# Machine Learning Project Report

## Group 11

**Ananya Kapoor**

**Mohammad Rafi Shaik**

**Sachini Weerasekara**

## Topic: Hand posture recognition

## Abstract

This work focusses on classification of hand postures, which has promising footprint on many fields such as human-computer interaction, healthcare, surveillance systems and many more. We built four machine learning models: Logistic Regression, Naïve Bayes, Support Vector Machine and Neural Networks to achieve the goal of hand posture recognition. We identified Discrete Nave Bayes with Laplace smoothing as the best performing model which results in AUC values for all the five classes.

## 1.0 Introduction

Posture recognition has gained increased attention in the fields of machine learning, and computer vision due to its promising applications in fields such as healthcare, human-computer-interaction, surveillance systems, and many more.

This work attempts to build a machine learning model to classify the hand posture of an individual to one of the 5 classes below.

- Class 1 - Fist (with thumbs out)
- Class 2 - Stop (hand flat)
- Class 3 - Point1 (point with pointer finger)
- Class 4 - Point2 (point with pointer and middle fingers)
- Class 5 - Grab (fingers curled as if to grab)

## 2.0 Data description

The dataset is sourced from UCI machine learning repository and can be accessed at https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures .

The dataset contains 78095 instances and 38 attributes.

The attributes can be categorized into three groups as below.

1. Xi - The x-coordinate of the ith unlabeled marker position.
2. Yi - The y-coordinate of the ith unlabeled marker position.
3. Zi - The z-coordinate of the ith unlabeled marker position.

Here, value of I varies from 0 to 11 th

A Vicon motion capture camera system was used to record 12 users performing 5 hand postures with markers attached to a left-handed glove. x, y, and z coordinates of 12 sensors attached to the hand form the set of predictor attributes. 3 coordinates per sensor form 36 predictors in total, out of which all are of the attribute type real. Dataset has some missing values and '?' is used to indicate a missing value. We followed both dropping columns with more than 15% and 50% missing values respectively and inputting the rest and evaluated the performance of each instance.

## 3.0 Methods

### 3.1 Logistic Regression

We deployed Logistic Regression using "One vs All" approach for the five classes: Fist, Stop, Point1, Point2, Grab. Five models were deployed for each class as separate probabilities of belonging to each class obtained using the "softmax" function. We implemented a baseline model and made improvements to it, in order to obtain better performance. Sections 3.1.1,3.1.2 and 3.1.3 discuss the improvements performed.

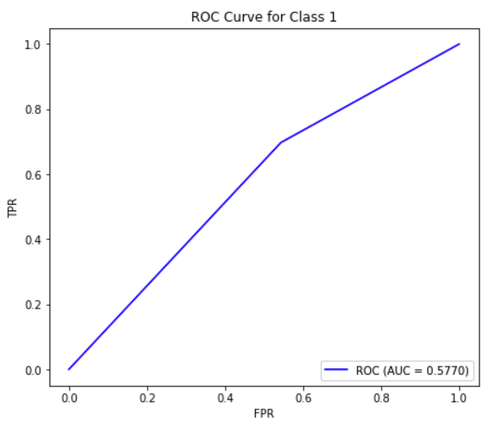
$$P(y_i = J \mid x_i) = \frac{e^{\theta_J^T x_i}}{\sum_{j=1}^{k} e^{\theta_j^T x_i}}$$

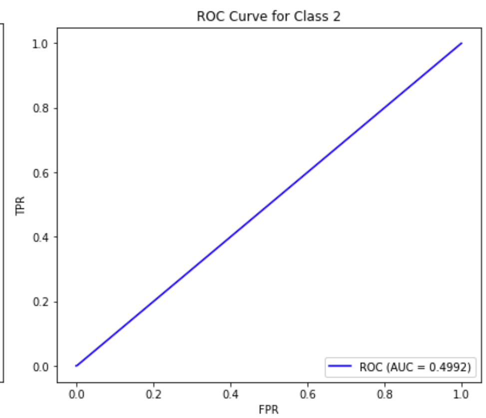
3.1.1 Baseline

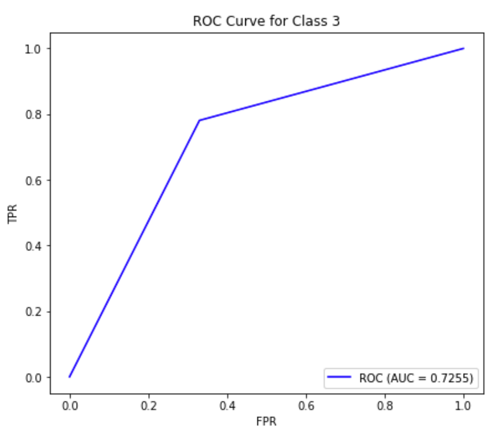
Baseline model used a learning rate of 0.0001, a tolerance of 0.0001.

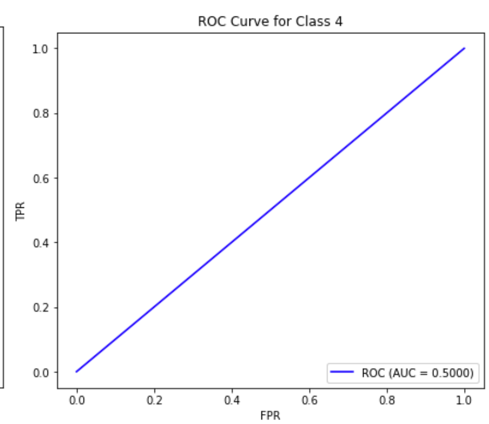
| Class | Precision | Recall | F1 score |
|---|---|---|---|
| 1 | 0.248 | 0.697 | 0.366 |
| 2 | 0.098 | 0.001 | 0.003 |
| 3 | 0.380 | 0.781 | 0.511 |
| 4 | 0.000 | 0.000 | 0.000 |
| 5 | 0.000 | 0.000 | 0.000 |

