# Report

## Data Mining CSE 572 Spring 2018

**Submitted to:**

**Prof. Ayan Banerjee**

**Ira A. Fulton School of Engineering**

**Arizona State University**

# Table of Contents

## 1. Introduction

The aim of this assignment is to carry out **user dependent and user independent analysis** using conventional classification methods like decision trees, support vector machines and neural networks on specific movement specific gesture data collected using the sensors. The data set under consideration is with conjecture with the results obtained after doing pca for the data as done in the previous phase. The data set is made suitable for creating a new feature matrix by using feature extraction methods. Then various classification techniques like neural networks, support vector machines and decision trees are applied on the new feature matrix. Henceforth, the values obtained for Accuracy, Precision, Recall and F1 measure from each classification technique are made use of to conclude the best method to identify the gestures.

## 3. Data Preprocessing

According to the ground truth folder provided in the data set. We identified the start and end rate of the eating action. We considered the formula (fps*frame start)/time elapsed and (fps*frame end)/time elapsed to obtain the starting and ending index of eating action and we labeled it as 1, where 1 is the label that is used to represent the eating data. Initially all the data is labelled 0 where 0 represents the label of non-eating data. Now after the above process is done, we obtain the class labels corresponding to each action performed by the user.

In the formula used,

Frame per second is 50
Time elapsed is 30 sec

### PCA Algorithm

- Calculate the covariance of the matrix of the data of the given data set
- calculate the eigen values of the covariance matrix
- identify the top 5 eigen values and take the corresponding eigen vectors

- the eigen vectors are multiplied with the covariance matrix to obtain the principal components of the data.

## Evaluate function:

- We used Evaluate function from the MATLAB site for calculating the accuracy measures. We have included the code of it along with the normal code.

## 4. Creation of New Feature Matrix

- **For phase 2 feature matrix**

  PCA is applied on the user separately and PCA components are obtained and the label obtained in the preprocessing is also saved to the new feature matrix. The same is repeated for every other user. Later it is passed to the classification methods such that 60 percent is for training and other 40 percent for testing

- **For phase 3 feature matrix**

  PCA is applied on the user by concatenating all the data and PCA components for the whole data are obtained. the label obtained in the preprocessing is also saved to the new feature matrix. Later it is passed to the classification methods such that 60 percent is for training and other 40 percent for testing.

# 5. Classification Techniques

## 5. 1 Terminology

**Confusion Matrix:**

A clean and unambiguous way to present the prediction results of a classifier is to use a confusion matrix.

It is needful to mention that True positive and true negatives are the observations which are vital for this analysis. We want to minimize false positives and false negatives.

True Positives (TP) - Correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

True Negatives (TN) - Correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

False positives and false negatives, these values generally come into existence when your actual class contradicts with the predicted class.

False Positives (FP) – When actual class is no and predicted class is yes.

False Negatives (FN) – When actual class is yes but predicted class in no.

Henceforth, we can calculate Accuracy, Precision, Recall and F1 score.

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Accuracy is computed in the following way.

**Accuracy = TP+TN/TP+FP+FN+TN**

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Precision is computed as follows. Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual Recall is computed as follows Recall = TP/TP+FN

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes into consideration both false positives and false negatives.

F1Score=2*(Recall*Precision)/(Recall+Precision)



## 5.2 Decision Trees

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

Decision tree is a conventionally used data mining method for modelling classification systems based on multiple parameters or for solving predictive problems. The technique is non-parametric and can efficiently deal with large, complicated datasets without imposing a complicated parametric structure. Decision trees can handle both categorical and numerical data.

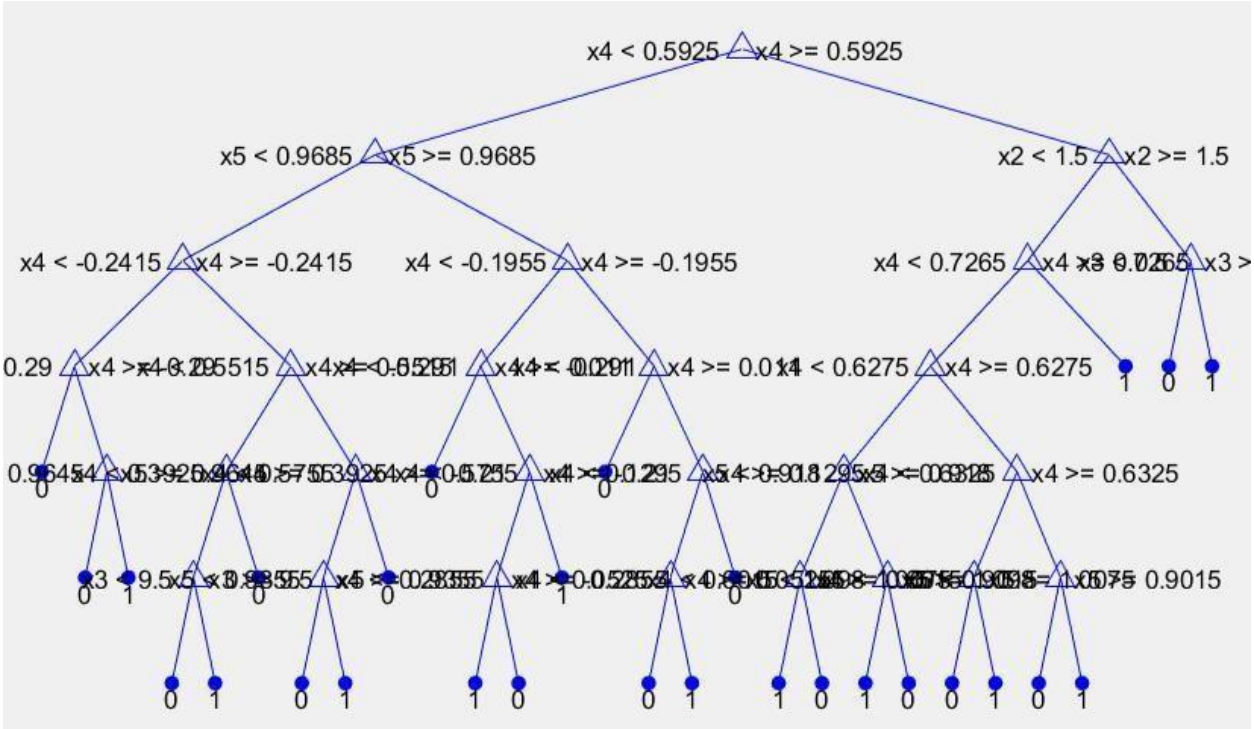
For the dataset under our consideration we have considered 60% of the data to be training dataset and 40% of the data to be test data.

