Breast Cancer Prediction Using Different Machine Learning Models

by

Khandker Al- Muhaimin
14101022
Tahsan Mahmud
14101224
Sudeepta Acharya
14101032
Ashiqul Islam
13301010

A thesis paper submitted to the Department of Computer Science and Engineering with total fulfillment of the requirements for the degree of B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
August 2019

1.1 Motivation

The sole motive to choose this topic was that a friend said about his cousin's mother was ill because of this disease. We found it very heart touching situation for our friend as well as us. We also research about this topic. Ultimately we fi out it is a fatal and common disease in today's era. Ninety-eight women ascertained from excessive-risk breast/ovarian melanoma clinics with breast cancer reporting at the least one other primary melanoma in themselves or in a relative with breast cancer were in comparison with 99 females with breast cancer who mentioned a loved ones history of breast melanoma only [2], [13].

1.2 Objective

The main goal of this paper is to analyze diffe r e n t algorithms and produce better result in the fi H of detecting breast cancer. We wanted to establish a comparative study that can help people to reduce the death of breast cancer patients through a time saving better clinical treatment and early awareness with an accurate, fast prediction. Algorithms are tested with regard to sensitivity, accuracy, time complexity, precision. As it is so vital issue in our current era, we eagerly want to study it more for better and reliable result

1.3 Thesis Orientation

The following chapter is concerning the related works carried out previous in the same subject with the aid of exclusive researchers and the basics of computing algorithm called machine learning. The proposed model is mentioned within the 3rd chapter and its sub-sectors provide an explanation for the implementation. The visualization of the dataset, data preprocessing, fundamental aspect of PCA, train-test split and some knowledge about used algorithms. The additional section includes the efficiency of performance metrics and the results of our exams. Chapter 4 concludes narrating the entire issues and comparison of the outcome of the one of a kind unit. Eventually, chapters end with paper entails few future effects and new views.

Chapter 2

Literature Review

Cancer is not a completely immune able disease. But some of the cancer has better treatment now and patients might get totally fi if properly treat. Cancer treatments are being frequently developed. progressively simpler and better-targeted treatments are offered. As treatment has changed progressively, the outcomes have developed significantly. The main kinds of cancer treatment are cancer surgery (surgical treatment), radiotherapy, chemo therapy and hormone therapy. Today numerous immunological therapies and ostensible good medicine delivery (or targeted medicine delivery) are used also. There is a spread of different cancer medicine available. They are mainly utilized in combination. Breast cancer is one of the most treatable cancers. Today there's an awesome menu of treatment selections that fight the complicated mixture of cells in every individual cancer.

2.1 Related Works

There are various modern facilities available to accurately predict breast cancer. Some of the works done on this fi are, Supervised hazard Predictor of Breast Cancer in the basis of Intrinsic Sub types [11], [8], [15], [16], [12]. It is a supervised work done on the basis of breast cancer risk. Improvement on today's standards for breast cancer prognosis and determination of chemotherapy profited by evolving a risk architecture is the main goal of this research. Another work on this particular criteria in cancer prognosis and determination done in the research [19]. They had used a variety of techniques, including Artificial Neural Networks (ANNs), Bayesian Networks (BNs), Support Vector Machines (SVMs) and Decision Trees (DTs). These predictive models are very beneficial for cancer research as they often produce effective and accurate decision making. The are many other work on breast cancer prediction [4], [15], [18], [7], [10], [6] using diff t algorithms. Deep Learning also a way to predict breast cancer that was used in another work [22].

2.2 Machine Learning(Applied in study)

Machine Learning is a part of scientific experimental method that helps a computer to find out and act like humans, and developing their learning over time in sovereign approach. It will feed data and information within the type of observations and real-world interactions. In our work we have a tendency to use a supervised machine learning methodology. Supervised Learning: Two strategies of classification, logistic regression and SVMs and Supervised Learning with Non-parametric learners: k-nearest neighbors, decision trees, random forests. Additionally, we used Gaussian process and Ada boost tree.

Chapter 3

Proposed Model

In our paper, we applied 9 algorithms on Wisconsin Breast Cancer Data Set (WBDC) [11] which consist of 569 patients for further accurate and quick detection of breast cancer. Performance metrics were used to compare the results of each algorithm with one another. The workflow is presented bellow in fig. 3.1

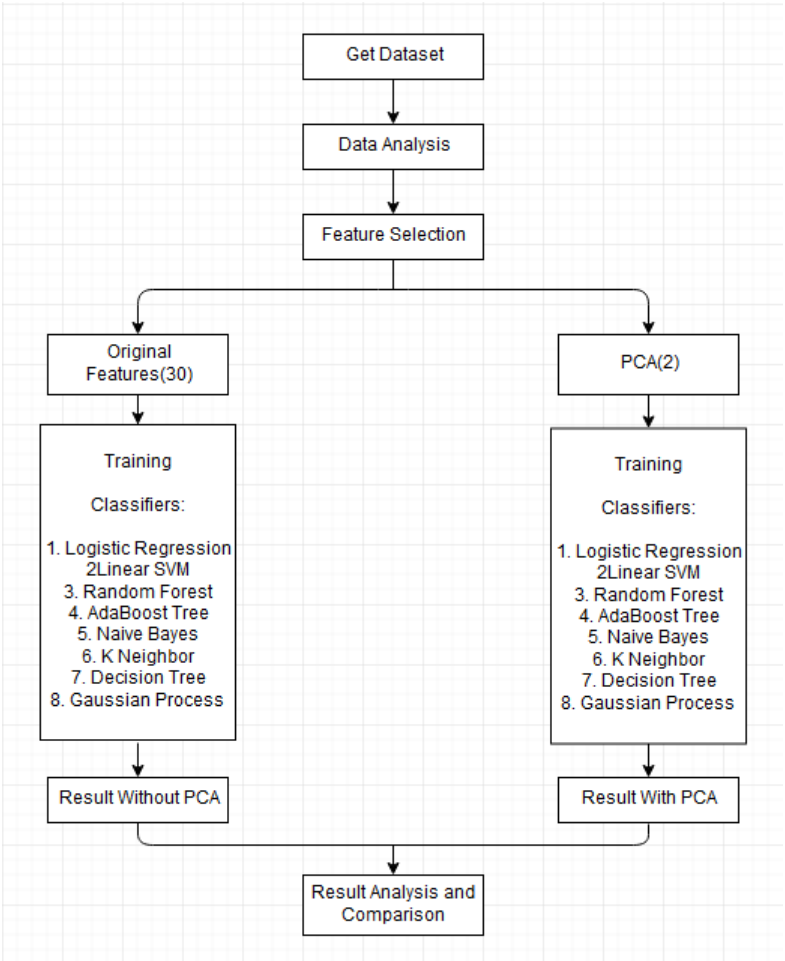


Figure 3.1: Proposed Model's Architecture

3.1 Dataset

The dataset had picked for our tests from Wisconsin Breast Cancer (Diagnostic) Data Set (WBCD). The dataset is openly accessible and well decorated. We can get it from the presumed storage which is UCI-Repository. Dr.William H.Wolberg, specialist at the University Of Wisconsin hospital at Madison, USA created WBCD. Dr. Wolfberg utilized Xcyt to investigate liquids tests taken from patients with strong bosom masses [14]. Xcyt is a simple to-utilize graphical PC program which is prepared to play out the examination of cytological highlights in light of computerized fi The dataset includes 569 examples and 32 characteristics of visually measured atomic features from a digitized picture of a fi needle aspirate (FNA) of a breast mass. FNA is a fi needle aspiration biopsy of a thyroid nodule is a simple and safe procedure performed in the doctor's office. Using this process doctors collects a sample to make a diagnosis or predicting disease such as breast cancer. Among the 569 examples, the class dispersion is 212 destructive tumors (dangerous) also, other 357 non-destructive tumors (amiable). Ten highlights are processed from every single cell in the example which is as per the following: 1. Radius (mean of distances from center to points on the perimeter) 2. Texture (standard deviation of gray-scale values) 3. Perimeter 4. Area 5. Smoothness (local variation in radius lengths) 6. Compactness (perimeter / area-1.0). 7. Concavity (severity of concave portions of the contour) 8. Concave points (number of concave portions of the contour) 9. Symmetry 10. Fractal dimension ("coastline approximation"- 1) The mean, standard error and "worst" or largest (mean of the three largest values) of all the features were registered for each picture, bringing about 30 features. All element esteems are recoded with four noteworthy digits. There are no missing values in dataset. There is mixture of both numerical and categorical features in the dataset. M = malignant or B = benign. The remainders of the features are numerical.

There are 32 attributes or column in our dataset. They behold diff t information of several data. The attributes are discussing bellow.

