Home Assignment 2_Part II report

Question 1: Using your algorithms for the emotion prediction with given EEG data

I picked up the EEG dataset from https://github.com/PerforMance308/EEG_Dataset by following below steps:

```
In [1]: import load_data

In [3]: data = load_data.read_data_sets(one_hot=True)

        Start downloading dataset...
        Successfully downloaded train 209530726 bytes.
        Start downloading dataset...
        Successfully downloaded test 144273978 bytes.
```

Figure 1: Downloading the EEG dataset files.

```
In [5]: import numpy
        train_x, train_label = data.train.next_batch(35000)
        test_x,test_labels = data.test.next_batch(5000)

        b = numpy.asarray(train_x)
        numpy.savetxt("trainingData.csv", b, delimiter=",")

        a = numpy.asarray(test_x)
        numpy.savetxt("testingData.csv", a, delimiter=",")

        a_label = numpy.asarray(test_labels)
        numpy.savetxt("testingLabels.csv", a_label, delimiter=",")

        b_label = numpy.asarray(train_label)
        numpy.savetxt("trainingLabels.csv", b_label, delimiter=",")
```

Figure 2: Saving the dataset files with the proper file format.

This is a **multiclass classification** problem.

I used the **Support Vector Machine** algorithm for predicting the emotions into 3 class labels that is "*positive*", "*neutral*", "*negative*". As, the given EEG dataset consists of 3 columns output (class label) matrix where I implemented the **data preprocessing** and converted that into single column where**, *positive* emotions has a class label '1', *neutral* emotion as '2' and *negative* emotion as '3'**.

In **MATLAB**, I used '**fitcecoc'** function for building an SVM algorithm model from training dataset and used this model to predict the emotions in testing dataset. So here, this model will be trained through a binary learner where, multiclass class labels are taken one by one in 2 class indices that is positive and negative class indices and as we are having 3 unique class labels, this model will be trained with 3 binary learners. See Figure 3 for the output of binary learners.

Home Assignment 2_Part II report

```
Command Window
>> SVM
Training binary learner 1 (SVM) out of 3 with 6317 negative and 6807 positive observations.
Negative class indices: 2
Positive class indices: 1

Training binary learner 2 (SVM) out of 3 with 6876 negative and 6807 positive observations.
Negative class indices: 3
Positive class indices: 1

Training binary learner 3 (SVM) out of 3 with 6876 negative and 6317 positive observations.
Negative class indices: 3
Positive class indices: 2
```

Figure 3: Output for 'fitcecoc' function to model SVM through binary learners.

So, as you can see first SVM binary learner used *neutral* and *positive* emotions, second SVM binary learner used *negative* and *positive* emotions whereas, third one used *negative* and *neutral* emotions as its class labels. Total training time taken is **439.2188 seconds**.

I am printing a table for true and predicted labels for 10 random instances plus overall and top1 model accuracy without feature selection. See Figure 4.

```
Command Window
ans =

  10×2 table

    TrueLabels    PredictedLabels
    __________    _______________

        2                2
        1                1
        3                1
        2                2
        2                2
        3                3
        1                2
        1                3
        1                1
        2                1

Total model accuracy without feature selection
    0.7873

Accuracy without feature selection
   94.8553
```

Figure 4: True and predicted output for 10 random samples and overall and top1 model accuracy without feature selection.

Without feature selection, one of the class labels was predicted with **94.85%** accuracy which can be called as Top1 accuracy. And overall accuracy is **78.73%.**

The confusion matrix for class labels without feature selection can be visualized in Figure 5.

Home Assignment 2_Part II report

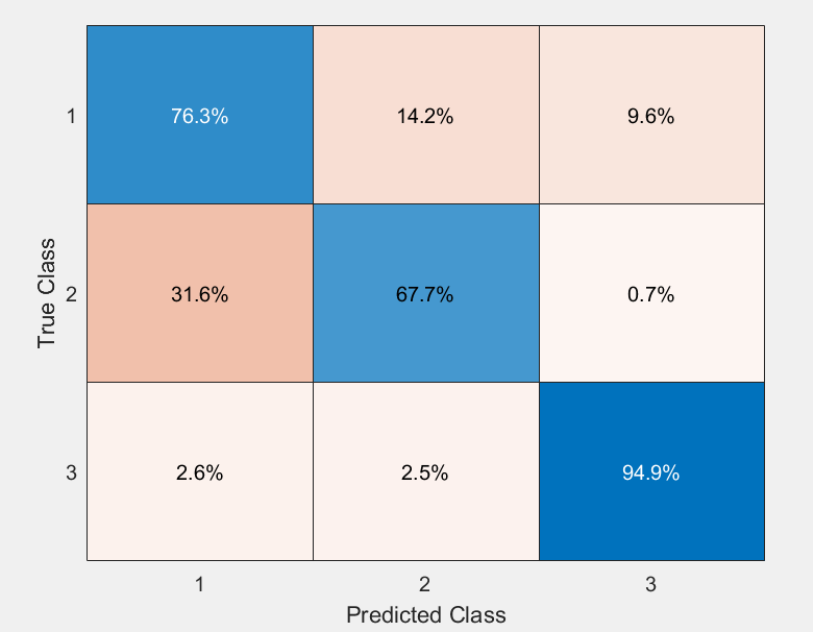

Figure 5: Confusion matrix for multiclass labels without feature extraction.

