Assignment-1: Key Findings & Insights in a Brief Report

Y.S SAI NITISH
M.Tech A.I
21677
IISc Bangalore
shivayegnam@iisc.ac.in

M. Samuel Jayakumar
M.Tech A.I
21181
IISc Bangalore
samuelj@iisc.ac.in

Abstract

This report entails briefly the findings obtained on implementing

- *Regression Analysis for linear models*
- *Regression Analysis using Non-Linear Models*
- *Multi-class classification using*
  1. *Baye's Classifier*
  2. *Linear classifier with one vs rest approach*
  3. *Multi-class Logistic Regression*
- *Multi-class classification of Kannada-MNIST dataset using*
  1. *Naive Baye's*
  2. *Logistic Regression*
  3. *Multi-class Baye's Classifier using GMM's*

Different metrics viz. classification accuracy, confusion matrix, F-1 Score, ROC cUrves, Likelihood Curve for EM and Empirical Risk were computed for evaluating the models. The same have been reported in the following sections.

1. Problem1: Linear Regression Using Linear Models

1.1. Problem Statement

The task is to predict the current health (as given by the target variable) of an organism given the measurements from two biological sensors measuring their bio-markers (negative indicates that it is lesser than the average case). With this data, we are expected to try our linear regression models on the training data and report the following metrics on the test split: (a) Mean Squared Error, (b) Mean Absolute Error, (c) p-value out of significance test.

1.2. Model Implemented:

The model used is linear and of the form "$W^T X$", which is an augmented format.

1.3. Data:

The input from sensors is X_i, a feature vector of dimension 2*1. The given data set has 10000 such data points. Each data point is augmented with a 1 and all the feature vectors are stacked to form a matrix 'X' of size $10000*3$ after augmenting the data vector with a 1. The target variable is denoted as Y_i of 1 dimension. So the stacked vector 'Y' is of size 10000*1.

1.4. Mathematics Involved:

- **Loss function used:** $||W^T X - Y||^2$
- The optimal W learned through data is: $W^* = (X^T X)^{-1} XY$

1.5. Results:

Parameter	Value
MSE for training data	10.119
MAE for training data	5.382
MSE for test data	5.0464
MAE for test data	1.7990
p-value	0.90847807

Table 1. Results on test data

1.6. Inferences:

It is clearly evident from the results that MSE & MAE are significantly higher in case of training data as compared to their testing counterpart. The reason behind might be a **Sampling Bias** in the test data. This can be explained by a simple example. If you are a student studying for an exam, and you understood only 40% of your syllabus. Fortunately

for you the examiner asks you question only on the things you learnt and you get a 100% result. This does not mean that you know the whole subject, just that the test was 'biased' for you.

2. Problem2: Linear Regression Using Non-Linear Models

2.1. Problem Statement

Here, we are expected to predict the lifespan of the above organism given the data from three sensors. In this case, the model is not linear. We are expected to try several (at least 3) non-linear regression models on the train split and report the following metrics on the test split (a) Mean Squared Error, (b) Mean Absolute Error, and (c) p-value out of significance test.

2.2. Models Implemented:

The models used are Non-Linear and are of the form $W^T\phi(X)$, where $\phi(X)$ is of the form

- $(X_1^3 \quad X_2^2 \quad X_1)$
- $((e^{X_1} \quad (e^{X_2})^2 \quad (e^{X_3})^3)$
- $(Sin(X_1) \quad Cos(X_2) \quad Tan(X_3))$

2.3. Data:

The input from sensors is X_i, a feature vector of dimension 3*1. The given data set has 10000 such data points. Each data point is augmented with a 1 and all the feature vectors are stacked to form a matrix 'X' of size 10000$*$4 after augmenting the data vector with a 1. The target variable is denoted as Y_i of 1 dimension. So the stacked vector 'Y' is of size 10000*1.