Here are the few decision tree samples

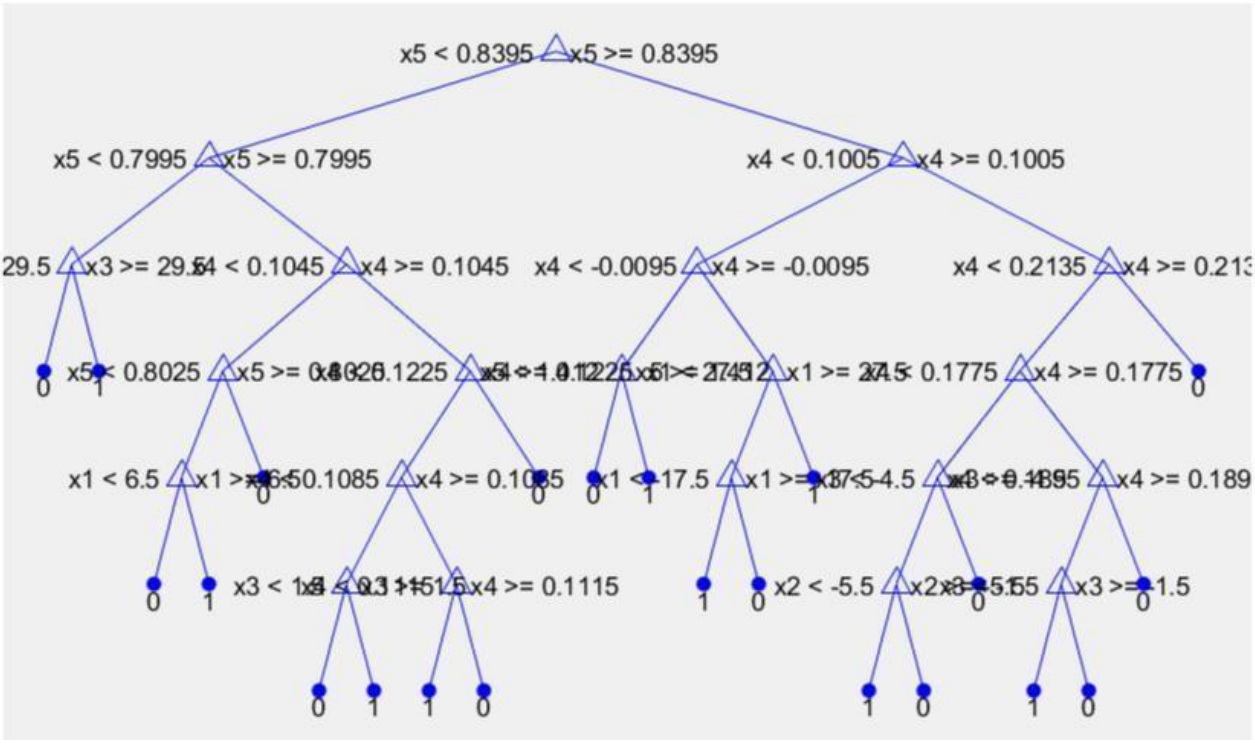


# Phase 2

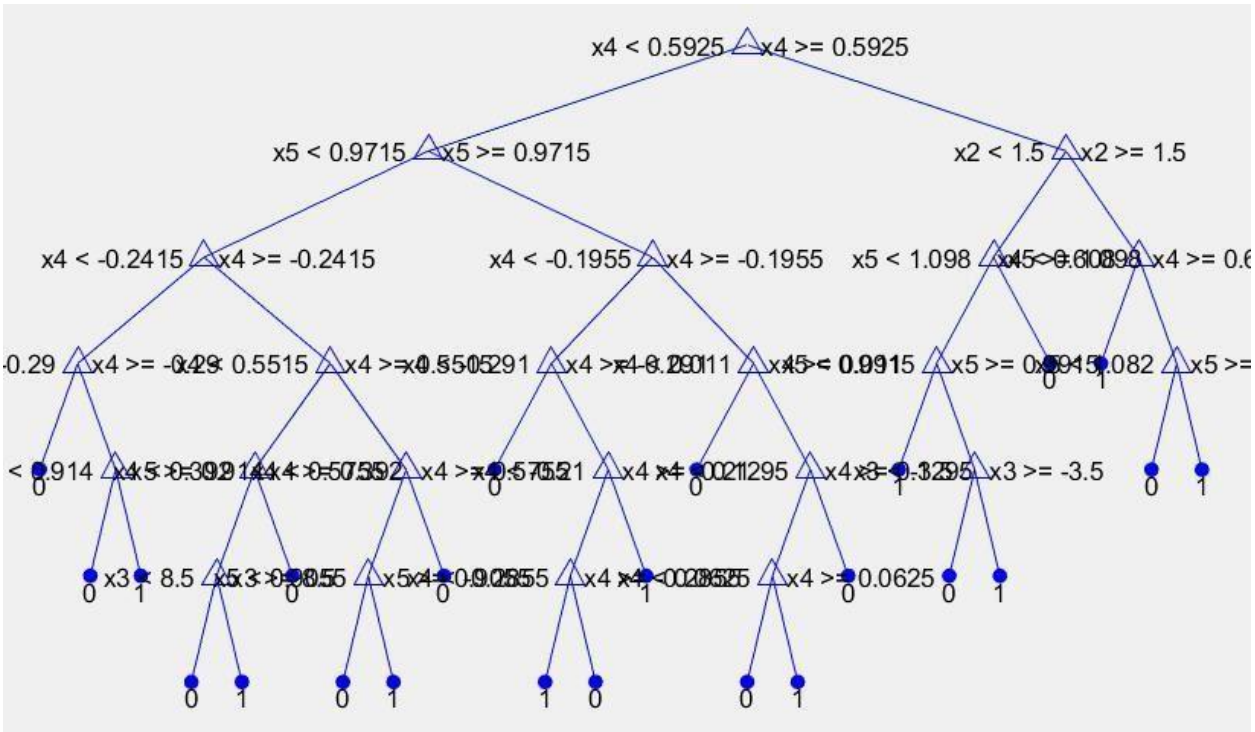
User 40


x4 < 0.5925
x4 >= 0.5925
x5 < 0.9685
x5 >= 0.9685
x2 < 1.5
x2 >= 1.5
x4 < -0.2415
x4 >= -0.2415
x4 < -0.1955
x4 >= -0.1955
x4 < 0.7265
x3 >
0.29
x4 < 0.6275
x4 >= 0.6275
x4 >= 0.6325
0.9015

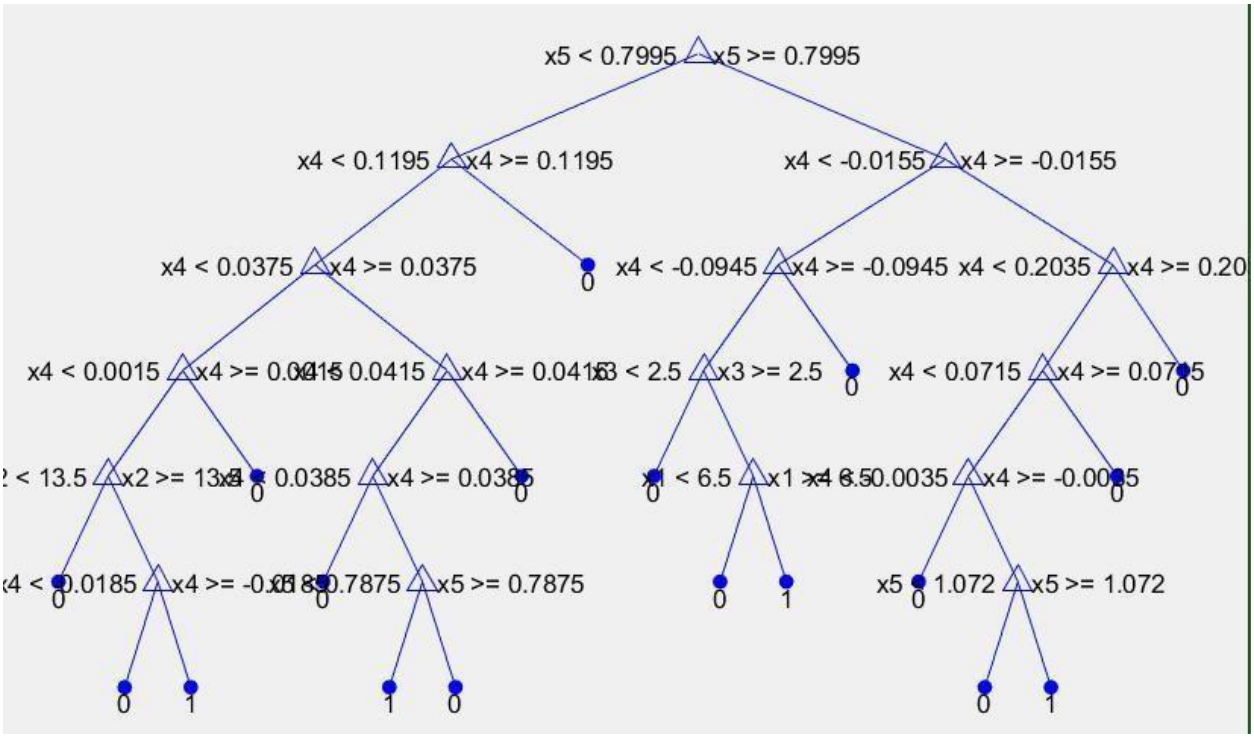
User 36


x5 < 0.8395
x5 >= 0.8395
x5 < 0.7995
x5 >= 0.7995
x4 < 0.1005
x4 >= 0.1005
x4 < 0.1045
x4 >= 0.1045
x4 < -0.0095
x4 >= -0.0095
x4 < 0.2135
x4 >= 0.1775
x1 < 6.5
x4 >= 0.1085
x2 < -5.5
x3 >= -1.5

