University of Victoria

ECE - 503

REPORT (Course Project)

Classification of Breast Cancer Dataset using PCA,Linear regression and Logistic regression

Submitted by

Sannath Reddy V (V00949217)

Introduction:

The project is about minimizing the classification errors by comparing the predicted labels produced by different classifiers which were part of the course. I have used breast cancer dataset which has 569 data instances and I will be applying 3 different algorithms to classify these data instances that has 30 features values from 569 patients. It is a binary classification as the patients are diagnosed as either benign or malignant. The entire dataset is divided into train and test instances as D_bc_tr and D_bc_te respectively. Here 569 instances have been divided into 480 train instances and 89 test instances to test the accuracy of each classifier that we create based on the train instances and their labels. Using the dataset, the problem of selection is to answer the correctness of the algorithms in predicting the labels. In other words, it is to test the different approaches that detect the benign and malignant patients based on the 30 features given in the dataset.

The algorithms which I worked on for the dataset are:

- Principal Component Analysis,
- Linear Model,
- Logistic Regression.

(Each of them is applied in the order of which they are taught in the course.)

As the predicted output can be either of the 2 classes (benign or malignant), the above-mentioned models can be considered as binary classifiers. As given in the lab experiment 4, the dataset contains radius as few of its 30 features (distances from centre to each instances) by which clustering of the data points in the space is already done. Hence, I have decided not to implement k-means clustering and have picked the other algorithms as the classification models. The output labels are binary values: -1 representing the features of the patient are recognised as malignant and 1 representing the features of the patient are recognised as benign. The first 30 values of an instance represent the 30 features of the patient and the 31st value gives the label of the instance which tells us whether the patient is malignant or benign. Initially, as a part of pre-processing, the train dataset is divided into benign and malignant instances in separate variables based on the labels as mentioned earlier.

As a part of experiment-4, the implementation of logistic regression on breast cancer dataset to classify the instances was already done. Now, the project tests the accuracy of the other algorithms in classifying the same dataset. The entire document is about figuring out which approach best suits the dataset which in turn gives the best algorithm out of the 3 algorithms learned in the course. Further section gives the technical details about each algorithm.

Technical details of Methods Implemented

1. Principal Component Analysis:

The main goal of PCA is to collect the features of every instance which are considered as dimensions in space and reduce the dimensions with minimum loss in actual data given. This is termed as feature extraction. The PCA gives the uncorrelated axes which are orthogonal to each other and represent the maximum variance in descending order.

About the method:

Given a dataset, PCA takes the dataset represented by a matrix with each row being considered as a dimension and each column as a data instance. For the breast cancer dataset, we have 30 dimensions(features) and 569 instances. When each instance is represented in the space as a vector, it turns out to be a complex problem dealing with 30 dimensions. This problem of high-dimensionality is termed as 'curse of dimensionality' which is addressed by PCA, a dimensionality reduction algorithm. Effective features that are observed in the lower dimensions are captured by the method to train the model that can classifies the incoming-unlabeled data.

PCA converts the dataset to new coordinate system in a way that the variances of the dataset lie on these coordinates. These coordinates are the principal components which are a result of the dimensionality reduction algorithm. First principal component gives the greatest variance by a scalar projection of the data and the variance decreases as the coordinates increases.
Eigen vectors and Eigen values are the core concepts for the construction of principal components in PCA. Briefly, eigen vectors give the direction of the new feature in space and eigen value gives the magnitude to that feature. Therefore, PCA gives the directions that maximizes the variance of data.

Steps in PCA:

Below are the main steps involved in the approach.

Pre-processing:
Calculating mean to centralize the data. This is done in order to remove the shift of data from origin which doesn't affect the uncorrelated axes produced by PCA.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

This step is applied all over the matrix for the breast cancer dataset i.e. for every dimension in the matrix.