2.4. Mathematics Involved:

- **Loss function used:** $||W^TX - Y||^2$
- The optimal W learned through data is: $W^* = (X^TX)^{-1}XY$

2.5. Results:

Parameter	Value
MSE for training data	0.0318
MAE for training data	0.2167
MSE for test data	0.01371
MAE for test data	0.0804
p-value	0.04049

Table 2. Results on test data for Cubic polynomial Regression

Parameter	Value
MSE for training data	0.023391
MAE for training data	0.227968
MSE for test data	0.0128735
MAE for test data	0.072282
p-value	3.07544451e-20

Table 3. Results on test data for Exponential Regression Model

Parameter	Value
MSE for training data	0.0341729
MAE for training data	0.222382
MSE for test data	0.01396497
MAE for test data	0.078601
p-value	1.8337783e-08

Table 4. Results on test data for Trigonometric Regression Model

2.6. Inferences:

- The P-value tests in all 3 cases imply that the null hypothesis i.e., the regression models used, can be rejected. But the reason behind the extremely low p-values might be because the data was normalised, which resulted in very low errors and consequently might have effected the p-values causing them to be low.

- On the other hand the MSE & MAE infer that Model-2 i.e., exponential model is most accurate out of all the three models. The P-value significance test also convinces that the model-2 is indeed an accurate model compared to the other two.

3. Problem3: Multi-Class Classification

3.1. Problem Statement

We have data from 10 sensors fitted in an industrial plant. There are five classes indicating which product is being produced. The task is to predict the product being produced by looking at the observation from these 10 sensors. Given this, we are expected to implement (a) Bayes' classifiers with 0-1 loss assuming Normal, exponential, and GMMs (with diagonal co-variances) as class-conditional densities.(b) Linear classifier using the one-vs-rest approach, and (c) Multi-class Logistic regressor with gradient descent.

3.2. Mathematics Involved

Maximum Likelihood Estimate for Normal Class Conditional Density

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T \tag{2}$$

Maximum Likelihood Estimate for Laplacian Class Conditional Density

$$\hat{\mu} = \text{median}(x_1, x_2, ..., x_n) \tag{3}$$

$$\hat{b} = \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{\mu}| \tag{4}$$

GMM update step:
E step: Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \tag{5}$$

M step: Re-estimate the parameters using the current responsibilities:

$$\mu_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{6}$$

$$\Sigma_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \tag{7}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{8}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{9}$$

Evaluate the **log-likelihood**:

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right) \tag{10}$$

Multi-class Logistic Regression:
For K = 1,2,...,m

$$P(Y = K \mid X = x) = \frac{e^{\beta_{0k} + \beta_{1k}x_1 + ... + \beta_{pk}x_p}}{\sum_{j=1}^{m} e^{\beta_{0j} + \beta_{1j}x_1 + ... + \beta_{pj}x_p}} \tag{11}$$

Where:

- K represents the class label
- m is the number of classes.
- $P(Y = K \mid X = x)$ is the probability of the observation x belonging to class K.
- $\beta_{0k}, \beta_{1k}, ..., \beta_{pk}$ are the coefficients for class K.
- $x_1, ..., x_p$ are the predictor variables.

3.3. Results

- **Performance Metrics for Classifier using Normal Distribution**

1. **Classification Accuracy:** 58.95333

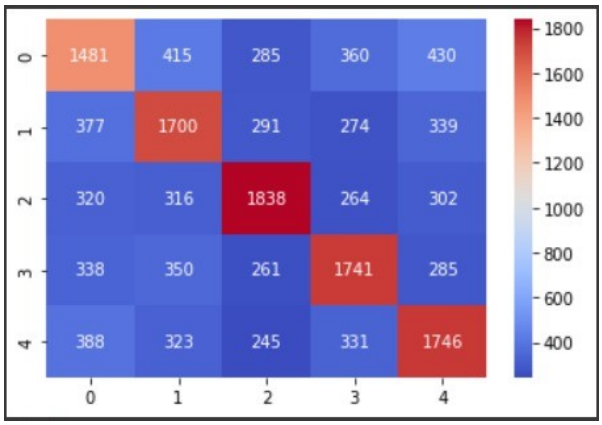
2. **Confusion Matrix**
$$\begin{bmatrix} 1617 & 370 & 259 & 338 & 387 \\ 381 & 1760 & 265 & 255 & 320 \\ 313 & 317 & 1873 & 233 & 304 \\ 344 & 349 & 237 & 1766 & 279 \\ 370 & 320 & 220 & 296 & 1827 \end{bmatrix}$$

3. **F1 Score** $\begin{pmatrix} 0.5394 \\ 0.5773 \\ 0.6356 \\ 0.6024 \\ 0.5941 \end{pmatrix}$

