BREAST CANCER WISCONSIN REPORT

By:

Bayat Vahabodin

Contents

Figures

Tables

Introduction

Breast cancer is a disorder in which the breast cells proliferate uncontrollably. There are several types of breast cancer. The kind of breast cancer is determined by which cells in the breast develop into cancer.

Breast cancer can start in any area of the breast. A breast is composed of three major components: lobules, ducts, and connective tissue. The lobules are the milk-producing glands. Ducts are tubes that transport milk to the nipple. The connective tissue (fibrous and fatty tissue) surrounds and binds everything together. Most breast cancers start in the ducts or lobules.

Breast cancer can spread outside of the breast via blood and lymph arteries. Breast cancer is considered to have metastasized when it spreads to other regions of the body.

The following are the most prevalent types of breast cancer:

- Invasive ductal carcinoma. The cancer cells start in the ducts and subsequently spread to other breast tissue regions. Invasive cancer cells can also move to other areas of the body, a process known as metastasis.
- Invasive lobular carcinoma. Cancer cells develop in the lobules and subsequently move to nearby breast tissues. These invasive cancer cells have the potential to spread to other places of the body as well.

Data Understanding

This assignment analyses the Breast Cancer Wisconsin (Diagnostic) Dataset, which contains 569 instances with 32 features for each one. Ten features are identified for each sample, which are categorized into: mean, standard error, and worst.

- Columns 2 to 12 are about the mean.
- Columns 13 to 22 are about the standard error.
- Columns 23 to 32 are about the worst (largest).

This data set was created by Dr. William H. Wolberg, a physician at the University of Wisconsin Hospital in Madison, Wisconsin, USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt, which can perform the analysis of cytological features based on a digital scan which is used to categorize tumors as benign or malignant.

Table 1: Attributes Description

Row	Attribute	Type	Description
1	ID number	Numeric (Ordinal)	Unique number for each person
2	Diagnosis	String (Nominal)	B = benign M = malignant
3	radius	Numeric	mean of distances from center to points on the perimeter
4	texture	Numeric	standard deviation of gray-scale values
5	perimeter	Numeric	-
6	area	Numeric	-
7	smoothness	Numeric	local variation in radius lengths
8	compactness	Numeric	perimeter^2 / area - 1.0
9	concavity	Numeric	severity of concave portions of the contour
10	concave points	Numeric	number of concave portions of the contour
11	symmetry	Numeric	-
12	fractal dimension	Numeric	"coastline approximation" - 1

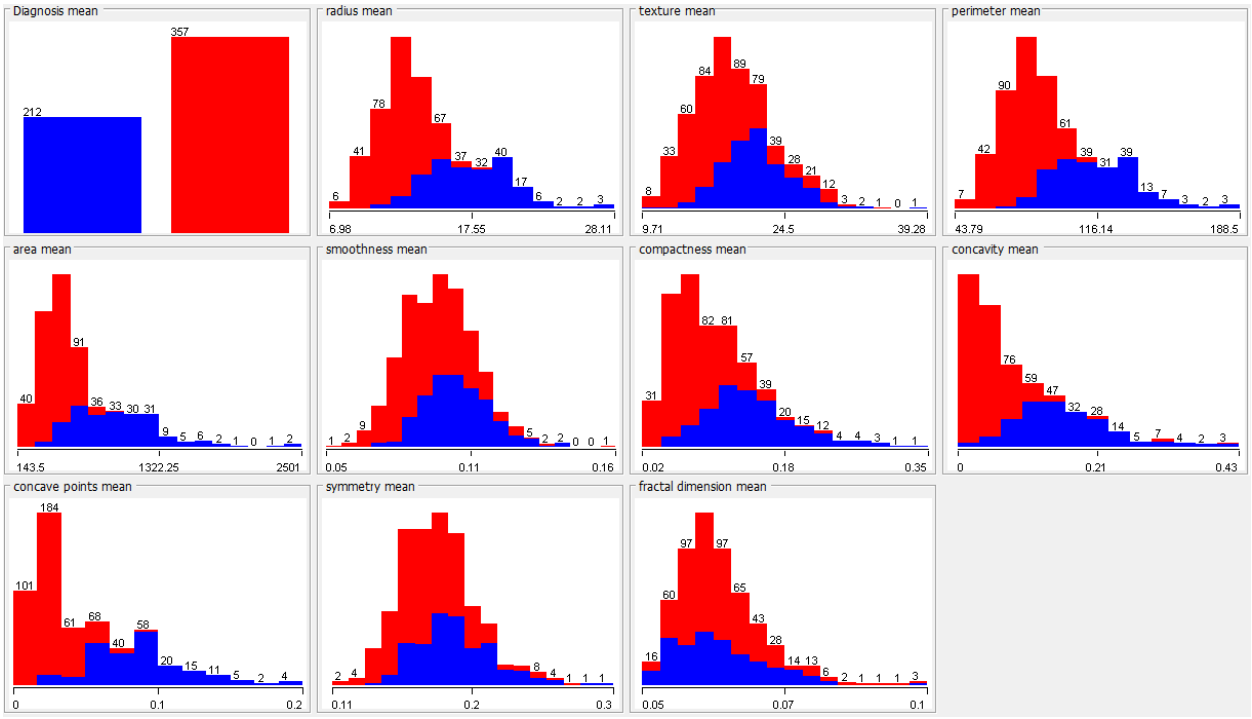

Figure 1: Attributes mean distributions.

