CSE583/EE552 Pattern Recognition and Machine Learning: Project #2

Due on February 15th, 2022 at 11:59pm

PROFESSOR Yanxi Liu Spring 22

Anish Phule **asp5607@psu.edu**

This report consists of:

Problem 1 - Feature Subset Selection:

1. Filter and Wrapper Criteria

2. Pipeline Description

3. Histogram of Most Discriminative features

4. Classification results:
   - Baseline
   - Filter only
   - Filter+Wrapper

5. Histogram of Most commonly selected Features

6. Subject Rates Comparison
7. Extra Credit

Problem 1

Feature Subset Selection

1. Filter and Wrapper criteria

Filter and wrapper are two different methods of feature selection, wherein Filter selects features based on a **prior** evaluation of the data w.r.t some criteria, while in wrapper we use the different features to check which features fit best, that is a **posterior** measure on the data.

Filter Criterion: For the filtering part, we use the Variance ratio test, wherein we calculate the variance ratio for every feature in the subset, and arrange them in an ascending order. The Variance ratio accounts for the means with between-class variance [1]. The expression for VR is as follows [2],

$$VR(F) = \frac{Var(S_F)}{1/C \sum_{k=1,2...C} Var_k(S_F)} \tag{1}$$

where $Var(S_F$ is the variance for feature F, and $Var_k(S_F)$ is the variance of elements that belong to feature F and class k out of C.
We then pick the top N features and perform classification based on these features. This significantly reduces the essential features required.

Wrapper Criterion: The wrapper method is based on the idea of Maximizing Classification rate, i.e. evaluating feature combinations for best number of features that is optimum as well as maximizing the output.

For the wrapper method, we take the top N features selected from the filtering method. We then evaluate the data for training accuracy for 1 feature using discriminant classifier for all features. The feature with the highest accuracy score is stored and we choose another feature from the remaining list. Together with the first feature and the second one, we evaluate the training accuracy again. The combination that gives the best score is again stored. We choose a third feature, and so on and so forth.

The process goes on until there is no increase in the training accuracy scores i.e. no more improvement in the results. The new number of features M, is then used to do the test classification and is made the model. Thus we get an optimum amount of features, significantly reducing model complexity.

2. Pipeline Description

The following text is a description of our classification code and its component codes.

- We start by setting up our runtime variables, setting how many maximum features we want(feature_select_count) to select through filtering and wrapping, and load out Taiji Dataset.
- The next part deals with the training and evaluation(testing) of data. This part loops over every one of the 10 subjects while leaving one subject out (LOSO).
- This happens in the 'split' function, where one subject that is chosen is used as test data while the other 9 are used as training data.
- We then normalize the data between 0 and 1 to remove any weighted means or variances.This happens on both test and train data.
- The following section deals with feature selection, that is filtering and wrapping. These are separate functions as defined below.

  - Filtering:
    - The filterMethod code takes in training data and labels, and sorts them according to ascending order of the labels. This makes it easier to perform analysis on the data.
    - We use the Variance ratio criterion, that the feature with the highest variance ratio is ranked up, and the rest are ranked in descending order.
    - We then return this descending order sorted index list and scores to the original code, where the train and test features corresponding to our filtered feature list is sorted and sent forward.
  - Wrapping:
    - The forwardSelection code implements the wrapping method. It takes the training data and labels, as well as the number of maximum features we want(feature_select_count) as input.
    - We create two vectors, feature index, which is an array of all features from 1 to 100(or 1 to feature_select_count) while final index is the vector of our finally selected indexes.
    - We now define our wrapping algorithm. We define a while loop with the condition, wherein if the new train accuracy score is the old accuracy score, the loop keeps running. The meaning of these scores is described in the points below.
    - Our final index is initially an array of zeros. We then populate the first element with all feature numbers from 1 to 1 to feature_select_count, and for every iteration, do classification and calculate a training accuracy score. The index with the highest score is stored. This highest score is stored in the new accuracy score, and the previous value of new accuracy score is stored in old accuracy score.
    - For the second iteration of final index value, now the second element is iterated and the combination of the first(stored index) and second(iterated index) is used for classification. The scores are evaluated and values stored as the above step.
    - Then the third item is iterated, then fourth, and so on and so forth.
    - This process goes on until there is no change in the new and old score values, that is, new train score = old train score. These are all the features that we actually need, and we take the final index vector and pass it on to our main code.
    - The main code then further chooses the indices as given by the wrapping code, for both train and test data.