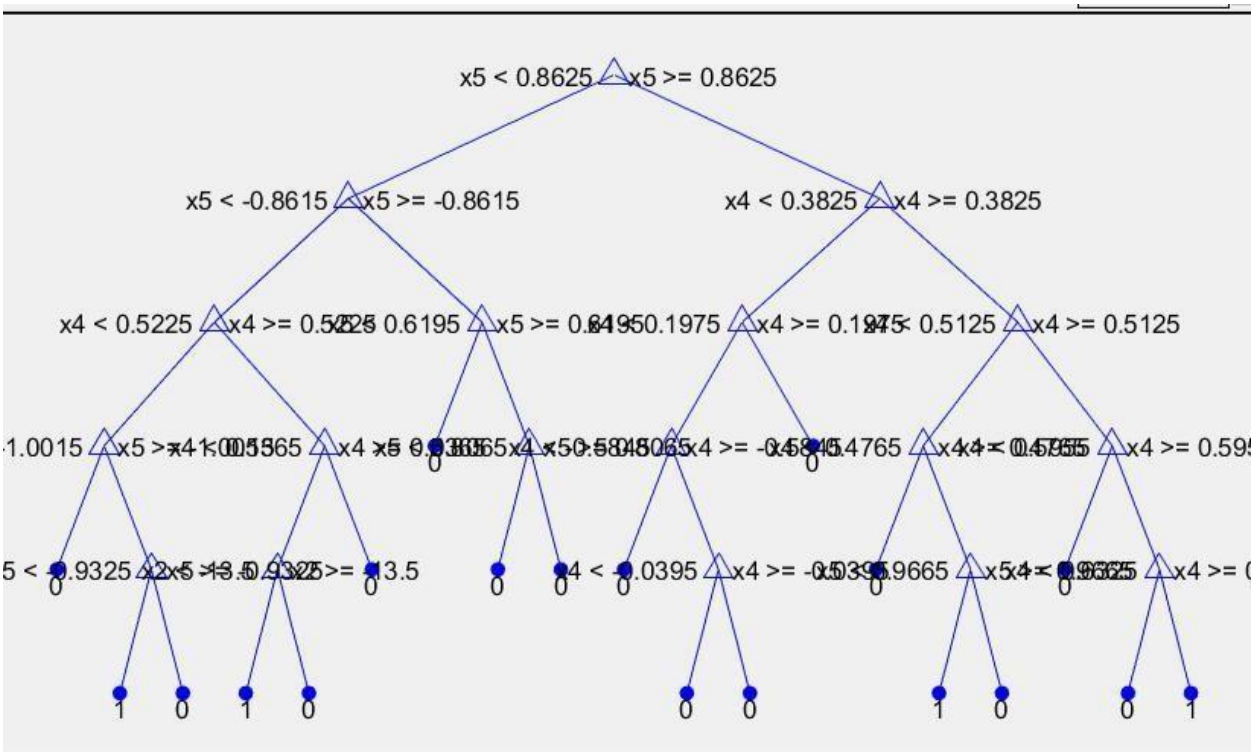
User 9


x4 < 0.5925
x4 >= 0.5925
x5 < 0.9715
x5 >= 0.9715
x2 < 1.5
x2 >= 1.5
x4 < -0.2415
x4 >= -0.2415
x4 < -0.1955
x4 >= -0.1955
x3 >= -3.5
x3 < 8.5
x4 >= 0.0625

User 9


x5 < 0.7995
x5 >= 0.7995
x4 < 0.1195
x4 >= 0.1195
x4 < -0.0155
x4 >= -0.0155
x4 < 0.0375
x4 >= 0.0375
0
x4 < -0.0945
x4 >= -0.0945
x4 < 0.2035
x4 >= 0.20
x4 < 0.0015
x4 >= 0.0015
0.0415
x4 >= 0.0415
x3 < 2.5
x3 >= 2.5
0
x4 < 0.0715
x4 >= 0.0715
0
< 13.5
x2 >= 13.5
0
0.0385
x4 >= 0.0385
0
0
6.5
0.0035
x4 >= -0.0035
0
0.0185
0
0.7875
x5 >= 0.7875
0
0
1
1.072
x5 >= 1.072
0
0
1
1
0
1
0
1

| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER 9 | 0.95 | 0.34665 | 0 | 0 |
| USER 10 | 0.785 | 0.29155 | 0.25 | 0.22545 |
| USER 11 | 0.85 | 0.12266 | 0.5 | 0.54545 |
| USER 12 | 0.875 | 0 | 0.625 | 0 |
| USER 13 | 0.925 | 0 | 0 | 0.66667 |
| USER 14 | 0.775 | 0.32555 | 0.25 | 0.25666 |
| USER 16 | 0.9625 | 0.33333 | 0.25 | 0.88889 |
| USER 17 | 0.8725 | 0.23455 | 0.5 | 0.11111 |
| USER 18 | 0.8 | 0.33333 | 0.625 | 0 |
| USER 19 | 0.925 | 0.23525 | 0.75 | 0 |
| USER 21 | 0.8625 | 0.31255 | 0 | 0.23325 |
| USER 22 | 0.9375 | 0.26525 | 0 | 0 |
| USER 23 | 0.86 | 0 | 0.5 | 0 |
| USER 24 | 0.9125 | 0.14555 | 0.25 | 0.66244 |
| USER 25 | missing | missing | missing | missing |
| USER 26 | 0.9125 | 0.33333 | 0 | 0.82255 |
| USER 27 | 0.7625 | 0.21345 | 0.5 | 0.54542 |
| USER 28 | 0.7875 | 0.34565 | 0 | 0 |
| USER 29 | 0.9125 | 0.11285 | 0.625 | 0 |
| USER 30 | 0.9 | 0.45639 | 0.5 | 0.11111 |
| USER 31 | 0.9 | 0.23666 | 0.5 | 0 |
| USER 32 | 0.8765 | 0.18333 | 0.625 | 0.24445 |
| USER 33 | 0.9225 | 0 | 0.25 | 0.33333 |
| USER 34 | 0.8265 | 0.24555 | 0 | 0.24554 |
| USER 36 | 0.8 | 0.31255 | 0 | 0.11133 |
| USER 37 | 0.85 | 0.32111 | 0.25 | 0.32323 |
| USER 38 | 0.9225 | 0.6 | 0 | 0.81115 |
| USER 39 | 0.7675 | 0 | 0.5 | 0 |
| USER 40 | 0.8125 | 0.33333 | 0.625 | 0 |
| USER 41 | 0.9265 | 0.24555 | 0.625 | 0.11255 |




**Phase 3**

| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER 9 | 0.723 | 0.44 | 0.40 | 57 |

## 5.3 Support Vector Machines

Support Vector machines(SVM) are supervised learning models with related learning algorithms that analyze data used for specific cases of regression analysis and classification analysis. A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two known classes. Given a set of training examples an SVM training algorithm builds a model that assigns new examples, making it a nonprobabilistic binary linear classifier. Initially we define an optimal hyperplane: maximize margin. Then we extend the above definition for non-linearly distinguishable problems: have a penalty term for misclassifications. Lastly, we map data to high dimensional space where it is easier to classify with linear decision surfaces: re-inculcate problem so that data is mapped implicitly to this space.

Below tables provides the values of Accuracy, Precision, Recall and F1 Score

of different groups obtained using SVM.

