CS5011 Machine Learning Contest Report

Srikanth Prabala, EE12B053
Rishiraj Surti, EE12B120
Rahul Vadaga, EE12B122
Team ID: 14

November 17, 2015

1 Dataset

The data provided (`train_X`) has 2048 attributes in all 14332 observations. There are 100 classes in the data. 80% of the data was used for training, and the remaining 20% was used for testing. This was done mainly for the purpose of evaluating the algorithms.

2 Algorithm

Instead of developing new methods, we thought of implementing the existing classification algorithms and take the maximum likely class labels from the outputs.
A histogram was made using `R` to study the class distributions. The plots have been generated for all the 3 different datasets (ref. Fig. 1, 2, 3).

We implemented the following algorithms:

- LibSVM, Gaussian, Polynomial, Sigmoid Kernels.
- Sklearn SVM, Gaussian Kernel
- KNN, k=7
- LDA
- Neural Network (Backpropagation)
- Bayes, Bernoulli and Multinomial

For the test data, we took the output which had a maximum vote among these algorithms. To try something different, we thought of giving priorities. But that didn't work out well. So instead, we repeated the outputs of some of our algorithms. As an example, from our experience, Neural Network works better. So we just repeated its output twice. If NN gave a wrong output, others would take care. But if it is a correct one, we made sure it gets noticed. In this case as the input data is very large so we have used large number of hidden neurons which tries to capture the data correctly. The learning rate is kept low

so that the function settles to a good optimum value.

Linear Discrimant Analysis (LDA) was performed on the dataset using `scikit-learn` package. As mentioned above, 80% of the data in `train_X` was used for training. We obtained an accuracy of 1.0 on the test data viz., 20% of the data in `train_X`.

We also used the `kNN` classifier for prediction. In order to determine the optimal number of neighbors, we used the `accuracy_score` metric from the `scikit-learn` package. We landed up on $k = 7$ as the optimal value, as it gave the highest accuracy score among all the others.

We also came to conclusion that SVM with Gaussian kernel gave better F1 measure than others. So our prediction was that the data was some kind of mixture of Gaussians. As the f1 measure of other teams revolved around the value 0.6, we gauged that it could be a mixture of "highly overlapping" gaussians, impossible to seperate beyond a certain point. We tried to work around the same idea.

3 Code

We had a private Github repository to work for this contest. The link to repo:
ml_contest
ml_contest_graphs
Please mail me at rishirajsurti.iitm@gmail.com with your Github username to gain access to repo. The file which generates the final output is "compare.py". This has an array of output files of algorithms implemented and chooses the most likely label among these.

