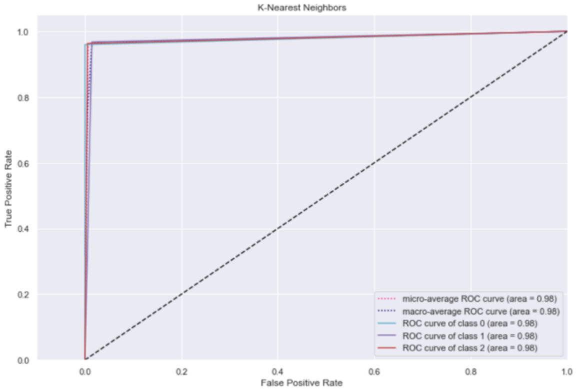


Figure 20 (ii): KNN ROC Curve

All these ROC curve figures represent results of confusion matrices of respective models. False Positive on the X-axis versus True Positive rate on Y-axis for a number of different candidate threshold values between 0.0 and 1.0, the graphs of micro and macro average curves.

The above ROC curves shows that DT is the best at predicting the three classes and can predict 99% accurately followed by KNN with 98% and GNB takes the last place with 82%. These results are replica of results from the confusion matrix. These above all explains that the three algorithms are capable of successfully classifying the level of lung cancer.

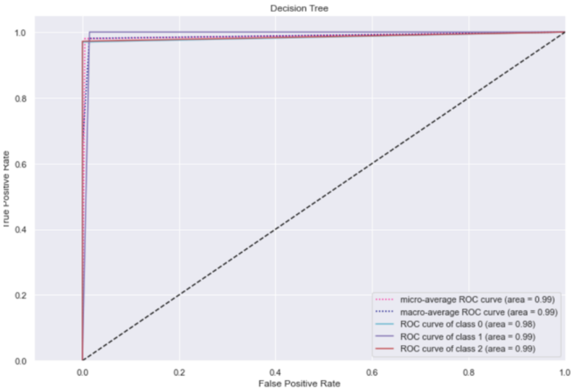


Figure 20 (iii): DT ROC Curve

A bar graph is plotted on F1 scores of each algorithm where Data set is on X-axis and Macro F1 Scores on Y-axis, as the algorithms are applied on original dataset and also PCA applied dataset, here only the PCA applied data set is considered in this report hence from the below figure 21 the applied PCA dataset scores are projected and found in second and third points of X-axis.

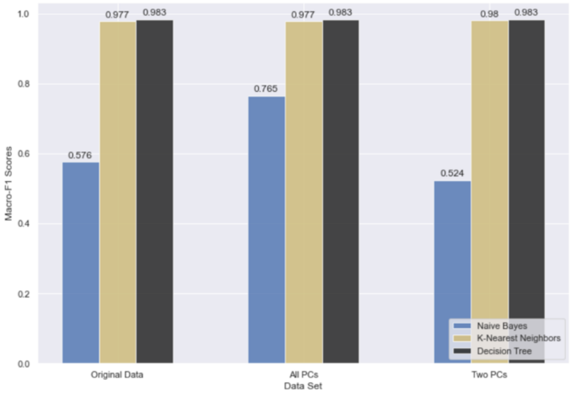


Figure 21: F1 Score Plot

From the graph it is observed that DT algorithm performs little better when compared to KNN algorithm in all types of datasets and importantly in applied PCA dataset too. GNB shows very little performance when compared to the two others. Whereas KNN shows better performance than GNB and little bit lesser performance than DT.

VIII. Conclusion

Here this report concludes that, PCA and three prominent ML classification algorithms applied on the lung cancer dataset. Lung cancer dataset holds the information on attributes of various factors to predict the level of lung cancer. Firstly, PCA is applied on the original dataset. The first three PCs apprehends 85.9% variance of the data. Hence, the feature set is reduced to number of PCs it holds the largest variance from 8 features. Experiments are performed on the respective PCs and different plots are generated to validate the obtained results from the different perspectives. Moving forward, three prominent ML classification algorithms GNB, KNN and DT are applied on the original dataset as well as the transformed dataset i.e., PCA applied dataset. Algorithms KNN and DT are tuned with hyperparameters, and performance evaluation is conducted by comparing confusion matrices, ROC curves and F1-Scores plot. It is observed that after hyperparameters tuning performance metrics score of the algorithm has improved. The GNB, KNN and DT algorithms performed well, and it has found that DT performance is better when compared with other two on both the datasets. Thus, performance level order from higher to lower is DT, KNN and GNB. Interestingly, after applying PCA the DT showed best performance metrics followed by KNN. To summarize, all three algorithms successfully predict the level of lung cancer for the lung cancer dataset.

References:
[1] Textbook
Advanced Statistical Approaches to Quality: A. Ben Hamza
Dataset:
[2]
https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link
[3]
https://scikit-learn.org/stable/
[4]
https://builtin.com/data-science/step-step-explanation-principal-component-analysis
[5]
https://erdogant.github.io/pca/pages/html/index.html