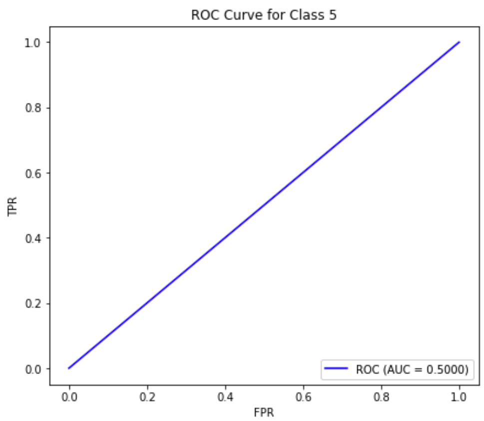
Below are the ROC curves and Area Under the Score (AUC) values foe each class for the baseline mode






### 3.1.2 Regularization with under-sampling

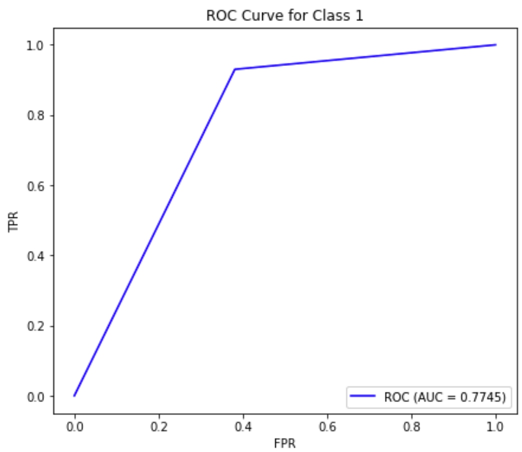
Due to the huge class imbalance when applying the “One vs All” approach, where 1 is given to the class of interest and all the other classes is referred to as 0. Therefore, we tried under-sampling and over-sampling methods. In addition, to avoid overfitting, we implemented L1 and L2 regularizations. This section presents the results after applying L2 regularization along with under-sampling. (We attained class balance (50% for each class) by removing the excess datapoints from class 0.)

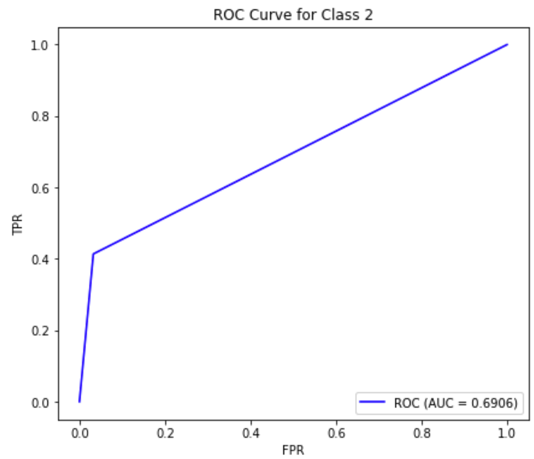
We used the same parameter values used for the baseline, along with L2 regularization to obtain below results for under-sampled data

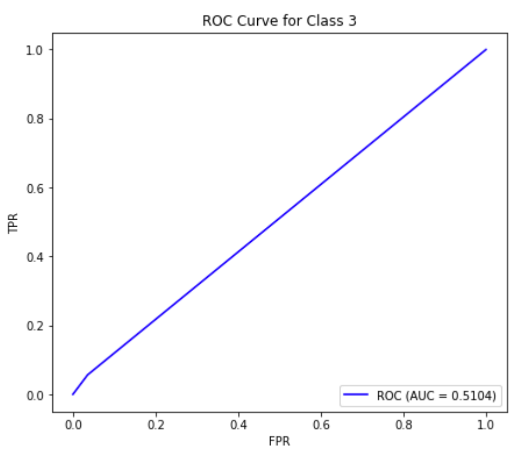
| Class | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|
| 1 | 0.386 | 0.930 | 0.545 | 0.7745 |
| 2 | 0.759 | 0.414 | 0.536 | 0.6906 |
| 3 | 0.291 | 0.056 | 0.095 | 0.5104 |
| 4 | 0.076 | 0.009 | 0.016 | 0.4919 |

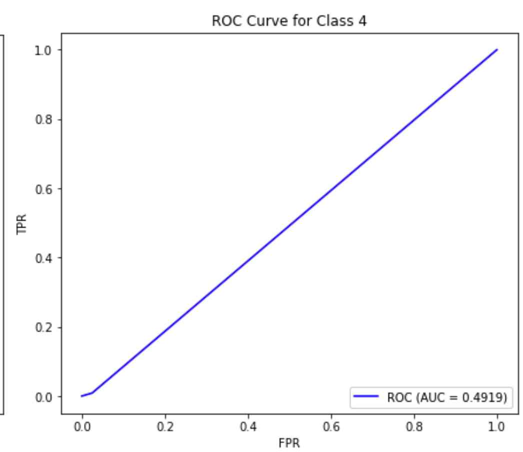
| | | | | |
|---|---|---|---|---|
| 5 | 0.255 | 0.426 | 0.319 | 0.5562 |

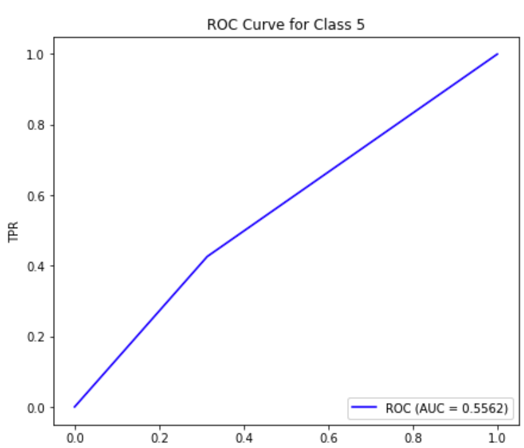
Given below are the ROC curves of the model.