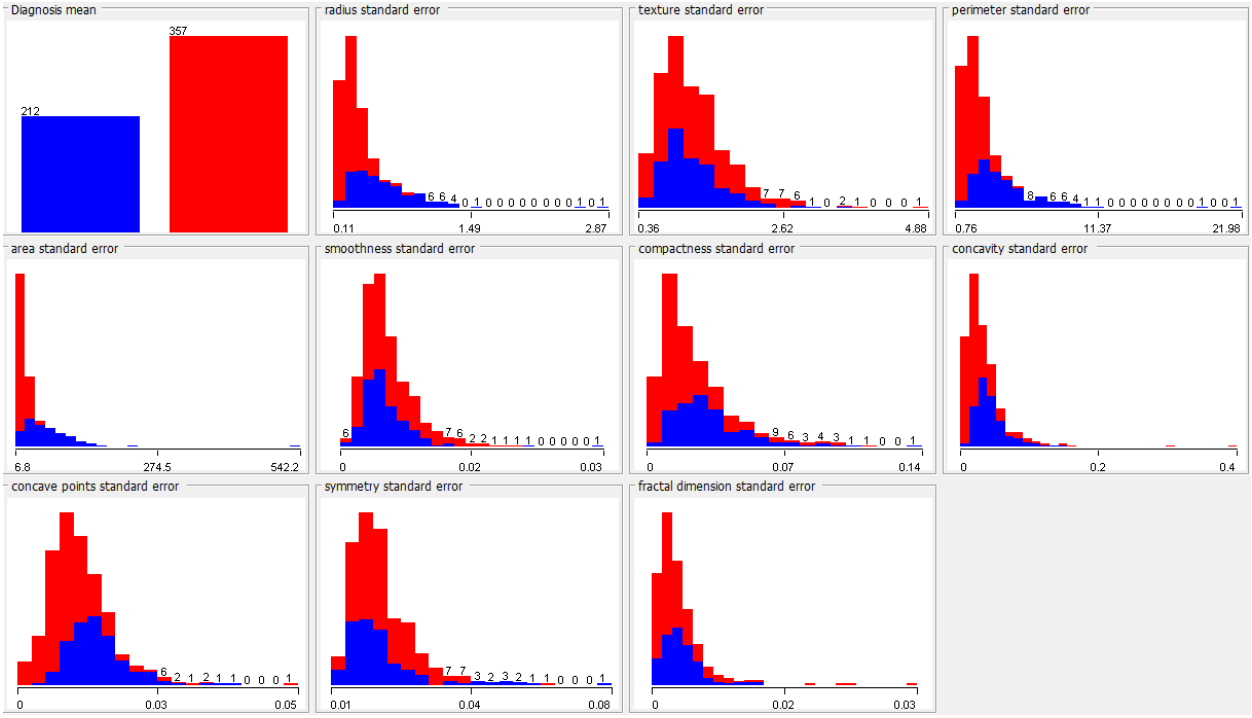

Figure 2: Attributes standard error distributions.

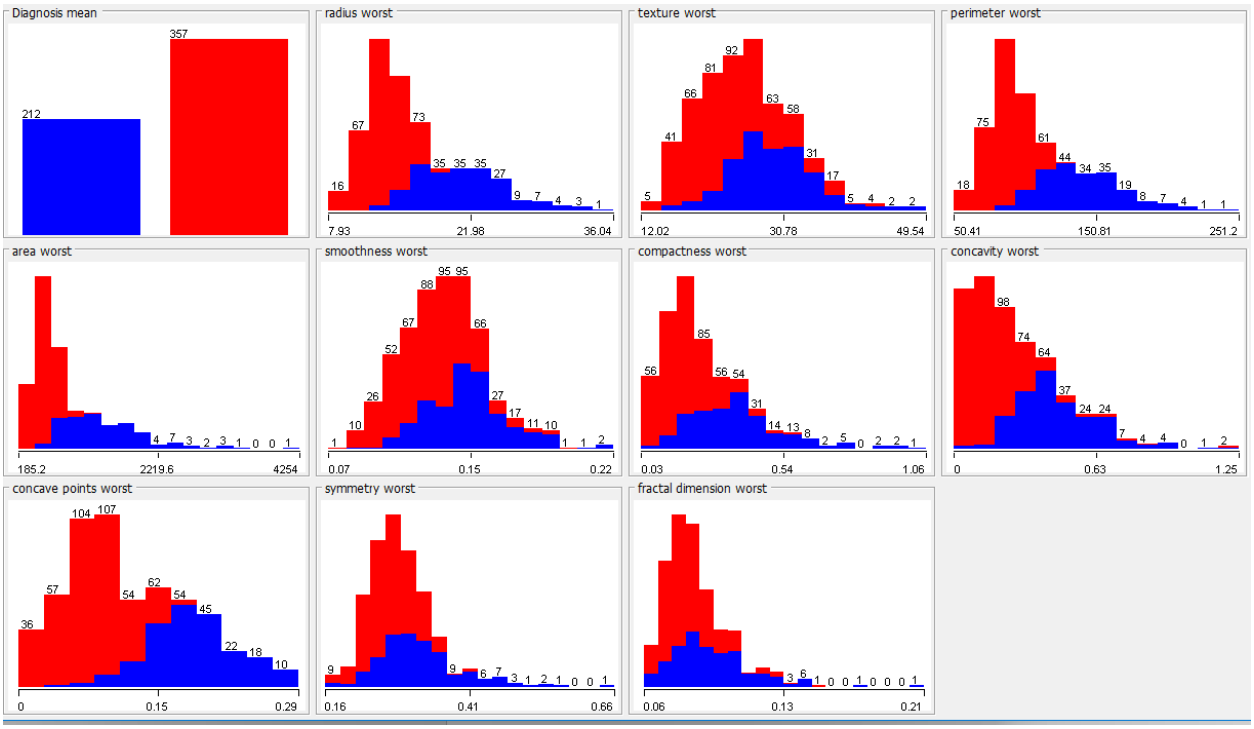

Figure 3: Attributes worst distributions.

From observing the attributes distributions, it is possible to see that they all follow a normal distribution with varying degrees of positive skewness. A data preparation step that could improve the interpretability of the data would be reducing the distribution skewness. Such methods would include taking the: square root, cube root, logarithm, or reciprocal of the data. However, for the purposes of this assignment, no data transformation will be applied.

Data Preparation

After loading the data into python using the Pandas library, it was checked for duplicates and missing values. Neither were found in the dataset.

The ID will not be useful as it has no effect on diagnosis, therefore, it was removed. The data was then split into two parts, the training set and test set. The remaining attributes are not on the same scale, hence the StandardScaler from scikit learn library was used to standardize the data. This is done by subtracting both sets by the training mean and then dividing by the training standard deviation. The mean and standard deviation from the training set is used so that there is no data leak between training and testing sets.

A copy of the data was then created to perform PCA and LDA separately for later comparison.

PCA

After performing PCA, the explained variance of each PC was plotted. This can be seen in Figure 4. From this, it was determined that only 7 principal components will be used. This equates to more than 91% of the variance explained.