- **Performance Metrics for Classifier using Laplacian Distribution**

1. **Classification Accuracy:** 56.70666

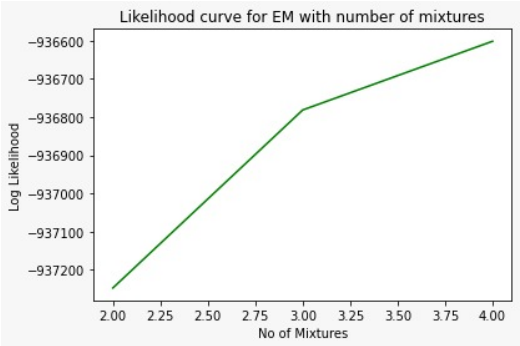
2. **Confusion Matrix**



3. **F1 Score** $\begin{pmatrix} 0.5042 \\ 0.5588 \\ 0.6168 \\ 0.5857 \\ 0.5692 \end{pmatrix}$

- **Performance Metrics for Classifier using a Gaussian Mixture Model with 3 components**

1. **Classification Accuracy:** 59.55333
2. **Confusion Matrix**
$$\begin{bmatrix} 6303 & 1497 & 1288 & 1425 & 1516 \\ 1470 & 6759 & 1096 & 1337 & 1357 \\ 1195 & 1111 & 7473 & 1030 & 1151 \\ 1395 & 1279 & 1044 & 7041 & 1266 \\ 1505 & 1416 & 1165 & 1231 & 6650 \end{bmatrix}$$
3. **F1 Score** $\begin{pmatrix} 0.5590 \\ 0.5822 \\ 0.6465 \\ 0.6128 \\ 0.5799 \end{pmatrix}$
4. **Likelihood Curve for GMM**



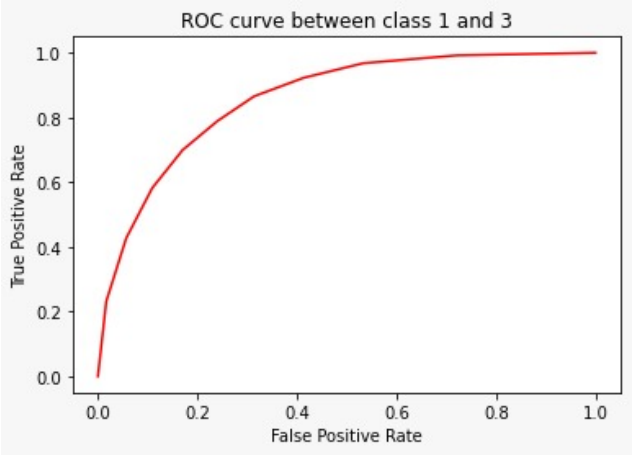
- **Performance Metrics for Linear Classifier using a One vs Rest Approach**

1. **Classification Accuracy:** 57.3200
2. **Confusion Matrix**
$$\begin{bmatrix} 1495 & 391 & 311 & 383 & 391 \\ 363 & 1698 & 312 & 285 & 323 \\ 293 & 304 & 1897 & 259 & 287 \\ 326 & 343 & 269 & 1749 & 288 \\ 339 & 326 & 265 & 344 & 1759 \end{bmatrix}$$
3. **F1 Score** $\begin{pmatrix} 0.5167 \\ 0.5620 \\ 0.6226 \\ 0.5835 \\ 0.5785 \end{pmatrix}$

- **Performance Metrics for Multi-Class Logistic Regressor**

1. **Classification Accuracy:** 57.32666
2. **Confusion Matrix**
$$\begin{bmatrix} 1518 & 390 & 297 & 375 & 391 \\ 377 & 1696 & 297 & 286 & 325 \\ 308 & 306 & 1881 & 256 & 289 \\ 339 & 345 & 262 & 1739 & 290 \\ 350 & 327 & 253 & 338 & 1765 \end{bmatrix}$$
3. **F1 Score** $\begin{pmatrix} 0.5178 \\ 0.5611 \\ 0.6239 \\ 0.5827 \\ 0.5794 \end{pmatrix}$
4. **Empirical Risk on Training Data :** 2.3013
5. **Empirical Risk on Testing Data :** 2.2518666