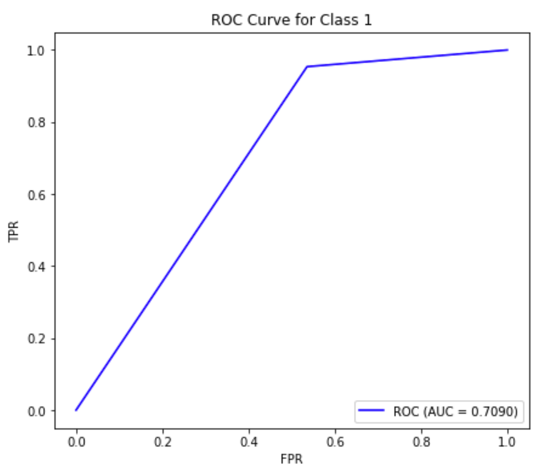



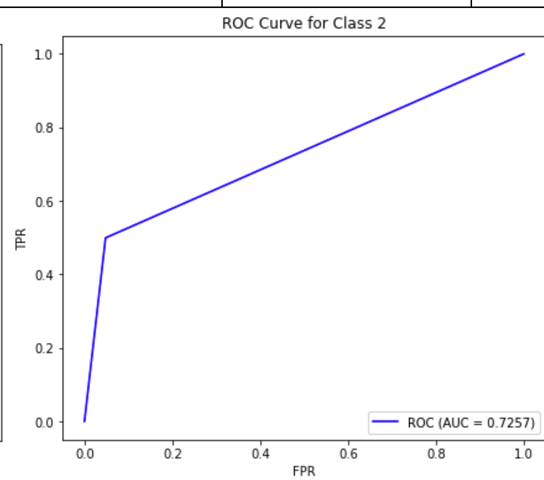



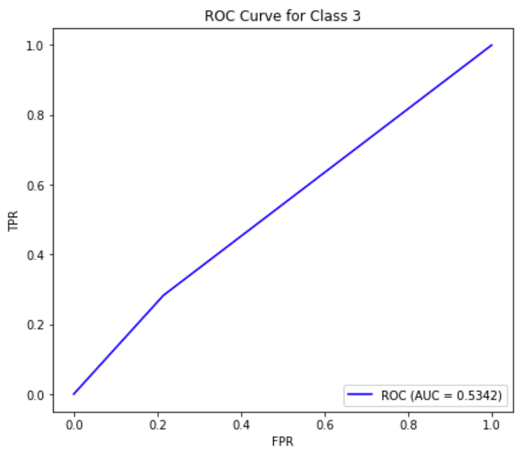

### 3.1.3 Regularization with over-sampling

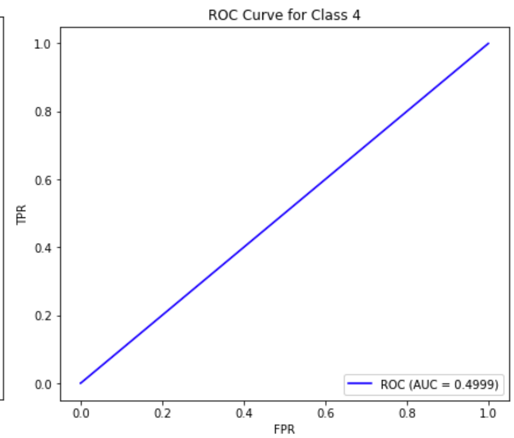
Next, we performed SMOTE over-sampling along with L1 and L2 regularization, where is randomly generated of the non-frequent class in order to achieve class balance. Below are the results for over-sampled data with L2 regularization.

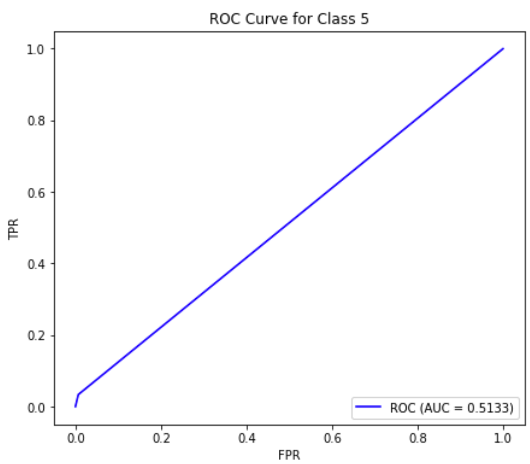
| Class | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|
| 1 | 0.314 | 0.954 | 0.473 | 0.7090 |
| 2 | 0.721 | 0.499 | 0.590 | 0.7257 |
| 3 | 0.254 | 0.283 | 0.268 | 0.5342 |
| 4 | 0.132 | 0.018 | 0.012 | 0.5 |
| 5 | 0.556 | 0.033 | 0.063 | 0.5133 |