1. Id: Showing ID number.
2. Diagnosis: The diagnosis of breast tissues (M = malignant, B = benign).
3. Radius_mean: mean of distances from center to points on the perimeter.
4. Texture_mean: standard deviation of gray-scale values.
5. Perimeter_mean: mean size of the core tumor.
6. Area_mean: measure the area.
7. Smoothness_mean: mean of local variation in radius lengths.
8. Compactness_mean: mean of perimeter / area - 1.0.
9. Concavity_mean: mean of severity of concave portions of the contour.
10. Concave points_mean: mean for number of concave portions of the contour.
11. Symmetry_mean: exactly like another when move in particular way.
12. Fractal_dimension_mean: mean for "coastline approximation" – 1.
13. Radius_se: standard error for the mean of distances from center to points on the perimeter.
14. Texture_se: standard error for standard deviation of gray-scale values.
15. Perimeter_se

16. Area_se
17. Smoothness_se: standard error for local variation in radius lengths.
18. Compactness_se: standard error for perimeter / area - 1.0.
19. Concavity_se: standard error for severity of concave portions of the contour.
20. Concave points_se: standard error for number of concave portions of the contour.
21. Symmetry_se
22. Fractal_dimension_se: standard error for "coastline approximation" – 1.
23. Radius_worst: "worst" or largest mean value for mean of distances from center to points on the perimeter.
24. Texture_worst: "worst" or largest mean value for standard deviation of gray-scale values.
25. Perimeter_worst
26. Area_worst
27. Smoothness_worst: "worst" or largest mean value for local variation in radius lengths.
28. Compactness_worst: "worst" or largest mean value for perimeter / area - 1.0.
29. Concavity_worst: "worst" or largest mean value for severity of concave portions of the contour.
30. Concave points_worst: "worst" or largest mean value for number of concave portions of the contour.
31. Symmetry_worst
32. Fractal_dimension_worst: "worst" or largest mean value for "coastline approximation" – 1.

3.2 Data Visualization

The mission of this paper is to analyze different algorithms and produce better result in the field of detecting breast cancer. We wanted to establish a comparative study that can help people to reduce the death of breast cancer patients through a time saving better clinical treatment and early awareness with an accurate, fast prediction. Algorithms are tested with regard to sensitivity, accuracy, time complexity, precision. As it is so vital issue in our current era, we eagerly want to study it more for better and reliable result.

3.2.1 Histogram

A bar chart may be a form of graph that's wide utilized in arithmetic, particularly in statistics. The bar chart represents the frequency of incidence of specific phenomena that lie at intervals a particular vary of values, that are organized in consecutive and field intervals. The frequency of the info incidence is pictured by a bar thence it's substantially sort of a chart. Figure 3.2 shows the class distribution of diagnosed malignant (M) and benign (B) tumors. Here we got 212 dangerous malignant tumors of around 38% and benign tumors 357 which is remaining 62% of the prescient class. The core highlights plotted against finding as observed on field 3.2 from that we can see the mean estimations of cell span, edge, territory, smallness, concavity and concave can be utilized in classification of the cancer. Bigger estimations of these parameters tend to demonstrate a relationship link with malignant tumors. The mean estimations of smoothness, fractal or symmetry measurement does not demonstrate a specific inclination of one finding over one another.

```
In [9]: ax = sns.countplot(data['diagnosis'], label='Count')
        B,M = data['diagnosis'].value_counts()
        print ("Bening", B)
        print ("Malignant", M)
```

```
Bening 357
Malignant 212
```

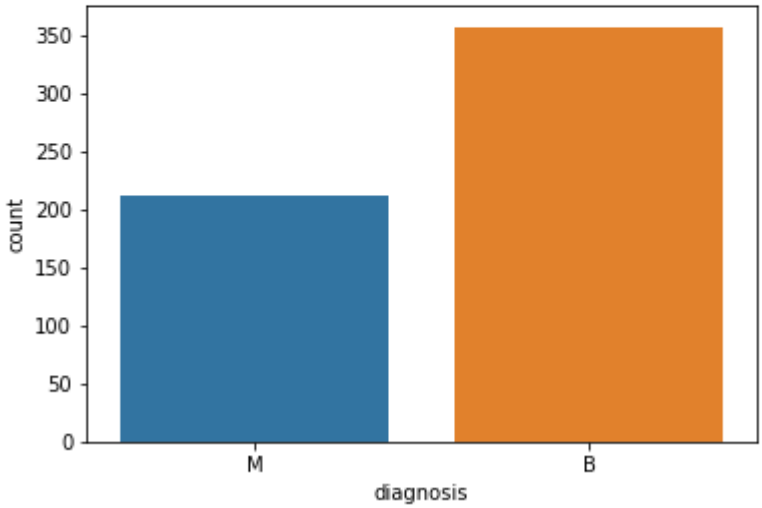


Figure 3.2: Class Distribution

3.2.2 Swarm Plot

The plot is similar to a strip plot with jitter, but the graphical presentation is more elegant. Swarm plot spreads out the points to avoid overlap and provides a nice visual look of the data. So it is more effective than strip plots for a well textured overview of the data.

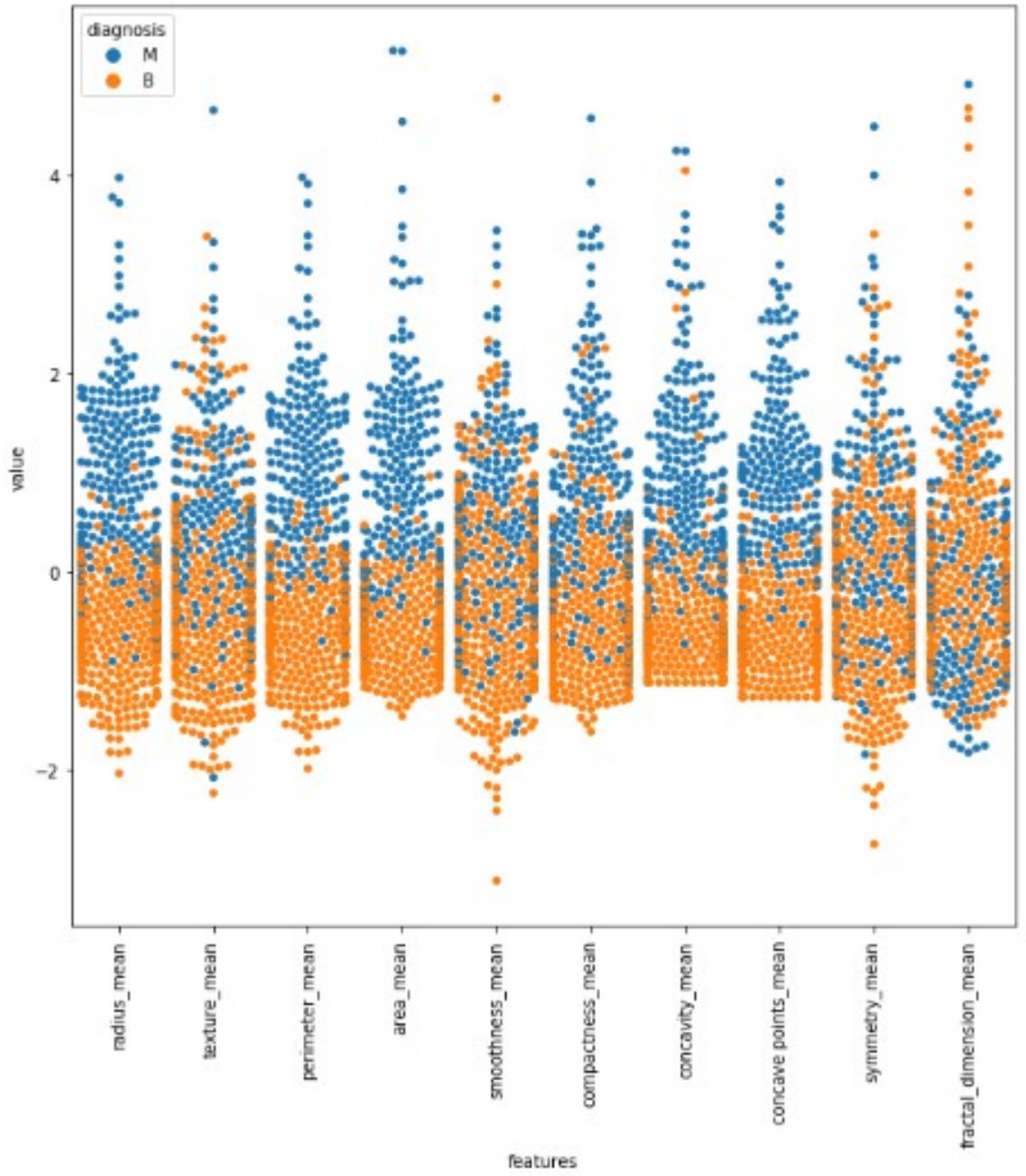


Figure 3.3: Swarm plot

3.2.3 Heatmap

A heat map is being called a two-dimensional illustration of data within which values are represented by colors. A straightforward heat map provides a direct visual outline of data. Additionally, elaborate heat maps enable the viewer to grasp advanced info sets.

There will be several ways to show heat maps. However, all of them share one matter in common. They use color to speak relationships between info values that might be a lot tougher to grasp if presented numerically in a very complex way.