- With the above values, we now move towards our model. We use linear discriminant classifier to perform training and testing on our data.

- The predict function uses the ClassificationModel code to evaluate the data. The function also evaluates overall accuracy per subject, subject training per-class accuracy, subject testing per-class accuracy, and also computes the time taken for training. This data is then printed.

- The above process, from point 3 happens for every one of the 10 subjects, thats is splitting, filtering, wrapping and model evaluation happens a total of 10 times.

- At the end, an overall train and test accuracy is calculated, and confusion matrice data is generated that can later be visualized in the visualize code.

- The above pipeline helps us select the optimum and most discriminant features that help get the best accuracy and reduce complexity.

3. Histogram of Most Discriminative features

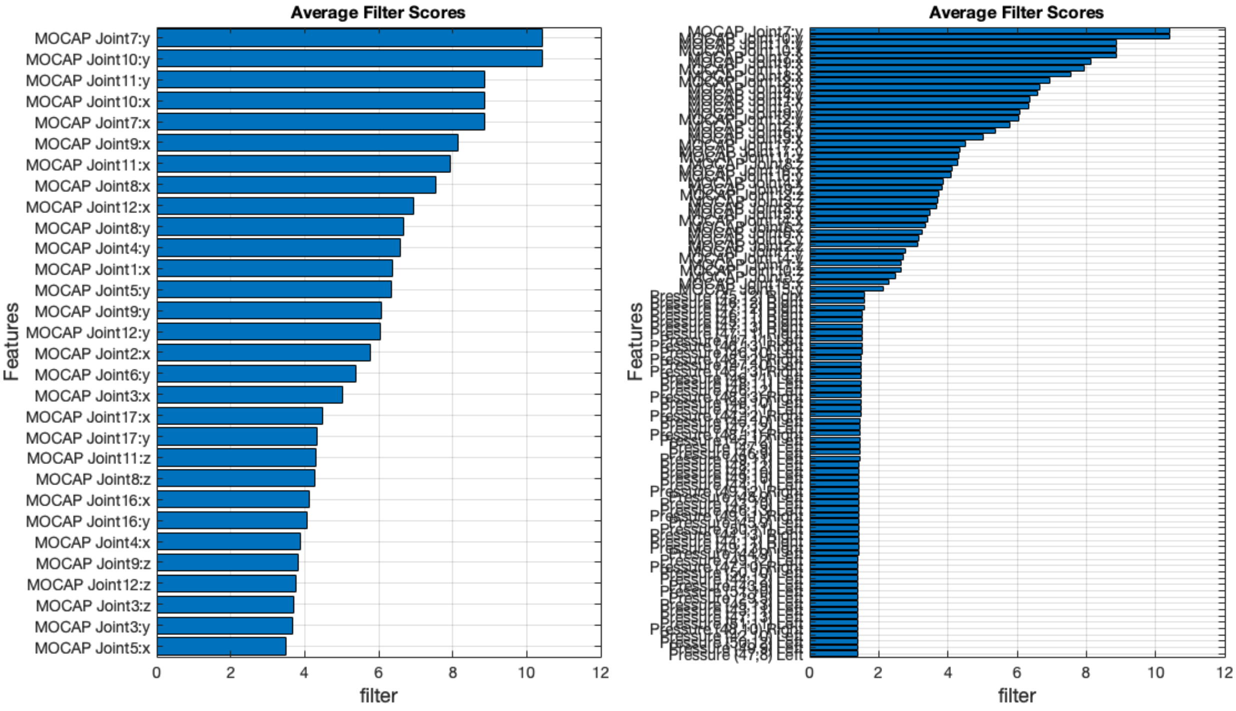

Figure 1: Histogram of the most discriminative features: (a)30 features, (b)100 features

We see a histogram of the most discriminative features. I have also included a histogram of 100 features, which shows how not all features actually contribute significantly to the filtering and training.