Correlation/Covariance Matrix :

Covariance[1] gives the measure of how much each of the dimensions vary from the mean with respect to each other. It gives the relation between 2 dimensions represented in the matrix. Considering A as the given matrix of size $d \times m$, covariance is computed by:

$$C = \frac{1}{m} AA^T$$

where d is the number of dimensions and m is the number of instances.
Eigen values and Eigen vectors, in short SVD:
Computing 30 eigen vectors and corresponding eigen values of the covariance matrix obtained in the above step. This is done by considering the characteristic equation of the covariance matrix which is represented as:

$$I - \lambda \mathrm{C}$$

where 'I' is Identity matrix and takes the shape of covariance matrix.
Determinant of the characteristic equation gives the eigen values of the covariance matrix.
eigs() is the function used in matlab to compute the eigen values of a matrix and it returns the values in descending order along with the eigen vectors. These eigen values and eigen vectors form the basis for yielding principal components.
Matlab also provides svd() that returns left-singular, right singular matrices along with a diagonal matrix containing singular values for the given matrix.

Principal Components:

As explained earlier the need for principal components, they are computed using the following formula:

$$f_i = U_q^T(x_i - \mu)$$

where q is the number of dimensions and U represents the matrix with eigen vectors (called singular matrix) computed in the previous step.
f_i gives the q-dimensional features of the dataset.
Therefore, the input matrix is decomposed based on the dimensions and can be represented in each dimension as,

1-dimensional representation of A = $[U_1^T A]$

2-dimensional representation of A = $\begin{bmatrix} U_1^T A \\ U_2^T A \end{bmatrix}$

q-dimensional representation of A = $\begin{bmatrix} U_1^T A \\ U_2^T A \\ \vdots \\ U_q^T A \end{bmatrix}$

The above steps are performed on the breast cancer dataset that extracted features from 30 dimensions, produced principal components and resulted in classification of instances into either benign or malignant.

2.Linear Regression:

Linear model can be expressed as designing a best fit line (considered as plane if number of attributes >2) for the given dataset which are scattered across the space. For a multivariate dataset plotted in space, linear model predicts the class of an instance based on the weight vector and bias the model produces from the entire dataset.

Dimensions of the weight vector depends upon the number of features input dataset has. The two parameters, weight and bias, are optimized to achieve the best prediction. This is ensured by minimizing the loss function. Loss function gives the distance between the predicted value from the actual value of an instance. Therefore, by minimizing the loss/objective function, we are actually trying to get the predicted value closer to the actual value.

Steps in Linear Model:

Given a training dataset D = {(x_n,y_n), n=1,2,3,…N} with $x_n \epsilon\, R^{d\times 1}$ and $y_n\ \epsilon\ R^{1\times 1}$, a linear model is represented by following function:

$$f(x,W,b) = W^T x + b$$

where weight W $\epsilon\ R^{d\times 1}$ and bias $b\ \epsilon\ R^{1\times 1}$.

The above function is represented as compact form by considering,

$$\widehat{W} = [w1 \quad w2 \quad \cdots \quad wn \quad b]^{\ T}$$

$$\text{and} \quad \hat{x} = \begin{bmatrix} \widehat{x_1^T} \\ \widehat{x_2^T} \\ \vdots \\ \widehat{x_n^T} \end{bmatrix} \text{ where } \widehat{x_n} = \begin{bmatrix} x_n \\ 1 \end{bmatrix}$$

this transforms the function as

$$\mathrm{f(x)} = \widehat{W}.\hat{x}$$

With 30 features in the breast cancer dataset, weight vector (along with bias) is produced as a column matrix with 31 values.

Computing the above function for an unknown instance gives a scalar.

Based on the sign of the scalar we predict the label of the unknown instance as follows, When performing binary classification, the new point x which is outside the dataset D:

$$\text{belongs to class } \mathcal{P} \text{ if } x^T w^* + b^* > 0$$
$$\text{belongs to class } \mathcal{N} \text{ if } x^T w^* + b^* < 0$$

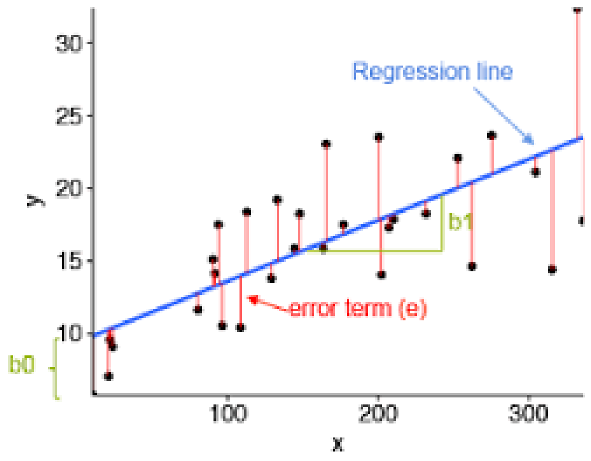
And the decision boundary is given by:

$$f(x) = w^{*T} x + b^* = 0$$

The above equation will define a flat plane which divides the input space into two parts. The points on the side of the plane will have *f(x)*values with same sign and the points on the other side of the plane will have *f(x)* values with opposite sign.