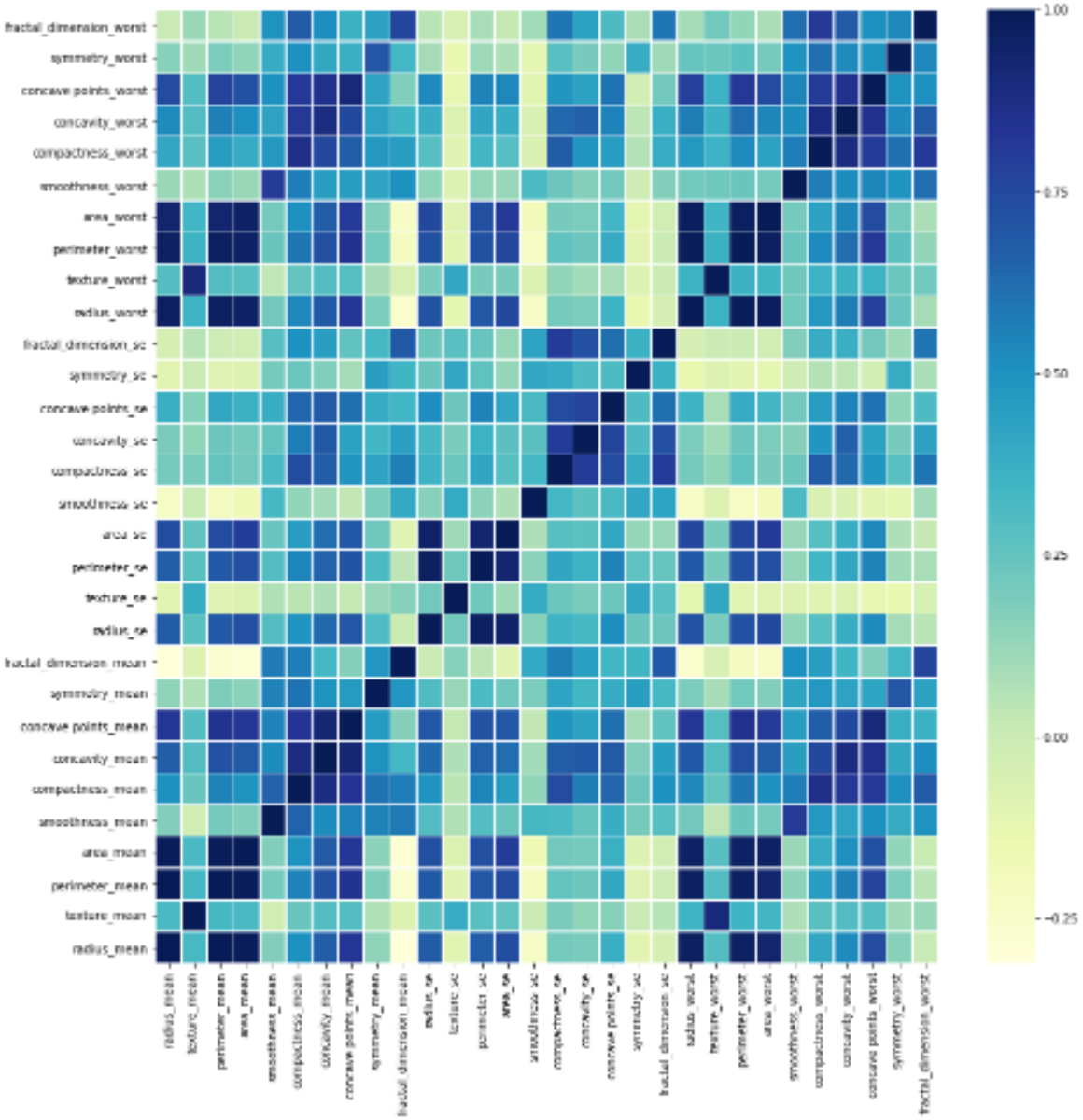


Figure 3.4: Correlation Between All Variables

3.3 Data Preprocessing

3.3.1 Categorical Variable Conversion

Any information attribution that is categorical in nature represents distinct values that belong to a particular finite set of classes or categories. These also are usually called categories or labels within the context of attributes or variables that are to be expected by a model (popularly known as response variables). These distinct values is text or numeric in nature (or even unstructured information like images!). There are 2 major categories of categorical information, nominal and ordinal. The dataset has numerical and categorical components combined. So there is two type of concept or symbol for two types of cancer. M stands for malignant and B stands for benign. All algorithms work better with numerical data. So we had to use "label encoder" was used to reshape non-numerical data to numerical value.

3.3.2 Feature Scaling

When operating with a learning model, it's vital to scale the options to a variety that's targeted around zero. this can be done in order that the variance of the options is within the same vary. If a feature's variance is orders of magnitude quite the variance of different options, that individual feature would possibly dominate different options within the dataset, that isn't one thing we wish happening in our model. The aim here is to attain mathematician with zero mean and unit variance. There are many ways of doing this, the 2 hottest are standardization and normalization. Standardization replaces the values by their Z scores.

$$X' = \frac{x - \bar{x}}{\sigma} \tag{3.1}$$

Standardization and Mean Normalization are being used for algorithms that presume zero centric data for example Principal Component Analysis(PCA).

3.3.3 Principal Component Analysis (PCA)

The principal component analysis is an approach to correlational analysis that considers the full variance within the knowledge, that is that the commonest correlational analysis and transforms the first variables into a smaller set of linear mixtures. The diagonal of the matrix consists of unities and therefore the full variance is brought into the issue matrix. The term issue matrix is that the matrix that contains the factor loadings of all the variables on all the factors extracted. The term, "factor loadings" is the simple correlations between the factors and the variables. Principal component analysis (PCA) could be a technique used for the identification of a smaller range of unrelated variables referred to as principal elements from a bigger set of information. The technique is wide wont to emphasize variation and capture sturdy patterns in an exceedingly knowledge set. Principal element analysis is suggested once the researcher's primary concern is to work out the minimum range of things that may account for the utmost variance within the knowledge in use in the specific statistical procedure, like in city studies. The eigenvalues talk over with the whole variance explained by every issue. the quality deviation measures the variability of the information. The task of principal part analysis is to spot the patterns within the data and to direct the info by lightness

their similarities and variations.

3.3.4 Train-Test Split

Training and check knowledge are common for supervised learning algorithms. Given a dataset, its split into coaching set and check set. within the world we've all types of knowledge like money data or client data. In Machine Learning, this is applicable to supervised learning algorithms. The coaching set contains a well-known output and therefore the model learns on this knowledge so as to be generalized to alternative data.

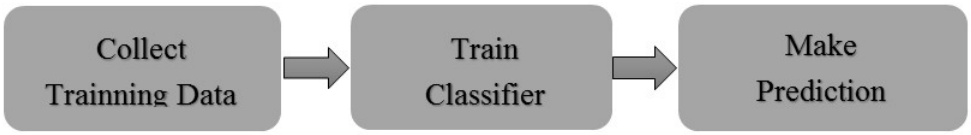


Figure 3.5: Class Distribution

A rule ought to create new predictions supported new knowledge. The observations within the coaching set to create the expertise that the rule uses to find out. In supervised learning issues, every observation consists of associate degree determined output variable and one or additional observed input variables. The check set could be a set of observations wont to measure the performance of the model exploitation some performance metrics. it's necessary that no observations from the coaching set are enclosed within the check set. If the check set will contain examples from the coaching set, it'll be tough to assess whether or not the rule has learned to generalize from the coaching set or has merely memorized it. A program that generalizes well are going to be ready to effectively perform a task with new knowledge. In distinction, a program that memorizes the coaching knowledge by learning an excessively complicated model might predict the worth of the response variable for the training set accurately however can fail to predict the value of the response variable for brand spanking new examples. Memorizing the coaching set is termed overfitting. A program that memorizes its observations might not perform its task well, because it might memories relations and structures that are noise or coincidence. reconciliation learning and generalization, or overfitting and under fitting could be a downside common to several machine learning algorithms. Regularization is also applied to several models to scale back overfitting. additionally, to the coaching and check knowledge, the third set of observations, known as a validation or hold-out set, is typically needed. The validation set is employed to tune variables known as hyper parameters, that management however the model is learned. The program continues to be evaluated on the check set to produce associate degree estimate of its performance within the real world; its performance on the validation set mustn't be used as an estimate of the model's real-world performance since the program has been tuned specifically to the validation knowledge. The train/test split was enforced through the train check split category of scikit-learn's model choice package into a 70:30 ratios with seventieth going in the coaching set and also the remainder of the package to the test set that is found to be ideal [23].

3.4 Algorithms

The architecture works along binary classification of tagged info. There are total 9 algorithms implemented to find out best one. The algorithms that had selected based on their ability of prediction. The machine learning algorithms used for this particular problem are Random Forest (RF). Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), Gaussian processes, Naive Bayes, Adaboost, Decision Tree. The output values of the algorithms were properly equated to decide the most accurate prediction of the disease.