4. Classification Results

- Baseline Results(No filter + no wrapper):

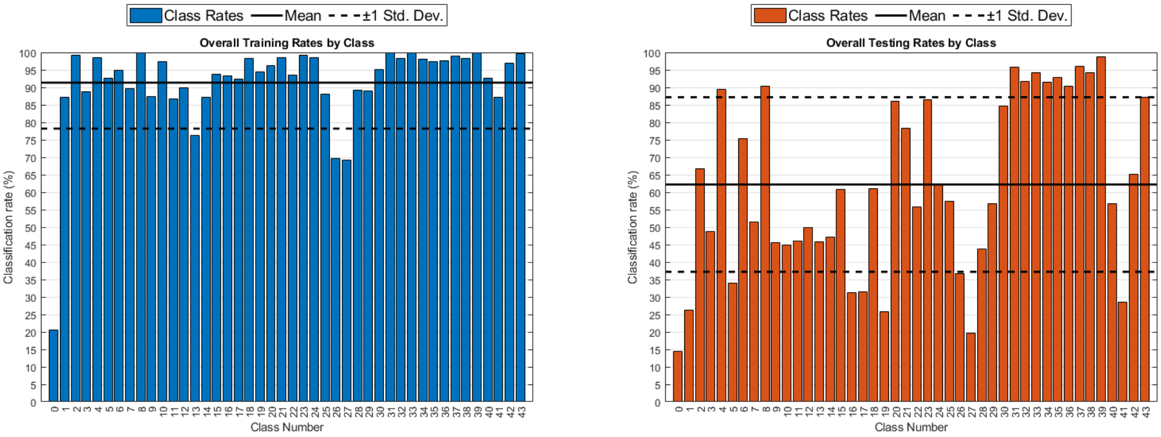

Figure 2: Baseline Classification rates + Std. Deviation (a)Train, (b)Test

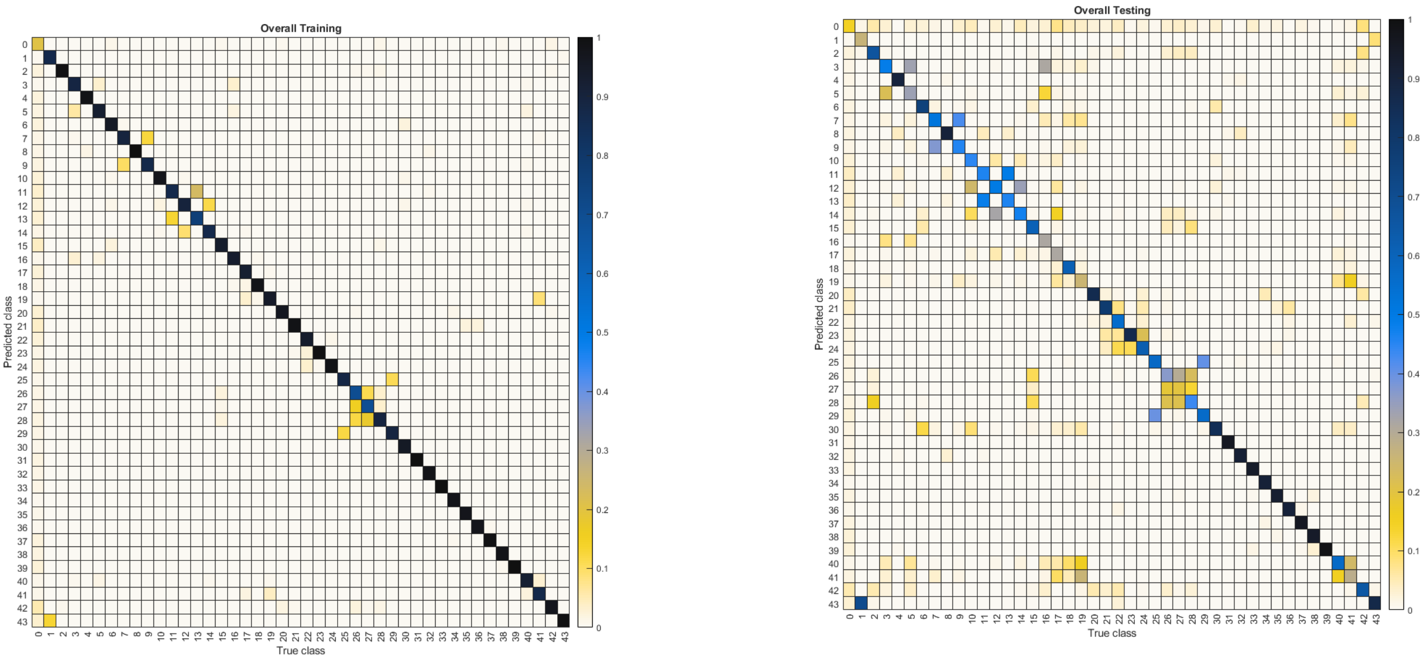

Figure 3: Baseline Confusion Matrices (a)Train, (b)Test

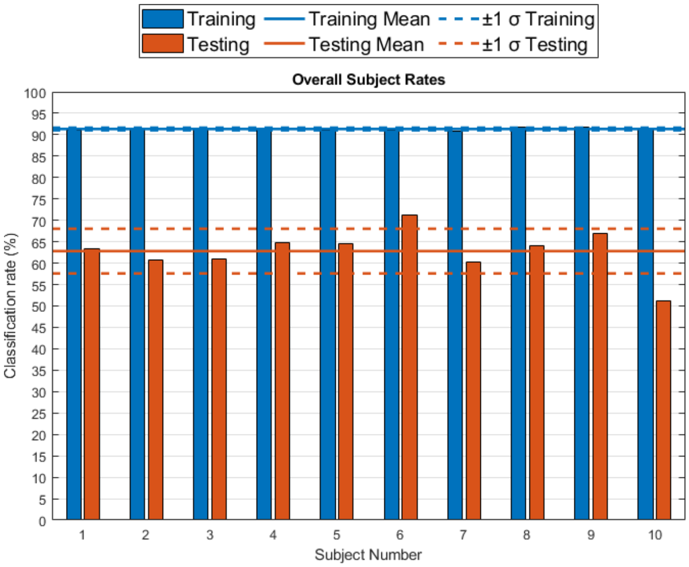

Figure 4: Subject rates

The above baseline results show us training and testing results for No filter and no wrapper applied to the data. As inferred from the confusion matrices and the classification rates, the training results are very high, as we are taking a lot of features, however the testing results are not great. The overall accuracy results, which are 91.34 and 62.81 percent for train and test respectively attest to the fact.

- Filter results(filter + no wrapper):

