VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School Of Information Technology and Engineering

April, 2019.

Stochastic and Statistical Classification of Breast Cancer

A CAL J-COMPONENT PROJECT REPORT

For Data Mining (ITA5007)

of

Master of Computer Application

In

Computer Application

By

SURAJ SAHA (18MCA0109)

RIYA SHAJI (18MCA0097)

ABHIJEET GIRI (18MCA0088)

TABLE OF CONTENTS

ABSTRACT

Today breast cancer is a serious menace for the living conditions of the people and now it stands as the next to the highest primary causes of women's death today , in countries like Nigeria where there is no service provided for the early detection of breast cancer the most common cancer in women is developing in Nigerian women. The maximum cases of breast cancer is found in Countries such as Belgium, Nethelands and Luxemborg . It is and was always wise to start cure before detection than after. Thus, a number of studies have been started using data mining techniques to understand the prediction of breast cancer and the various type of cancer the person is suffering from. This paper will illustrate about the breast cancer prediction by using data mining methods to contrive an efficient way to predict breast cancer in women. The objective of this paper is comparing and identifying an accurate model based on various patients' clinical records to predict the incidence of breast cancer. We will implement various data mining models and perform the comparison between all those methods so as to find out the optimal one for the classification of cancer. Due to the high impact on the capability and effectiveness of the learning process, feature space is profoundly discussed in this paper .A hybrid between principal component analysis (PCA) and affiliated data mining models is proposed for testing the significance of feature space reduction, which places a principle component analysis method to contract the feature space. The result achieved by this analysis determines comprehensive trade-off between these strategies and also imparts a described evaluation on the models. It is expected that in real pertinence, people like physicians and patients can gain betterment from the feature recognition outcome to prevent the breast cancer.

KEYWORDS Breast Cancer, KNN, SVM, Artificial Neural Network, Logistic Regression

INTRODUCTION

According to WHO, millions of people died suffering from cancer throughout the world with a calculated increase of 50% in developing countries and an increase of 72% in total number of deaths because of this serious cancer. Estimations provided by Parkin in [1] only 5% of global funds for cancer control is possessed by developing nations and only few human and material resources are availed for them.. Breast cancer is caused when the cells in breast starts growing out of control, then these cells forms a tumour. There are number of factors known as the risk factors which causes the cancer, these are distinguished as either modifiable (this could be in human control like the daily routines, addictions, perils from environment, etc) or the non-modifiable f (the one which is not in human control like getting from family ancestors, gender etc). According to the Joint Venture Group on "Hormonal Factors in Breast Cancer" which was presented in 2002, being female and of former age was declared as the primary risk factor for this. Other factors that are possible for this risk include the genealogy of breast cancer, age when the first menstruation occurs which is known as menarche , the particular age when first birth was given to a baby, weight of the body, menopause age, consumption of alcohol, higher hormonal levels and diet. In the United States, 1,665,540 was recorded as the new number of cases found and 585,620 people died in 2014 suffering from cancer. Approximately 30% of cancer diagnosed in women results in breast cancer, which leads to approximately 15% of deaths which are caused suffering from cancer in the year of 2014. With the enhancement in biomedical and computer technologies, different clinical factors related to breast cancer have been reported. Moreover, the implementation and improvement of next generation medical facilities requires a large amount of funding and cost which can be provided by the authority only when there is a assurance in the data presented by the developers and data scientists. This global crisis led to the development of optimised data mining tricks and methods for classifying and predicting the presence of cancer or the type of cancer the patient is suffering from. Alongside, the improvement of naive statistical tools, affirmative and cutting edge methods like deep learning been also been trained properly for its prediction and classification capacity overweighs the former tools.

PROBLEM STATEMENT

In this problem, we are using Wisconsin Diagnostic Breast Cancer (WDBC) data set that provides many attributes for the classification on diagonisis of Breast Cancer types, i.e, Malignant and Bengin. Therefore we will be using different classification methods like logistic regression, SVM, KNN and atlast we will be implementing it by deep learning approach., seems optimal for large number of data set the classification approach we will be Using is Neural Network and Stochastic Gradient Descent as they will be the optimal one. Moreover, in order to avoid the problem of Dimensionality, we have to decrease thenumber of attributes as possible which could affect the result most using Principal Component Analysis (PCA).

TRAINING DATASET DESCRIPTION

The data used for training the models is Wisconsin Breast Cancer Data Set collected from Kaggle repositories. The following dataset seems to contain good quality of data with less noisy and disrupted frequency of variations. Wisconsin data contains 31 data attributes that are generally responsible for detection the type of Cancer cell. Here two types of cancer cell are being considered Malignant and Bengnin Cell. Total number of training tuples provided in this dataset are 927242.