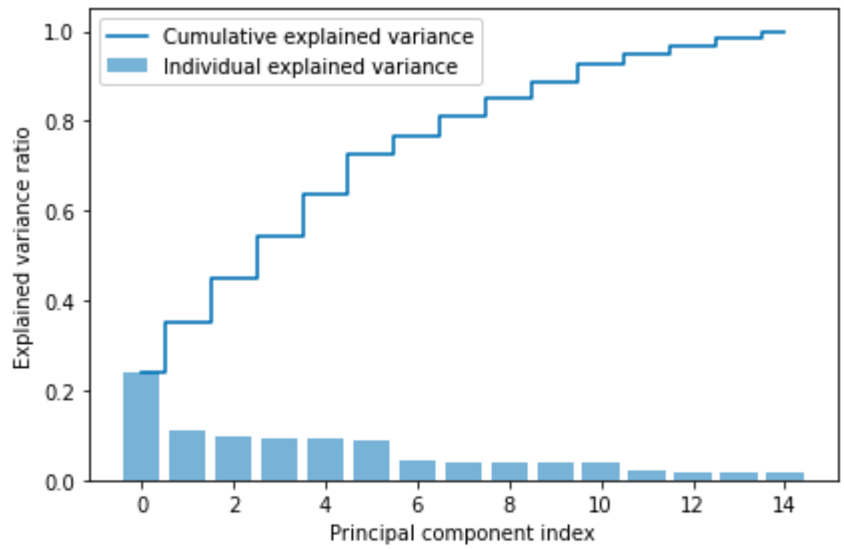






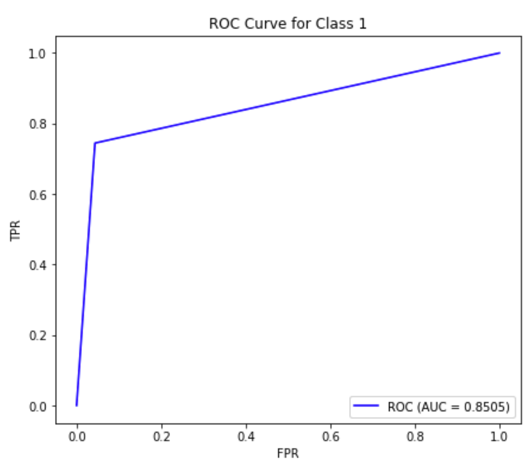

### 3.1.4 PCA with L2 regularization

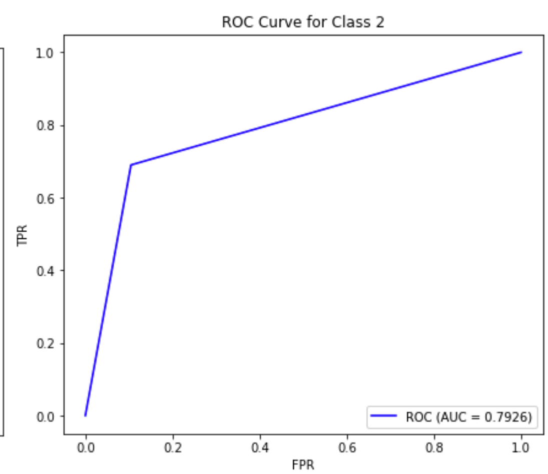
Finally, we performed PCA on the features to do feature reduction and implemented the model on principal components along with L2 regularization. It was observed that, 95% variance is explained by 11-12 principal components, as shown below.

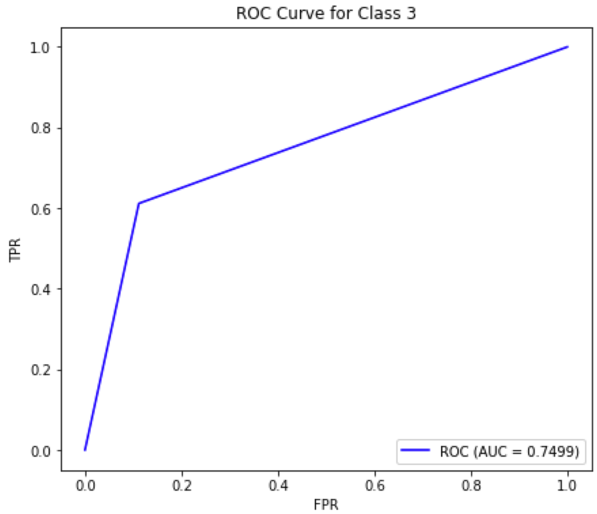


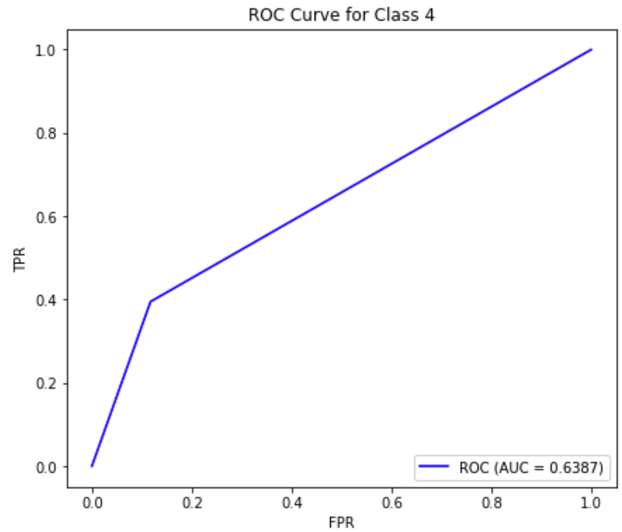

We used 95% explained variance to choose the principal components and obtained the below performance matrices, which is the best we have got for Logistic Regression.

| Class | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|
| 1 | 0.815 | 0.744 | 0.778 | 0.8505 |
| 2 | 0.620 | 0.690 | 0.653 | 0.8 |
| 3 | 0.586 | 0.612 | 0.598 | 0.7499 |
| 4 | 0.441 | 0.395 | 0.416 | 0.6387 |
| 5 | 0.538 | 0.555 | 0.546 | 0.7174 |






### 3.2 Naïve Bayes

We have implemented NB-Gaussian approach and NB-Discrete with/without Laplace smoothing multi-class classification.

3.2.1 Naïve Bayes – Guassain :

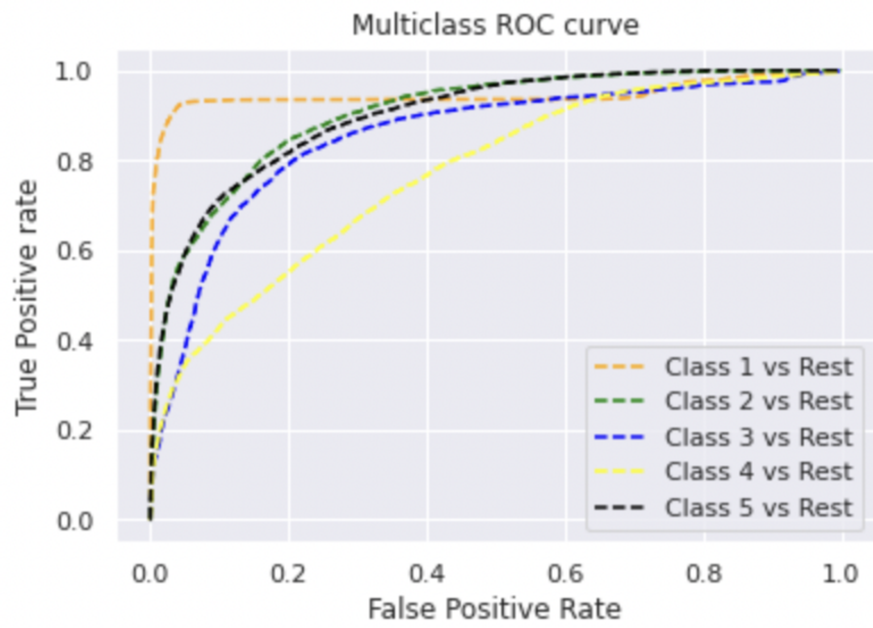
Using NB Guassain on 5-fold cross evaluation our model is performing as follows

The following are performance metrics of our model

**Accuracy: 0.669**

| Class | Precision | Recall | F1 Score |
|---|---|---|---|
| 1 | 0.867 | 0.915 | 0.890 |
| 2 | 0.566 | 0.767 | 0.651 |
| 3 | 0.610 | 0.672 | 0.640 |
| 4 | 0.551 | 0.314 | 0.400 |
| 5 | 0.720 | 0.651 | 0.683 |