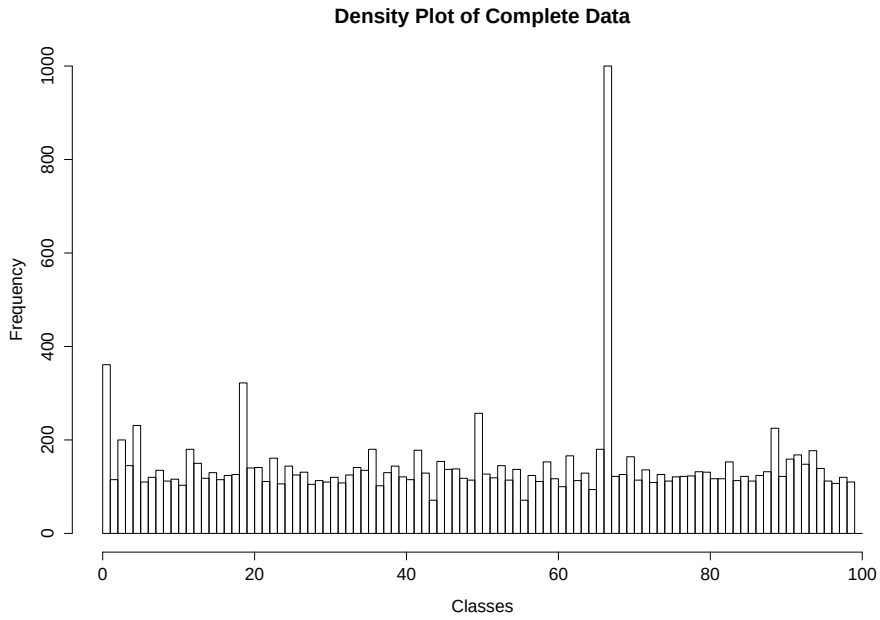

Figure 1: Histogram showing class frequency of complete data

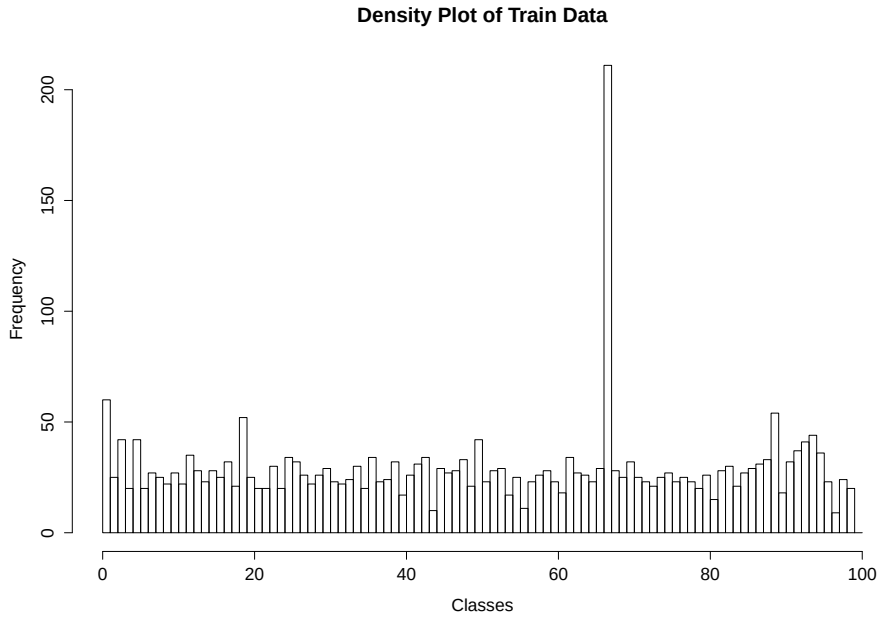

Figure 2: Histogram showing class frequency of training data

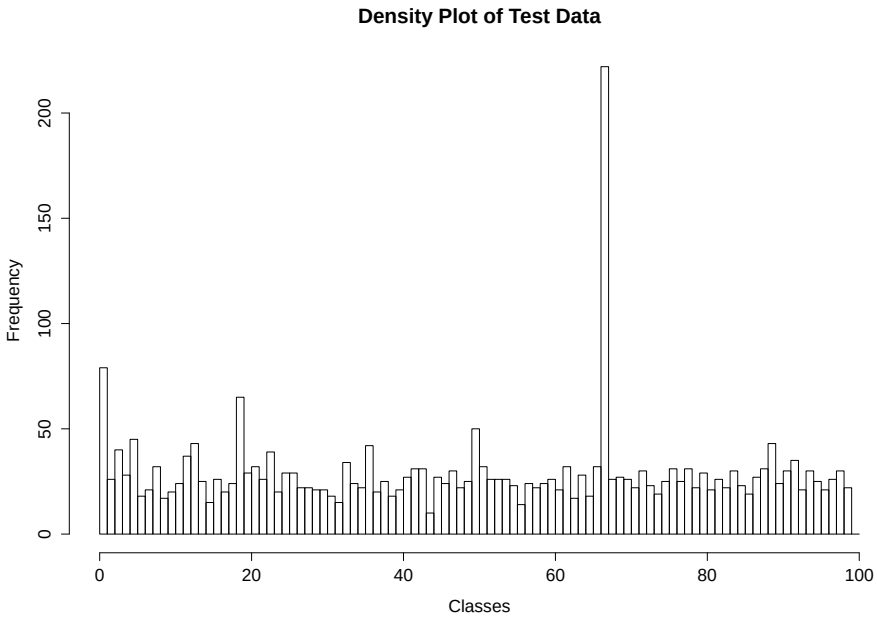

Figure 3: Histogram showing class frequency of test data