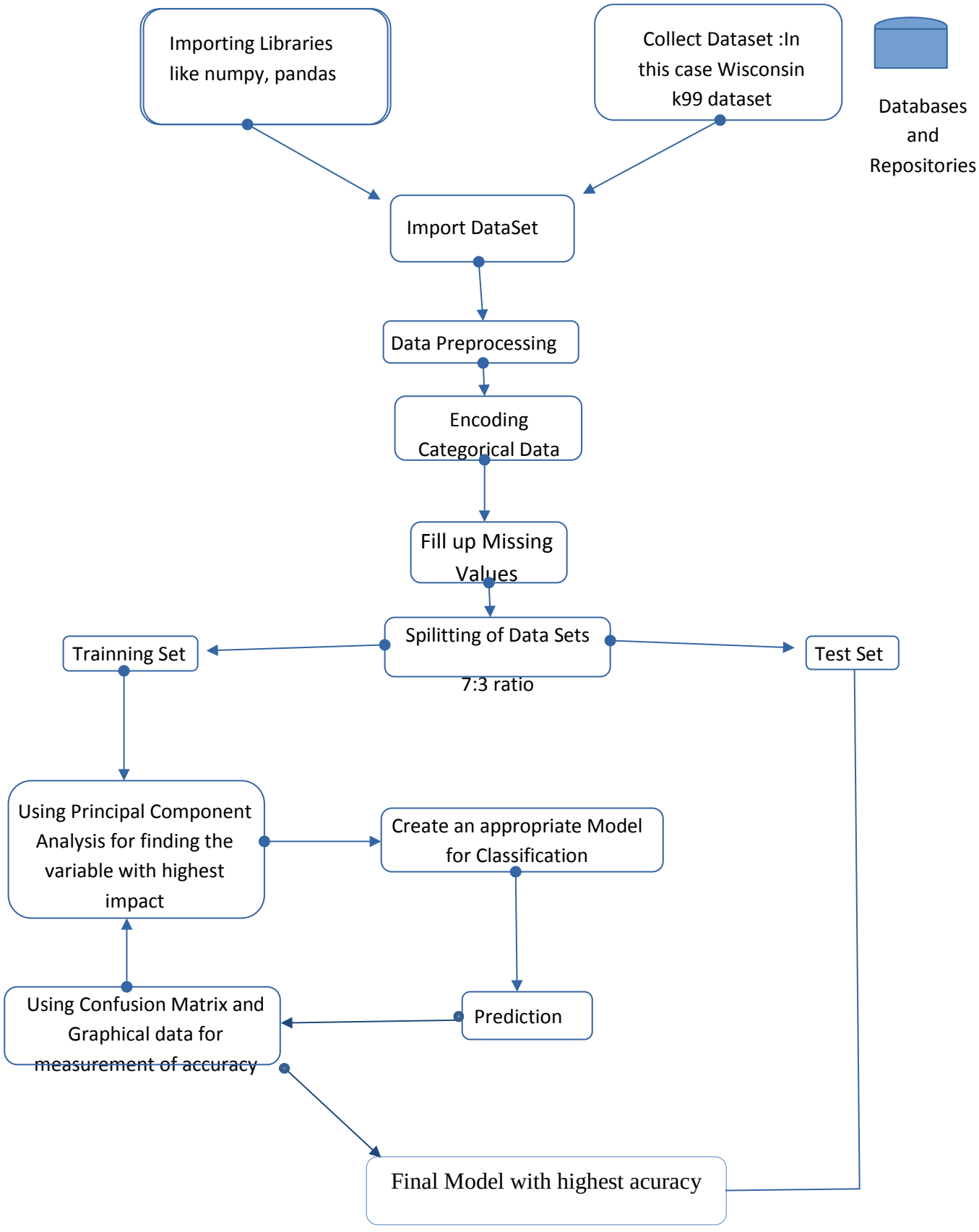
ATTRIBUTES

Attribute	Description	Mean	Standard Deviation
1. "diagnosis"	contains breast cancer tissues types (M = malignant, B = benign)		
2. " radius_mean"	mean distances on the perimeter to the point center	14.12729	3.524048
3."texture_meanstandard"	deviation of gray-scale values	19.2896	4.30103
4. "perimeter_mean"	mean size of the core tumor	91.9690	24.2989
5. "area_mean"		654.8891	351.9141
6. "smoothness_mean"	mean of local variation in radius lengths	0.09636	0.0140
7. "compactness_mean"	mean of perimeter^2 / area - 1.0	0.10434	0.05281
8. "concavity_mean"	mean of severity of concave portions of the contour	0.08879	0.07971
9. "concave points_mean"	mean for number of concave portions of the contour	0.04891	0.0388
10. "symmetry_mean"		0.06279	0.00706
11. "fractal_dimension_mean"	mean for "coastline approximation" - 1	0.40517	0.27731

12. “radius_sestandard”	error for the mean of distances from center to points on the perimeter	1.21685	0.55164
13. “texture_sestandard”	error for standard deviation of gray-scale values	2.86605	2.02185
14. “perimeter_se”		40.337	45.4910
15. “area_se”		0.007040	0.0030
16. “smoothness_sestandard”	error for local variation in radius lengths	0.0254	0.01790
17. “compactness_sestandard”	error for perimeter^2 / area - 1.0	0.03189	0.0301
18. “concavity_sestandard”	error for severity of concave portions of the contour	0.01179	0.0061
19. “concave points_sestandard”	error for number of concave portions of the contour	0.0205	0.00826
20. “symmetry_se”		16.269	4.83324
21. “fratal_dimsion_sestandard”	error for "coastline approximation" - 1	25.6772	6.14625
22. “radius_worst”	"worst" or largest mean value for mean of distances from center to points on the perimeter	107.2612	33.6025
23. “texture_worst”	"worst" or largest mean value for standard deviation of gray-scale values	880.5831	569.356
24. “perimeter_worst”		0.13236	0.0228
25. “area_worst”		0.2542	0.15733
26. “smoothness_worst”	“worst” mean value for local variation in radius lengths	0.2721	0.2086