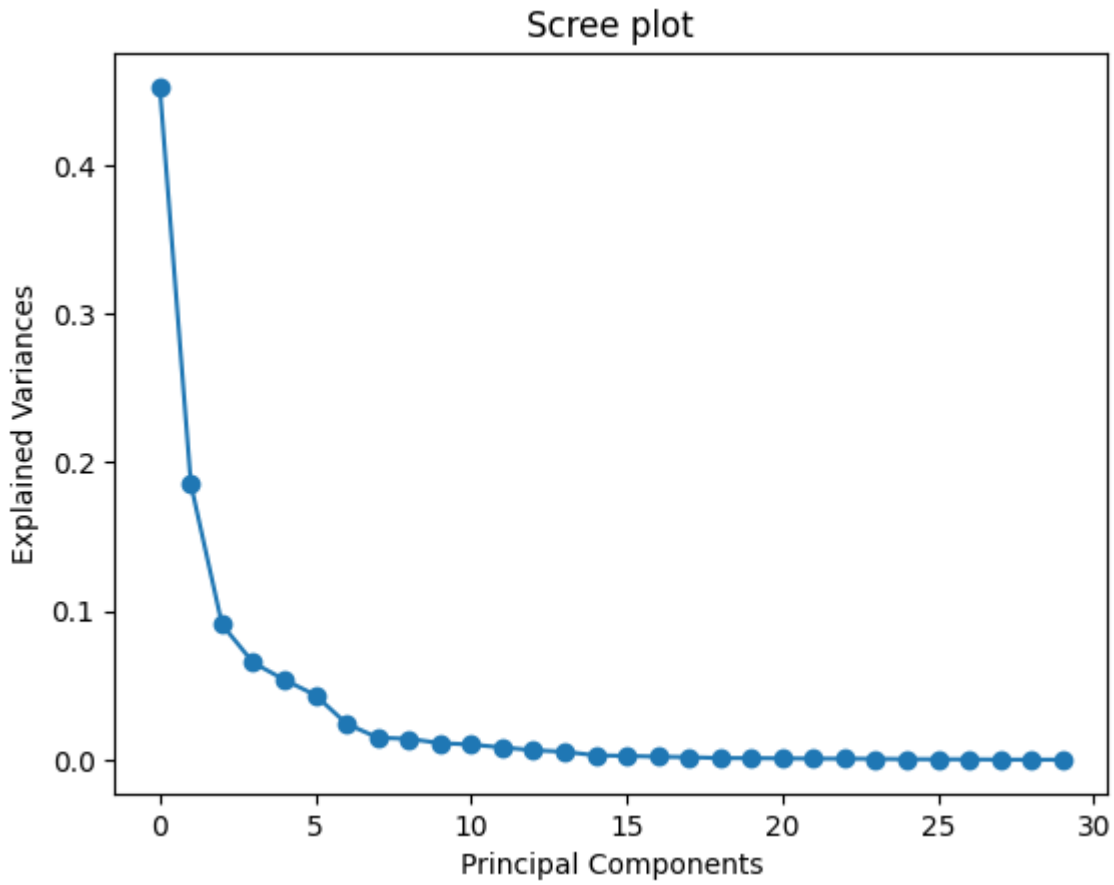

Figure 4: Scree plot for the principal components of the original dataset

LDA

Linear Discriminant Analysis (LDA) is a dimensionality reduction technique that is commonly used for supervised classification problems. It is a linear method that tries to find a linear combination of features that maximizes the separation between different classes. Unlike PCA which is an unsupervised method that tries to find the directions of maximum variance in the data, LDA is a supervised method that tries to find the directions of maximum class separation.

In summary, LDA is a supervised method that is used for classification problems and aims to maximize class separation, while PCA is an unsupervised method that is used for dimensionality reduction and aims to find the directions of maximum variance in the data.

Modeling & Evaluation

In this assignment, we compared the results of two dimensionality reduction techniques: PCA and LDA. Additionally, another test with neither was carried. The models used for this comparison were: kNN, Decision Tree, Random Forest, MLP, and SVM.

kNN

k-Nearest Neighbors (kNN) is a non-parametric, instance-based learning algorithm. Given a new observation, it finds the k-number of training examples that are closest to it and uses the majority class among those k-neighbors as the prediction. The value of k is usually determined using cross-validation or the elbow method.

This method was tested with k between 1 and 30. The k with highest accuracy when performing no dimensionality reduction was 3. This will also be used when comparing PCA and LDA. After fitting the data, with the original dataset, after PCA, and after LDA, we found the accuracy to be 98%, 97%, and 95% respectively. The other parameters and confusion matrices can be seen through Figure 5 to Figure 10.

K Nearest Neighbour Algorithm Accuracy is 98%

.....

Classification Report for K Nearest Neighbour:

.....

	precision	recall	f1-score	support
B	0.97	1.00	0.99	104
M	1.00	0.96	0.98	67
accuracy			0.98	171
macro avg	0.99	0.98	0.98	171
weighted avg	0.98	0.98	0.98	171

Figure 5: Classification Report for kNN Algorithm on original dataset

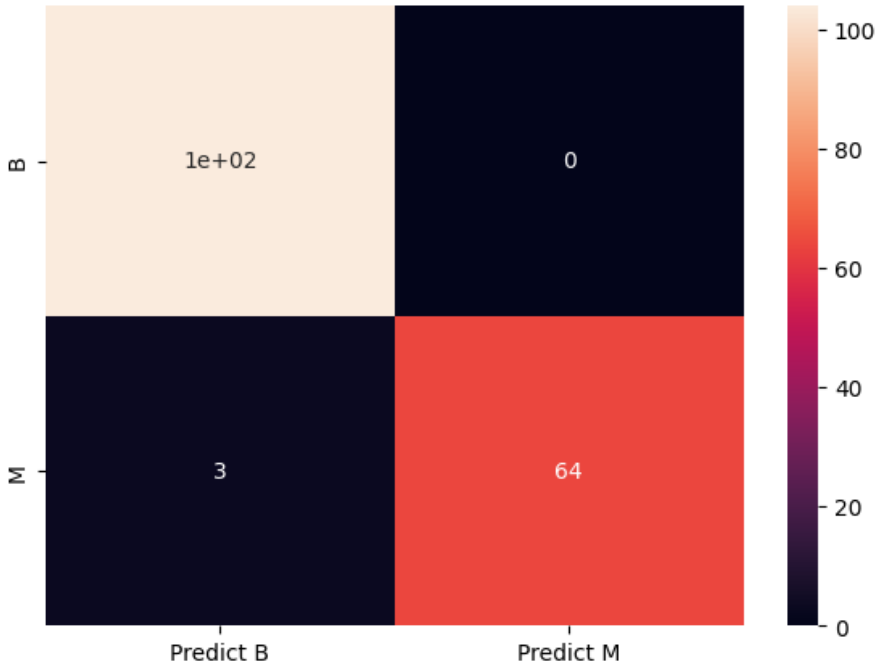

Figure 6: Confusion matrix for kNN Algorithm on original dataset

K Nearest Neighbour Accuracy After Applying PCA is 97%

.....

Classification Report After Applying PCA for K Nearest Neighbour:

.....

	precision	recall	f1-score	support
B	0.96	0.99	0.98	104
M	0.98	0.94	0.96	67
accuracy			0.97	171
macro avg	0.97	0.97	0.97	171
weighted avg	0.97	0.97	0.97	171

Figure 7: Classification Report for kNN Algorithm after PCA

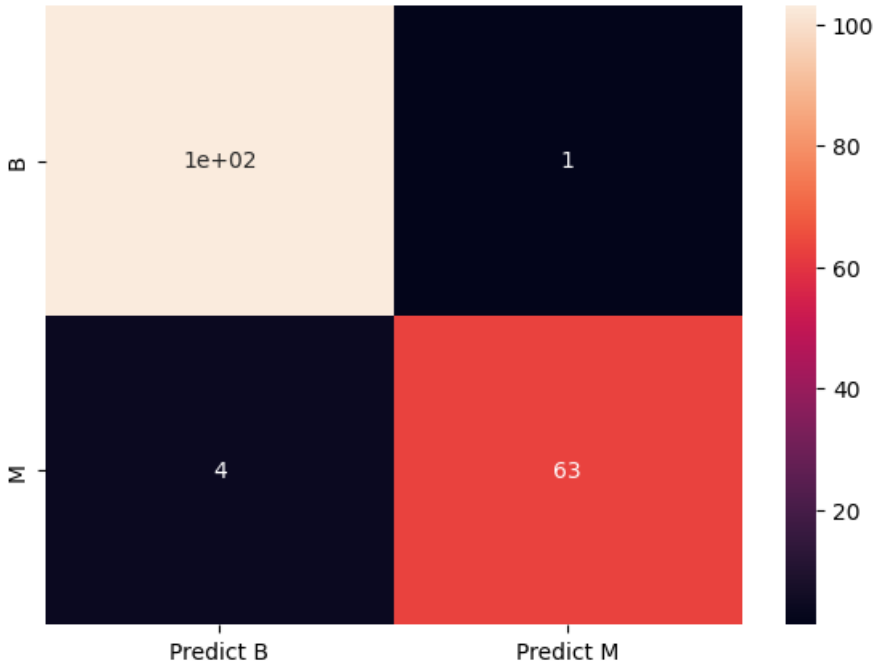

Figure 8: Confusion matrix for kNN Algorithm after PCA

K Nearest Neighbour Accuracy After Applying LDA is 95%

.....