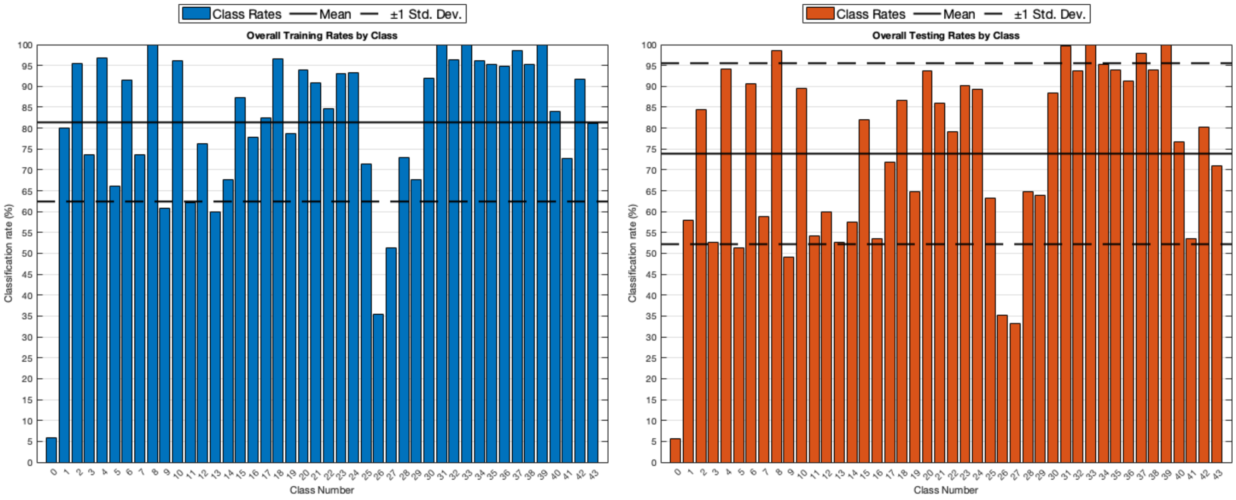

Figure 5: Filter Classification rates + Std. Deviation (a)Train, (b)Test

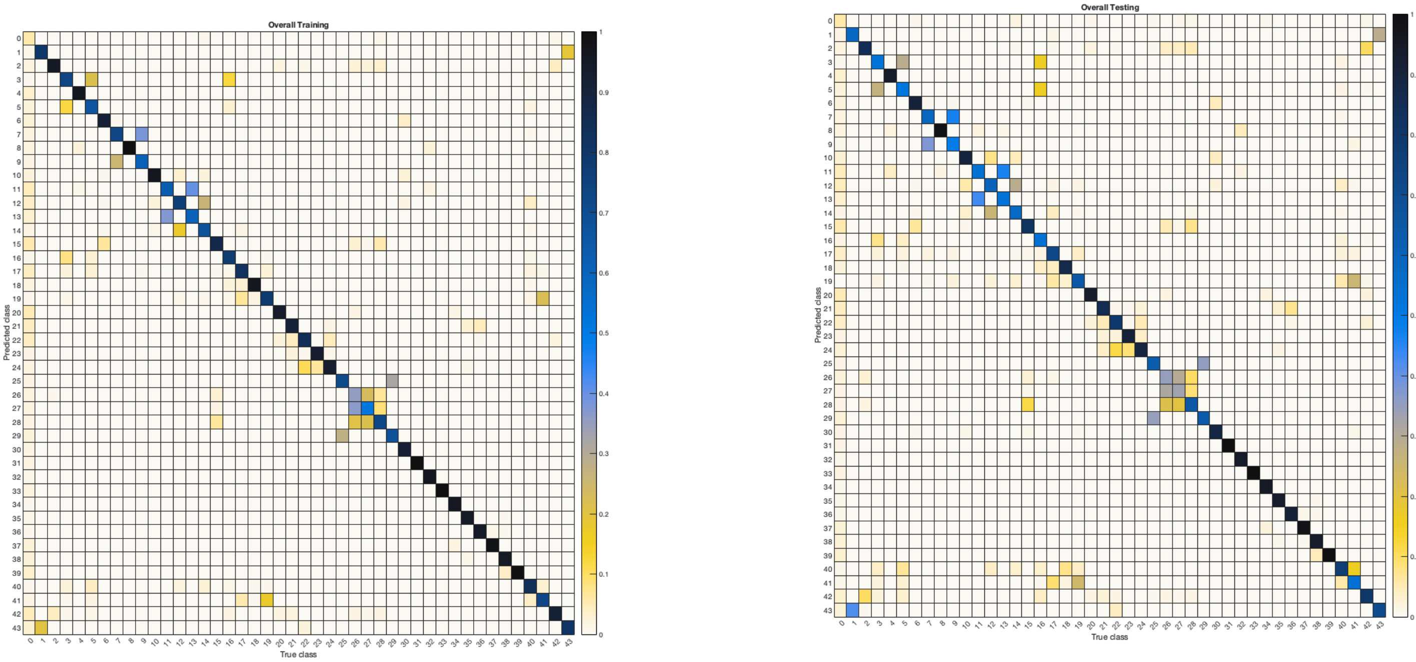

Figure 6: Filter Confusion Matrices (a)Train, (b)Test

The above results are for the model with filter applied on the features. The filter is based on the variance ratio, and selects 100 features with the top scores. This might give us worse training results, but significantly reduces dimensionality and increases the testing results.
As clear from the classification rates figures 4 and confusion matrices in Figure 5, the testing rates have significantly increased as compared to baseline results. The overall results are 81.34 and 74.13 percent for train and test respectively.

- Wrapper results(filter + wrapper):