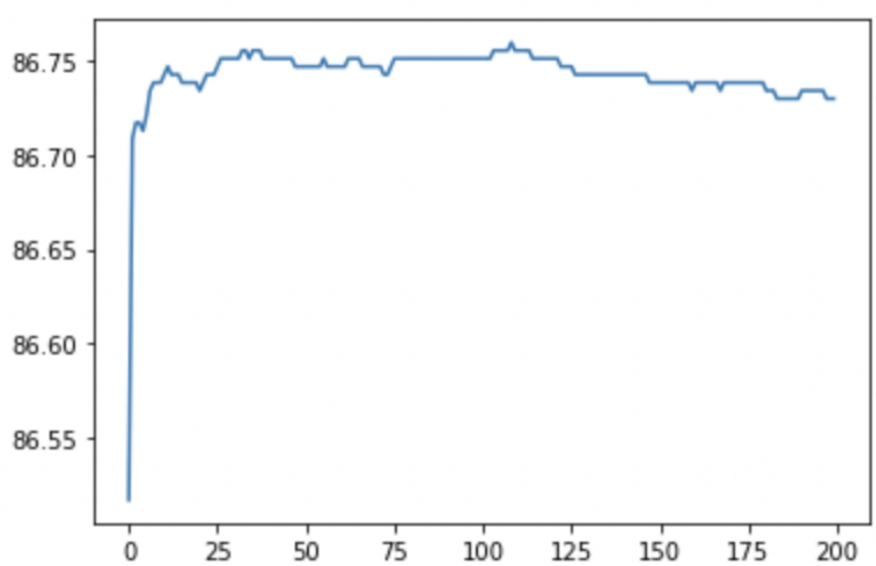
The ROC curve for this model is:




3.2.2 Naïve Bayes – Discrete with Laplace smoothing:

Here we need to figure out hyperparameter alpha.

So, we tried to fit in various alphas for various models and compare performance metrics among them. We finally chose the best alpha as 0.001, from best performing model

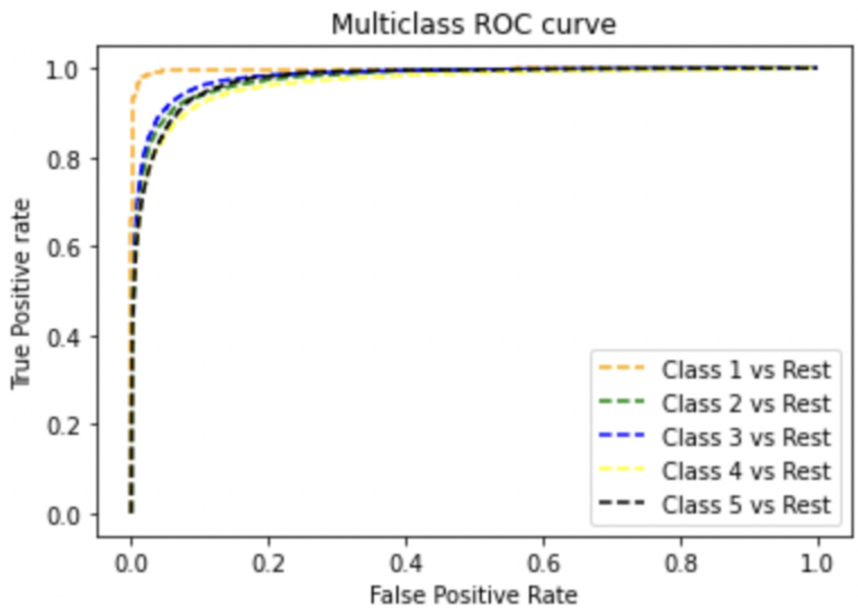



The following are performance metrics of our model

**Accuracy: 0.868**

| Class | Precision | Recall | F1 Score |
|---|---|---|---|
| 1 | 0.987 | 0.930 | 0.958 |
| 2 | 0.817 | 0.880 | 0.847 |
| 3 | 0.875 | 0.870 | 0.873 |
| 4 | 0.840 | 0.810 | 0.825 |
| 5 | 0.825 | 0.846 | 0.835 |

Here is the ROC curve for our model



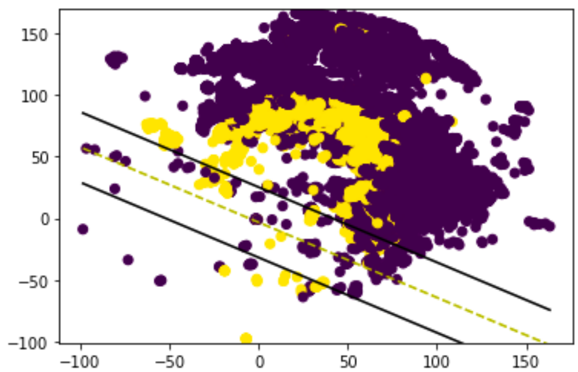

### 3.3 Support-Vector Machine

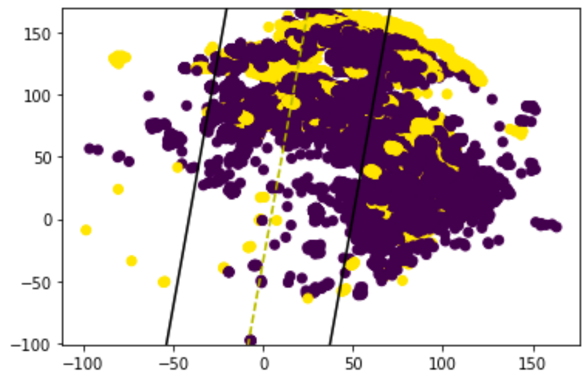
We could only perform Hard Margin linear SVM on the dataset. Because of the lack of computing power, we could not perform Soft margin SVM and non-linear kernel SVM.
To adjust Hard Margin SVM to support multi class classification, we implemented five different SVMs for each class, and picked the class that gives the maximum positive distance from the boundary as the class of the test sample.