With the above implementation, classification for the breast cancer instances is done with the help of linear model by calculating the optimized weight vector and bias by minimizing the error function.

Below is the graph[2] that represents the linear modelling in 2D:



The output for the dataset is based on the sign of f(x) where negative values are considered as benign and positive values are considered as malignant.

3.<u>Logistic Regression</u>:

Logistic Regression is another kind of classifier which performs the same task of classifying instances based on the features in a different approach. Logistic regression is closely related to linear regression and perfectly fits the problem as the main purpose of logistic regression is categorical classification (binary, ternary, ..) whereas linear regression can also be used for the data when the response variable is continuous.

Steps in Logistic Regression:

Logistic regression uses sigmoid function as the objective function and computes the error or loss for the predicted values(labels) and uses conditional probability to classify instances.
Sigmoid function is given by,

$$h(x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

Given train data *D* = {(***x****ₙ*, *yₙ*), *n* = 1, 2, …, *N*} where label y_n takes value 1 for class *P* or –1 for class *N*, logistic regression models the conditional probability as ,

$$P({}^{y}/{}_{x})\begin{cases} h(x) & for\ y = 1 \\ 1 - h(x) & for\ \ y = \ -1 \end{cases}$$

Weight and bias of the above function represent the same functionality as mentioned in linear regression and can take the same compact form.

$$\widehat{W} = [w1 \quad w2 \ \cdots \ wn \quad b]^{\,T}$$

and $\hat{x} = \begin{bmatrix} \widehat{x_1^T} \\ \widehat{x_2^T} \\ \vdots \\ \widehat{x_n^T} \end{bmatrix}$ where $\widehat{x_n} = \begin{bmatrix} x_n \\ 1 \end{bmatrix}$

Considering the above compact form, the model increases its accuracy by minimizing the probability of observed data as,

$$\mathrm{f}(\widehat{w}) = \frac{1}{N}\sum \ln(1 + e^{-y_n \widehat{w}^T \hat{x}_n})$$

where N is included to make the calculations easier without affecting the results and y_n is label of the instance.
Optimizing weight vector and minimizing the error function is the goal of logistic regression by considering the conditional probabilities for the unknown instances which are out of the dataset. The above function can be further simplified to get the label of an instance by considering the sign of $\widehat{w}^T \hat{x}_n$ i.e.