It shows the positive rate accuracies according to the class labels. **76.3% for positive class labels, 67.7% for neutral class labels and 94.9% for negative class labels**.

With Feature extraction algorithm (**Minimum Redundancy Maximum Relevance** (MRMR)), I ranked the features according to their importance for building a model and I used top 280 features for building a better SVM model. See Figure 6 for feature importance score vs feature rank for 310 features.

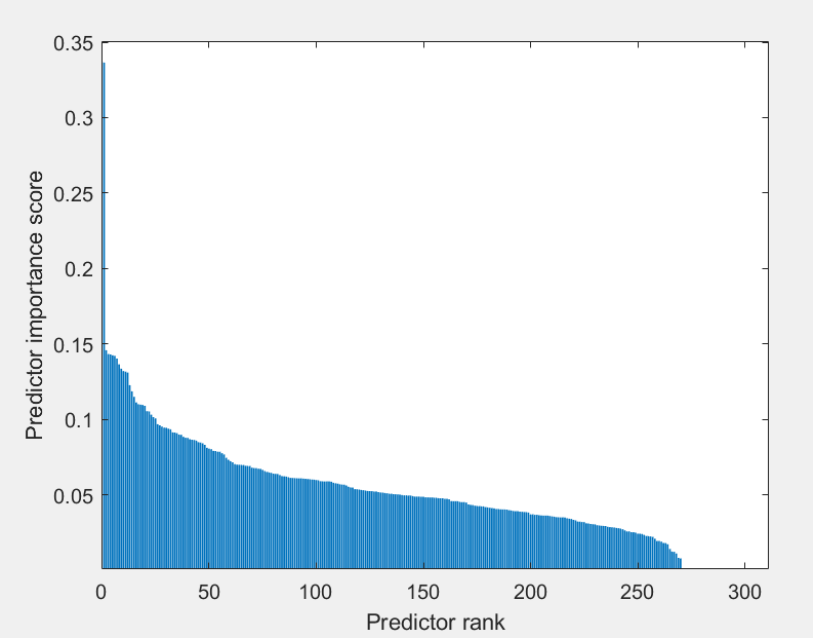

Figure 6: Feature importance score vs feature rank graph

The output for SVM model with feature extraction method can be seen in Figure 7.

Home Assignment 2_Part II report

```
Command Window

ans =

  10×2 table

    TrueLabels    PredictedLabels
    __________    _______________

        3               3
        2               2
        2               2
        2               2
        2               2
        1               2
        1               2
        2               2
        2               2
        3               3

Total model accuracy without feature selection
    0.8056

Accuracy with feature selection
   93.7938
```

Figure 7: An output with feature extraction in SVM model.

With feature extraction MRMR algorithm, total training time taken is **597.6719 seconds**, total model accuracy increased to **80.56%** and average top 1 accuracy is now **93.79%**. From Figure 8 confusion matrix for SVM model with feature selection, we can see that accuracy for *"positive"* and *"neutral"* class labels has been increased significantly.

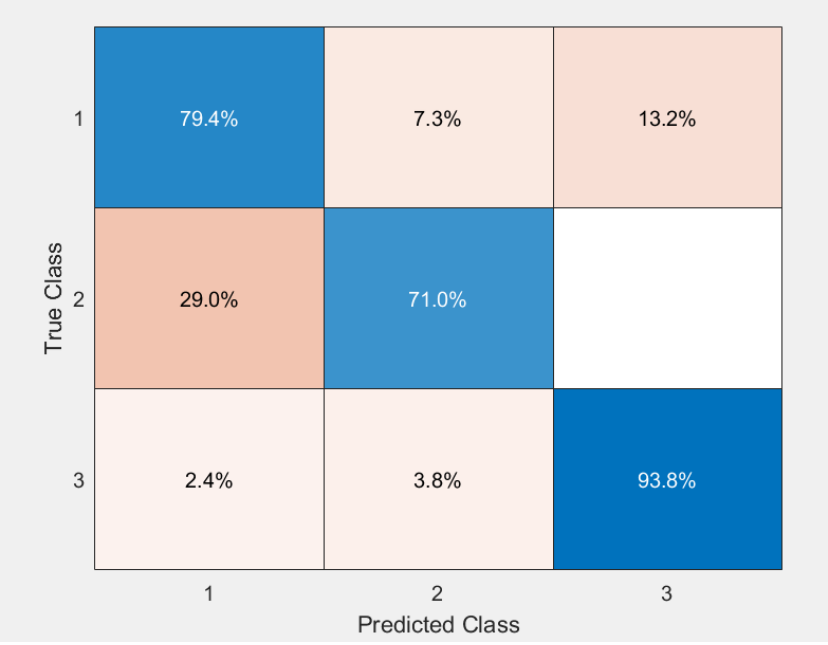

Figure 8: Confusion matrix with MRMR feature selection technique for SVM model

Home Assignment 2_Part II report

- **Steps to run an SVM algorithm** MATLAB code for emotion prediction on EEG dataset:

1. Run the ‘SVM.m’ file in MATLAB command window.