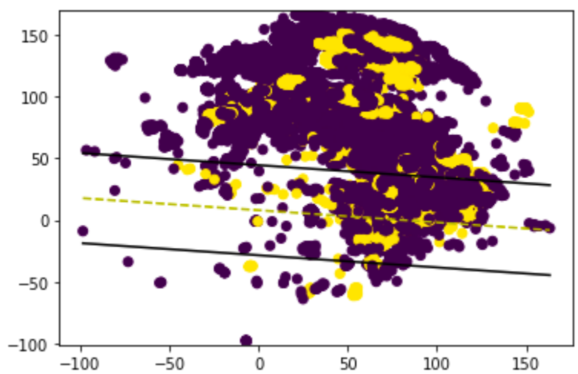
$$D = argmax(w_i^T X + b_i)$$

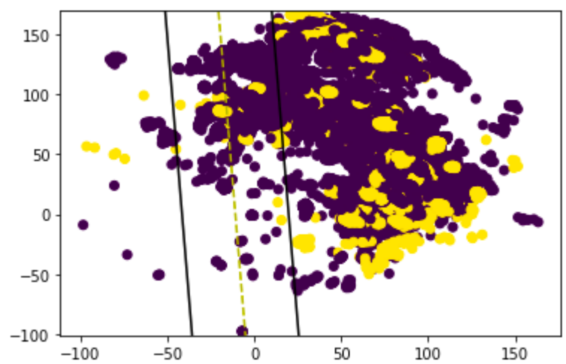
Due to the complicated data structure, Hard Margin linear kernel SVM did not seem to be performing the classification well.

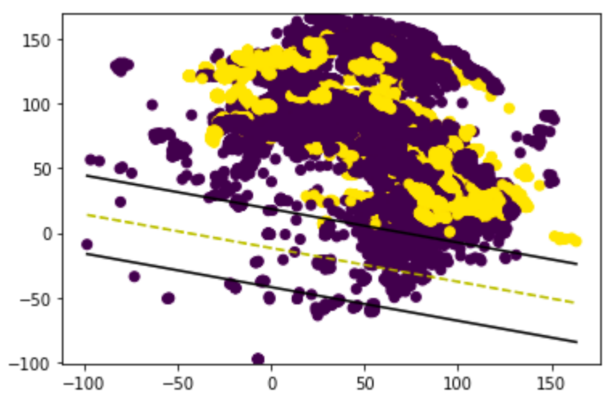
However, given below are the visualizations of the five Hard Margin Linear SVMs created.




### 3.4.0 Neural Network

We created the Neural Network model on our dataset for three combinations

Model 1- one hidden layer with 8 nodes

Model 2 - one hidden layer with 10 nodes

Model 3 - 2 hidden layers with 10 and 8 nodes

We found that the Model 3 has shown improvement with adding more hidden layers and Model 2 is performing better than Model 1 by just adding 2 more nodes.
Model 1 performed with an accuracy of prediction was 0.64 , Model 2 with an accuracy of prediction was 0.68 and Model 3 with an accuracy of prediction was 0.73

**3.4.0 Results**

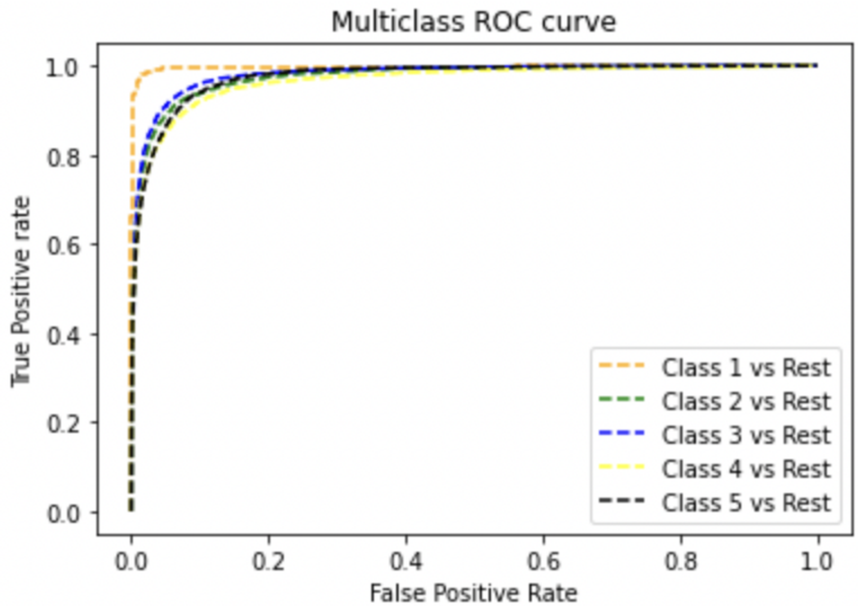
Out of the four models we implemented, discrete Naïve Bayes with Laplace smoothing and Logistic Regression with PCA and L2 regularization seem to be the top performing ones. Below is a comparison of the two models. Results for other implementations are given under each model in the previous sections of the report

Results for Logistic Regression with PCA and L2 regularization

| Class | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|
| 1 | 0.815 | 0.744 | 0.778 | 0.8505 |
| 2 | 0.620 | 0.690 | 0.653 | 0.8 |
| 3 | 0.586 | 0.612 | 0.598 | 0.7499 |
| 4 | 0.441 | 0.395 | 0.416 | 0.6387 |
| 5 | 0.538 | 0.555 | 0.546 | 0.7174 |

Discrete Naïve Bayes with Laplace smoothing

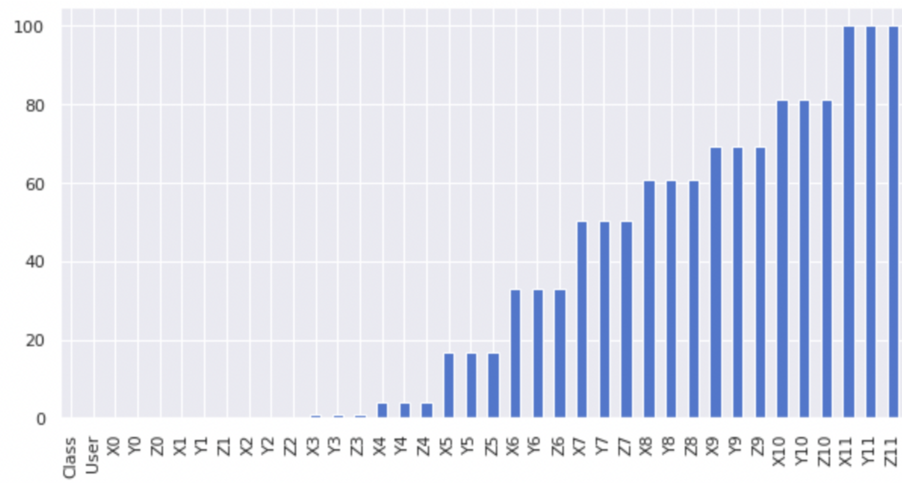
| Class | Precision | Recall | F1 Score |
|---|---|---|---|
| 1 | 0.987 | 0.930 | 0.958 |
| 2 | 0.817 | 0.880 | 0.847 |
| 3 | 0.875 | 0.870 | 0.873 |
| 4 | 0.840 | 0.810 | 0.825 |
| 5 | 0.825 | 0.846 | 0.835 |