3.4.1 Logistic Regression

Logistic Regression is a useful and helpful Machine Learning algorithms in the field of binary classification. It is an easy to use Algorithm that can be implemented as a performance baseline. It is easy to evaluate and it may do good enough in several problems. For that reason, its will be very wrathful if Machine Learning engineer is familiar to its theory. The building block theory of Logistic Regression can utilize deep learning to make neural networks. Logistic Regression for 2 features is:

$$log\frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x \tag{3.2}$$

here,
p(x) dependent variable
x independent variable
β_0 intercept
β_1 slope co-efficient.
Input (X) are added linearly using coefficient output to specify an output(Y). It's not similar to linear regression reason is the result generate a binary output (0 or 1) rather than a numeric output. As a result, the logistic function is relatively known as the sigmoid function which does the conversion. It is commonly used to do binary classification tasks and works more than fine when correlated characteristic are minimized.

3.4.2 Support Vector Machine

support vector machine algorithm helps to find a hyper plane in an n-dimensional space (n is the amount of attributes) that unambiguously categorize the data points. In order to split the two category of data points, there are several probable hyper planes that might be selected. Our goal is to define a plane that can give utmost margin, the utmost extent among data points of both classes. Uprising the margin extent will provide some extra support by which the time ahead data points can be categorized with more faith. Hyper planes are the selection area that facilitate to categorize the data points. Data points pass on both part of the hyper plane can be valued to transformed classes. The proportion of the hyper plane rely on the amount of attributes. If the amount of input value is 2, then the hyper plane will be a line. If the amount of input value is 3, then the hyper plane creates a two-dimensional plane. It will be tough to visualize when the amount of value outrun 3. Support vectors are data points that are nearer to the

hyper plane and sway the place and alignment of the hyper plane. utilizing these support vectors, we can limit up the margin of the classifier. Deleting the support vectors will replace the place of the hyper plane. We use these discussed matters in order to help build SVM.

3.4.3 Random Forest

Random Forest make a jungle in some manner and do it random. There is a straight forward communication among the amount of trees in the jungle and the output it can produce: the bigger the amount of trees, the further correct the result is. It may use in both categorize and regression tasks. Overfitting is a vital casualty that may create the output worse. In Random Forest algorithm, if there are plenty trees in the jungle, the classifier never over fit the structure. Also the classifier of Random Forest can manage missing output. Classifier also be modeled as categorical output.

3.4.4 AdaBoost

AdaBoost is small form of Adaptive Boosting. Ada Boosting was the first actual perfect boosting algorithm evolve for binary categorization. Additionally, it is the best entry point to understand boosting. Furthermore, now a days boosting tactic stand on AdaBoost, specially, stochastic gradient boosting machines. Commonly, AdaBoost work with small decision trees. Lately, the first tree will build as performance of the tree on every training case is used. By utilizing weight of it we can say how much attention the next tree will get. Each training instance get attention by doing this. So, training value that is difficult to determine is being set more weight and easy to predict instances are given less weight. Every one of component in the training dataset is weighted. The starting weight is given as:

$$weight(x_i) = \frac{1}{n} \tag{3.3}$$

Where x is the i-th instruct component and n is the amount of instruct instances. A faint classifier is set on the instruct info utilizing the weighted samples. Only binary classification enigma is validated. So every single judgment stump creates one final result on one input variable. Outputs will be a+1.0 / a-1.0 value for the starting or after that one's class value. The miss categorize scale is computed to get trained model. Mostly, er will be,

$$er = \frac{c - N}{N} \tag{3.4}$$

Where er is the miss categorize scale and c is the amount of training element guessed by the model. N is the combined amount of training element. A few advantages of AdaBoost are, speedy, not very confusing and not difficult to program, no parameters to tune (except T), fl - can combine with any learning algorithm.

3.4.5 Naive Bayes

Naive Bayes is very straightforward but strong algorithm used for prediction as well as classification. the other values of attributes. It is built upon Bayes probability theorem. It mainly used to make text categorization which includes top level volume training data sets. Some examples are spam strain, nostalgic analysis and categorize news clause. It is good for both simplicity and efficacy. Speedy to build structure and make guessing with Naive Bayes algorithm. Naive Bayes is the best and trusty algorithm to be decide for cracking text classification problem.
The equation of Bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.5}$$

Here,

P(A|B) posterior probability,

P(B|A) likelihood,
P(A) class prior probability
and P(B) is predictor prior probability.
A benefit of Naive Bayes classifier is, comparably relaxed algorithm to use and memorize. Faster to guess classes through this algorithm than other classification algorithms. This algorithm can easily be trained with a small attribute set. Naive Bayes algorithm can get itself educated with probability of an element with unique features which can be found in a specific group. Finally, can titled it as a probabilistic classifier.

3.4.6 K-Nearest Neighbors

The k-nearest neighbors (KNN) is a very easy, efficient-to-implement supervised algorithm that can utilize as a way to solve both classification and regression problems. An algorithm called supervised when a machine learning algorithm depends on tagged input value to educate itself a function that make an appropriate result when insert a new untagged value. 1. The KNN Algorithm Load the input value 2. Start by putting K to a selected amount of neighbors 3. Equate the space among the query and current example in the data 3.1. 3.2 plus space in between and the index of the particular value to a sequential collection. 4. Sort the sequential set of spaces and indices from smallest to largest by the spaces. 5. Chose the 1st K entries from the sorted set 6. Take in the tags of the chosen K entries 7. If regression, send the mean of the K tags 8. If classification, send the mode of the K tags.

3.4.7 Decision Tree

A decision tree is like a flow of couple of chart structure in which an inner node present variables. Here branch specify a decision rule, and every one of leaf node specify result. The upper node inside decision tree is called root node. It gains knowledge of division of the attribute value. It makes division of tree in reverse way named recursive partitioning. This chart of flow structure helps us in judgement taking. It's view shape like a flowchart diagram which comfortably copy the human like dreaming. So for this reason decision trees are not so complex to comprehend and believe. Decision Tree inspection is a normal, guess worthy structure tool that has software spread in various area. Mostly, decision trees are created through a mathematical approach that define path to divide a data set with respect on various rules. It is a broadly used and very realistic methods for supervised learning. Decision Tree is a non-parametric supervised learning process which utilize for both classification and regression sectors. The approach is to make a structure that define the result of a target variable through processing simple decision regulation permitted from the data values. The decision regulations are normally in shape of if or else statements. The deeper the tree, the more complicated the regulations and fitting the structure. In general idea work after every decision tree algorithm is given aside: (a) Select the best variable using variable choosing techniques to split the records.
(b) Make that variable a decision node and split the dataset into little subsets.
(c) Continue tree making by reprise this process recursively for every child as long as one of the term will match: i. All the tuples from to the exact variable value. ii. No be left attributes. iii. No left out instances.

3.4.8 Gaussian Process

Gaussian processes are a powerful algorithm for both regression and classification. A Gaussian process is a probability distribution over possible functions. Gaussian processes are parametric: they are principled on the statement that underlying info is generally circulated and normally jointly distributed. This is a very strong assumption, of course, which will rarely occur fully in practice. The closer to these assumptions a dataset holds, the better the performance GP will have A Gaussian process state a circulation over functions, p(f), Here f is a function Mapping for a few input space X to R
f: $X \rightarrow R$
Here f is an infinite-dimensional amount
Now
for an n-dimensional vector of function values assessed
at n points, Note F is a arbitrary variable.
p(f) is a Gaussian process if for any finite subset,
the unique dispersion of that finite subset p(f) has a multivariate Gaussian distribution. The greatest practical advantage of Gaussian processes is that they can give a reliable estimate of their own uncertainty. It has been flourishing in both supervised and unsupervised machine learning assignment. Still their equational obsession has restricted to realistic applications.

Chapter 4

Result and Discussion

After the successful implementation of machine learning model it is very important to measure that how effective the model is and how the model performs on the dataset that we have chosen for our research. We noticed that algorithms performance is different. For with PCA it's something and for without PCA its's something else. In our thesis, various execution parameter to figure out which machine learning algorithm will be the best option for the detection of breast cancer. From WBCD dataset, we have taken 80% of the data for training purpose and rest 20% was used for testing. Both random state and 10 old cross validation were deployed to find out the best possible result and the more satisfying result.

4.1 Performance Metrics

This paper mainly focuses on the comparison of different classification problems and from that classified performance matrix basically focus on classification. In order to detect of breast cancer, the tagged variable 1(Malignant) means it is a positive instance and that clearly refers the patient is having a breast cancer. On the contrary, 0(Benign) means it is a negative instance and that indicates the patient having no breast cancer.

4.1.1 Confusion Matrix

Confusion matrix is always recognized as easily understandable matrix while it is arguably the most common matrix to determine the precision and rightness of a prototype. A confusion matrix is a summary of prediction results on a classification problem. Confusion matrix structure accommodates the users to conceptualize the effectiveness of confusion matrix. Here instances are actual classes which are represented by each row of the matrix in fi 4.1. In contrast, a single column adheres the exemplification in a pre-defined class or in the opposite way.

	Predictive Negative	Predictive Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Figure 4.1: Confusion Matrix

Important terms of a Confusion Matrix and their related discussion are given below for better understanding of a reader:

True Positives (TP):

Basically, in this situation both the anticipate class and the real class is correct (true) (1), i.e., while a classifier predicts the sufferer obtain a complication of breast cancer and actually the patient has a breast cancer. So, the classifier is predicting correct decision in this situation.