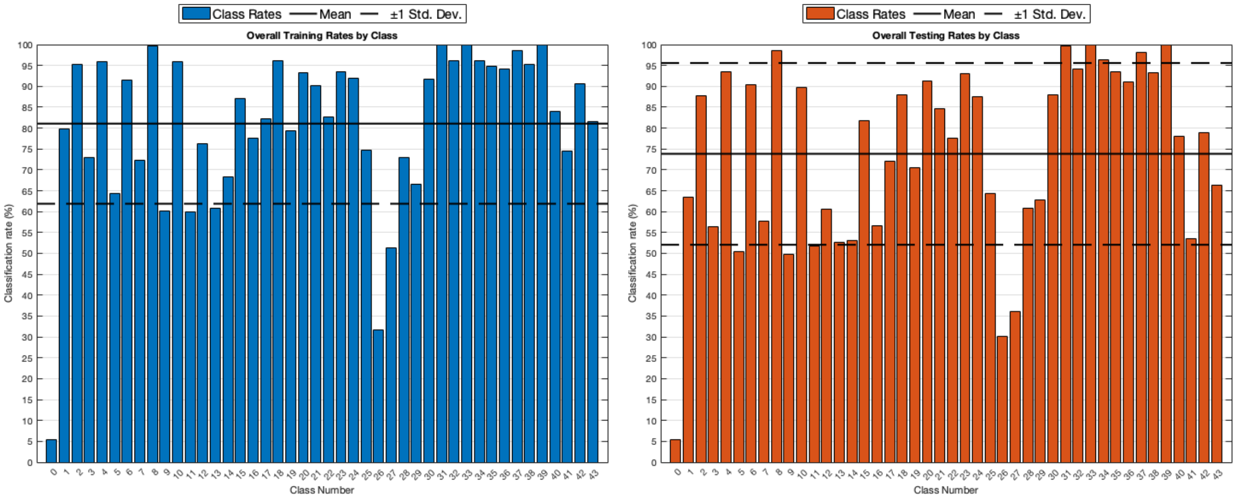

Figure 7: Wrapper Classification rates + Std. Deviation (a)Train, (b)Test

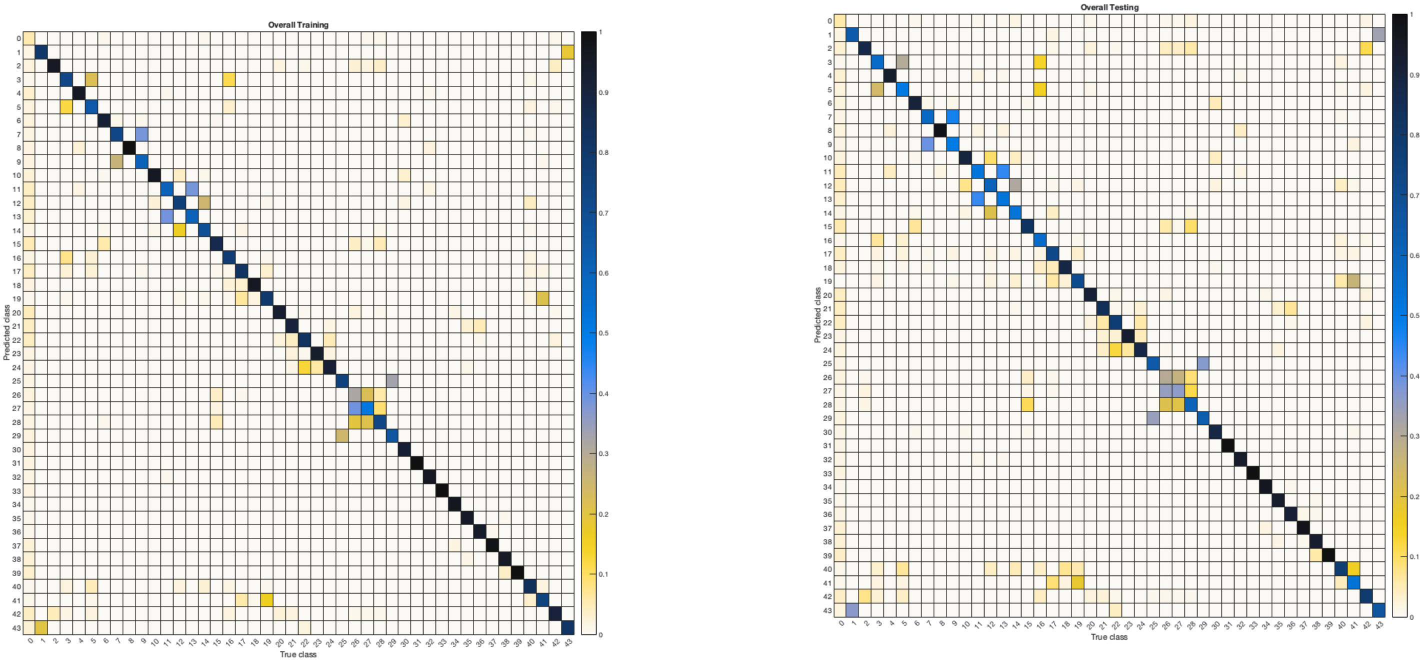

Figure 8: Wrapper Confusion Matrices (a)Train, (b)Test

The purpose of using the wrapper method is to optimize the number of features. The model does the same, that is use less than 100(around 70-80) features for each subject and gives us the same results as filtering. The test results are still better than baseline, thus being a better and less complex model. The overall accuracy rates are 81.02 and 74.03 percent respectively.