# Phase 2

| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER 9 | 0.85 | 0.24655 | 0.25 | 0 |
| USER 10 | 0.725 | 0.29155 | 0 | 0.12445 |
| USER 11 | 0.8825 | 0 | 0 | 0.14995 |
| USER 12 | 0.9125 | 0.12225 | 0.625 | 0.12234 |
| USER 13 | 0.925 | 0 | 0.25 | 0.66667 |
| USER 14 | 0.8 | 0.23455 | 0 | 0 |
| USER 16 | 0.9625 | 0.33333 | 0.2 | 0.88889 |
| USER 17 | 0.7335 | 0.32225 | 0.5 | 0.22555 |
| USER 18 | 0.9 | 0.3 | 0.625 | 0.88997 |
| USER 19 | 0.825 | 0.13335 | 0.5 | 0.43335 |
| USER 21 | 0.9625 | 0.12245 | 0 | 0.23325 |
| USER 22 | 0.9225 | 0.23333 | 0.625 | 0.11875 |
| USER 23 | 0.9 | 0.13333 | 0.25 | 0.22455 |
| USER 24 | 0.8125 | 0 | 0 | 0 |
| USER 25 | missing | missing | missing | missing |
| USER 26 | 0.8 | 0.12234 | 0 | 0.72255 |
| USER 27 | 0.7625 | 0.31245 | 0.5 | 0.44455 |
| USER 28 | 0.8725 | 0.34565 | 0.625 | 0.22445 |
| USER 29 | 0.9225 | 0 | 0 | 0.33333 |
| USER 30 | 0.8 | 0.41233 | 0 | 0 |
| USER 31 | 0.8 | 0.14555 | 0.5 | 0.11111 |
| USER 32 | 0.9125 | 0.33333 | 0.25 | 0.24445 |
| USER 33 | 0.8665 | 0.12225 | 0.5 | 0.12245 |
| USER 34 | 0.7625 | 0 | 0.625 | 0.33333 |
| USER 36 | 0.8 | 0.31255 | 0.5 | 0.14555 |
| USER 37 | 0.95 | 0.12245 | 0.25 | 0 |
| USER 38 | 0.9225 | 0 | 0.625 | 0.24455 |
| USER 39 | 0.7675 | 0.6 | 0.5 | 0.33333 |
| USER 40 | 0.8125 | 0.31255 | 0 | 0 |
| USER 41 | 0.8165 | 0.24555 | 0.5 | 0 |
| | | | | |

## Phase 3

| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER 9 | 0.9125 | 0.23250 | 0 | 0.11345 |

## 5.4 Neural Networks

A neural network, is a mathematical model idealized by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist process to computation. In most cases a neural network is an adaptive system that transforms its structure during a learning phase. Neural networks are used to model interconnected relationships between inputs and outputs or to find different patterns in data.

**Accuracy:**

For our model on an average, we have got which means our model is approx.

**Precision:**

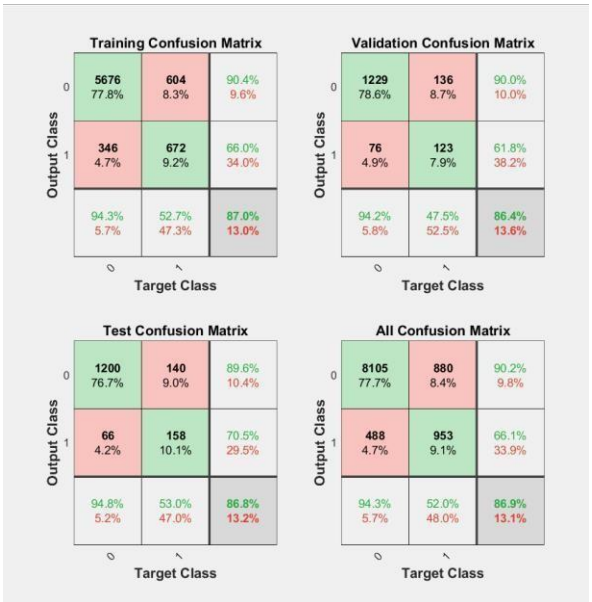
On an average we have got precision value as

**Recall:**

For our model on an average we have got Recall value as

**F1 Score:**

For our model on an average we got F1 Score value



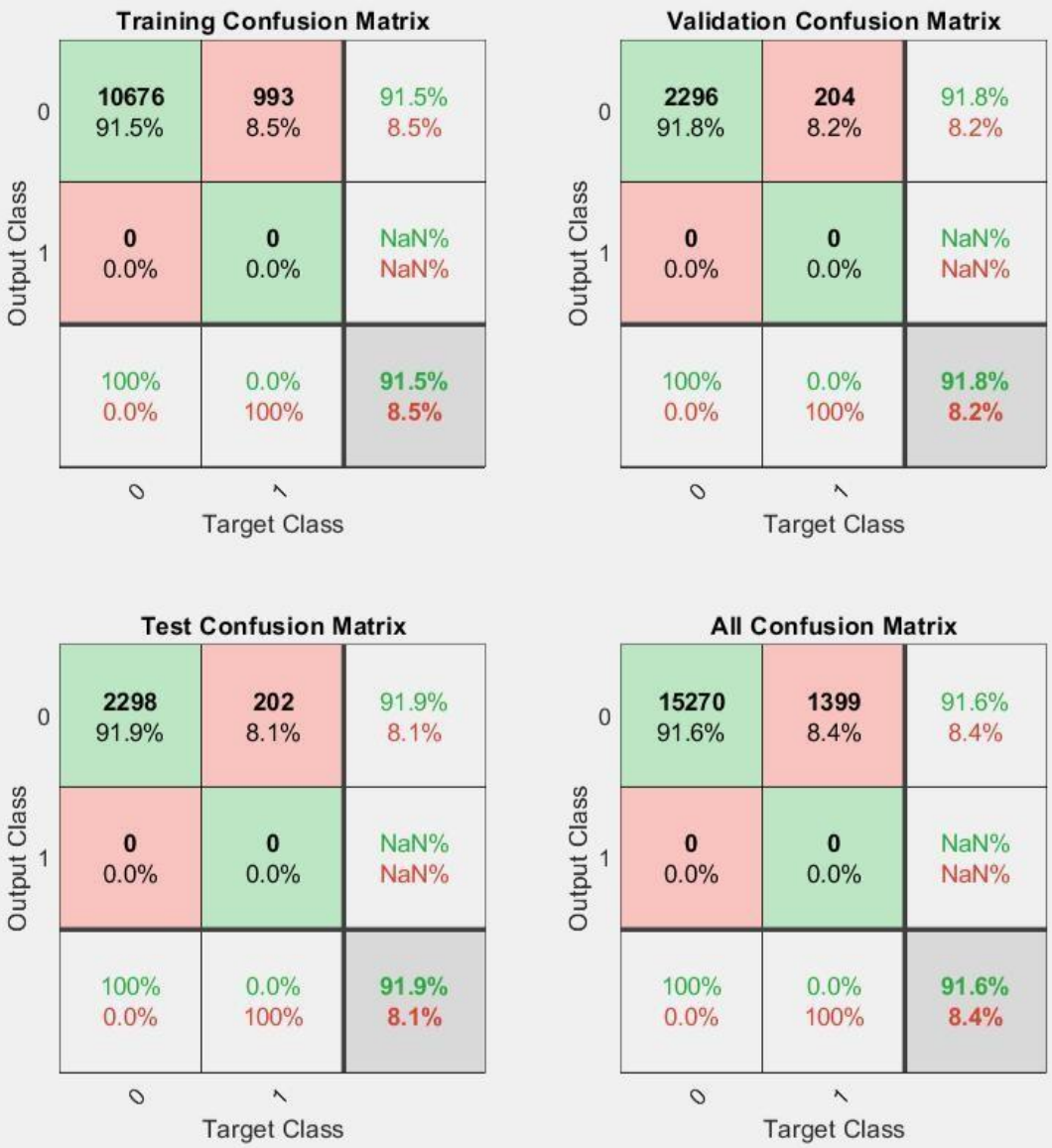
Below sample images gives the values of Confusion Matrix, Accuracy, Precision, Recall and F1 Score of one group obtained using neural networks.