27. "compactness_worst"	"worst"mean value for perimeter^2 / area - 1.0	0.11460	0.0657
28. "concavity_worst"	worst mean value for severity of concave portions of the contour	0.2900	0.0618
29. "concave points_worst"	"worst" mean value for number of concave portions of the contour	0.083945	0.01806
30. "symmetry_worst"			
31. "fractal_dimension_worst"	"worst" mean value for the "coastline approximation"		

FLOW CHART


Importing Libraries like numpy, pandas
Collect Dataset :In this case Wisconsin k99 dataset
Databases and Repositories
Import DataSet
Data Preprocessing
Encoding Categorical Data
Fill up Missing Values
Spilitting of Data Sets
7:3 ratio
Trainning Set
Test Set
Using Principal Component Analysis for finding the variable with highest impact
Create an appropriate Model for Classification
Using Confusion Matrix and Graphical data for measurement of accuracy
Prediction
Final Model with highest acuracy

MODULE DESCRIPTION

Logistic Regression:

Dichotomy is the prediction of two values in a problem, for this type of prediction we generally use the logistic regression. As the name suggest regression is used for prediction of range values of domains, but logistic regression is used for classification problem especially when there is a presence of two output classes. It is a type of nonlinear regression unlike simple linear regression and multiple linear regressions. These are possible two reasons why we cannot use linear regression for the classification of binary value.

- Generally a linear regression will predict values whose limits of range lies beyond 0 and 1.
- As dichotomous research experiments can have any one of the two possible values, the outputs will not be able to normally be distributed around the predicted line.

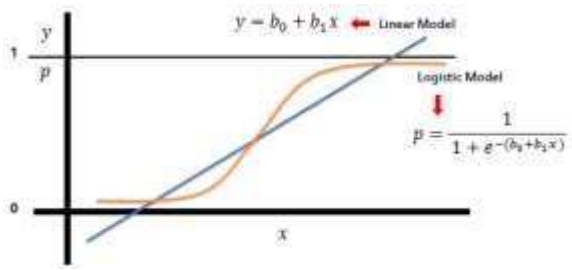


Fig. 1

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

K-Nearest Neighbours:

K nearest neighbour is defined as a simple supervised algorithm which can train itself with all the given cases provided by the supervisor and then classify it to a new test case on previous features based on similarity measure (e.g., distance functions such as Euclidean or Hamilton distance). KNN algorithm can also be used for clustering problems too which is an unsupervised learning algorithm Algorithm:

Mostly all the cases are distinguished by a majority vote of its nearest neighbours, with the case being designated to the class most common amongst its K nearest neighbours are measured by the distance function. If K = 1, then the case is simply designated to the class of its nearest neighbour .

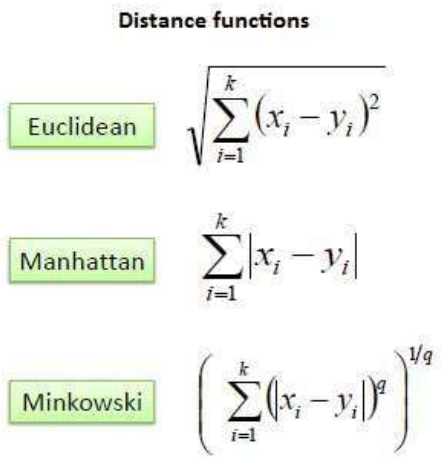


Fig. 2

It should also be notable here that all three distance measures are valid only for the continuous variables. For the occurrence of categorical variables the Hamming distance should be used. It also gives the issue of standardization of the numerical variables lying between 0 and 1 when the numerical and categorical values are mixed in the dataset .

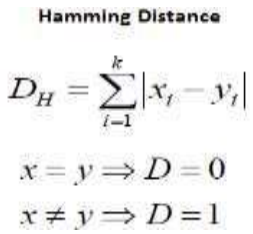


Fig. 3

<u>Support Vector Machine:</u>

The purpose of the support vector machine algorithm is finding in an N-dimensional space a hyperplane, here 'n' is the number of features that plainly distinguishes the data points.

There exist many combinations of hyperplanes that could be chosen in order to separate the two classes of data points. Our initiative is to find a maximum margin plane i.e. the highest distance between any two data points of both the classes. Increasing the margin to maximum distance gives us some reinforcement so that data in near future can be classified with more level of confidence.