**4.0 Explanatory data analysis:**

As part of Data Exploration, we implemented the following steps: we generated summary statistics for numerical columns and created a few graphs to understand data better.

1. As a first step, we looked for different datatypes each attribute holds the data. And we generated summary statistics for numerical variables and created a few graphs to understand their distribution, range, and bad values
2. Here is basic bar plot representing the missing values proportion in our data set. We got co-ordinates of few markers have high percentage of missing values. So, we are going to have different thresholds to impute and drop and compare model performance

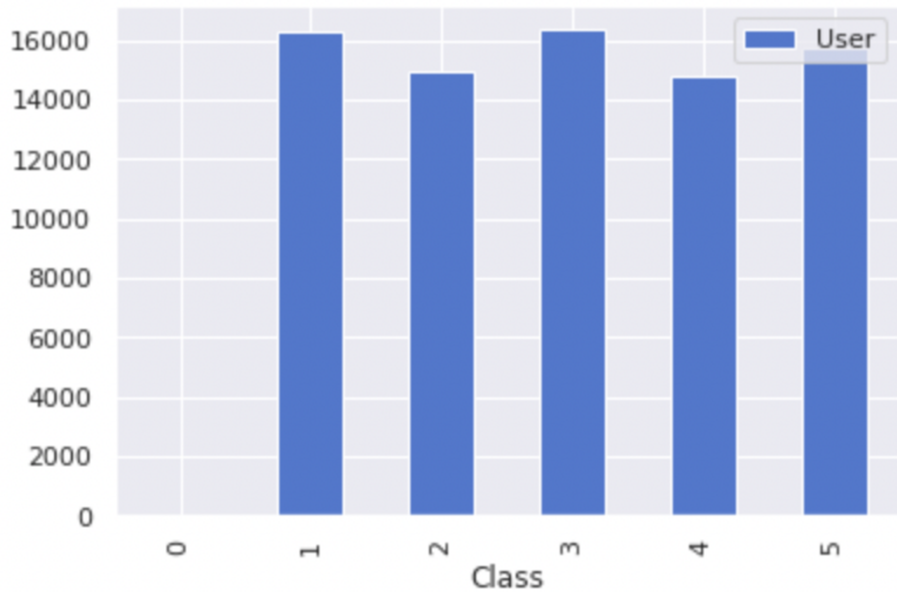



3. Here is the co-relation matrix representing one-one relationship between each numerical variable in our data set.

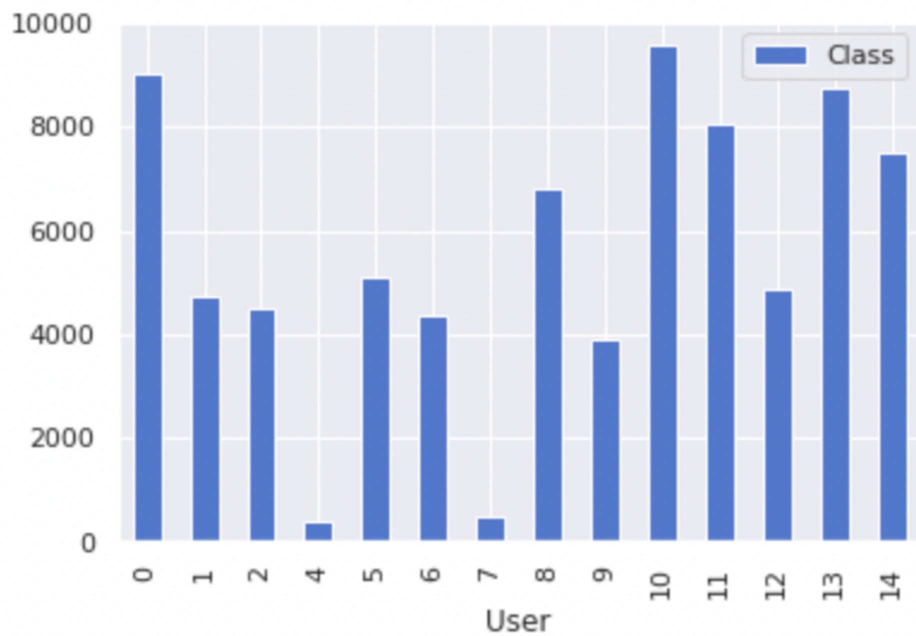
We do not observe huge co-relation between variables in considering dropping redundant variables

