Saranya Soundarrajan *SJSU ID: 011502985*

IMAGE CLASSIFICATION REPORT

The purpose of the assignment is to develop predictive models that can determine, given an image, which one of 11 classes it is. The method uses **Voting Classifier (Randomforest, KNN, XGBoost, Adaboost, Extratrees) & Variance Threshold** for feature selection to predict the class labels. The code is written in Python using the scikit-learn, imblearn library. Code is uploaded in .py format. Below are the steps followed to achieve the F1 Score of **.8484** securing 1st rank while writing the report.

Approach:

1. Used SMOTE over sampler for imbalanced data.
2. Used VarianceThreshold for feature selection.
3. Used VotingClassifier with several classifiers *(onevsrest KNN, onevsrest ExtraTrees, XGBoost, Randomforest, Adaboost***)** with weights attached (1,2,1,2,1) to classifiers.

Step 1 : Analyzing Document

Data was imbalanced in nature, which meant minority class will not be predicted well. Images are described as 887-dimensional features, we also have to find a way to reduce the dimensions, by removing the irrelevant features or creating a feature subset of relevant ones.

Step 2 : Normalize the data

Loading the train.dat file and train.labels file. Used **StandardScalar** to center sparse data to scale sparse inputs as the features are on different scales.

Step 3 : Feature Selection

Feature selector (VarianceThreshold) removes all low-variance features. Out of 887 features, only 48 features are selected from the train dataset.

Step 4 : Split Train/Test Data & Run Classifier Algorithms

Training data is split into train (80%) and test data (20%) to test the accuracy of different Classifiers. XGBoost, ExtraTrees, KNN, Random Forest, SVM with default parameters was trained and crossvalidated to get the accuracy of the classifiers. Random Forest, ExtraTrees, XGBoost achieved the highest accuracy of 83%, followed by KNN (77%).

```
Accuracy: 0.8314 (+/- 0.0171)
Mean fit time: 398.2691 ms
```

```
Accuracy: 0.7733 (+/- 0.0134)
Mean fit time: 2.2926 ms
```

Step 5 : Dealing with Imbalanced data

Data was imbalanced in nature, which meant minority classes are not well represented (Labels 10, 11). Used **SMOTE over sampler,** which performs over sampling by creating synthetic samples for minority class. Assigned ratio for minority class. Applied VotingClassifier on Random forest, KNN, XGBoost, ExtraTrees, Adaboost. Randomforest was able to give the highest accuracy for the majority classes and KNN was able to give highest accuracy minority class (10,11). So combined all classifiers with Voting and applied weights on each classifier after doing crossvalidation which significantly increased accuracy to 84%.

```
Accuracy: 0.8437 (+/- 0.0166)
Mean fit time: 773.5820 ms
```

Step 6 : Result

Re-trained model on full training set, using best parameters and applied re-trained model to test set. The output predicted is then stored onto prediction.dat, achieving .8484 on CLP Leaderboard.

Final Leaderboard:

Rank	F1	User ID	Submission Count
1	0.8484	11502985	17
2	0.8477	11810721	15
3	0.8451	12424126	8