False Positives (FP):

This situation refers the classifiers anticipate class is correct(true) (1) but the actual class is wrong(false) (0), i.e., while a classifier predicts that a sufferer has a complication of breast cancer but actually the sufferer has no breast cancer. So, the classifier is unable to predict correct decision for that case.

True Negatives (TN):

True Negatives state that the anticipate class and the real class are false (0), i.e., while a classifier predicts that a sufferer has a no complication of breast cancer and actually the sufferer does not contain breast cancer. Therefore, this classifier is predicting the correct decision.

False Negatives (FN):

Essentially, this situation refers the classifiers anticipate class is false (0) but the real class is correct(true) (1), i.e., while the classifier predicts that the sufferer has no complication of breast cancer but actually the patient has a breast cancer. So, the classifier is unable to predict correct decision similarly to False Positives.

Thus, the classifier's accuracy is higher when more TP and TN are found inside confusion matrix. Similarly, accuracy of a classifier

decreases when the amount of FP and the FN increases in a Confusion Matrix. So, the best situation would be when none of the FP and FN would be founded inside the model. If it happens, the model can give us the 100% accuracy.

4.1.2 Accuracy

Accuracy means is proportion of correct anticipation assembled by the classification data model over the complete number of anticipation that the classifier assembled. If the target variable classes in a dataset are nearly balanced, we can expect a good accuracy. Ex: In a cancer dataset, 60% data is Benign and rest 40% is Malignant.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

4.1.3 Precision

Precision is called that which generate ratio of True Positives to the summation of True Positives and False Positives. Simply high precision means that an algorithm generated mostly appropriate results than inappropriate.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

4.1.4 Recall

Recall is a measure of the proportion of patients that were predicted to have the complications among those patients that actually have the complications. A high recall show that an algorithm generated maximum appropriate results. The formula of recall is given below-

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

4.1.5 F1 Score

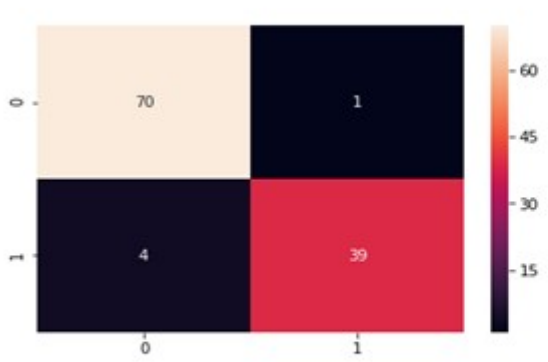
F1 Score is the Harmonic Mean between precision and recall. Additionally, weighted average of precision and recall is known as F1 score. The span of F1 score is from 0 to 1. F1 score represents how accurate the classifier is and also shows how durable that is at the same time.

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

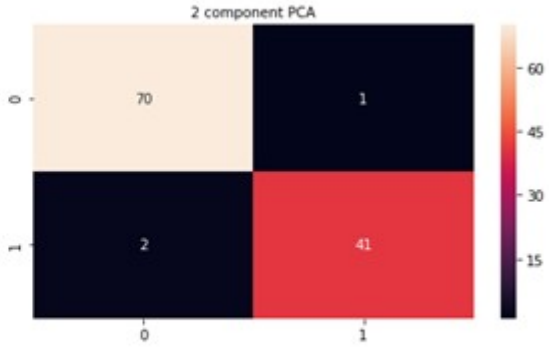
4.2 Model Performances

4.2.1 Logistic Regression (LR)

The Accuracy of Logistic Regression model before applying principal component analysis was 95.61% whereas after applying principal component analysis, the Accuracy increased suddenly to 97.36%. Along with Accuracy, other performance metrics Precision, Recall and F1 score raise after the introduction of PCA which can be seen from fi 4.2a representing the confusion matrix of the model before PCA and fi ure 4.2b representing after the introduction of PCA. Performance of Logistic Regression before and after using PCA is given in fi e 4.2c and 4.2d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.95	0.99	0.97	71
M	0.97	0.91	0.94	43
avg / total	0.96	0.96	0.96	114

```
[[70  1]
 [ 4 39]]
Logistic Regression accuracy is 0.956140350877193
```

(c) Result Without PCA

	precision	recall	f1-score	support
B	0.97	0.99	0.98	71
M	0.98	0.95	0.96	43
avg / total	0.97	0.97	0.97	114

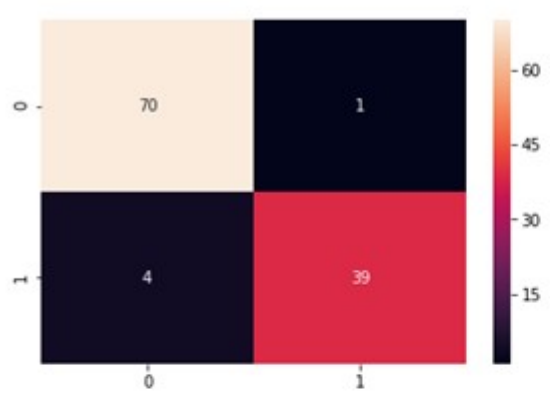
```
[[70  1]
 [ 2 41]]
LogisticRegression accuracy is  0.9736842105263158
```

(d) Result With PCA

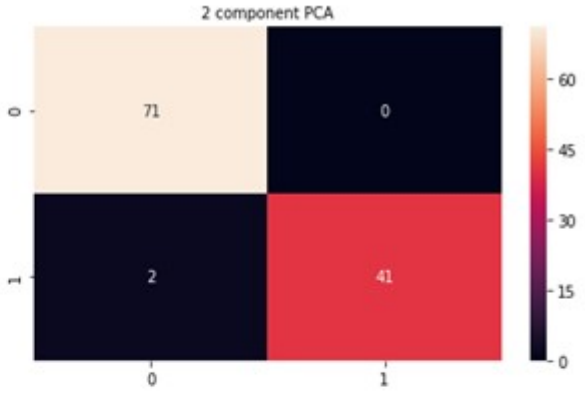
Figure 4.2: Performance Comparison of Logistic Regression

4.2.2 Linear Support Vector Machine (LSVM)

The Accuracy of Linear SVM model before applying principal component analysis was 95.61% whereas after applying principal component analysis, the Accuracy increased significantly to 98.24%. Along with Accuracy, other performance metrics Precision, Recall and F1 score raise after the introduction of PCA which can be seen from fi 4.3a representing the confusion matrix of the model before PCA and fi 4.3b representing after the introduction of PCA. Performance of Linear Support Vector Machine before and after using PCA is given in fi 4.3c and 4.3d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.95	0.99	0.97	71
M	0.97	0.91	0.94	43
avg / total	0.96	0.96	0.96	114

```
[[70  1]
 [ 4 39]]
Linear SVM accuracy is 0.956140350877193
```

(c) Result Without PCA

	precision	recall	f1-score	support
B	0.97	1.00	0.99	71
M	1.00	0.95	0.98	43
avg / total	0.98	0.98	0.98	114

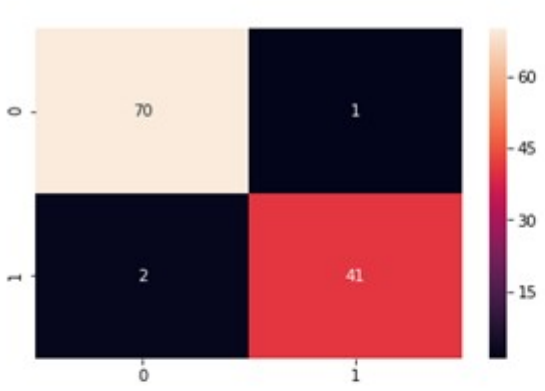
```
[[71  0]
 [ 2 41]]
LinearSVM accuracy is  0.9824561403508771
```

(d) Result With PCA

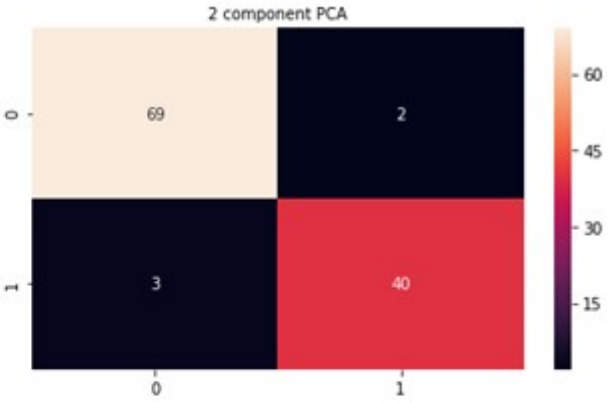
Figure 4.3: Performance Comparison of Linear SVM

4.2.3 Random Forest (RF)

The Accuracy of Random Forest model before applying principal component analysis was 97.36% whereas after applying principal component analysis, the Accuracy decreased slightly to 95.61%. Along with Accuracy, other performance metrics Precision, Recall and F1 score goes slightly downward after the introduction of PCA which can be seen from fi 4.4a representing the confusion matrix of the model before PCA and fi 4.4b representing after the introduction of PCA. Performance of Random Forest before and after using PCA is given in fi ure 4.4c and 4.4d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.97	0.99	0.98	71
M	0.98	0.95	0.96	43
avg / total	0.97	0.97	0.97	114

```
[[70  1]
 [ 2 41]]
RandomForest accuracy is  0.9736842105263158
```