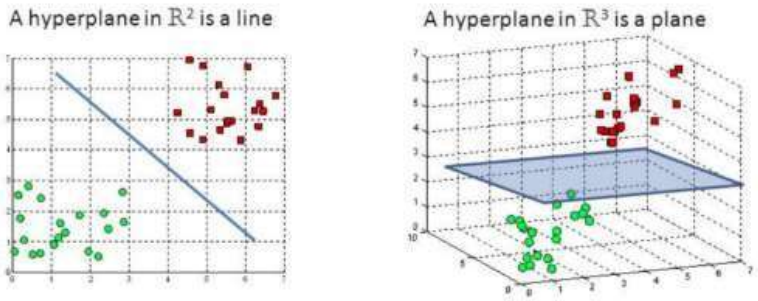


Fig. 4. Hyperplanes in 2D and 3D Hyperspace

Hyperplanes are used as decision boundaries which help in classifying the data points. Data points which falls on either side of the hyperplane is considered of different classes. Therefore the number of features represents the dimension of Hyperplanes. When two input features are considered, that results the hyperplane in just a line. When three input features are taken, then the hyperplane acts as a 2D plane. Situations where number of features is more than 3, it becomes difficult to predict. Support vectors are that data point which are nearer to the hyperplane and makes its impact on hyperplane's position and orientation. We maximize the margin of the classifier by implementing these support vectors. The position of the hyperplane changes when we delete the consecutive support vectors. These points help in the implementation of our SVM.

Updating the Cost Function and Gradient

According to the SVM algorithm, maximum margin between the data points and the hyperplane is maintained. In order to achieve the maximum margin, loss function is used.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \qquad c(x, y, f(x)) = (1 - y * f(x))_+$$

If the cost is 0 we need to consider there is same sign before the predicted value and the actual value. In case they are not same, then we need to calculate the loss value. Addition of a regularization variable is maintained with the cost function. The very purpose of the regularization variable is to maintain a balance between the maximization of margin and loss.

.

DATA SET

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
842302	M	17.99	10.38	122.8	1001	0.1184
842517	M	20.57	17.77	132.9	1326	0.08474
84300903	M	19.69	21.25	130	1203	0.1096
84348301	M	11.42	20.38	77.58	386.1	0.1425
84358402	M	20.29	14.34	135.1	1297	0.1003
843786	M	12.45	15.7	82.57	477.1	0.1278
844359	M	18.25	19.98	119.6	1040	0.09463
84458202	M	13.71	20.83	90.2	577.9	0.1189
844981	M	13	21.82	87.5	519.8	0.1273
84501001	M	12.46	24.04	83.97	475.9	0.1186
845636	M	16.02	23.24	102.7	797.8	0.08206
84610002	M	15.78	17.89	103.6	781	0.0971
846226	M	19.17	24.8	132.4	1123	0.0974
846381	M	15.85	23.95	103.7	782.7	0.08401
84667401	M	13.73	22.61	93.6	578.3	0.1131
84799002	M	14.54	27.54	96.73	658.8	0.1139
848406	M	14.68	20.13	94.74	684.5	0.09867
84862001	M	16.13	20.68	108.1	798.8	0.117
849014	M	19.81	22.15	130	1260	0.09831

8510426	B	13.54	14.36	87.46	566.3	0.09779
8510653	B	13.08	15.71	85.63	520	0.1075
8510824	B	9.504	12.44	60.34	273.9	0.1024
8511133	M	15.34	14.26	102.5	704.4	0.1073
851509	M	21.16	23.04	137.2	1404	0.09428

LITERATURE SURVEY

The paper [3] presents a study about breast cancer prediction based on data mining methods for identifying an effective way for prediction and classification of presence of Breast Cancer. The establishment of the author from this paper is comparing and identifying an accurate model for the prediction of the incidence of breast cancer based on various patients' clinical records. Four data mining models such as "Support Vector Machine (SVM)", "Naive Bayes" classifier, "AdaBoost tree" and the artificial neural network (ANN),. To test the significance of feature space reduction, a hybrid between principal component analysis (PCA) and affiliated data mining models is proposed, which places a principle component analysis method to contract the feature space. To appraise the performance of these models, two extensively used test data sets are used, Wisconsin Diagnostic Breast Cancer (1995) and the Wisconsin Breast Cancer Database (1991). 10-fold cross-validation method is utilized so as to project the test error of each individual model. The results gained by this analysis establish a comprehensive trade-off between these approaches and also provides a comprehensive evaluation of the models.

The author of this paper [4] surveys on different processes and ways of data mining, which are considered explicitly for the prediction of breast cancer presence. Varieties of data mining techniques are highlighted and a detailed description has been highlighted such as Decision Tress, Artificial Neural Network, Genetic Algorithm and Support Vector Machine. Every techniques and methods presented had its own advantage and disadvantage. This research paper collaborate the reviews which are carried out by different experts who contribute in the field of data analysis and mining of data from systems. From the review presented above, we can assume that there is still a lack of early "sensitivity", "specificity" and also the data diagnosis of the breast cancer.

In this paper [5] two distinct classification techniques for mining of data are used for prediction of the risk of breast cancer and the perfection of these are analyzed to figure out the most accurate classifier. The results from the experiment displays that the better model is "J48 decision tress" for the prediction of the risk of having breast cancer and as per the records for both the models the value of of accuracy, recall, precision and error rates . Therefore, an efficient classifier for breast cancer risks has been determined and also by increasing the number of samples for the training data set we could increase the attribute numbers which could result in the development of a more definite model.