Classification Report After Applying LDA for K Nearest Neighbour:

.....

	precision	recall	f1-score	support
B	0.97	0.95	0.96	104
M	0.93	0.96	0.94	67
accuracy			0.95	171
macro avg	0.95	0.95	0.95	171
weighted avg	0.95	0.95	0.95	171

Figure 9: Classification Report for kNN Algorithm after LDA

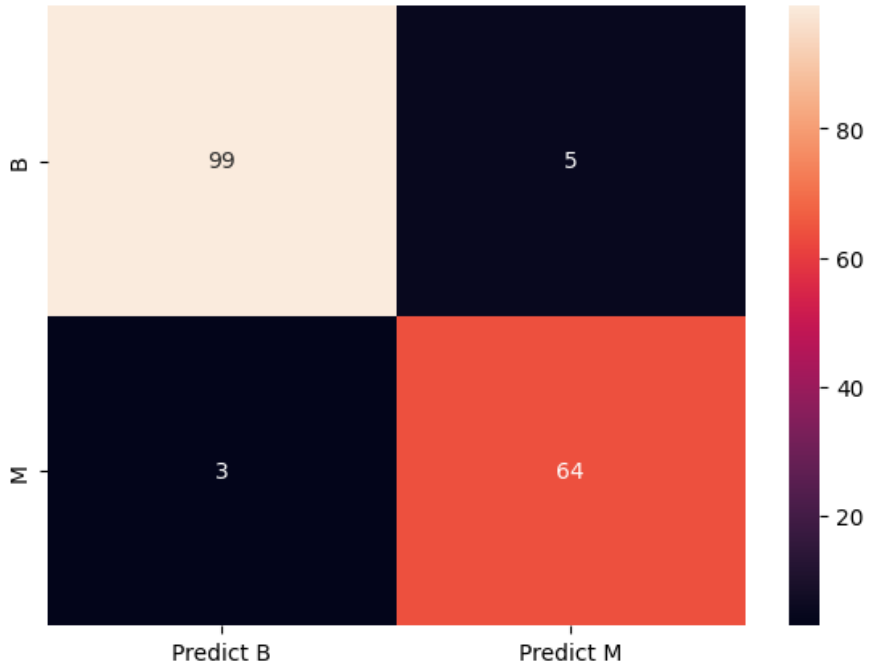

Figure 10: Confusion matrix for kNN Algorithm after LDA

Decision Tree Method

A Decision Tree is a flowchart-like structure in which an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition based on the attribute value. It partitions the tree recursively in a manner called recursive partitioning.

```
Decision Tree Algorithm Accuracy is 92%
.............................

Classification Report for Decision Tree Algorithm:
.............................
              precision    recall  f1-score   support

           B       0.97      0.90      0.94       104
           M       0.86      0.96      0.91        67

    accuracy                           0.92       171
   macro avg       0.92      0.93      0.92       171
weighted avg       0.93      0.92      0.92       171
```

Figure 11: Classification Report for Decision Tree Algorithm on original data

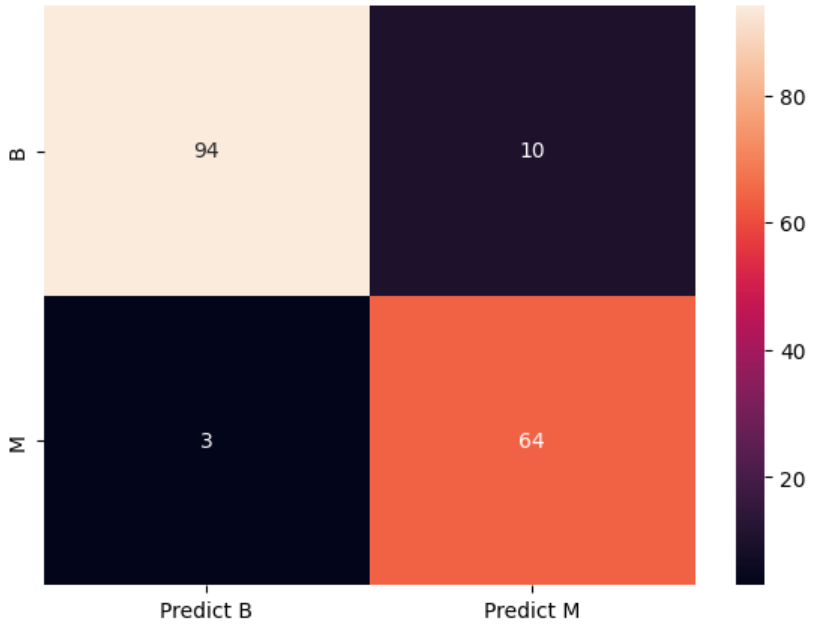

Figure 12: Confusion matrix for Decision Tree Algorithm on original data

Decision Tree Algorithm Accuracy After Applying PCA is 92%

.....

Classification Report for Decision Tree Algorithm After Applying PCA:

.....

	precision	recall	f1-score	support
B	0.94	0.92	0.93	104
M	0.88	0.91	0.90	67
accuracy			0.92	171
macro avg	0.91	0.92	0.91	171
weighted avg	0.92	0.92	0.92	171

Figure 13: Classification Report for Decision Tree Algorithm after PCA

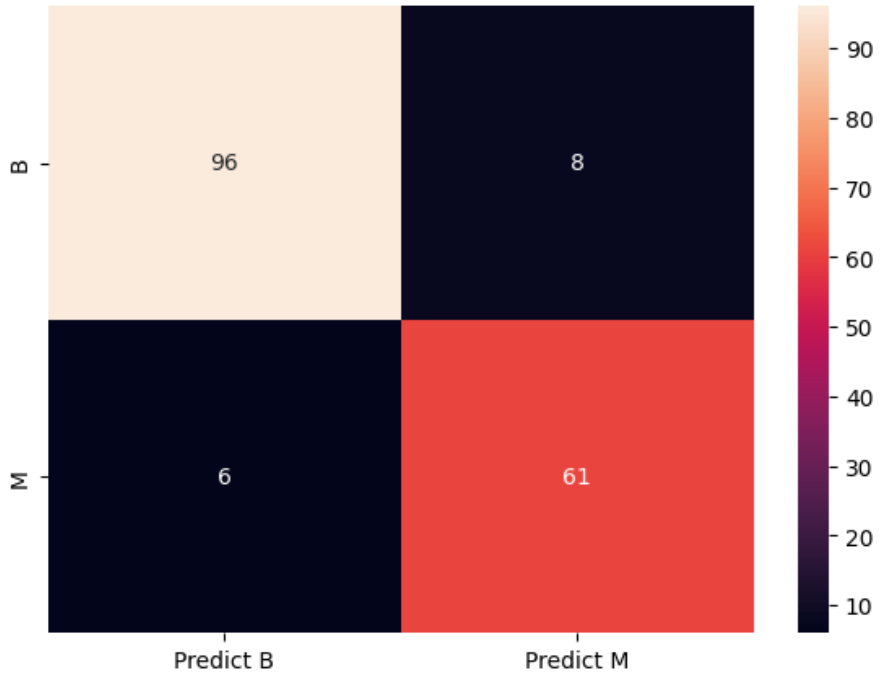

Figure 14: Confusion matrix for Decision Tree Algorithm after PCA

```
Decision Tree Algorithm Accuracy After Applying LDA is 94%
.....................................................................

Classification Report for Decision Tree Algorithm After Applying LDA:
.....................................................................
              precision    recall  f1-score   support

           B       0.96      0.93      0.95       104
           M       0.90      0.94      0.92        67

    accuracy                           0.94       171
   macro avg       0.93      0.94      0.93       171
weighted avg       0.94      0.94      0.94       171
```