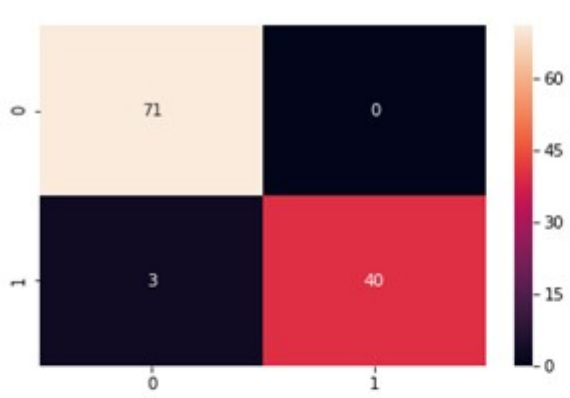
	precision	recall	f1-score	support
B	0.96	0.97	0.97	71
M	0.95	0.93	0.94	43
avg / total	0.96	0.96	0.96	114

```
[[69  2]
 [ 3 40]]
RandomForest accuracy is  0.956140350877193
```

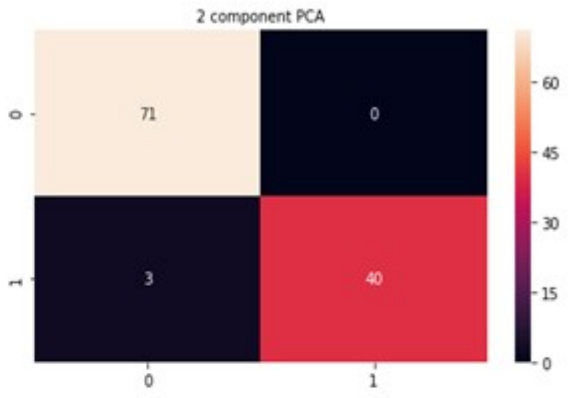
Figure 4.4: Performance Comparison of Random Forest

4.2.4 AdaBoost Tree

The Accuracy of Ada-Boost Tree model before applying principal component analysis was 97.36% and even after applying principal component analysis, the Accuracy we found was same, that means 97.36%. Along with Accuracy, other performance metrics Precision, Recall and F1 score remain same after the introduction of PCA which can be seen from fi 4.5a representing the confusion matrix of the model before PCA and fi 4.5b representing after the introduction of PCA. Performance of Ada-Boost Tree before and after using PCA is given in fi 4.5c and 4.5d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.96	1.00	0.98	71
M	1.00	0.93	0.96	43
avg / total	0.97	0.97	0.97	114

```
[[71  0]
 [ 3 40]]
AdaBoost accuracy is 0.9736842105263158
Here is Kfold:  0.9736842105263158
```

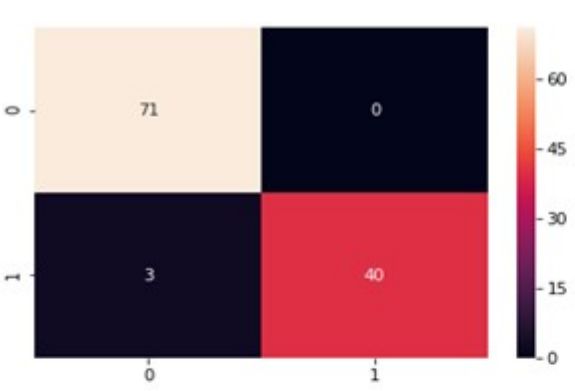
	precision	recall	f1-score	support
B	0.96	1.00	0.98	71
M	1.00	0.93	0.96	43
avg / total	0.97	0.97	0.97	114

```
[[71  0]
 [ 3 40]]
AdaBoost accuracy is 0.9736842105263158
Here is Kfold:  0.9736842105263158
```

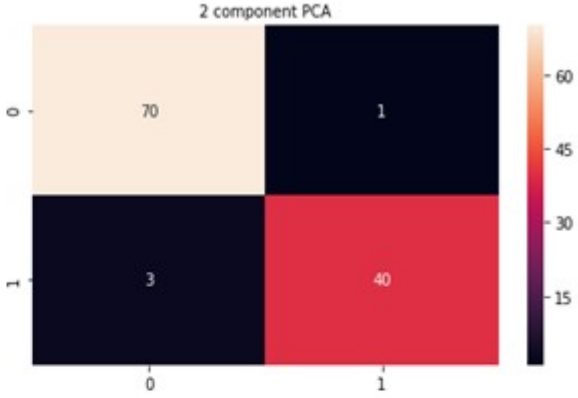
Figure 4.5: Performance Comparison of AdaBoost Tree

4.2.5 Naive Bayes

The Accuracy of Naive Bayes model before applying principal component analysis was 97.36% whereas after applying principal component analysis, the Accuracy decreased slightly to 96.49%. Along with Accuracy, other performance metrics Precision, Recall and F1 score goes slightly downward after the introduction of PCA which can be seen from fi 4.6a representing the confusion matrix of the model before PCA and fi 4.6b representing after the introduction of PCA. Performance of Naïve Bayes before and after using PCA is given in fi 4.6c and 4.6d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.96	1.00	0.98	71
M	1.00	0.93	0.96	43
avg / total	0.97	0.97	0.97	114

```
[[71  0]
 [ 3 40]]
NaiveBayes accuracy is 0.9736842105263158
```

(c) Result Without PCA

	precision	recall	f1-score	support
B	0.96	0.99	0.97	71
M	0.98	0.93	0.95	43
avg / total	0.97	0.96	0.96	114

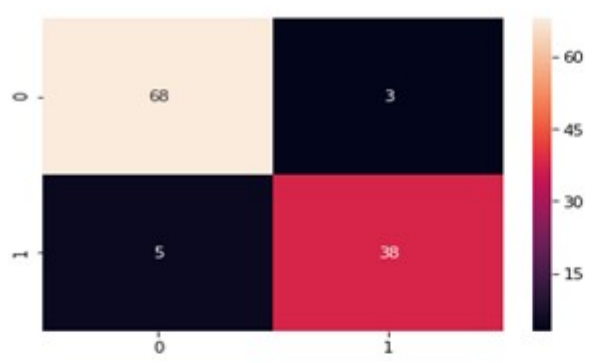
```
[[70  1]
 [ 3 40]]
NaiveBayes accuracy is  0.9649122807017544
```

(d) Result With PCA

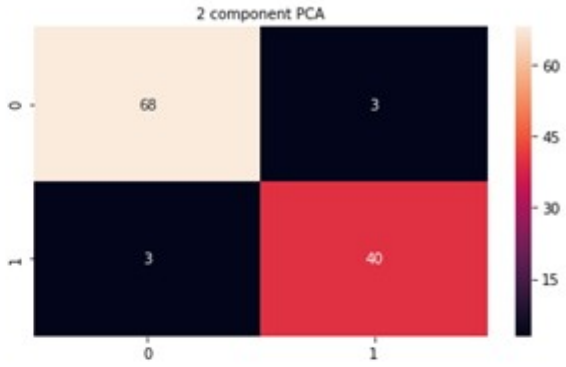
Figure 4.6: Performance Comparison of Naive Bayes

4.2.6 K Neighbor Classifier

The Accuracy of K Neighbor model before applying principal component analysis was 92.98% whereas after applying principal component analysis, the Accuracy increased significantly to 94.73%. Along with Accuracy, other performance metrics Precision, Recall and F1 score raise after the introduction of PCA which can be seen from fi 4.7a representing the confusion matrix of the model before PCA and fi 4.7b representing after the introduction of PCA. Performance of K Neighbor Classifier before and after using PCA is given in fi 4.7c and 4.7d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.93	0.96	0.94	71
M	0.93	0.88	0.90	43
avg / total	0.93	0.93	0.93	114

```
[[68  3]
 [ 5 38]]
KNeighbour accuracy is 0.9298245614035088
```

(c) Result Without PCA

	precision	recall	f1-score	support
B	0.96	0.96	0.96	71
M	0.93	0.93	0.93	43
avg / total	0.95	0.95	0.95	114

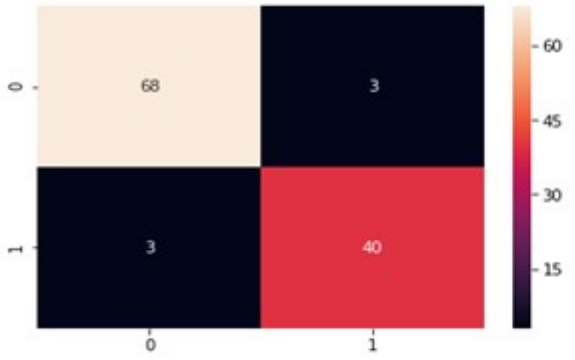
```
[[68  3]
 [ 3 40]]
KNeighbour accuracy is  0.9473684210526315
```