# Phase 2

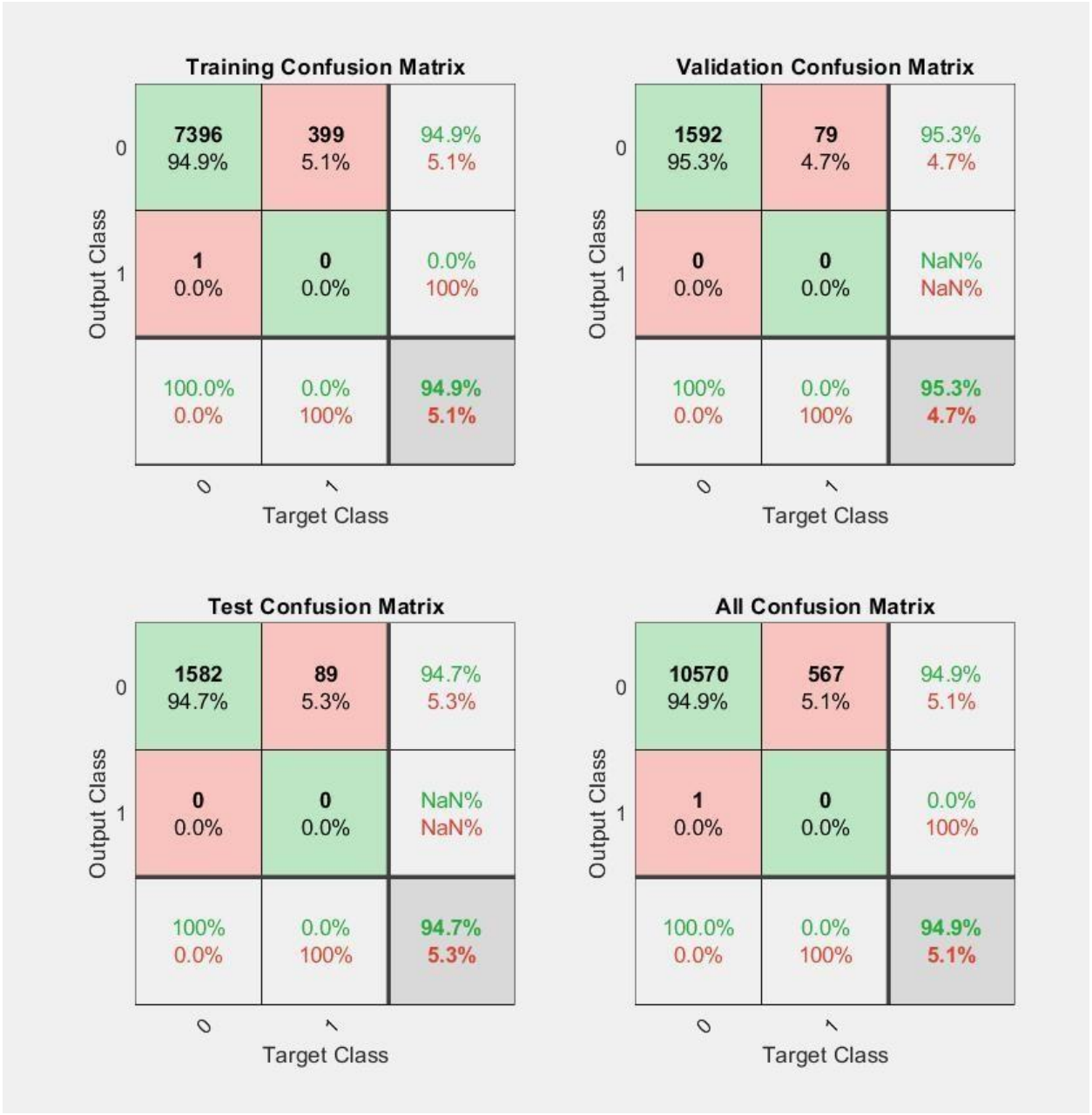
**User 38**



**User 36**


Training Confusion Matrix
10676
91.5%
993
8.5%
91.5%
8.5%
0
0.0%
0
0.0%
NaN%
NaN%
100%
0.0%
0.0%
100%
91.5%
8.5%
Output Class
Target Class
Validation Confusion Matrix
2296
91.8%
204
8.2%
91.8%
8.2%
0
0.0%
0
0.0%
NaN%
NaN%
100%
0.0%
0.0%
100%
91.8%
8.2%
Output Class
Target Class
Test Confusion Matrix
2298
91.9%
202
8.1%
91.9%
8.1%
0
0.0%
0
0.0%
NaN%
NaN%
100%
0.0%
0.0%
100%
91.9%
8.1%
Output Class
Target Class
All Confusion Matrix
15270
91.6%
1399
8.4%
91.6%
8.4%
0
0.0%
0
0.0%
NaN%
NaN%
100%
0.0%
0.0%
100%
91.6%
8.4%
Output Class
Target Class

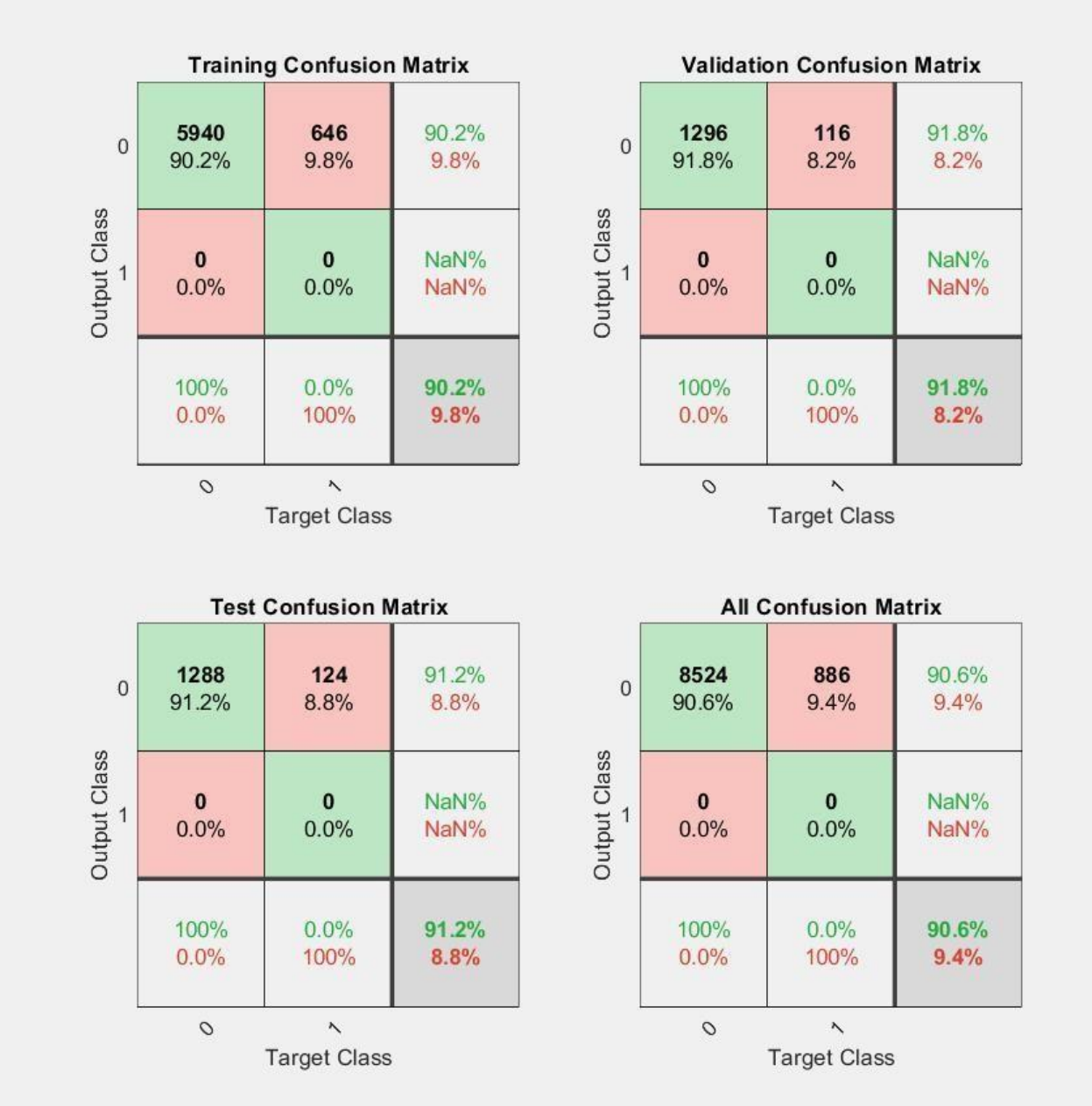
User 39
.