In [6], the author for the prognosis of breast tumour has applied the genetic systems. This system combines the decision tree, ANN and logistic regression . A total of 695 records of patients from the University of Wisconsin who are suffering from breast cancer been taken for their research. They have also taken 9 indicator variables with 1 result variable for the information investigation with 10fold cross approval . The researchers claimed an accuracy of 98.9% which has been given by the genetic prediction model presented by them.

The author of the paper in [7] used gene expression data for the cancer prediction. They formed the minimum gene probability . Suggestions given by the author includes “gene selection” and “gene ranking” scheme. Based on the ranking =-[;’[[p[--scores, presence of “malignant” cancer cell was detected. Methods like T-test and class seperability has been also conducted by them for the support of theory.

The author of this paper [8] carried out applied “Support Vector Machine” as a mapping which is non-linear in manner for moving the data stored for training in order to highlight higher dimensional spaces. The search for linear optimal hyper plane was granted by this new dimension . The SVM by using the edges and the support vector discovered the hyper plane .

PROPOSED WORK

BY USING LOGISTIC REGRESSION

```
import numpy as np import
matplotlib.pyplot as plt import
pandas as pd

dataset = pd.read_csv('data.csv')
X = dataset.iloc[:, 2:32].values y
= dataset.iloc[:, 1].values

from sklearn.preprocessing import Imputer
imputer = Imputer(missing_values = 'NaN', strategy = 'mean', axis = 0)
imputer = imputer.fit(X[:, 0:29]) X[:, 0:29] = imputer.transform(X[:, 0:29])

from sklearn.preprocessing import LabelEncoder
labelencoder_y = LabelEncoder() y =
labelencoder_y.fit_transform(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

from sklearn.preprocessing import StandardScaler sc_X
= StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

''' Applying logistic regression'''
from sklearn.linear_model import LogisticRegression classifier
= LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)
'''Confusion Matrix'''
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
#-----------------------------------------------------------------------------

#-----------------------------------------------------------------------------
'''Applying PCA2 to find out most determining factor'''
from sklearn.decomposition import PCA pca =
```

```
PCA(n_components = 2)
#pca=PCA(n_components=None)
X_train = pca.fit_transform(X_train) X_test
= pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_

'''Apply logistic Regression Again''' from
sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)
'''Confusion Matrix'''
from sklearn.metrics import confusion_matrix
cm2 = confusion_matrix(y_test, y_pred)
#-----------------------------------------------------------------------------
'''Applying PCA3 to find out most determining factor'''
from sklearn.decomposition import PCA pca =
PCA(n_components = 3)
#pca=PCA(n_components=None)
X_train = pca.fit_transform(X_train) X_test
= pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
'''Apply logistic Regression Again''' from
sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)
'''Confusion Matrix'''
from sklearn.metrics import confusion_matrix
cm3 = confusion_matrix(y_test, y_pred)
#-----------------------------------------------------------------------------
'''Applying PCA5 to find out most determining factor'''
from sklearn.decomposition import PCA pca =
PCA(n_components = 5)
#pca=PCA(n_components=None)
X_train = pca.fit_transform(X_train) X_test
= pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
'''Apply logistic Regression Again'''
```

```
from sklearn.linear_model import LogisticRegression classifier
= LogisticRegression(random_state = 0) classifier.fit(X_train,
y_train)

y_pred = classifier.predict(X_test)
'''Confusion Matrix'''
from sklearn.metrics import confusion_matrix
cm5 = confusion_matrix(y_test, y_pred)
#-----------------------------------------------------------------------------




'''Visualising the Training set results''' from
matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step =
0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green'))) plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max()) for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
c = ListedColormap(('red', 'green'))(i), label = j) plt.title('Logistic
Regression (Training set)') plt.xlabel('X') plt.ylabel('Y')
plt.legend() plt.show()

''' Visualising the Test set results'''
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step =
0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green', 'blue'))) plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max()) for i, j in enumerate(np.unique(y_set)):      plt.scatter(X_set[y_set
== j, 0], X_set[y_set == j, 1],                  c = ListedColormap(('red', 'green', 'blue'))(i),
label = j) plt.title('Logistic Regression (Test set)') plt.xlabel('PC1') plt.ylabel('PC2') plt.legend()
plt.show()
```