(d) Result With PCA

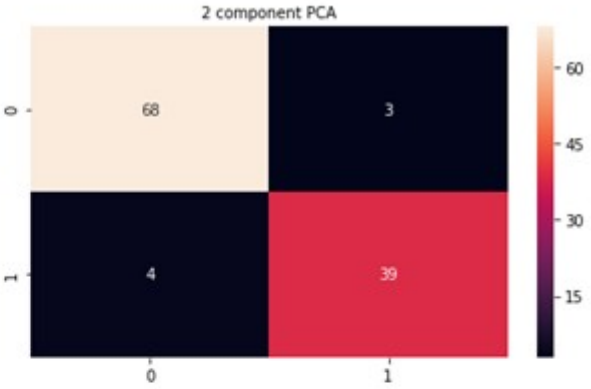
Figure 4.7: Performance Comparison of K Neighbor

4.2.7 Decision Tree (DT

The Accuracy of Decision Tree model before applying principal component analysis was 94.73% whereas after applying principal component analysis, the Accuracy decreased slightly to 93.85%. Along with Accuracy, other performance metrics Precision, Recall and F1 score goes down after the introduction of PCA which can be seen from fi 4.8a representing the confusion matrix of the model before PCA and fi 4.8b representing after the introduction of PCA. Performance of Decision Tree before and after using PCA is given in fi 4.8c and 4.8d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.96	0.96	0.96	71
M	0.93	0.93	0.93	43
avg / total	0.95	0.95	0.95	114

```
[[68  3]
 [ 3 40]]
DecisionTree accuracy is 0.9473684210526315
```

(c) Result Without PCA

	precision	recall	f1-score	support
B	0.94	0.96	0.95	71
M	0.93	0.91	0.92	43
avg / total	0.94	0.94	0.94	114

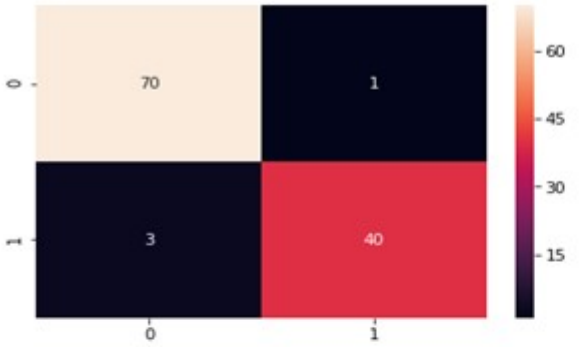
```
[[68  3]
 [ 4 39]]
DecisionTree accuracy is  0.9385964912280702
```

(d) Result With PCA

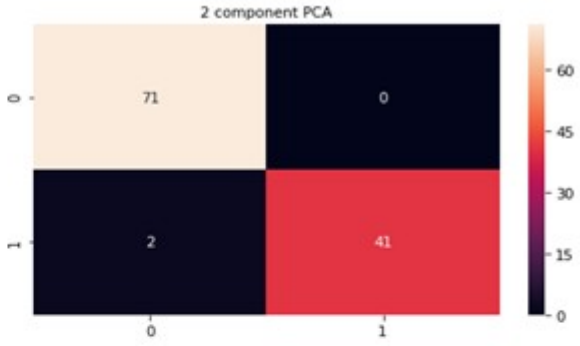
Figure 4.8: Performance Comparison of Decision Tree

4.2.8 Gaussian Process

The Accuracy of Gaussian Process model before applying principal component analysis was 96.49% whereas after applying principal component analysis, the Accuracy increased to 98.24%. Along with Accuracy, other performance metrics Precision, Recall and F1 score raised after the introduction of PCA which can be seen from fi 4.9a representing the confusion matrix of the model before PCA and fi 4.9b representing after the introduction of PCA. Performance of Gaussian Process before and after using PCA is given in fi 4.9c and 4.9d for better understanding and clarity.



(a) Confusion Matrix Without PCA



(b) Confusion Matrix With PCA

	precision	recall	f1-score	support
B	0.96	0.99	0.97	71
M	0.98	0.93	0.95	43
avg / total	0.97	0.96	0.96	114

```
[[70  1]
 [ 3 40]]
GaussianProcess accuracy is 0.9649122807017544
```

(c) Result Without PCA

	precision	recall	f1-score	support
B	0.97	1.00	0.99	71
M	1.00	0.95	0.98	43
avg / total	0.98	0.98	0.98	114

```
[[71  0]
 [ 2 41]]
GaussianProcess accuracy is  0.9824561403508771
```

(d) Result With PCA

Figure 4.9: Performance Comparison of Gaussian Process

4.2.9 Dimensionality Reduction

The objective of this paper is to analyze diff t algorithms and produce better result in the fi of detecting breast cancer. We wanted to establish a comparative study that can help people to reduce the death of breast cancer patients through a time saving better clinical treatment and early awareness with an accurate, fast prediction. Algorithms are tested with regard to sensitivity, accuracy, time complexity, precision. As it is so vital issue in our current era, we eagerly want to study it more for better and reliable result.

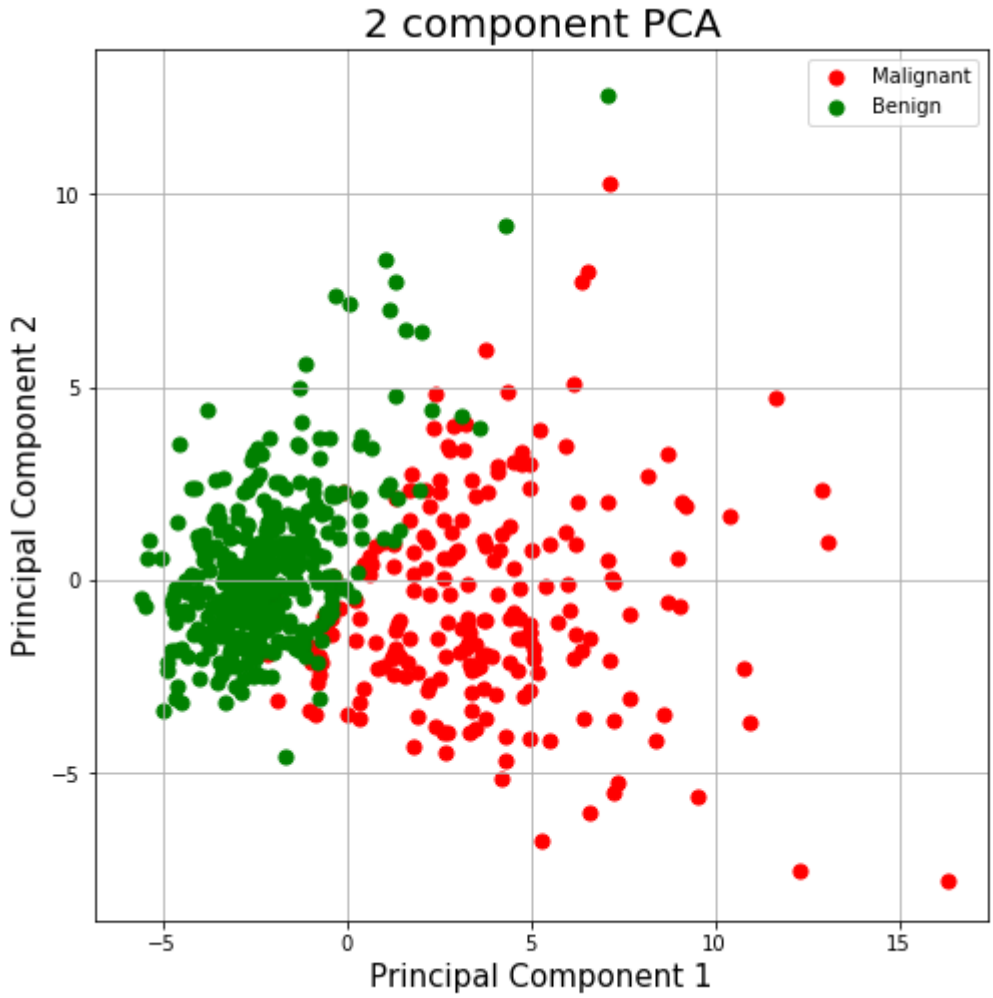


Figure 4.10: Graphical Representation Of B and M Using PCA

4.2.10 Graphical Comparison Among The Algorithms

Here, we have used 8 diff t kinds of algorithms to compare the accuracy with and without PCA. With PCA, Linear Support Vector Machine (LSVM) and Gaussian Process generate the value of 0.98 which is the most accurate. And without PCA, Random Forest (RF), Ada-Boost Tree, Naive Bayes generate 0.97 which is the most accurate. X axis is the name of the algorithms when Y axis is the rate of the accuracy.