ROC CURVE Between Class 1 & Class 3



3.4. Inferences:

- Baye's classifier (with 0-1 loss function) with **Normal** class conditional density is **more accurate** than Baye's classifier with**Laplacian** class conditional density implying that the actual class densities of the data are more close to a Normal than to Laplacian.
- Though a Gaussian Mixture Model is having slightly higher accuracy, the difference is not widespread. This suggests that the data is particularly not complex or diverse, and can be well approximated by a a single Gaussian distribution
- It is observed that the accuracy of both linear classifier with one-vs-rest approach 57.32% and multi-class logistic regression 57.3266% is similar. Possible reason is that the number of classes in the problem is small and the differences in the two methods are not significant enough to affect the accuracy. Other reason might

be that the decision boundaries for each class are simple and not highly no-linear or curved, because if the data can be well separated by a linear boundary both the models will perform the similarly.

- **ROC curve for Multi-Class Logistic Regression** The ROC curves show that the classifier performs better than a random classifier(as it is above the x=y line), but not the best. This could be due to the assumption that the hypothesis function is a softmax function, may not be accurate enough.

4. Problem4: Multi-class Classification on MNIST Dataset

4.1. Problem Statement

In this problem, we consider an image dataset called Kannada-MNIST. This dataset contains images (60,000 images with 6000 per class) of digits from the south Indian language of Kannada. The task is to build a 10-class classifier for the digits. Classification schemes used are: (a) Naive Bayes' with Normal as Class conditional, (b) Logistic regressor with gradient descent, and (c) Multi-class Bayes' classifier with GMMs with diagonal co-variances for class conditionals. The data is split into test and training data in the following ratios 20:80, 30:70, 50:50, 70:30 & 90:10 for different evaluations.

The given input is images of size 28*28 so the dimension of feature vector is 784.

4.2. Mathematics Involved

- **Naive Baye's Classifier** The Naive Bay's algorithm assumes that the individual features of input feature vector are independent of each other. In this problem class conditional for each feature is assumed to be gaussian and parameters (mean and variance) are estimated using the univariant equivalents of the equations (5) and (6)

$$P(Y \mid X_1, X_2, ..., X_n) = \frac{P(Y) \cdot \prod_{i=1}^{n} P(X_i \mid Y)}{\prod_{i=1}^{n} P(X_i)} \tag{12}$$

Where:

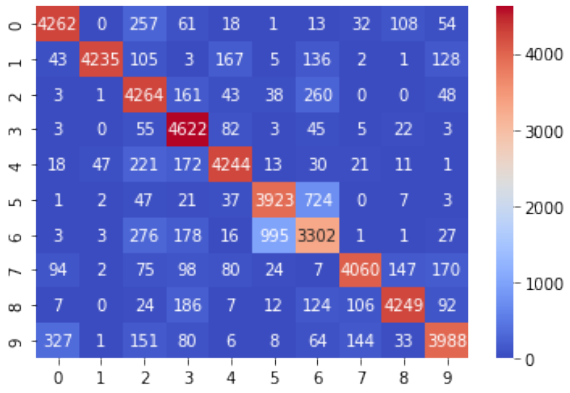
  - Y represents the class label.
  - $X_1, X_2, ..., X_n$ are the predictor variables.
  - $P(Y)$ is the prior probability of class Y.
  - $P(X_i \mid Y)$ is the conditional probability of predictor variable X_i given class Y.
  - $P(X_1, X_2, ..., X_n)$ is the probability of the predictor variables.

- **Multi-class Logistic Regression:**
  - Normalized each feature of train and test data points by subtracting the mean and dividing by standard deviation of train dataset features.
  - **Advantages of Normalization:**
    * Handling outliers: Normalizing data handles outliers by reducing the impact of extreme values on the analysis. By transforming the data to a common scale, outliers' impact is decreased and the analysis is more robust to extreme values.
    * normalizing data is an indispensable artifact that helps in improving the accuracy, reliability, and reproducibility of data analyses and ML models.
  - Rest of the implementation is same as that of p3

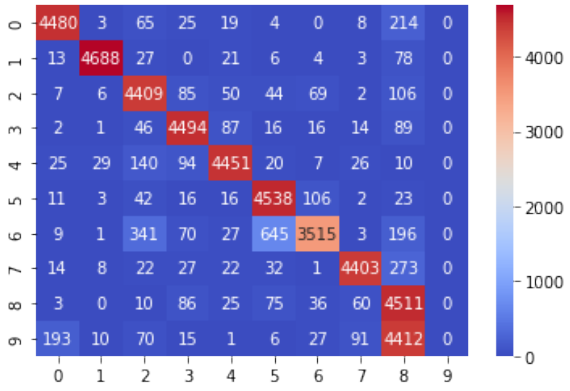
- **Multi-class Bayes' classifier with GMMs**
  - Implemented the update equations (5) to (11)