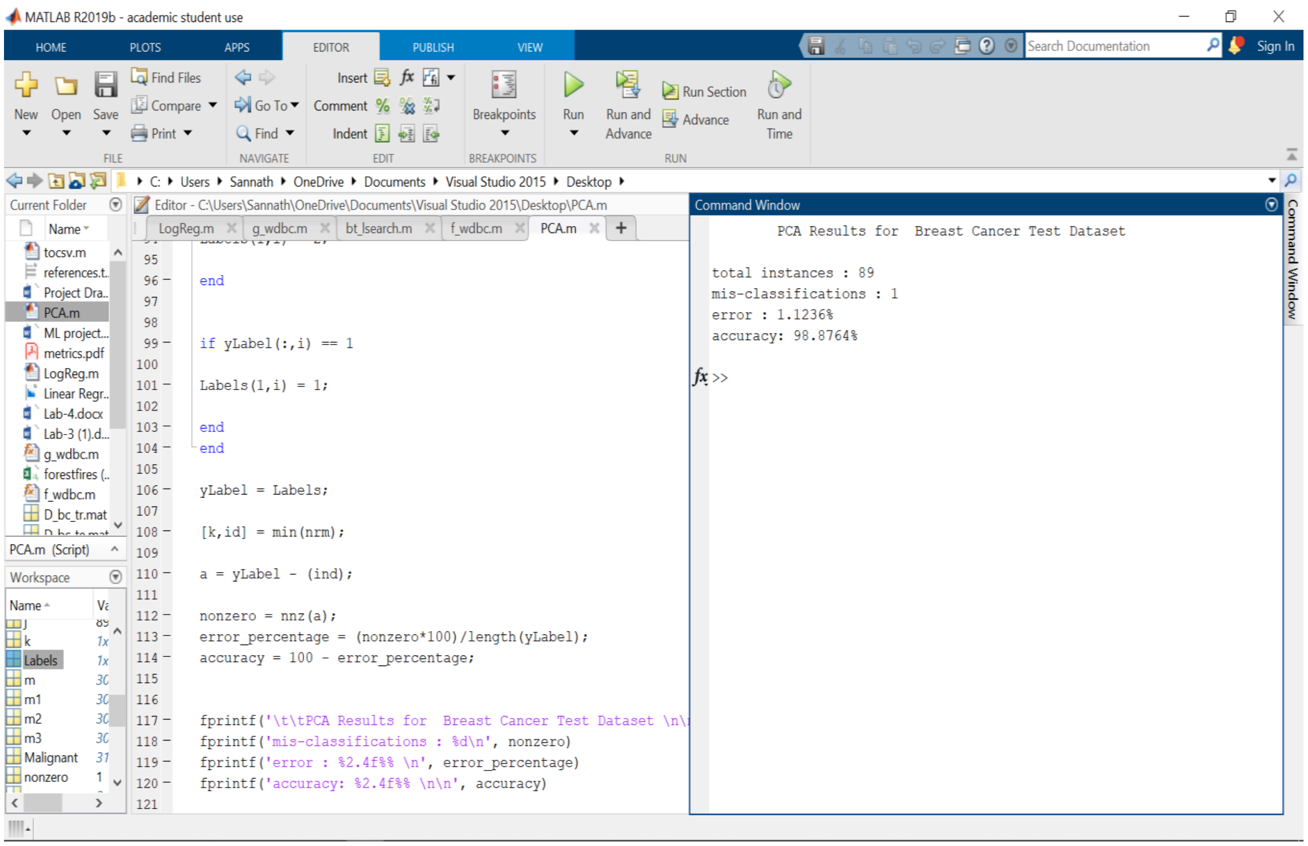
$$x \text{ belongs to} \begin{cases} class\ P\ if\ \ \widehat{w}^T \hat{x} > 0 \\ class\ N\ if\ \ \widehat{w}^T \hat{x} < 0 \end{cases}$$

Results of the classifiers on Breast Cancer dataset:

PCA output:

By implementing PCA on breast cancer dataset, the model was trained on the 480 data instances in D_bc_tr.mat and was tested on D_bc_te.mat for labels.

Out of 89 test instances, PCA successfully predicted the labels 88 instances correctly with the accuracy rate of 98.8%. The following is the output produced by PCA:



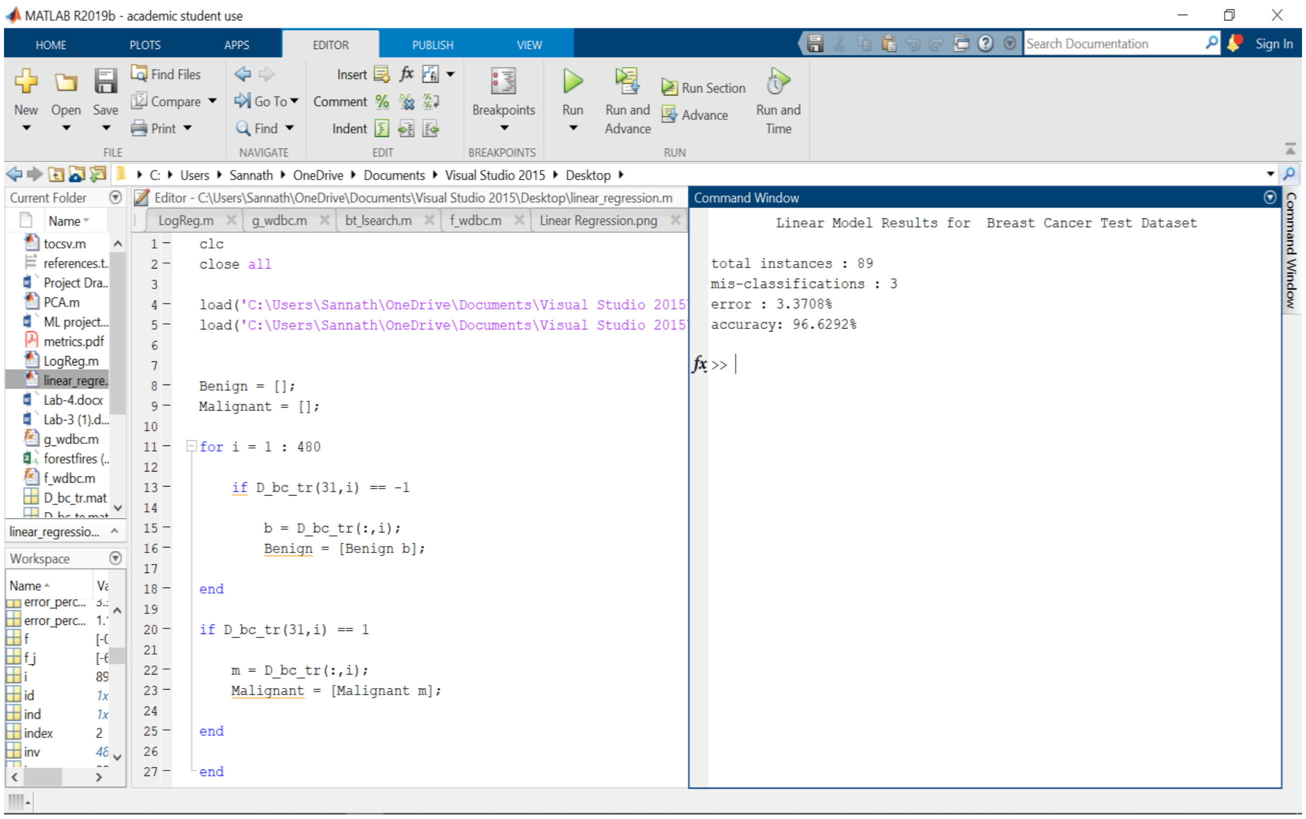
```
end


if yLabel(:,i) == 1

Labels(1,i) = 1;

end
end

yLabel = Labels;

[k,id] = min(nrm);

a = yLabel - (ind);

nonzero = nnz(a);
error_percentage = (nonzero*100)/length(yLabel);
accuracy = 100 - error_percentage;


fprintf('\t\tPCA Results for  Breast Cancer Test Dataset \n\
fprintf('mis-classifications : %d\n', nonzero)
fprintf('error : %2.4f%% \n', error_percentage)
fprintf('accuracy: %2.4f%% \n\n', accuracy)
```

```
        PCA Results for  Breast Cancer Test Dataset

total instances : 89
mis-classifications : 1
error : 1.1236%
accuracy: 98.8764%

>>
```

Linear Regression Output:

Linear model mis-classified 3 labels out of 89 test instances for the same dataset.

Below is the screenshot of the output of linear regression,



```
clc
close all

load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015
load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015


Benign = [];
Malignant = [];

for i = 1 : 480

    if D_bc_tr(31,i) == -1

        b = D_bc_tr(:,i);
        Benign = [Benign b];

end

if D_bc_tr(31,i) == 1

    m = D_bc_tr(:,i);
    Malignant = [Malignant m];

end

end
```

```
        Linear Model Results for  Breast Cancer Test Dataset

total instances : 89
mis-classifications : 3
error : 3.3708%
accuracy: 96.6292%

>>
```

<u>Logistic Regression</u>:

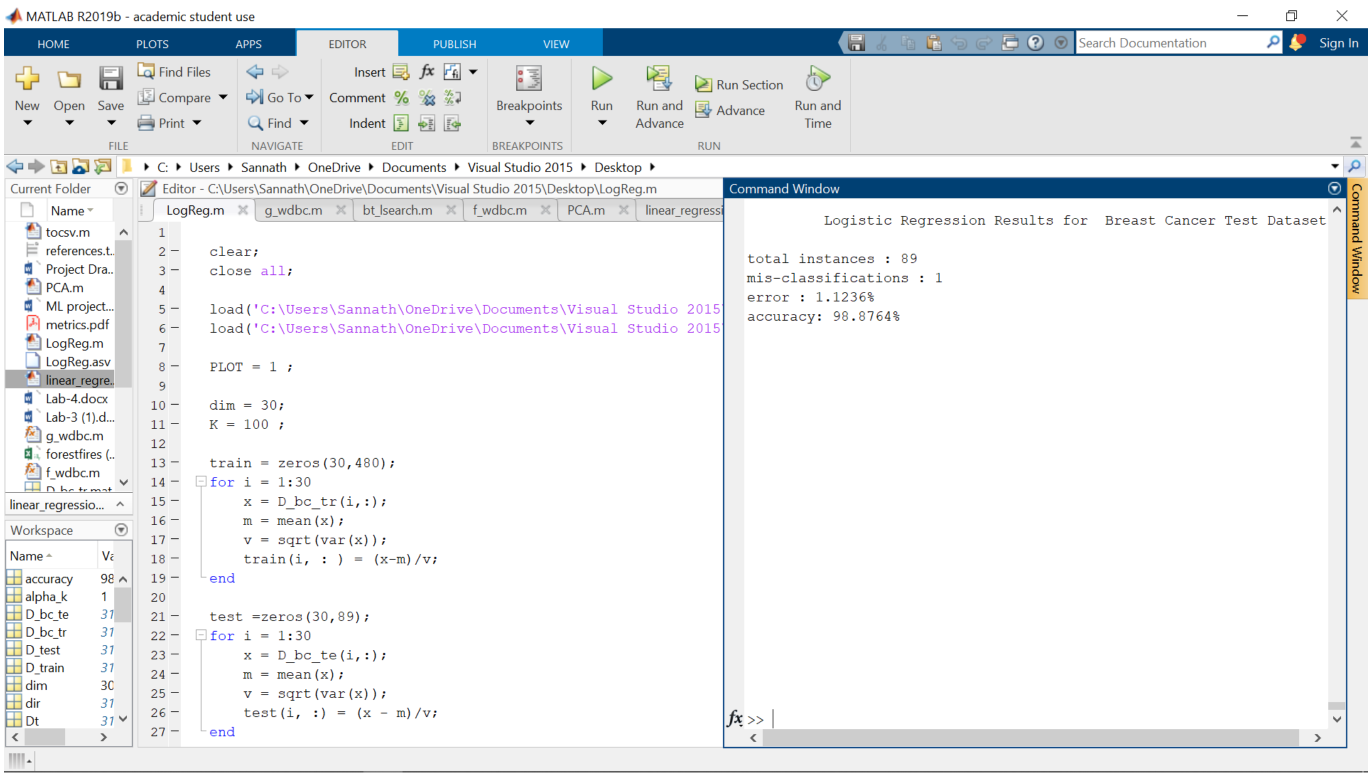
Reusing the code submitted for experiment-4, I have used 13 iterations (5<iteration<13 has same error rate). As we have observed that with a greater number of iterations, the classifier keeps getting better.

with k=1, it produced more than 50% error rate,

with k=5, wrongly predicted labels count came down to 2,

with k=13, we have only 1 mis-classified instance.

Below is the output screen for k=13,


MATLAB R2019b - academic student use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Editor - C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\LogReg.m
LogReg.m
g_wdbc.m
bt_lsearch.m
f_wdbc.m
PCA.m
Command Window

```
clear;
close all;

load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015
load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015

PLOT = 1 ;

dim = 30;
K = 100 ;

train = zeros(30,480);
for i = 1:30
    x = D_bc_tr(i,:);
    m = mean(x);
    v = sqrt(var(x));
    train(i, : ) = (x-m)/v;
end

test =zeros(30,89);
for i = 1:30
    x = D_bc_te(i,:);
    m = mean(x);
    v = sqrt(var(x));
    test(i, :) = (x - m)/v;
end
```

```
         Logistic Regression Results for  Breast Cancer Test Dataset

total instances : 89
mis-classifications : 1
error : 1.1236%
accuracy: 98.8764%
```

Conclusion:

Based on the outputs of the 3 algorithms, PCA and Logistic regression can be considered as the best among the 3 classifiers.
When performance is taken into, PCA is the best approach as logistic had to perform 13 iterations to yield the same results as PCA.