Figure 15: Classification Report for Decision Tree Algorithm after LDA

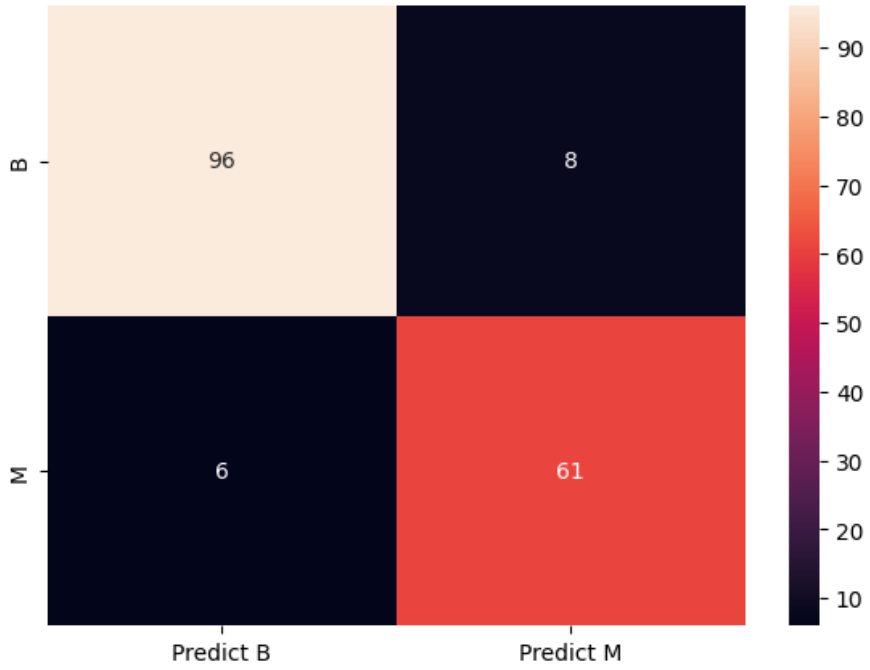

Figure 16: Confusion matrix for Decision Tree Algorithm after LDA

Random Forest Method

Random Forest is an ensemble method that creates a set of decision trees and merges them together to get a more accurate and stable prediction. It works by training multiple decision trees using different subsets of the training data, and then averaging the predictions made by each tree. This helps to reduce the variance of a single decision tree and can often result in a more accurate model.

The Random Forest approach achieved 96% accuracy on the original dataset. After PCA the accuracy was 95%, and after LDA, it was 94%. The other parameters and confusion matrices can be seen through to Figure 22.

```
Random Forest Algorithm Accuracy is 96%
.............................

Classification Report for Random Forest Algorithm:
.............................
              precision    recall  f1-score   support

           B       0.97      0.96      0.97       104
           M       0.94      0.96      0.95        67

    accuracy                           0.96       171
   macro avg       0.96      0.96      0.96       171
weighted avg       0.96      0.96      0.96       171
```

Figure 17: Classification Report for Random Forest Algorithm on original dataset

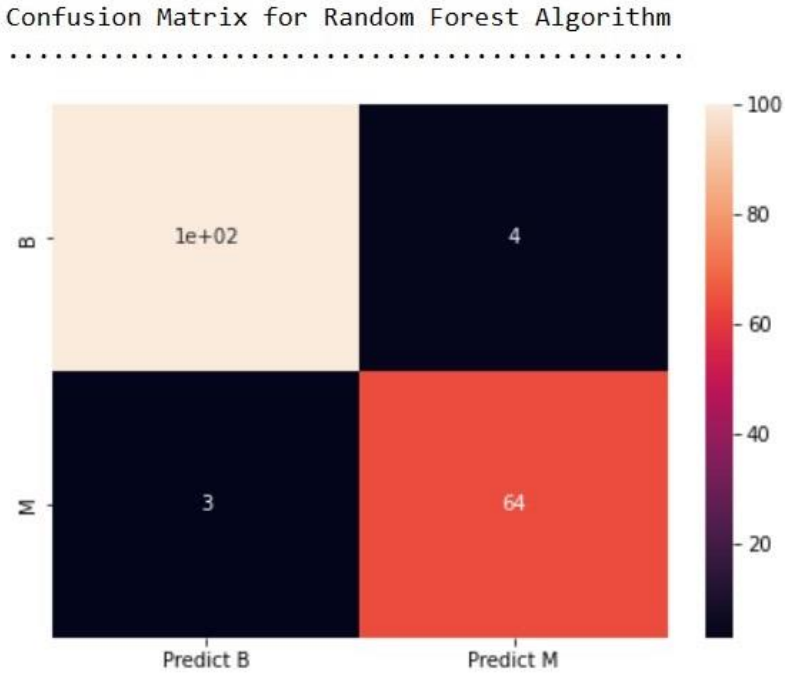

Figure 18: Confusion Matrix for Random Forest Algorithm on original data

Random Forest Algorithm Accuracy After Applying PCA is 96%

.....

Classification Report for Random Forest Algorithm After Applying PCA:

.....

	precision	recall	f1-score	support
B	0.95	0.98	0.97	104
M	0.97	0.93	0.95	67
accuracy			0.96	171
macro avg	0.96	0.95	0.96	171
weighted avg	0.96	0.96	0.96	171

