

Weekly Report 3

BTP

Team Members -

Tushar Gupta - 18085073

Tushar Singh Diwakar - 18085074

Vipin Sharma - 18085080

After feature extraction we worked with three algorithms/approaches to classify different genres. The three approaches as described in previous mail (Weekly Report 2) were:

1. **Ensembling Approach** - We created an ensemble of the of classifiers like Random Forest Classifier , Support Vector Machines , Logistic Regression , Decision Classifier , Naive Bayes and KNeighbours Classifier .This ensemble classifier use the prediction of each classifier and run a majority voting heuristic to obtain the optimal class label for given test input. The weights used to average the classifiers are proportional to the accuracy of the classifiers. We got an accuracy score of 0.46 using it.

Link To the Python Notebook -

<https://colab.research.google.com/drive/1RMYXrQ4r3wRvUSKpSqh8qPtcw-Q7fw-N?usp=sharing>

2. Convolutional Neural Network -

Convolution: This step involves sliding a matrix filter (say 3x3 size) over the input image which is of dimension image width x image height. The filter is first placed on the image matrix and then we compute an element-wise multiplication between the filter and the overlapping portion of the image, followed by a summation to give a feature value.

Pooling: This is a way to reduce the dimension of the feature map obtained from the convolution step, formally known as the process of down sampling

Non-linear Activation: we can apply an activation function such as ReLU on each element of the feature map.

The final layer of the neural network outputs the class probabilities (using the softmax activation function) for each of the seven possible class labels

We got an accuracy score of 0.65 using it.

Link To the Python Notebook -

https://colab.research.google.com/drive/1iiZLMkM503il7hDn1du_FPkxwRtuWocq?usp=sharing

3. Light Gradient Boosting Machine - LightGBM is a gradient boosting framework based on decision trees to increase the efficiency of the model and reduce memory usage. LightGBM splits the tree leaf-wise as opposed to other boosting algorithms that grow tree level-wise. It chooses the leaf with maximum delta loss to grow. Since the leaf is fixed, the leaf-wise algorithm has lower loss compared to the level-wise algorithm. Leaf-wise tree growth might increase the complexity of the model and may lead to overfitting in small datasets.

We got an accuracy score of 0.88 using it, which is highest among the three approaches.

Link To Python Code -

https://colab.research.google.com/drive/16kYILIKLhFsJCy7JVr_4creYAI0IRJXC?usp=sharing