Accuracy comparison:

PCA	: 98.87%
Linear Model	: 96.62%
Logistic Regression	: 98.87% (for iterations=5)
	: 97.75% (for iterations=13)

MATLAB Code:

Principal Component Analysis:

```
clc
close all

load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\D_bc_tr.mat');
load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\D_bc_te.mat');

Benign = [];
Malignant = [];

for i = 1 : 480

  if D_bc_tr(31,i) == 1

    b = D_bc_tr(:,i);
    Benign = [Benign b];

  end

  if D_bc_tr(31,i) == -1

    m = D_bc_tr(:,i);
    Malignant = [Malignant m];

  end

end

X = [Benign(1:30,:) Malignant(1:30,:)];

d = 48;
q = 8;
m1 = [];
u = [];

m2 = [];
u1 = [];

m3 = [];
u2 = [];

for j = 0 : 1
```

```
if j == 0

d = 180;
X1 = X(:,1:180);
m = mean(X1,2);
A = X1 - m;
C = 1/d*(A*A');

[uq, S] = eigs(C,q);
m2 = [m2 m];
u1 = [u1 uq];

end

if j == 1

d = 300;
X1 = X(:,181:end);
m = mean(X1,2);
A = X1 - m;
C = 1/d*(A*A');

[uq, S] = eigs(C,q);
m3 = [m3 m];
u2 = [u2 uq];

end
end

u = [u1 u2];
m1 = [m2 m3];
test_bc = D_bc_te(1:30,:);

for j = 1:89
for i = 0:1

Uq = u(:,(i*q)+1:(i+1)*q);
f_j = Uq'*(test_bc(:,j)-m1(:,i+1));
x_vec = Uq*f_j+m1(:,i+1);
nrm(i+1,j) = norm(test_bc(:,j)-x_vec);

end
end
```

```
yLabel = D_bc_te(31,:);
Labels = zeros(1,89);

for i = 1 : 89

if yLabel(:,i) == -1

Labels(1,i) = 2;

end


if yLabel(:,i) == 1

Labels(1,i) = 1;

end
end

yLabel = Labels;

[k,id] = min(nrm);

a = yLabel - (ind);

nonzero = nnz(a);
error_percentage = (nonzero*100)/length(yLabel);
accuracy = 100 - error_percentage;


fprintf('\t\tPCA Results for  Breast Cancer Test Dataset \n\ntotal instances : 89\n')
fprintf('mis-classifications : %d\n', nonzero)
fprintf('error : %2.4f%% \n', error_percentage)
fprintf('accuracy: %2.4f%% \n\n', accuracy)
```

<u>Linear Regression</u>:

```
clc
close all

load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\D_bc_tr.mat');
load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\D_bc_te.mat');


Benign = [];
Malignant = [];

for i = 1 : 480

  if D_bc_tr(31,i) == -1

     b = D_bc_tr(:,i);
     Benign = [Benign b];

end

if D_bc_tr(31,i) == 1

  m = D_bc_tr(:,i);
  Malignant = [Malignant m];

end

end

x1 = Benign(1:30,:);
x2 = Malignant(1:30,:);
y1 = [ones(300,1);-ones(180,1)];
y2 = [ones(180,1);-ones(300,1)];

x1_A = [x1 x2];
x1_A(31,:) = [ones(1,300) ones(1,180)];

inv = x1_A';
w1_vec = pinv(inv)*y1;
w1 = w1_vec(1:30,:);
b1 = w1_vec(31,:);

x2_A = [x2 x1];
```

```
x2_A(31,:) = [ones(1,180) ones(1,300)];

inv = x2_A';
w2_vec = pinv(inv)*y2;
w2 = w2_vec(1:30,:);
b2 = w2_vec(31,:);

weight = [w1 w2];
bias = [b1;b2];

Label = D_bc_te(31,:);

L_1 = [];
L_2 = [];
test1 = [];
test2 = [];

for i = 1 : 89

   if Label(:,i) == -1

      l2  = -1;
      L_2 = [L_2 l2];
      test = D_bc_te(1:30,i);
      test1 = [test1 test];

   end

   if Label(:,i) == 1

      l1  = 1;
      L_1 = [L_1 l1];
      test = D_bc_te(1:30,i);
      test2 = [test2 test];

   end

end

L = [L_1 L_2];
XTe = [test1 test2];

test = [XTe];
Label = [ones(1,57) ones(1,32)+1];
```

```
ek = zeros(2,89);
predict_wrong = 0;

for col = 1:89

   f = weight'*test(:,col)+bias;
   [maxvalue index] = max(f);
   ek(index,col)  = 1;

   if index ~= Label(col)

       predict_wrong = predict_wrong +1;

   else

   end

end


ek;
predict_wrong;

error_percent = predict_wrong/89*100;
accuracy = 100 - error_percent;


fprintf('\t\tLinear Model Results for  Breast Cancer Test Dataset \n\ntotal instances : 89\n')
fprintf('mis-classifications : %d\n', predict_wrong)
fprintf('error : %2.4f%% \n', error_percent)
fprintf('accuracy: %2.4f%% \n\n', accuracy)
```