BY USING KNN

```
import numpy as np import
matplotlib.pyplot as plt import
pandas as pd

dataset = pd.read_csv('data.csv')
X = dataset.iloc[:, 2:32].values y
= dataset.iloc[:, 1].values

from sklearn.preprocessing import Imputer
imputer = Imputer(missing_values = 'NaN', strategy = 'mean', axis = 0)
imputer = imputer.fit(X[:, 0:29]) X[:, 0:29] = imputer.transform(X[:, 0:29])

from sklearn.preprocessing import LabelEncoder
labelencoder_y = LabelEncoder() y =
labelencoder_y.fit_transform(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)


# Fitting K-NN to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2) classifier.fit(X_train,
y_train)

# Predicting the Test set results y_pred
= classifier.predict(X_test)

# Making the Confusion Matrix from
sklearn.metrics import confusion_matrix cm
= confusion_matrix(y_test, y_pred)



'''
#Applying PCA2 to find out most determining factor
from sklearn.decomposition import PCA pca =
```

```
PCA(n_components = 2)
#pca=PCA(n_components=None)
X_train = pca.fit_transform(X_train) X_test
= pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_ from
sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train) y_pred = classifier.predict(X_test) from
sklearn.metrics import confusion_matrix
cm2 = confusion_matrix(y_test, y_pred)
'''

'''Applying PCA3 to find out most determining factor
from sklearn.decomposition import PCA pca =
PCA(n_components = 3)
#pca=PCA(n_components=None)
X_train = pca.fit_transform(X_train) X_test
= pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_ from
sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train) y_pred = classifier.predict(X_test) from
sklearn.metrics import confusion_matrix
cm3 = confusion_matrix(y_test, y_pred)
'''


'''Applying PCA5 to find out most determining factor
from sklearn.decomposition import PCA pca =
PCA(n_components = 5)
#pca=PCA(n_components=None)
X_train = pca.fit_transform(X_train) X_test
= pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_ from
sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2) classifier.fit(X_train,
y_train)
y_pred = classifier.predict(X_test) from
sklearn.metrics import confusion_matrix cm5
= confusion_matrix(y_test, y_pred)
```

```
"""Visualising the Training set results"' from
matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step =
0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green'))) plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max()) for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
c = ListedColormap(('red', 'green'))(i), label = j) plt.title('Logistic
Regression (Training set)') plt.xlabel('X') plt.ylabel('Y')
plt.legend() plt.show()

"' Visualising the Test set results"'
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step =
0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green', 'blue'))) plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max()) for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green', 'blue'))(i), label = j)
plt.title('Logistic Regression (Test set)') plt.xlabel('PC1')
plt.ylabel('PC2') plt.legend() plt.show()
```

BY USING SVM CLASSIFICATION

```
import numpy as np import
matplotlib.pyplot as plt import
pandas as pd

dataset = pd.read_csv('data.csv')
X = dataset.iloc[:, 2:32].values y
= dataset.iloc[:, 1].values

from sklearn.preprocessing import Imputer
```

```
imputer = Imputer(missing_values = 'NaN', strategy = 'mean', axis = 0)
imputer = imputer.fit(X[:, 0:29]) X[:, 0:29] = imputer.transform(X[:, 0:29])

from sklearn.preprocessing import LabelEncoder
labelencoder_y = LabelEncoder() y =
labelencoder_y.fit_transform(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

# Fitting SVM to the Training set from
sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0) classifier.fit(X_train,
y_train)

# Predicting the Test set results y_pred
= classifier.predict(X_test)

# Making the Confusion Matrix from
sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
'''
#Applying PCA2 to find out most determining factor from
sklearn.decomposition import PCA
pca = PCA(n_components = 2) X_train =
pca.fit_transform(X_train) X_test =
pca.transform(X_test) explained_variance =
pca.explained_variance_ratio_ from sklearn.svm
import SVC classifier = SVC(kernel = 'linear',
random_state = 0) classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix
cm2 = confusion_matrix(y_test, y_pred)
"""
```

```
#Applying PCA3 to find out most determining factor
from sklearn.decomposition import PCA pca =
PCA(n_components = 3) X_train =
pca.fit_transform(X_train) X_test =
pca.transform(X_test) explained_variance =
pca.explained_variance_ratio_ from sklearn.svm
import SVC classifier = SVC(kernel = 'linear',
random_state = 0) classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix
cm3 = confusion_matrix(y_test, y_pred)
"""
#Applying PCA5 to find out most determining factor
from sklearn.decomposition import PCA pca =
PCA(n_components = 5) X_train =
pca.fit_transform(X_train) X_test =
pca.transform(X_test) explained_variance =
pca.explained_variance_ratio_ from sklearn.svm
import SVC classifier = SVC(kernel = 'linear',
random_state = 0) classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test) from
sklearn.metrics import confusion_matrix
cm5 = confusion_matrix(y_test, y_pred)
'''
'''Visualising the Training set results''' from
matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step =
0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green'))) plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max()) for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],                    c =
ListedColormap(('red', 'green'))(i), label = j) plt.title('SVM (Training set)') plt.xlabel('X')
plt.ylabel('Y') plt.legend() plt.show()

''' Visualising the Test set results'''
```