Figure 19: Classification Report for Random Forest Algorithm after PCA

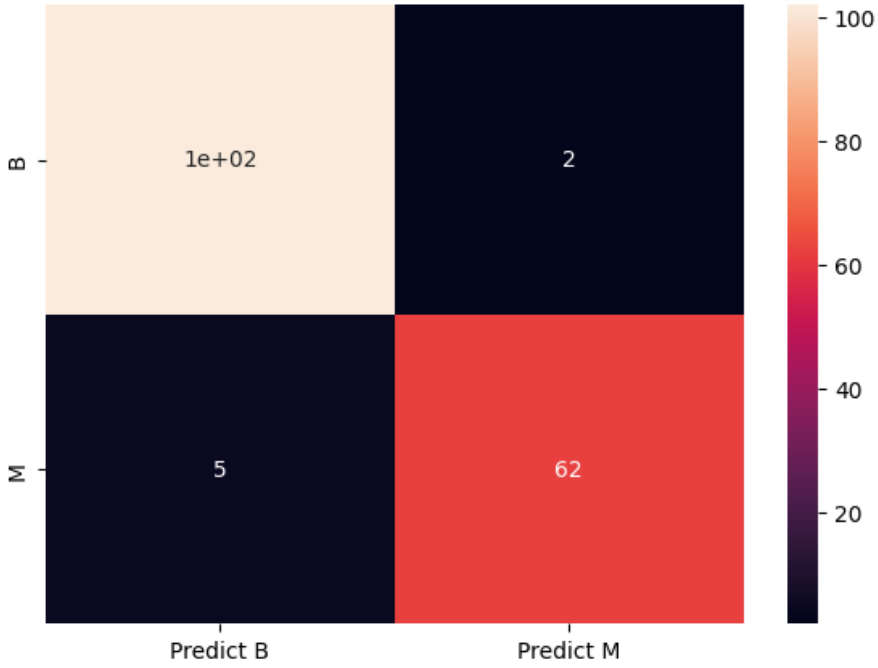

Figure 20: Confusion Matrix for Random Forest Algorithm After Applying PCA

```
Random Forest Algorithm Accuracy After Applying LDA is 94%
.....................................................................

Classification Report for Random Forest Algorithm After Applying LDA:
.....................................................................
              precision    recall  f1-score   support

           B       0.96      0.93      0.95       104
           M       0.90      0.94      0.92        67

    accuracy                           0.94       171
   macro avg       0.93      0.94      0.93       171
weighted avg       0.94      0.94      0.94       171
```

Figure 21: Classification Report for Random Forest Algorithm after LDA

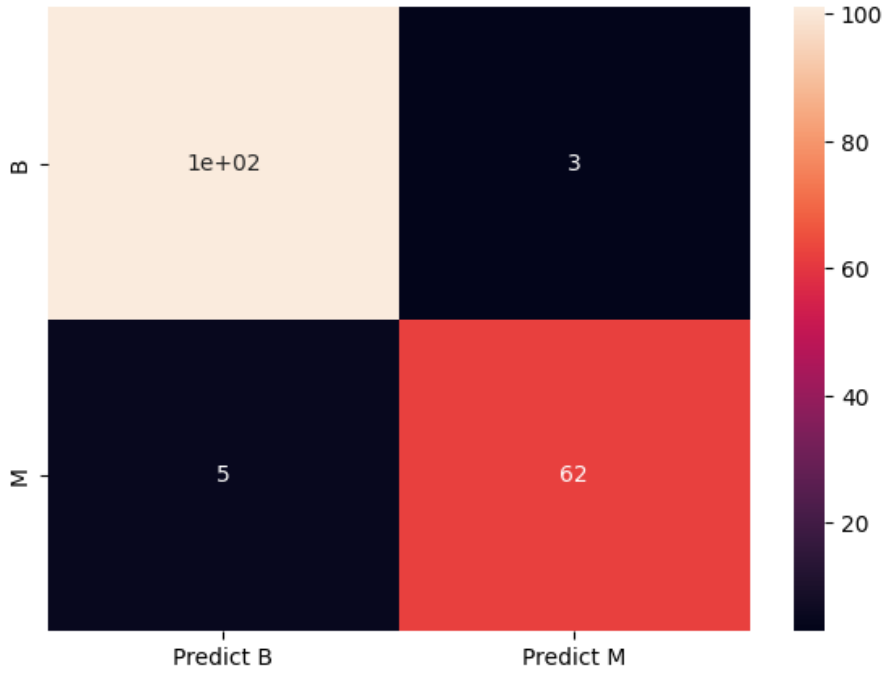

Figure 22: Confusion Matrix for Random Forest Algorithm After Applying LDA

MLP

A Multi-layer Perceptron (MLP) is a type of feedforward artificial neural network. It consists of an input layer, one or more hidden layers, and an output layer. Each layer is made up of a set of neurons, which use a non-linear activation function to produce an output. The goal of training an MLP is to adjust the weights and biases of the neurons so that the network can accurately map inputs to outputs.

```
MLP Algorithm Accuracy is 96%
..............................

Classification Report for MLP Algorithm:
..............................
              precision    recall  f1-score   support

           B       0.98      0.95      0.97       104
           M       0.93      0.97      0.95        67

    accuracy                           0.96       171
   macro avg       0.95      0.96      0.96       171
weighted avg       0.96      0.96      0.96       171
```

Figure 23: Classification Report for MLP Algorithm on original data

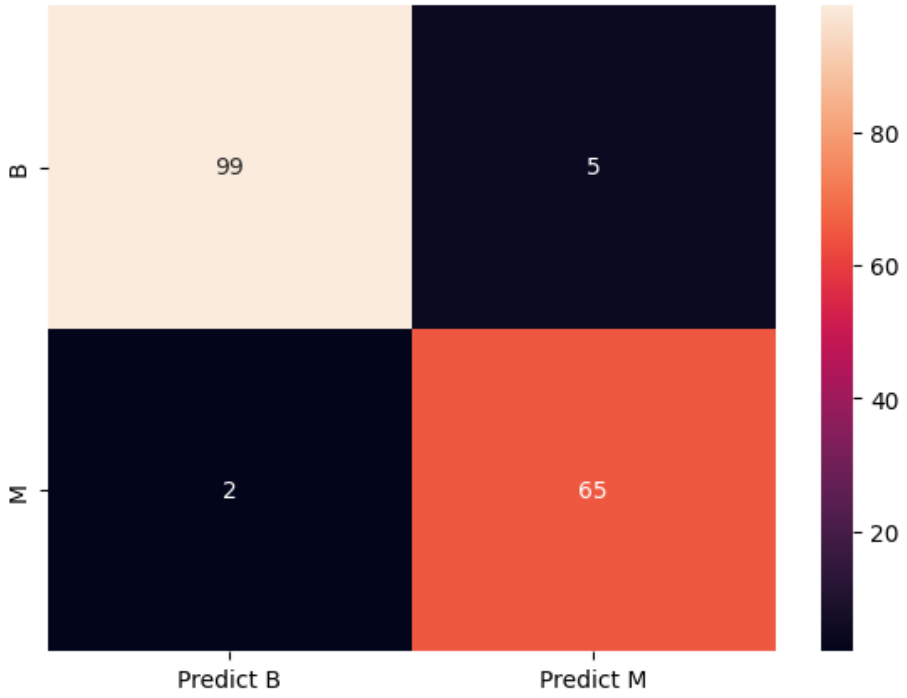

Figure 24: Classification Report for MLP Algorithm on original dataset

```
MLP Algorithm Accuracy After Applying PCA is 95%
.............................

Classification Report for MLP Algorithm After Applying PCA:
.............................
              precision    recall  f1-score   support

           B       0.97      0.94      0.96       104
           M       0.91      0.96      0.93        67

    accuracy                           0.95       171
   macro avg       0.94      0.95      0.95       171
weighted avg       0.95      0.95      0.95       171
```

