ML Classification

1 Variables

The data consists of 294 features and we need to use it to classify classes from A-E. On looking at the variable we can categorize them as:

- Unary – 11 features, for e.x. column 59 has only 0s.
- Binary – 277 features.
- Ternary – 4 features (column 4, 23, 36, 43)
- Continuous – 3 features (3, 64, 294)

Since, the unary variables don't add any value for classification, we will ignore the classes.

If we check the classes, the 5 classes are unequally distributed. With 70% of the samples belonging to class 'C' and least samples belonging to class 'A'.

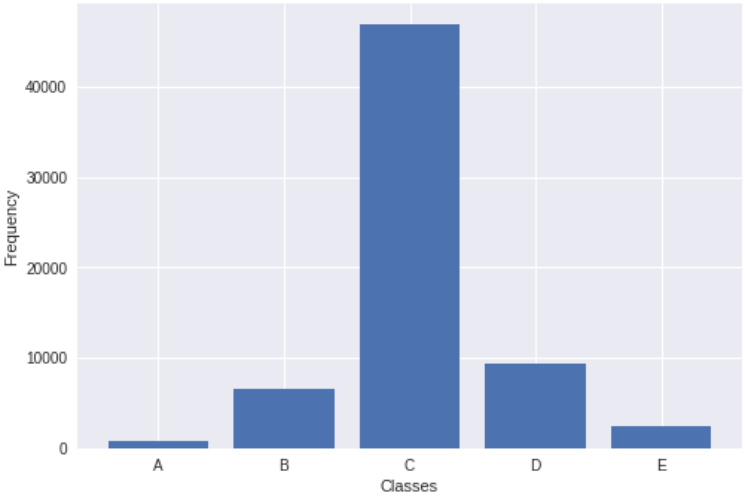

Fig 1. No. samples in different classes.

2 Challenges

- Unbalanced data
- High dimension

3 Classification

Before moving to solution, I want to apply different classifiers to have an insight about how classes are separated in the dataset. I am using the below classifiers:

- A linear classifier: I want to check if the classes are linearly separable and use a Linear SVM classifier. Although it is highly unlikely, but it can provide us insights about the classes.

- A generative model: A generative model estimates the joint probability distribution of data P(X, Y) between the observed data X and corresponding labels Y. Since, most of the features are binary, I want to check using Naïve Bayes classifiers if there is enough information in the dataset such that the joint probabilities can be found which can distinguish between classes.

- An ensemble model: Random Forests creates a set of classifiers by random subset of features. They have shown good results on structured datasets. I will be interested in seeking its performance on the dataset.

- A neural network model: Since the data is structured and non-sequential, I have used simple feedforward algorithm as prediction models.

3.1 Discussing results

- F1 score is used to compare the performance of classifiers as the classes are unbalanced.
- Looking at the performance of classifiers for each class, classifiers have very high precision and recall for class 'C' but for other classes the values are very bad.
- Linear SVM is worst among the three classifiers in classifying under-represented classes and overall performance.
- The Naïve Bayes classifier has better accuracies for under presented classes but comparatively bad for class 'C'.
- Random Forest is slightly better in terms of final/overall accuracy than the other two classifiers.
- class_weights: for unbalanced data, a parameter class_weights are present in SVM, Random Forest and neural network. Using these parameters helped in having better performance for under-represented classes.
- The ensemble of the four models provides only slightly better precision than random forest.
- Feedforward neural networks performed better than Convolution Neural Networks and LSTMs.

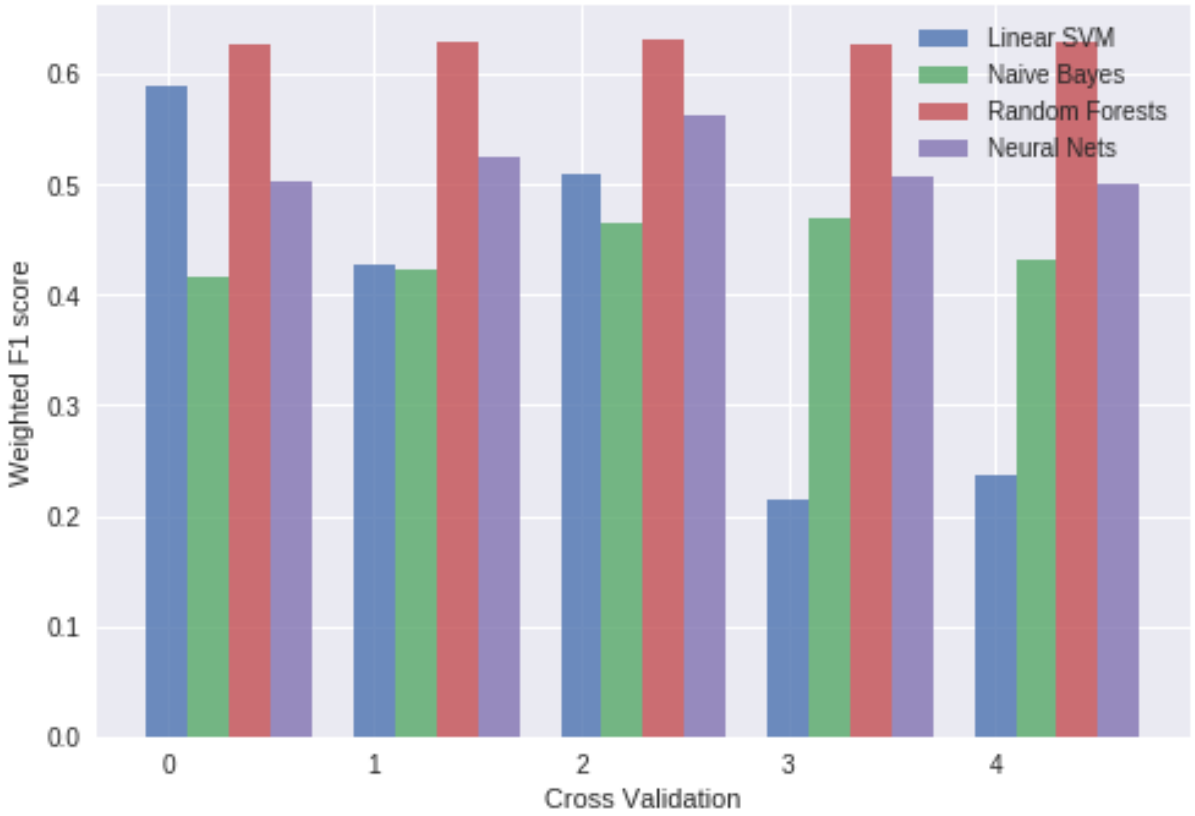

Fig 2. Comparison of weighted F1 score of different classifiers.

For finding the best classifiers from the set of 4 classifiers, I have used Friedman Rank test. As it is evident from the bar plot of different classifiers, Random Forests were best at predicting the models followed by Feedforward Neural Network, Naïve Bayes and SVM.

4 Further analysis

- Down sampling/up sampling: To mitigate the unbalanced samples across different classes, one of ideas is to down sample the majority class or up sample the less frequent class.

- Dimensionality reduction: I have removed few unary features, but we can further remove few dimensions based on random forest feature importance. This will help our classifiers to learn better.

- I tried training simple Convolution Neural Network (CNN) by reshaping the features and LSTM. LSTM was not able to learn at all and CNN also learned the majority class. Also, I had problem training due to computational problems. I was interested in using Autoencoders as classifier or dimensionality reduction but I did not had enough of computation power to test it.