| | Class | User | X0 | Y0 | Z0 | X1 | Y1 | Z1 | X2 | Y2 | Z2 | X3 | Y3 | Z3 | X4 | Y4 | Z4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Class** | | | | | | | | | | | | | | | | | |
| **User** | -0.03 | | | | | | | | | | | | | | | | |
| **X0** | 0.08 | -0.01 | | | | | | | | | | | | | | | |
| **Y0** | 0.08 | 0.07 | -0.29 | | | | | | | | | | | | | | |
| **Z0** | 0.05 | 0.19 | -0.24 | 0.55 | | | | | | | | | | | | | |
| **X1** | -0.01 | -0.04 | 0.12 | -0.00 | 0.01 | | | | | | | | | | | | |
| **Y1** | 0.14 | 0.04 | -0.01 | 0.13 | 0.21 | -0.29 | | | | | | | | | | | |
| **Z1** | 0.09 | 0.18 | 0.03 | 0.23 | 0.46 | -0.26 | 0.57 | | | | | | | | | | |
| **X2** | -0.05 | -0.06 | 0.11 | 0.01 | 0.01 | 0.14 | -0.03 | -0.01 | | | | | | | | | |
| **Y2** | 0.17 | 0.00 | -0.01 | 0.08 | 0.15 | -0.05 | 0.11 | 0.18 | -0.32 | | | | | | | | |
| **Z2** | 0.10 | 0.16 | 0.02 | 0.19 | 0.42 | -0.01 | 0.18 | 0.46 | -0.30 | 0.59 | | | | | | | |
| **X3** | -0.05 | -0.05 | 0.09 | 0.01 | 0.02 | 0.13 | -0.03 | -0.02 | 0.14 | -0.06 | -0.03 | | | | | | |
| **Y3** | 0.17 | -0.01 | -0.01 | 0.06 | 0.13 | -0.05 | 0.10 | 0.16 | -0.06 | 0.13 | 0.18 | -0.35 | | | | | |
| **Z3** | 0.09 | 0.13 | 0.02 | 0.16 | 0.39 | -0.01 | 0.18 | 0.44 | -0.03 | 0.19 | 0.45 | -0.33 | 0.61 | | | | |
| **X4** | -0.05 | -0.03 | 0.08 | 0.01 | 0.01 | 0.11 | -0.03 | -0.01 | 0.12 | -0.06 | -0.03 | 0.12 | -0.07 | -0.04 | | | |
| **Y4** | 0.18 | -0.03 | -0.00 | 0.04 | 0.11 | -0.05 | 0.09 | 0.14 | -0.06 | 0.14 | 0.17 | -0.06 | 0.15 | 0.19 | -0.37 | | |
| **Z4** | 0.07 | 0.10 | 0.03 | 0.14 | 0.35 | -0.00 | 0.16 | 0.39 | -0.02 | 0.18 | 0.41 | -0.02 | 0.19 | 0.43 | -0.34 | 0.62 | |

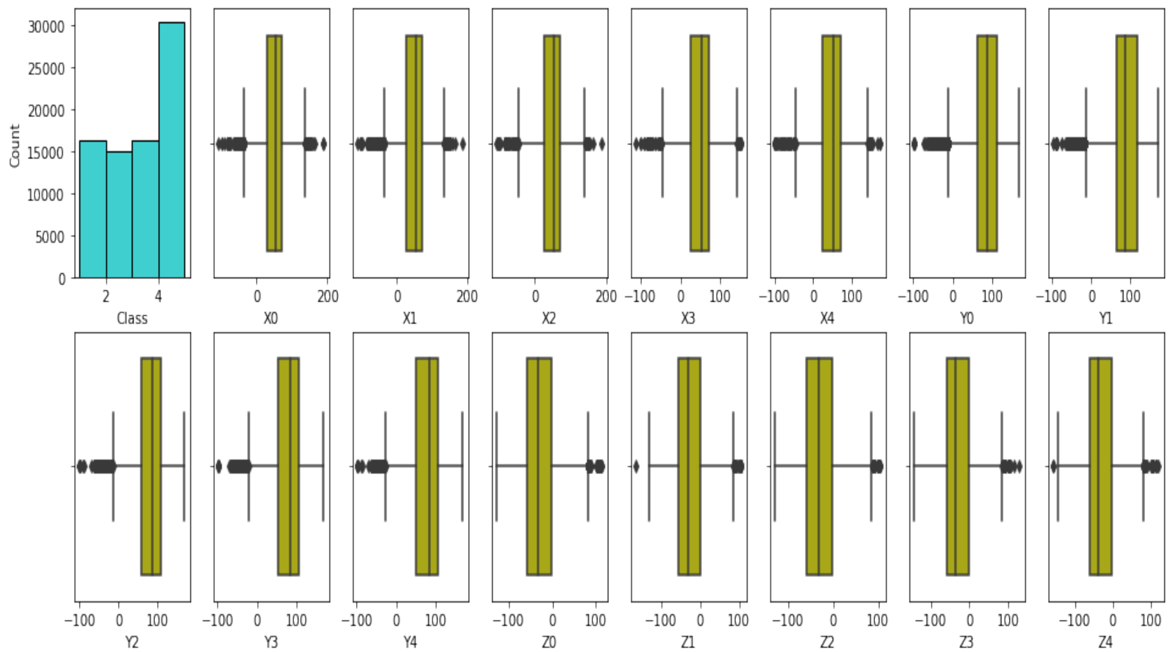
4. Here is the distribution of Class and User variables over our dataset



User
16000
14000
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
Class

Class
10000
8000
6000
4000
2000
0
0
1
2
4
5
6
7
8
9
10
11
12
13
14
User




SUMMARY

1. The range of X parameters is between -99 to +190.
2. The range of Y predictors is between +51 to +171 .
3. The range of Z coordinates is between -165 to +129.
4. We find significant outliers in X0,X1,X2,X3,X4.
5. We found Y0,Y1 , Y2 , Y3 ,Y4 are left skewed
6. We found Z0,Z1,Z2,Z3 to have certain values on the higher side.

**4.1 Data Cleaning:**

As part of Data pre-processing on our dataset, we implemented the following steps

i. Fix illegal values

The first row of data is invalid, which is a dummy row. So, we dropped it.

Missing values in dataset are represented by ‘?’ in our data set, so replacing ‘?’ in our data with **NaN** so we can impute accordingly.

ii. Handling missing values

We have a huge range of missing values for our attributes. So, we decided to check different thresholds (0, 15, 25, 50) to consider in our models. After careful consideration, we choose to go ahead with drop columns with more than 15% of missing values.
And we used MEAN with respect to Class for imputing missing values in our predictors

## 5.0 Discussion

We implemented four models: Naïve Bayes, Logistic Regression, SVM and Neural Networks for the classification problem at hand. For each model, we first implemented a baseline, and then kept improving the models to achieve better performance. Out of the four models implemented, Discrete Naïve Bayes with Laplace smoothing and Logistic Regression on PCA with L2 regularization, and Neural Network with 2 hidden layers with 10 and 8 nodes emerged as the top three performing models for the problem. Even though we believe non-linear kernel SVM would be a good candidate model, we could not perform non-linear kernel SVM, because of the computational complexities.
The identified best performing model is Discrete Naïve Bayes with Laplace smoothing, which has close to 0.9 ROC for almost all the five class.