<u>Logistic Regression</u>:

```
clear;
close all;

load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\D_bc_tr.mat');
load('C:\Users\Sannath\OneDrive\Documents\Visual Studio 2015\Desktop\D_bc_te.mat');

PLOT = 1 ;

dim = 30;
K = 13 ;

train = zeros(30,480);
for i = 1:30
   x = D_bc_tr(i,:);
   m = mean(x);
   v = sqrt(var(x));
   train(i, : ) = (x-m)/v;
end

test =zeros(30,89);
for i = 1:30
   x = D_bc_te(i,:);
   m = mean(x);
   v = sqrt(var(x));
   test(i, :) = (x - m)/v;
end

y_train = D_bc_tr(31,:);
y_test = D_bc_te(31,:);

D_train = [train; y_train];
D_test = [test; y_test];


eps = 10^-9;
W = zeros(dim+1,1);

h = zeros(1,K);
k=1;

gk = g_wdbc(W, D_train);
dir = -gk;
```

```
alpha_k = bt_lsearch(W,dir,'f_wdbc','g_wdbc', D_train);

while(norm(alpha_k*dir) >=eps && (k<K))

   W = W +alpha_k*dir;
   gk = g_wdbc(W,D_train);
   dir = -gk;
   alpha_k = bt_lsearch(W,dir,'f_wdbc','g_wdbc',D_train);
   h(k) = f_wdbc(W,D_train);
   k=k+1;

end

Dt = [test; ones(1,89)];
Result = W'*Dt;
TestLabel = zeros(1,length(Result));
FalsePos = 0
FalseNeg = 0 ;

for ii = 1:length(Result)
   if Result(ii)<0
      TestLabel(ii) = -1;
      if y_test(ii) > 0
         FalseNeg = FalseNeg + 1;
      end
   else
      TestLable(ii) = 1;
      if y_test(ii) < 0
         FalsePos = FalsePos + 1;
      end
   end
end


error_percent = (FalsePos/89)*100;
accuracy = 100-error_percent;
fprintf('\t\t Logistic Regression Results for  Breast Cancer Test Dataset \n\ntotal instances :
89\n')
fprintf('mis-classifications : %d\n', FalsePos)
fprintf('error : %2.4f%% \n', error_percent)
fprintf('accuracy: %2.4f%% \n', accuracy)
```

```
function g = g_wdbc(w,D)
X = D(1:30, :);
y = D(31, :);
N = length(y)
g = zeros(31,1);
wt = w(:)';
for i = 1:N
   xi = [X(:,i);1];
   ei = exp(-y(i)*(wt*xi));
   ci = -y(i)*ei/(1+ei);
   gi = ci*xi;
   g = g + gi;
end
g= g/N;
```

```
function a = bt_lsearch(x,s,F,G,p1,p2)
rho = 0.1;
gma = 0.5;
x = x(:);
s = s(:);
a = 1;
parameterstring = '';

if nargin > 4,
   if ischar(p1),
      eval([p1 ';']);
   else
      parameterstring =',p1';
   end
end
```

```
function f = f_wdbc(w,D)

X = D(1:30,:);
y = D(31,:);
N = length(y);
f = 0;
wt = w(:)';
```

```
for i = 1:N
    xi = [X(:,i);1];
    fi = log(1+exp(-y(i)*(wt*xi)));
    f = f + fi;
end
f = f/N
```

References:

[1] http://www.cse.psu.edu/~rtc12/CSE586/lectures/pcaLectureShort_6pp.pdf

[2]http://www.sthda.com/english/articles/40-regression-analysis/167-simple-linear-regression-in-r/

[3] 'ECE-403/ 503-Course Notes', by Dr. Wu-Sheng Lu, 2019.

[4] 'ECE-403/ 503-Lab Manual', by Dr. Wu-Sheng Lu, 2019.