Figure 25: Classification Report for MLP Algorithm after PCA

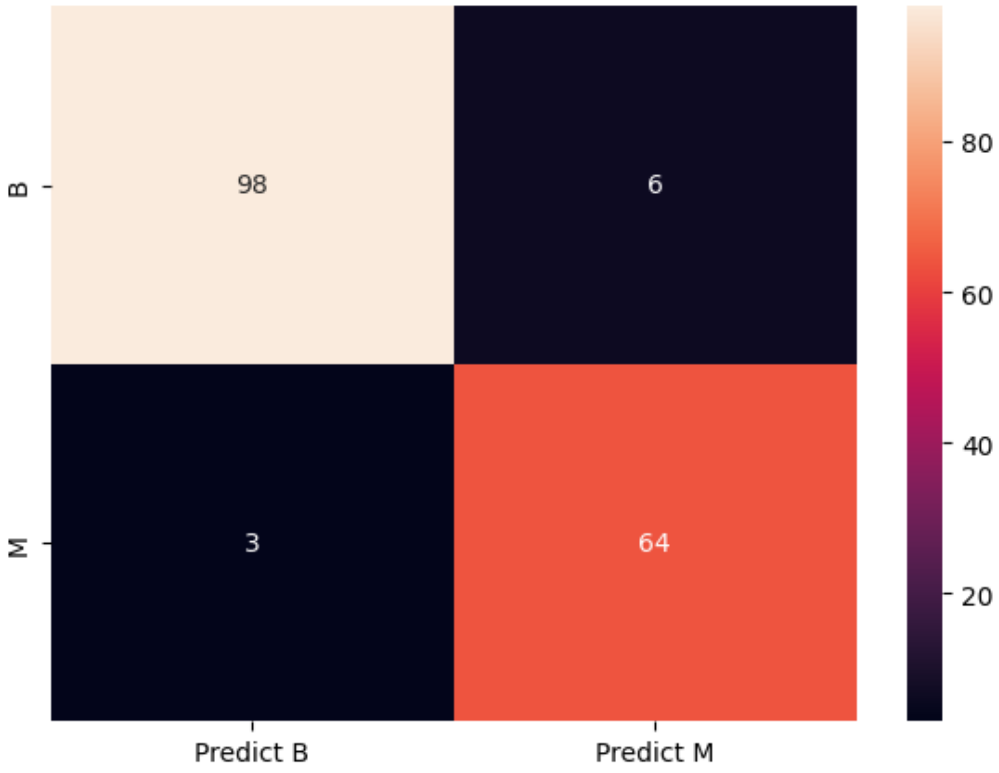

Figure 26: Confusion Matrix for MLP Algorithm after PCA

```
MLP Algorithm Accuracy After Applying LDA is 96%
.....................................................................

Classification Report for MLP Algorithm After Applying LDA:
.....................................................................
              precision    recall  f1-score   support

           B       0.95      0.98      0.97       104
           M       0.97      0.93      0.95        67

    accuracy                           0.96       171
   macro avg       0.96      0.95      0.96       171
weighted avg       0.96      0.96      0.96       171
```

Figure 27: Classification Report for MLP Algorithm after LDA

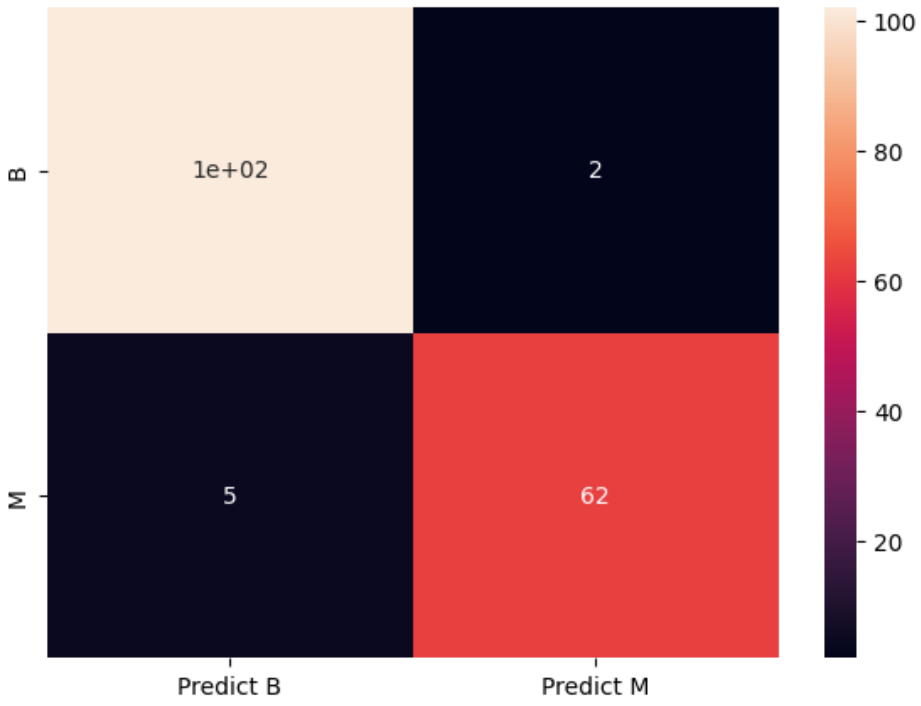

Figure 28: Confusion Matrix for MLP Algorithm after LDA

SVM

Support Vector Machine (SVM) is a linear model for classification and regression problems. It can handle linear and non-linear classification problems. The basic idea of SVM is to find a hyperplane that separates the data into classes. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points from either class. This distance is also known as the functional margin.