Training Confusion Matrix
7396
94.9%
399
5.1%
94.9%
5.1%
1
0.0%
0
0.0%
0.0%
100%
100.0%
0.0%
0.0%
100%
94.9%
5.1%
Output Class
Target Class
Validation Confusion Matrix
1592
95.3%
79
4.7%
95.3%
4.7%
0
0.0%
0
0.0%
NaN%
NaN%
100%
0.0%
0.0%
100%
95.3%
4.7%
Output Class
Target Class
Test Confusion Matrix
1582
94.7%
89
5.3%
94.7%
5.3%
0
0.0%
0
0.0%
NaN%
NaN%
100%
0.0%
0.0%
100%
94.7%
5.3%
Output Class
Target Class
All Confusion Matrix
10570
94.9%
567
5.1%
94.9%
5.1%
1
0.0%
0
0.0%
0.0%
100%
100.0%
0.0%
0.0%
100%
94.9%
5.1%
Output Class
Target Class

User 40

| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER 9 | 0.9125 | 0.23250 | 0 | 0.11345 |
| USER 10 | 0.8 | 0.2 | 0.5 | 0.12445 |
| USER 11 | 0.9 | 0.33333 | 0 | 0.24685 |
| USER 12 | 0.8225 | 0.12225 | 0.25 | 0.82235 |
| USER 13 | 0.8155 | 0.14555 | 0.3 | 0 |
| USER 14 | 0.7955 | 0 | 0.625 | 0.12245 |
| USER 16 | 0.9345 | 0.12255 | 0.25 | 0.65645 |
| USER 17 | 0.9 | 0.23355 | 0.2 | 0 |

| | | | | |
|---|---|---|---|---|
| USER 18 | 0.8 | 0.32355 | 0.5 | 0.24465 |
| USER 19 | 0.815 | 0 | 0 | 0.34345 |



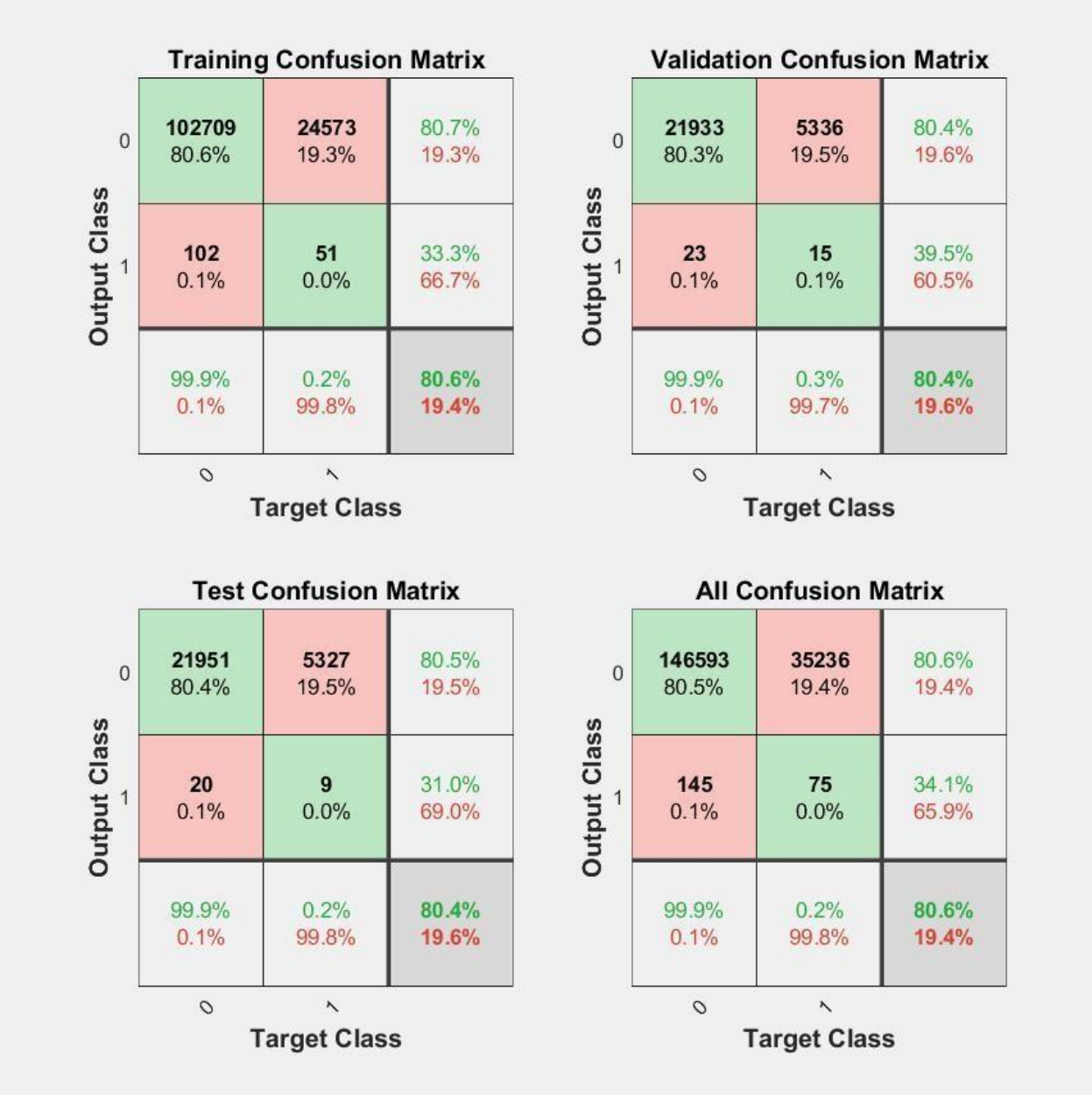

Below tables provides the values of Accuracy, Precision, Recall and F1 Score of different groups obtained using Neural Networks.

| | | | | |
|---|---|---|---|---|
| USER 21 | 0.8625 | 0 | 0.265 | 0 |
| USER 22 | 0.9665 | 0.23333 | 0.625 | 0.12245 |
| USER 23 | 0.7995 | 0.19995 | 0.625 | 0 |
| USER 24 | 0.8605 | 0.12235 | 0 | 0.23465 |
| USER 25 | missing | missing | missing | missing |
| USER 26 | 0.9 | 0.24245 | 0.3 | 0.82825 |

| | | | | |
|---|---|---|---|---|
| USER 27 | 0.9225 | 0.31245 | 0.5 | 0.4 |
| USER 28 | 0.8125 | 0.34565 | 0 | 0.22450 |
| USER 29 | 0.9665 | 0.19985 | 0.625 | 0 |
| USER 30 | 0.9 | 0.33333 | 0.5 | 0.33333 |
| USER 31 | 0.8125 | 0.12225 | 0 | 0 |
| USER 32 | 0.9225 | 0.45545 | 0 | 0 |
| USER 33 | 0.7995 | 0 | 0.625 | 0.34345 |
| USER 34 | 0.7625 | 0.34435 | 0.625 | 0.33333 |
| USER 36 | 0.9255 | 0.31255 | 0 | 0.12445 |
| USER 37 | 0.9 | 0.2 | 0.25 | 0.86285 |
| USER 38 | 0.9225 | 0.6 | 0.25 | 0.11111 |
| USER 39 | 0.8655 | 0 | 0.5 | 0 |
| USER 40 | 0.8 | 0.32455 | 0.625 | 0.33333 |
| USER 41 | 0.7925 | 0.28625 | 0.5 | 0.45455 |

‘

## Phase 3 Combined




| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER all | 0.806 | 0.55 | 0.60 | 0.45 |

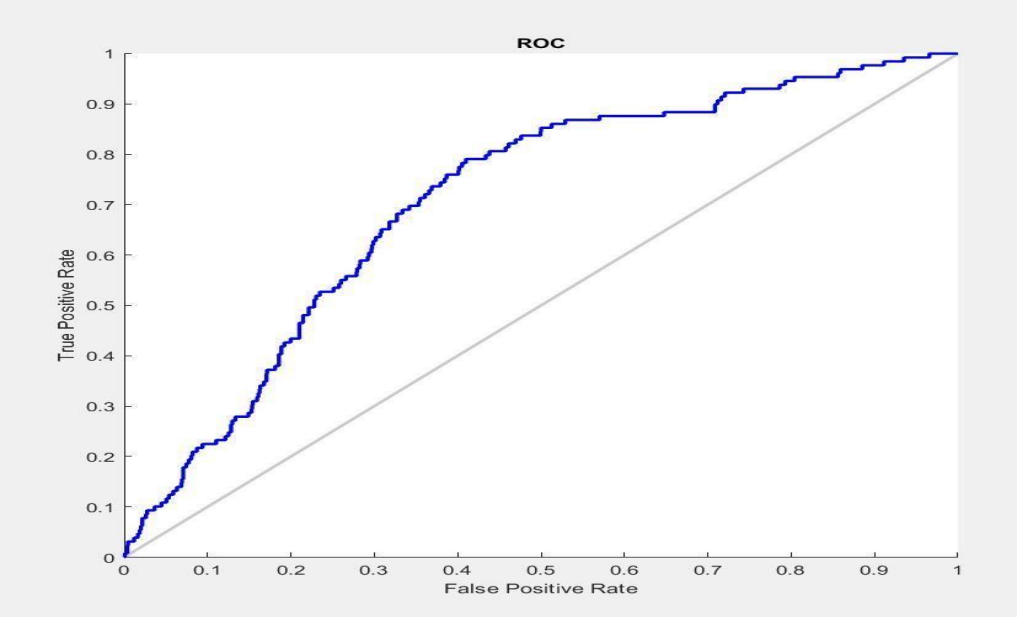
**ROC CURVE:**

A Receiver Operating Characteristic (ROC) Curve is a method to compare diagnostic tests. It is a comparison of the true positive rate against the false positive rate. A ROC plot shows:

- It is immediately apparent that a ROC curve can be used to select a threshold for a classifier which maximises the true positives, while minimising the false positives.
- Test accuracy: ROC curves also give us the ability to assess the performance of the classifier over its entire operating range. .