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step =
0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green', 'blue'))) plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max()) for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green', 'blue'))(i), label = j)
plt.title('SVM (Test set)') plt.xlabel('PC1') plt.ylabel('PC2') plt.legend()
plt.show()
```

OUTPUT

Feature Space Reduction

We have often heard about the term "Curse of Dimensionality" in fields like Machine Learning and Data Science. It is used to refer the condition that arises when number of attributed taken for regression and classification is too much the model fails to perform in its optimal structure perfectly. Too many attributes makes the data abrupt and it needs more time and memory at the time of classification, thus compromising the resources. In order to avoid it we take features from the data sets that are highly correlated with the output. Principal Component Analysis is one of the commonly used statistical tools that are used to find the most viable features from the given set of features that makes its most impact on the result. Other tool like Linear Discriminant Analysis too helps in finding the best features from the data sets.

Results and Comparison

For the evaluation of performance and model selection we are going to look at the Confusion Matrix and compare the F_{score}. A Confusion matrix consists of four cells when the number of classification to be made is two in our case, and nine when number of classification is three and so on. It generally consist of TP,TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively. True positive and True negative are the positive and negative tuples respectively that are being correctly labelled by classifier. False positive and false negative are positive and negative tuples respectively that are either mislabelled or incorrectly labelled by the classifier. From the confusion matrix we are going to find the accuracy rate, error rate and consequently f_{score} for each model.

$$\boldsymbol{accuracy\ recognition\ rate} = (\boldsymbol{t^p + t_n})/\boldsymbol{p + n}$$

$$\boldsymbol{error\ rate} = (\boldsymbol{F^p + F_n})/(\boldsymbol{p + n})$$

$$\boldsymbol{recall\ rate} = \boldsymbol{t^p}/\boldsymbol{p} \qquad \boldsymbol{precision} = \boldsymbol{T_p}/(\boldsymbol{T_p + F_p})$$

$$\boldsymbol{F_{score}} = (\boldsymbol{2 \times precision \times recall})/(\boldsymbol{precision + recall})$$

Logistic Regression

	0	1
0	61	6
1	3	44

Fig. 1

	0	1
0	65	2
1	3	44

Fig.2

	0	1
0	65	2
1	2	45

Fig.3

Figure 1 is the Confusion matrix when we have considered 5 attributes as principal components. Clearly accuracy is 92% but still we reduce the principal component to 3 when we find the confusion matrix in Figure 2, which indeed gives a better result of 95%. In order to plot the result on graph we made principal component to 2, for which accuracy of 96.4% is acquired.

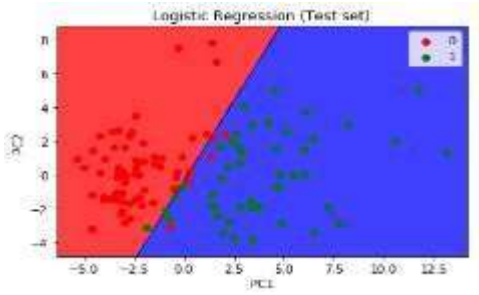


Fig. 4 **Fig. 5**

Here, we can observe how the logistic regression helps us in the classification of data, both for test and training dataset. The point of colour red in the blue section signifies the mismatch labelled from the classifier in the figure 4. The straight line between the red and blue region is the section difference between the properties of Malignant and Bengnin Cancer cell.

KNN

	0	1
0	61	6
1	3	44

	0	1
0	65	2
1	4	43

	0	1
0	67	0
1	5	42

Fig. 6 **Fig. 7** **Fig.8**

Figure 6 descibes the Confusion Matrix when the number of principal component is considered 5 and Figure 7, when 3 pricipal components are taken. For the former we get an accuracy of 92.1% and for the other its 94%. When number of pricipal components is taken as 2, we get an accuracy rate of 95.61%. Thus for the Wisconsin dataset performs better with logistic regression.

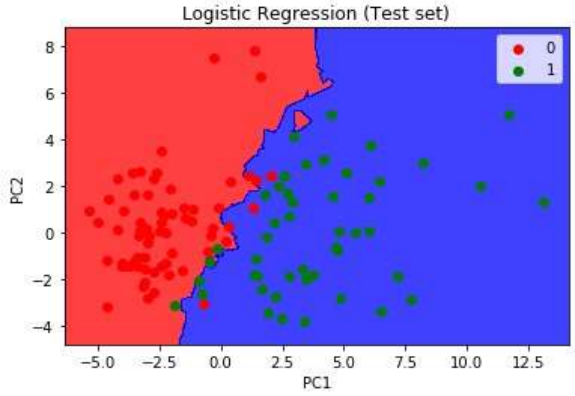


Fig. 9 **Fig.10**

For KNN we can observe, that line of separation in this case in not linear i.e., a straight line. In our experiment we considered 5 neighbours for grouping of similar data. This makes the data classification on the graph more of a non-linear type. Generally K-Nearest Neighbours work more

accurately than logistic regression, but for Wisconsin Data arise an exception. In figure 9 we can observe.

SVM

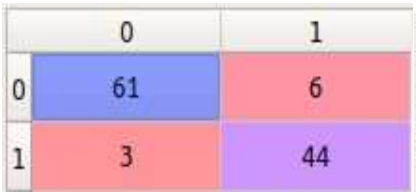