5. Histogram of Most commonly selected Features

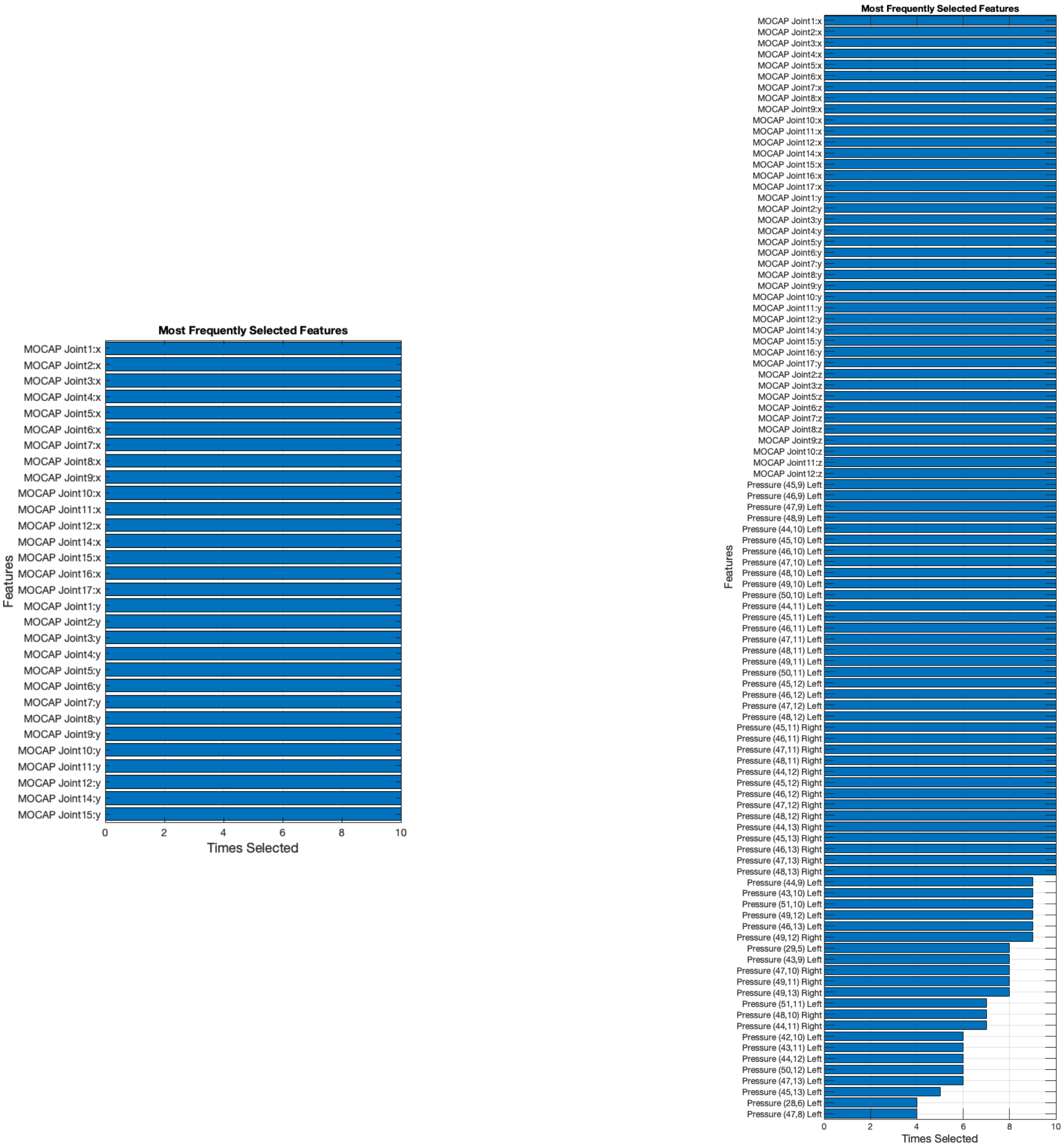

Figure 9: Most selected features for 100 features selected (a)Top 30, (b)All 100

The above figure shows histograms for the most commonly selected features for 100 features, top 30 and 100 of which are shown,

6. Comparison of Subject Rates

We compare the subject rates for all three scenarios.