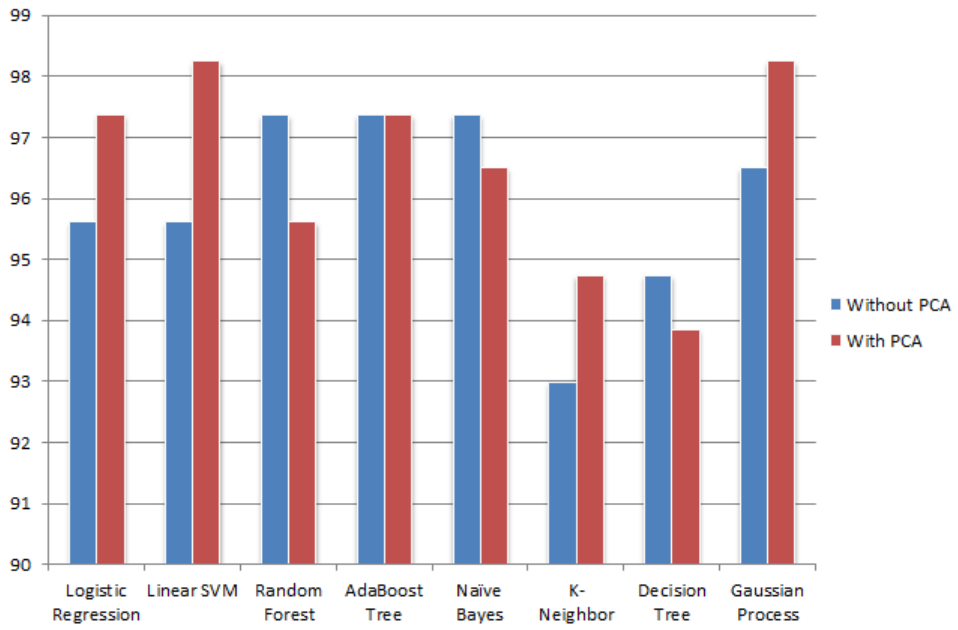


Figure 4.11: Graphical Representation Of B and M Using PCA

4.2.11 Comparison And Analysis Between The Algorithms

We have already discussed about all algorithms in chapter 3. Here we will analyse and compare among all these eight algorithms that we have used for our research work. This comparison is based on some core characteristics like, what is the average predictive accuracy, how fast the classifiers train and make predictions, what happens when there is a small dataset. Average predictive accuracy of Logistic Regression, Naïve Bayes, K- Nearest Neighbours and Decision Tree are comparatively lower. On the contrary, Support Vector Machine, Random Forest, AdaBoost Tree and Gaussian Process classifiers have higher accuracy. In our research work also noticed the upper scenario as SVM, Gaussian Process and Random Forest gave the highest accuracy. Training speed also defers for diff t algorithms. Training speed of Logistic Regression, Naïve Bayes and Decision Tree are faster and rest classifiers training speed is much slower. K- Nearest Neighbour classifier doesn't need any training. Moreover, Gaussian Process classifier's training speed becomes slower when it deals with a huge dataset. Most of the classifiers predict Fast but Random Forest and Gaussian Process classifiers predict at a average speed. On the other hand, prediction speed of K- Nearest Neighbour classifier is slower.
Logistic Regression, SVM, Naïve Bayes, K- Nearest Neighbours and Gaussian Process performs well with a small number of observations whereas Random Forest, AdaBoost Tree and Decision tree need a comparatively big number of observations to make a satisfactory result.

	Average Predictive Accuracy	Training Speed	Prediction Speed	Performs Well with small number of Observations
Logistic Regression	Lower	Fast	Fast	Yes
SVM	Higher	Slow	Fast	Yes
Random Forest	Higher	Slow	Moderate	No
AdaBoost Tree	Higher	Slow	Fast	No
Naïve Bayes	Lower	Fast	Fast	Yes
K- Nearest Neighbours	Lower	No training needed	Slow	Yes
Decision Tree	Lower	Fast	Fast	No
Gaussian Process	Higher	Slow for huge datasets	Moderate	Yes

Chapter 5

Conclusion

Breast cancer, a vital but secretive cancer among women. In rural area mostly women are afraid and ashamed to share this problem with others. So sometimes it's become vital issue when they can't get proper treatment in due time. So it is very important to get proper and perfect and fast detection of the disease.

We used supervised method for our research. We have used eight algorithms to detect the breast cancer more accurately. we used SVM, Random Forest, K Neighbor Classifier, Logistic Regression, Adaboost, Gaussian Process, Decession Tree, Naïve Bayes algorithms. We analysed the results of all algorithms and tried to find out the best possible one. In our work, SVM and Gaussian Process performed best with PCA and Logistic Regression and Random Forest without PCA.

At the end, we tried our best to fi out a suitable classifier that will save the patients by detecting a breast cancer correctly. Everything is controlled by our Almighty but what we can do is, we can try our best to solve breast cancer problems by its early diagnosis. We used a very standard dataset which is widely renowned but the dataset was not that much big. In near future, we will try to enhance our work by managing a comparatively big dataset and adding some more functionalities like the stage detection of breast cancer and so on.

Bibliography

[1] J. Peto, N. Collins, R. Barfoot, S. Seal, W. Warren, N. Rahman, D. F. Easton, C. Evans, J. Deacon, and M. R. Stratton, "Prevalence of brca1 and brca2 gene mutations in patients with early-onset breast cancer", *Journal of the National Cancer Institute*, vol. 91, no. 11, pp. 943–949, 1999.

[2] H. A. Shih, K. L. Nathanson, S. Seal, N. Collins, M. R. Stratton, T. R. Rebbeck, and B. L. Weber, "Brca1 and brca2 mutations in breast cancer families with multiple primary cancers", *Clinical cancer research*, vol. 6, no. 11, pp. 4259–4264, 2000.

[3] S. Ramaswamy and C. M. Perou, "Dna microarrays in breast cancer: The promise of personalised medicine", *The Lancet*, vol. 361, no. 9369, pp. 1576–1577, 2003.

[4] J. A. Cruz and D. S. Wishart, "Applications of machine learning in cancer prediction and prognosis", *Cancer informatics*, vol. 2, p. 117693510600200030, 2006.

[5] W. Yi and W. Fuyong, "Breast cancer diagnosis via supp ort vector machines", in *2006 Chinese Control Conference*, IEEE, 2006, pp. 1853–1856.

[6] K. Polat and S. Güneş, "Breast cancer diagnosis using least square support vector machine", *Digital signal processing*, vol. 17, no. 4, pp. 694–701, 2007.

[7] C.-L. Huang, H.-C. Liao, and M.-C. Chen, "Prediction model building and feature selection with support vector machines in breast cancer diagnosis", *Expert Systems with Applications*, vol. 34, no. 1, pp. 578–587, 2008.

[8] J. Thongkam, G. Xu, and Y. Zhang, "Adaboost algorithm with random forests for predicting breast cancer survivability", in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 3062–3069.

[9] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis", *Expert systems with applications*, vol. 36, no. 2, pp. 3240–3247, 2009.

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

[11] J. S. Parker, M. Mullins, M. C. Cheang, S. Leung, D. Voduc, T. Vickery, S. Davies, C. Fauron, X. He, Z. Hu, *et al.*, "Supervised risk predictor of breast cancer based on intrinsic subtypes", *Journal of clinical oncology*, vol. 27, no. 8, p. 1160, 2009.

[12] L. Vanneschi, A. Farinaccio, G. Mauri, M. Antoniotti, P. Provero, and M. Giacobini, "A comparison of machine learning techniques for survival prediction in breast cancer", *BioData mining*, vol. 4, no. 1, p. 12, 2011.

[13] S. Narang, H. K. Verma, and U. Sachdev, "Breast cancer detection using art2 model of neural networks", *International Journal of Computer Applications*, vol. 57, no. 5, 2012.

[14] G. I. Salama, M. Abdelhalim, and M. A.-e. Zeid, "Breast cancer diagnosis on three diff t datasets using multi-classifiers", *Breast Cancer (WDBC)*, vol. 32, no. 569, p. 2, 2012.

[15] L. G. Ahmad, A. Eshlaghy, A. Poorebrahimi, M. Ebrahimi, A. Razavi, *et al.*, "Using three machine learning techniques for predicting breast cancer recurrence", *J Health Med Inform*, vol. 4, no. 124, p. 3, 2013.

[16] B. Gayathri, C. Sumathi, and T. Santhanam, "Breast cancer diagnosis using machine learning algorithms-a survey", *International Journal of Distributed and Parallel Systems*, vol. 4, no. 3, p. 105, 2013.

[17] J. Kim and H. Shin, "Breast cancer survivability prediction using labeled, unlabeled, and pseudo-labeled patient data", *Journal of the American Medical Informatics Association*, vol. 20, no. 4, pp. 613–618, 2013.

[18] B. Zheng, S. W. Yoon, and S. S. Lam, "Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms", *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.

[19] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction", *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015.

[20] P. H. Abreu, M. S. Santos, M. H. Abreu, B. Andrade, and D. C. Silva, "Predicting breast cancer recurrence using machine learning techniques: A systematic review", *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 52, 2016.

[21] H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel, "Using machine learning algorithms for breast cancer risk prediction and diagnosis", *Procedia Computer Science*, vol. 83, pp. 1064–1069, 2016.

[22] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer", *arXiv preprint arXiv:1606.05718*, 2016.

[23] K.-w. Chau, *Use of meta-heuristic techniques in rainfall-runoff modelling*, 2017.