SVC is a branching method of SVM used for classification. This was the algorithm used in this assignment. The results can be seen through Figure 29 to Figure 34.

```
SVM Algorithm Accuracy is 98%
..............................

Classification Report for SVM Algorithm:
..............................
              precision    recall  f1-score   support

           B       0.99      0.97      0.98       104
           M       0.96      0.99      0.97        67

    accuracy                           0.98       171
   macro avg       0.97      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```

Figure 29: Classification Report for SVC Algorithm on original dataset

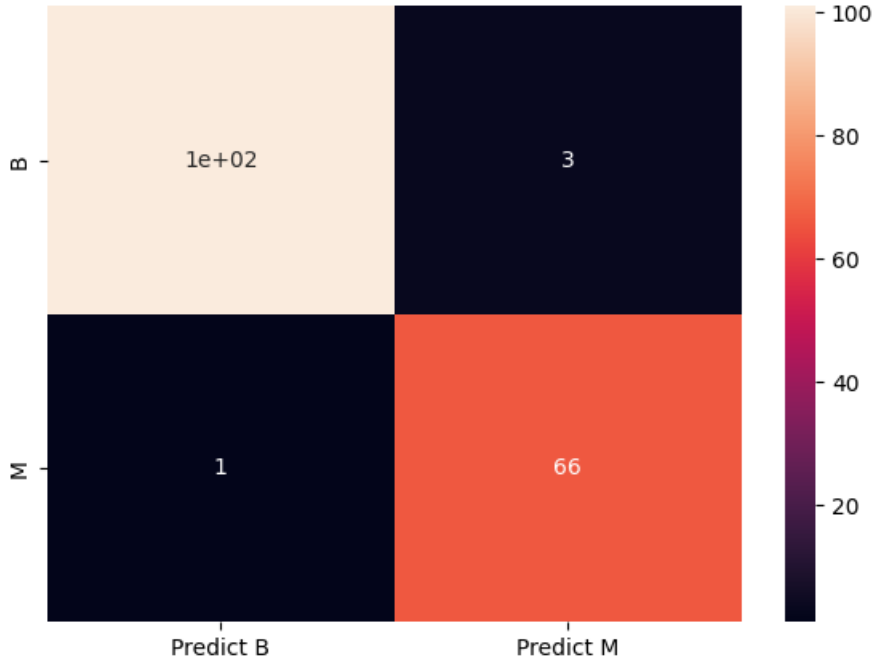

Figure 30: Confusion Matrix for SVC Algorithm on original data

SVC Algorithm Accuracy After Applying PCA is 96%

.....

Classification Report for SVC Algorithm After Applying PCA:

.....

	precision	recall	f1-score	support
B	0.98	0.96	0.97	104
M	0.94	0.97	0.96	67
accuracy			0.96	171
macro avg	0.96	0.97	0.96	171
weighted avg	0.97	0.96	0.96	171

Figure 31: Classification Report for SVC Algorithm After Applying PCA

Confusion Matrix for SVC Algorithm After Applying PCA

.....

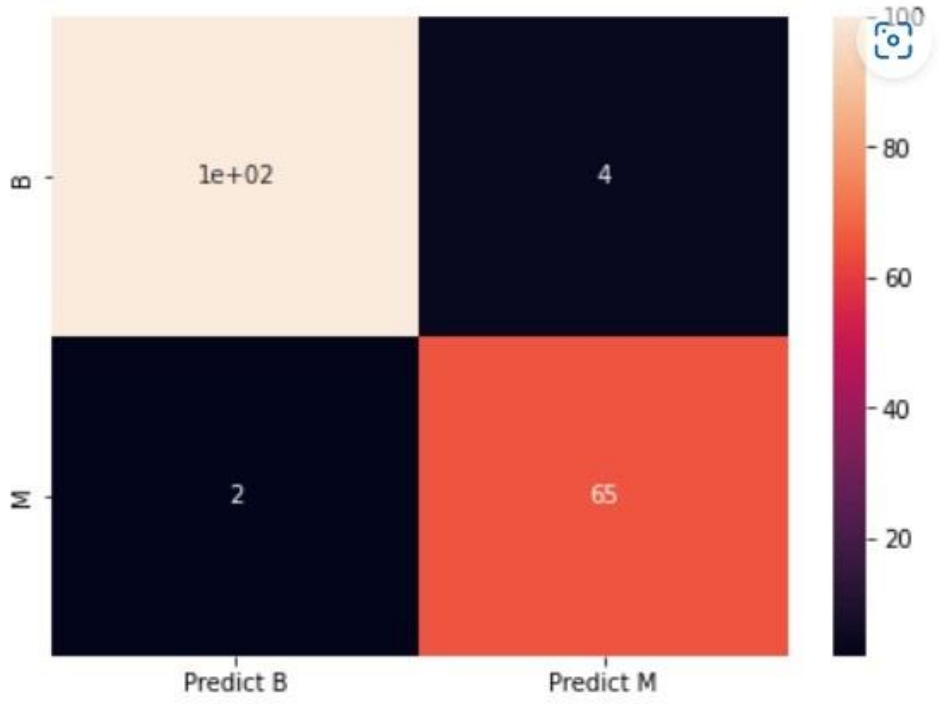

Figure 32: Confusion Matrix for SVC Algorithm After Applying PCA

```
SVM Algorithm Accuracy After Applying LDA is 97%
.....................................................................

Classification Report for SVM Algorithm After Applying LDA:
.....................................................................
              precision    recall  f1-score   support

           B       0.95      1.00      0.98       104
           M       1.00      0.93      0.96        67

    accuracy                           0.97       171
   macro avg       0.98      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171
```

Figure 33: Classification Report for SVC Algorithm After Applying LDA

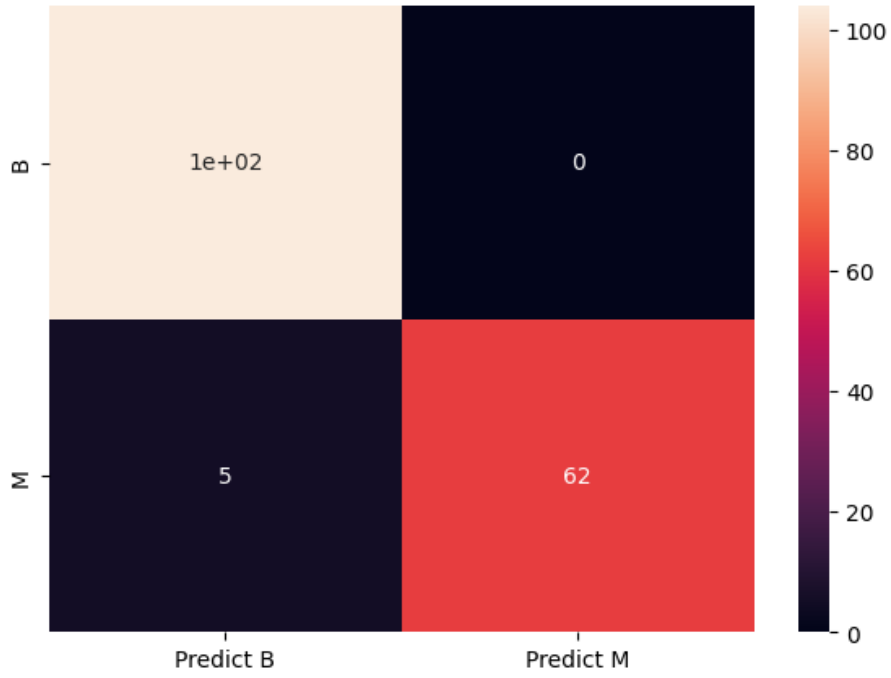

Figure 34: Confusion Matrix for SVC Algorithm After Applying LDA

Accuracy Results

Table 2 shows how the accuracy for each model varied when using dimensionality reduction methods (PCA, and LDA), compared to when they were not used.

Table 2: Models accuracies for original data, PCA, and LDA.

Model	Original data	PCA	LDA
kNN	98%	97%	95%
Decision Tree	92%	94%	94%
Random Forest	95%	96%	94%
MLP	96%	95%	96%
SVM	98%	96%	97%

Conclusion

In this assignment, the Breast Cancer Wisconsin (Diagnostic) Dataset was analyzed using two different dimensionality reduction techniques, PCA and LDA, as well as machine learning models, such as kNN, Decision Tree, Random Forest, MLP, and SVM. The data was pre-processed by removing the ID column and standardizing the remaining attributes.

The results of the analysis showed that PCA and LDA both maintained similar accuracy of the machine learning models when compared to using the original dataset. PCA had a similar performance impact compared to LDA. The kNN and SVM models had the highest accuracy of 98% on the original dataset and still the highest performance after applying PCA and LDA according to Table 2. The results of this analysis demonstrate the effectiveness of dimensionality reduction techniques in significantly decreasing the complexity of the problem while not impacting much the performance of machine learning models on this dataset.