4.3. Results & Plots

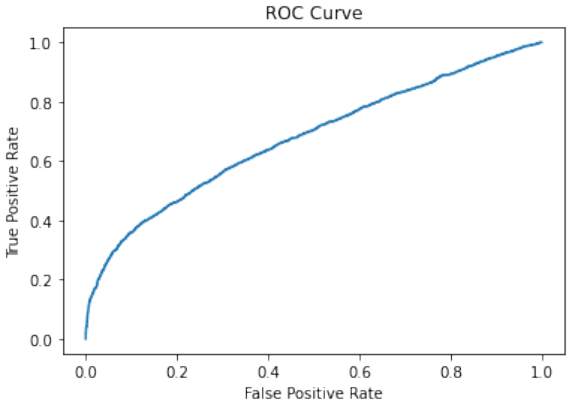
- **Metrics for Naive Baye's Classifier with 20:80 split**



- **Metrics for Multi-Class Logistic Regressor with 20:80 split**



- **ROC Curve**



4.4. Inferences

Naive Baye's

- According to the metrics table, it seems that the accuracy scores are comparable across various train-test splits. Nonetheless, there is disparity in the F1 scores for individual classes within each split, and this could be attributed to an uneven distribution of the different classes during the random split.
- This implies that the random split might not have distributed the various classes uniformly between the training and testing datasets, resulting in differences in the F1 scores for each class.

Multi-Class Logistic Regressor:

- Based on the metric table, it seems that there is a slight rise in accuracy as the train-test split increases.
- The reason for this could be that by using more training data to fit the logistic regression model, the model gains access to more information about the underlying patterns and relationships in the data. This, in turn, enables the model to make more precise predictions on new, unseen data.
- It is crucial to keep in mind that the increase in accuracy may not follow a linear trend and could eventually level off or even decline if the volume of training data becomes excessively large, or if the model becomes overfit to the training data.

Comparision Between Train & Test Empirical Risk

- It is typically anticipated that the empirical risk for the train dataset is lower than that of the test dataset. However, from the metric table, it can be observed that for smaller train-test splits, the empirical risk for the test data is lower than that of the empirical risk of the train dataset. In certain cases, this could be due to random chance, particularly with small datasets or with specific types of data distributions.

Multi-class Bayes' classifier with GMMs

- Effect of Curse Of Dimensionality
  - In the case of Gaussian Mixture Models (GMMs), the curse of dimensionality can have a significant impact on their performance.
  - One of the main challenges of GMMs is that they require estimating a large number of parameters, particularly as the dimensionality of the input data increases. Specifically, the number of parameters in a GMM scales quadratically with the dimensionality of the data, which can quickly become computationally infeasible for high-dimensional data,
  - In the current problem, we have high dimensional data(784 features), which is making it computationally infeasible. Possible fix for this is to use PCA to reduce the number of dimensions, which has been done in P5.

Metric	*20:80*	*30:70*	*50:50*	*70:30*	*90:10*
Classification Accuracy	95.78 %	96.10 %	96.36 %	96.6 %	96.51 %
F1 scores	nan 0.9358 0.9778 0.8862 0.9288 0.9350 0.8948 0.8186 0.9354 0.6130	nan 0.8897 0.9756 0.8554 0.9241 0.9342 0.8542 0.6854 0.9047 0.5639	nan 0.9255 0.9727 0.7962 0.9144 0.9314 0.7280 0.7119 0.9382 0.6383	nan 0.9189 0.9675 0.7228 0.9039 0.9225 0.6153 0.6720 0.9360 0.6476	nan 0.8729 0.9716 0.7634 0.9207 0.9094 0.7805 0.2648 0.9197 0.5796
Empirical risk on Train data	343.224	375.492	437.307	467.441	492.266
Empirical risk on Test data	339.9393	380.013	436.502	469.898	502.641