Fig. 11

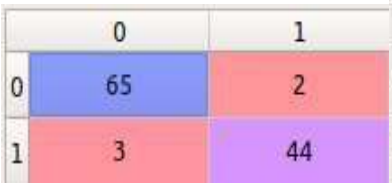


Fig. 12

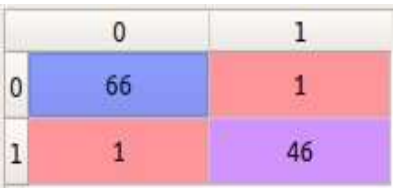


Fig. 13

An accuracy rate of 92.1% is observed when number of principal components is 5 in figure 11. When number of features taken for model training is 3 we get an accuracy rate of 95% which is gives us a better result than Logistic regression and KNN combined. On reducing the number of features to 2, we get a accuracy rate of 98.24%.

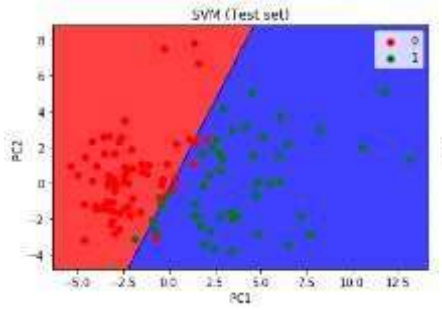


Fig. 14

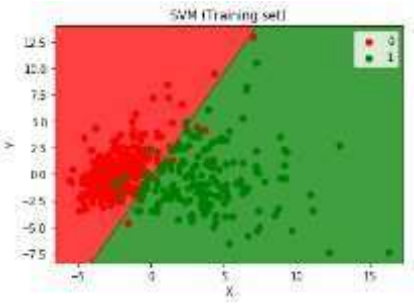


Fig. 15

From the previous graphs of KNN we have seen being non-linear makes the tool less accurate than the logistic regression, which in turn has linear seperability. In Support Vector Machine we can observe that, due to its linear seperability of classes make it the most accurate of the three models we have used so far.

Model	Accuracy Recognition Rate(%)	Error Rate (%)	Sensitivity or Rate Recall	Precision	Fscore
Logistic Regression					
pcm=5	92	7.89	0.91	0.95	0.92
pcm=3	95	4.38	**0.97**	0.95	0.95
pcm=2	**96.4**	3.5	**0.97**	**0.97**	**0.97**
KNN					
pcm=5	92.1	7.89	0.91	**0.95**	0.92
pcm=3	94.73	5.26	0.97	0.94	0.95
pcm=2	**95.61**	4.38	**1**	0.93	**0.96**
SVM					
pcm=5	92.1	7.89	0.91	0.95	0.92
pcm=3	95.6	4.38	0.97	0.95	0.95
pcm=2	**98.2**	1.75	**0.98**	**0.98**	**0.98**

Table 2: A Broad Classification Of models.

CONCLUSION

In the era of data intelligence, we are bounded by the need of data from all directions. Thus being a fundamental need of today's industrial and commercial purposes, data must be of good quality and should be error free. The above project is being used in the medical research. Breast Cancer has always been a major threat to the world and the human beings. Prediction of such major hazard has been a priority and challenge for the developers. From the tables mentioned in this paper has proved that the bounds of data has even surpassed the traditional statistical learning methods to classify. Thus we need deep learning and neural nets methods and an efficient one for better classification. In this paper we have highlighted an accuracy rate of 98.47% implemented by Neural Network. Principal Component Analysis plays a major role in dimension reduction technique, gives us some advantages in terms of classifications. The reduction method can be further be reduced in a more correlative manner by using either LDA or naive-PCA methods.

REFERENCES

[1]American Cancer Soceity(2005), "Breast Cancer Facts & Figures 2005-2006", http://web.archive.org/web/20070613192148/ http://www.cancer.org/downloads/STT/CA/

FF2005BrFacspdf2005.pdf; 13June2007;Retrieved 2013/02/26

[2]American Cancer Society (2007), **"Cancer Facts & Figures 2007"**; 10 April 2007; http://web.archive.org/web/20070410025934/, http://www.cancer.org/downloads/STT/CAFF2007PWSecured.pdf. Retrieved 2012-11-26.

[3]Haifeng Wang, Sang Won Yoon, **"Breast Cancer prediction using Data Mining Methods"**, *Proceedings of the 2015 Industrial and Systems Engineering Research Conference*, S Centikaya and J.K Ryan, eds. October 2015

[4] M Deepika, L Mary Gladence, and R Madhu Keerthana, "**A Review on Prediction Of Breast Cancer Using Various Data Mining Techniques**", *Research Journal of Pharmaceutical, Biological and Chemical Sciences*, ISSN: 0975-8585, pg-808, Januray- February 2016.

[5] Peter Adebayo Idowu, Kehinde Oladipo Williams, Jeremiah Ademola Balogun and Adeniran Ishola Oluwaranti, "**Breast Cancer Risk Prediction Using Data Mining Classification Techniques",** *Transactions on Network and Communication*, ISSN: 2054- 7420, Vol. 3, Issue 2, March 10 2015.

[6] RuiXu, Anagnostopoulos, Wunsch And Gc, D.C.I.I, "**Multiclass Cancer Classification Using Semi supervised Ellipsoid ARTMAP and Particle Swarm Optimization with Gene Expression Data"**, *IEEE/ACM Transactions On Computational Biology And Bioinformatics*, Vol.4, No.1, Pp. 65-77, 2007.

[7]Lipo Wang, Feng Chu, And Wei Xie, "**Accurate Cancer Classification Using Expressions Of Very Few Genes"**, *IEEE/ ACM Transactions On Computational Biology And Bioinformatics,* 4, 40-52, 2007.

[8]XiaoweiSonga, Arnold Mitnitskib,c, JafnaCoxb, Kenneth Rockwood, **"Comparison of Machine Learning Techniques with Classical Statistical Models in Predicting Health Outcomes"**, *MEDINFO 2004 M. Fieschi et al. (Eds) Amsterdam: IOS Press © 2004 IMIA*