The greater the area encompassed under the curve, the more accurate the test. A perfect test has an area under the ROC curve (AUROCC) of 1. The diagonal line in a ROC curve denotes perfect chance.

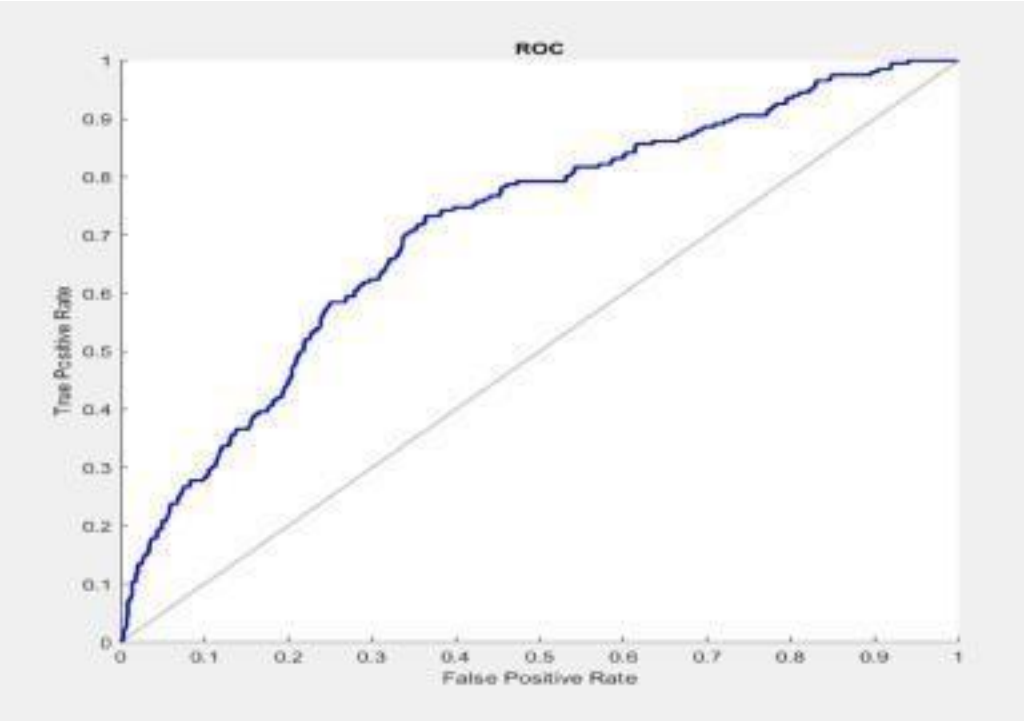
# Phase 2



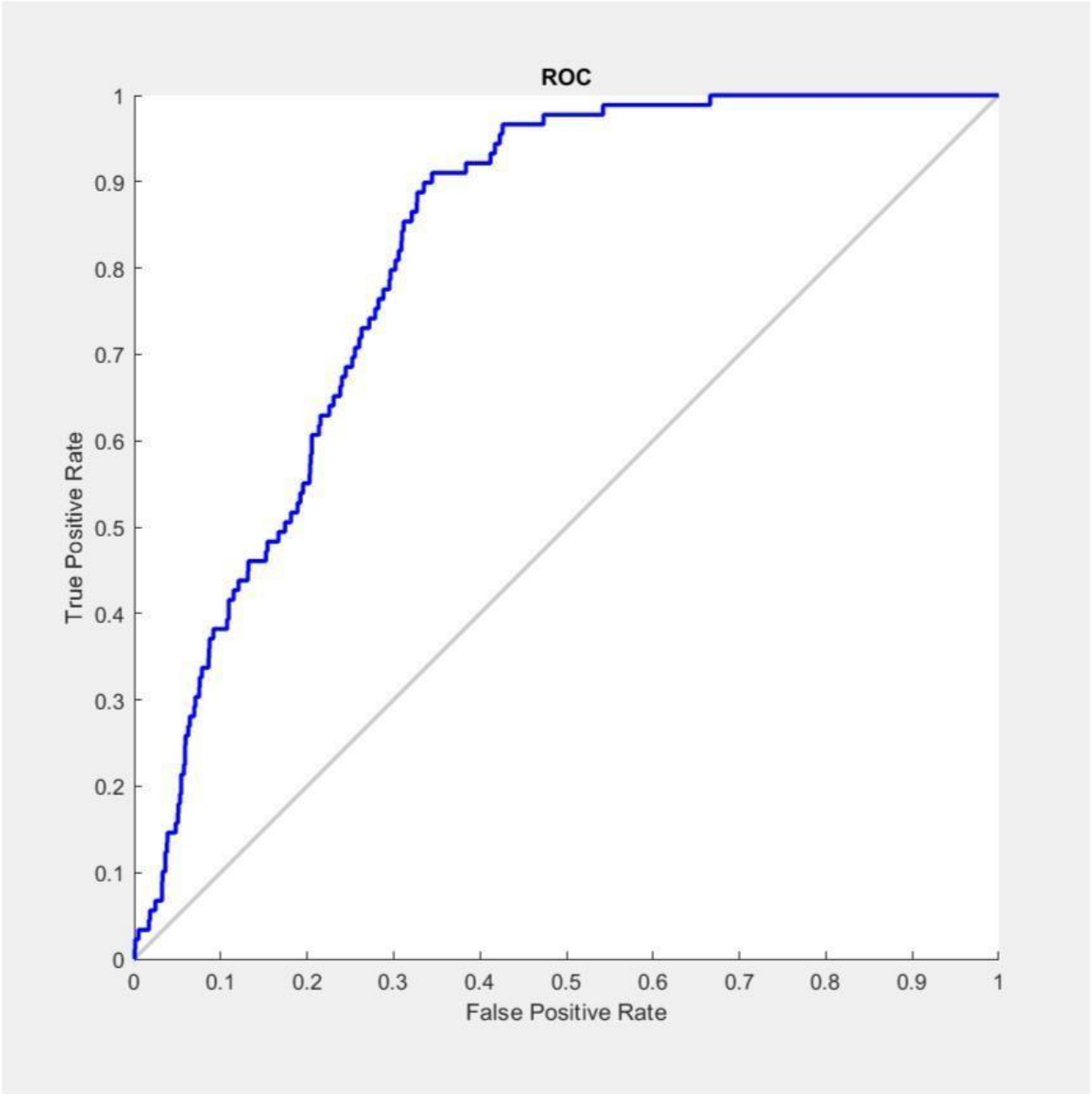
**User 37**



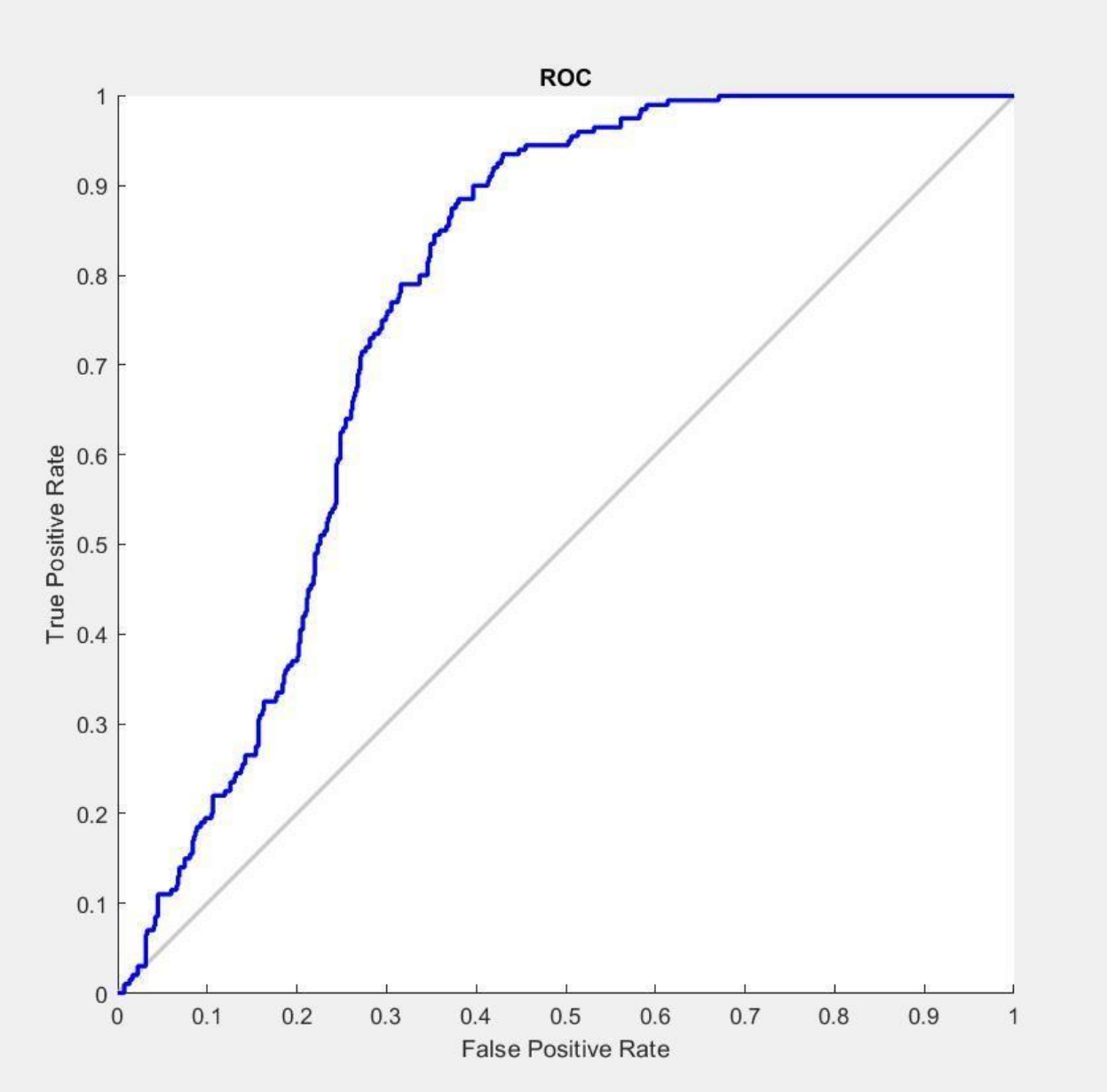
**User 36**



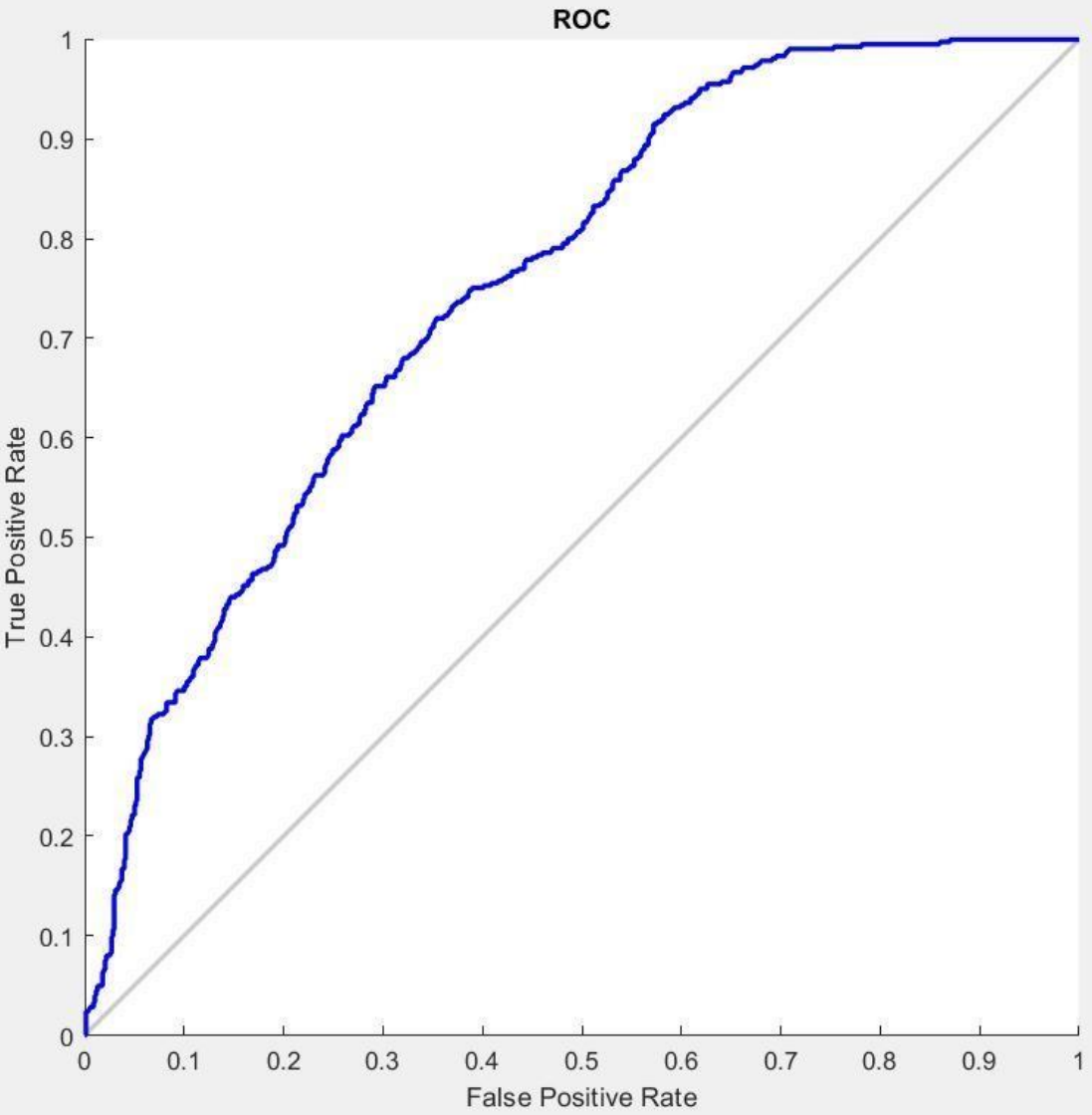

**User 36**


ROC
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

**User 22**


ROC
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

**User 21**

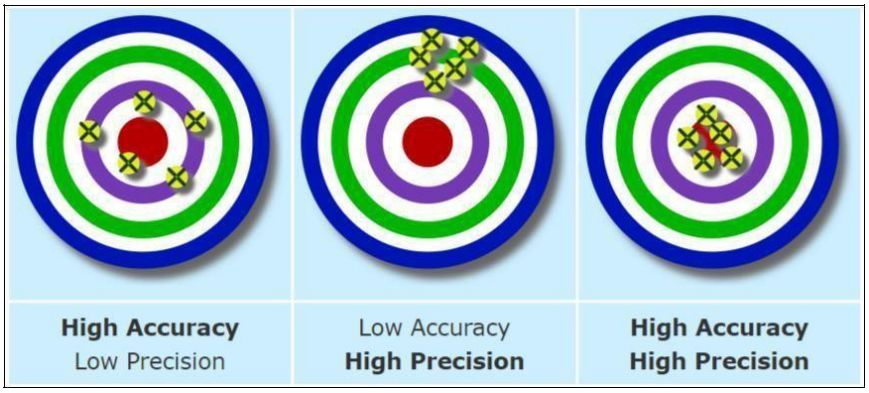

ROC
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

## Phase 3

| USER | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| USER | 0.60 | 0.27 | 0.42 | 0.19 |

## 6. Summary

Accuracy is how near a calculated value is to the actual value and Precision is how close the calculated values are to each other. The result on variations of Accuracy and Precision on using the classification tech nique for the given dataset is shown as below.




In this report, we observed different classification techniques which are better suited on our dataset. Each technique has different Accuracy, Precision, Recall and F1 Score. But the results indicate that the classification accuracy is better for Neural Network than Decision tree and Support Vector Machine.