Table 5. Metrics of Multi-class Regression for P4

Metric	*20:80*	*30:70*	*50:50*	*70:30*	*90:10*
Classification Accuracy	87.12708 %	86.06190 %	83.76666 %	85.80555 %	83.36666%
F1 scores	0.8562 0.8910 0.9291 0.8285 0.8870 0.8955 0.8017 0.6946 0.8896 0.9054	0.8551 0.8950 0.9294 0.8330 0.8852 0.8966 0.8019 0.7007 0.8893 0.9042	0.8538 0.8967 0.9273 0.8393 0.8783 0.8923 0.8046 0.7043 0.8871 0.9053	0.8524 0.8926 0.9182 0.8417 0.8817 0.8890 0.8006 0.6984 0.8878 0.9032	0.8537 0.8898 0.9171 0.8544 0.8849 0.8889 0.8065 0.7174 0.8716 0.8975

Table 6. Metrics of Naive Bayes' Classifier with Normal Class Conditionals for P4

5. Problem5: Multi-class Classification with Condensed Data

5.1. Problem Statement

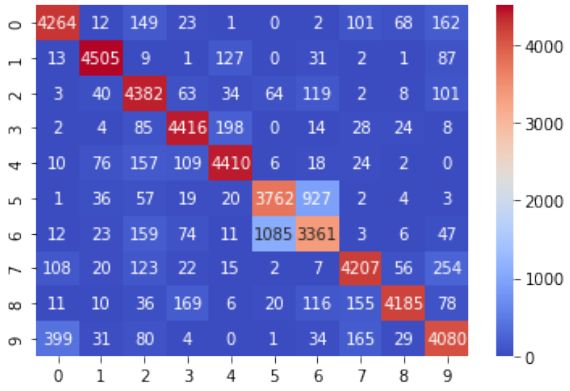
The aim of this task is to categorize the Kannada PCA MNIST dataset into ten distinct classes. In this instance, the Kannada MNIST dataset is compressed using PCA to two dimensions, and each data point has two features. The provided datasets are then utilized to train and test various classifiers, and the outcomes are compared to p4.

5.2. Implementation

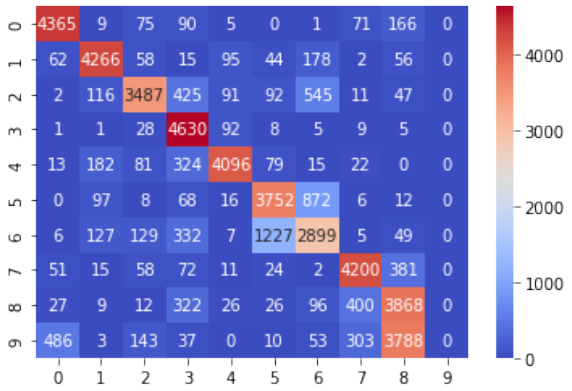
The implementation is same as that of P4.

5.3. Results & Plots

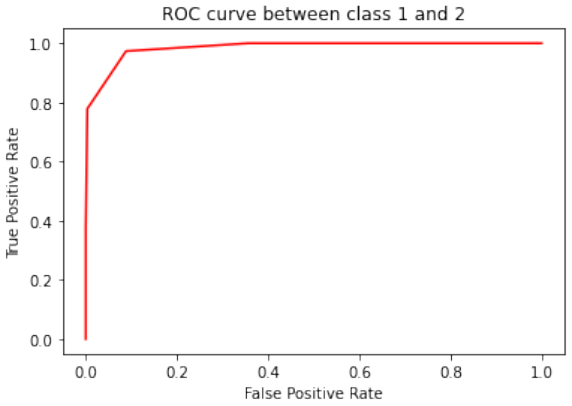
- **Metrics for Naive Baye's Classifier with 20:80 split**



- **Metrics for Multi-Class Logistic Regressor with 20:80 split**



- **ROC Curve**



5.4. Inferences

Naive Baye's

- The observed accuracies are marginally lower, which is not surprising since there is a loss of information when the data is compressed using PCA

Multi-Class Logistic Regression

- The observed accuracies are lower than those of p4, which is expected since there is a loss of information when the data is compressed using PCA.
- ll other observations are similar to that of p4.

Multi-class Bayes' classifier with GMM

- Issue of the curse of dimensionality is fixed here as the number of dimensions is decreased to 2.
- Accuracy is less due to loss of information due to a decrease in dimensions.
- Increase in log-likelihood after each iteration is observed as expected from GMM algorithm, this observation is captured in the log-likelihood curves in the plots.

6. References

1. Pattern Classification Book by David G. Stork, Peter E. Hart, and Richard O. Duda

2. Pattern Recognition and Machine Learning Book by Christopher Bishop

3. Towards Data Science - Gaussian Mixture Models & EM Algorithm.

Metric	*20:80*	*30:70*	*50:50*	*70:30*	*90:10*
Classification Accuracy	86.9 %	86.8 %	86.6 %	86.6 %	86.38 %
F1 scores	$\begin{bmatrix}0.84620.8879\\0.9451\\0.8718\\0.9125\\0.9155\\0.7700\\0.7143\\0.8854\\0.9129\end{bmatrix}$	$\begin{bmatrix}0.8462\\0.8884\\0.9468\\0.8763\\0.9161\\0.9194\\0.7705\\0.7133\\0.8860\\0.9149\end{bmatrix}$	$\begin{bmatrix}0.84290.8854\\0.9457\\0.8784\\0.9176\\0.9202\\0.7783\\0.7213\\0.8855\\0.9197\end{bmatrix}$	$\begin{bmatrix}0.83560.8830\\0.9448\\0.8769\\0.9094\\0.9206\\0.7842\\0.7264\\0.8804\\0.9182\end{bmatrix}$	$\begin{bmatrix}0.83850.8755\\0.9470\\0.8774\\0.9051\\0.9099\\0.8032\\0.7336\\0.8751\\0.9228\end{bmatrix}$

Table 7. Metrics after PCA for Naive Bayes

Metric	*20:80*	*30:70*	*50:50*	*70:30*	*90:10*
Classification Accuracy	86.9 %	86.8 %	86.6 %	86.6 %	86.38 %
F1 scores	$\begin{bmatrix}nan0.8913\\0.8887\\0.7840\\0.8347\\0.8855\\0.7435\\0.6137\\0.8534\\0.5879\end{bmatrix}$	$\begin{bmatrix}nan0.8872\\0.8921\\0.7911\\0.8403\\0.8883\\0.7473\\0.6144\\0.8585\\0.5909\end{bmatrix}$	$\begin{bmatrix}nan0.8894\\0.8949\\0.8116\\0.8405\\0.8919\\0.7589\\0.6322\\0.8559\\0.5982\end{bmatrix}$	$\begin{bmatrix}nan0.8872\\0.8959\\0.8182\\0.8290\\0.8895\\0.7646\\0.6389\\0.8571\\0.5971\end{bmatrix}$	$\begin{bmatrix}nan0.8892\\0.9092\\0.8191\\0.8198\\0.8721\\0.7931\\0.6473\\0.8581\\0.6001\end{bmatrix}$
Empirical risk on Train data	569.30068	660.75209	797.96023	879.72766	954.49764
Empirical risk on Test data	559.13571	656.43788	790.89523	889.27140	975.18455

Table 8. Metrics after PCA for Multi-Class Logistic Regression

Metric	*20:80*	*30:70*	*50:50*	*70:30*	*90:10*
Classification Accuracy	68.46 %	64.01 %	65.99 %	73.00 %	70.16 %
F1 scores	$\begin{bmatrix}0.7117\\0.8281\\0.9168\\0.7474\\0.7812\\0.8473\\0.6557\\0.6097\\0.8794\\0.8820\end{bmatrix}$	$\begin{bmatrix}0.7057\\0.7812\\0.9055\\0.7019\\0.8407\\0.8706\\0.6920\\0.5824\\0.8897\\0.8909\end{bmatrix}$	$\begin{bmatrix}0.7835\\0.8625\\0.9099\\0.6673\\0.8718\\0.8781\\0.7139\\0.5710\\0.8792\\0.8990\end{bmatrix}$	$\begin{bmatrix}0.7915\\0.8756\\0.8876\\0.7328\\0.8080\\0.8661\\0.7335\\0.6040\\0.8704\\0.8907\end{bmatrix}$	$\begin{bmatrix}0.8233\\0.8846\\0.9132\\0.7728\\0.8490\\0.8613\\0.7404\\0.6278\\0.8760\\0.8949\end{bmatrix}$

Table 9. Metrics after PCA for GMM