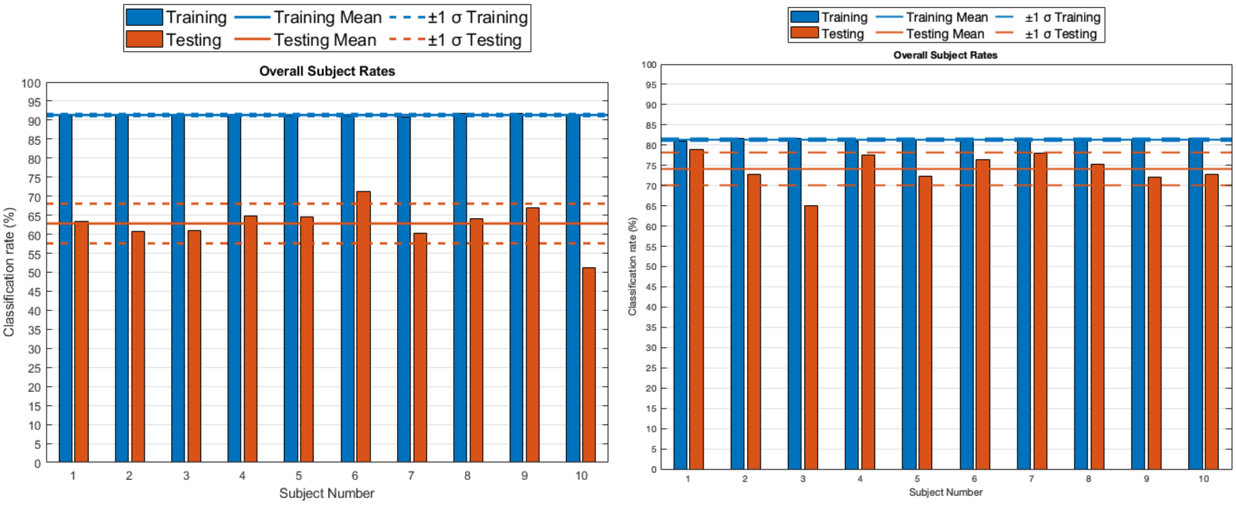

Figure 10: Subject rates (a)Baseline (b)Filter only

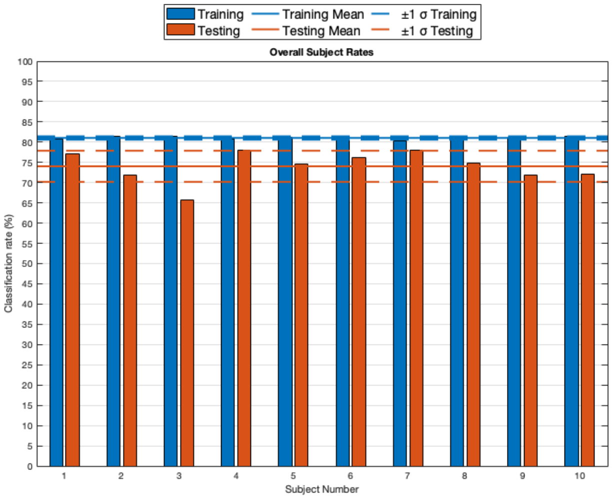

Figure 11: Subject rates(filter+wrapping)

The Subject rates gives us an idea of how the classification is working for every subject, that is training and testing rates for every subject. In the baseline results, we see very high training rates(averaging at around 92 percent), but very low testing rates(averaging at around 63 percent).
On the contrary, our filter and filter+wrapper methods, although have relatively worse training rates(averaging at around 82 percent) they also have relatively higher training rates(averaging at about 74 percent).

Thus, our new model with the filter and wrapper methods is successful at giving better results, with lesser number of features for all subjects.

Problem 1 continued on next page. . .

7. Extra Credit: Decision Tree model

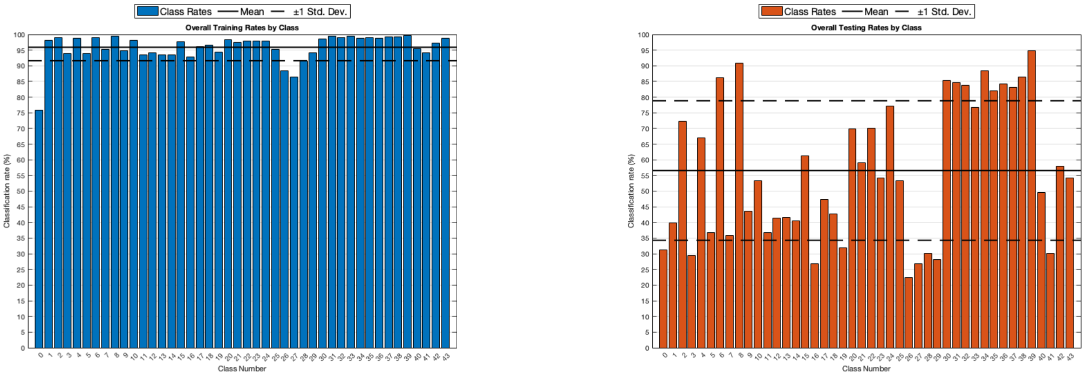

Figure 12: Decision Tree Classification rates + Std. Deviation (a)Train, (b)Test

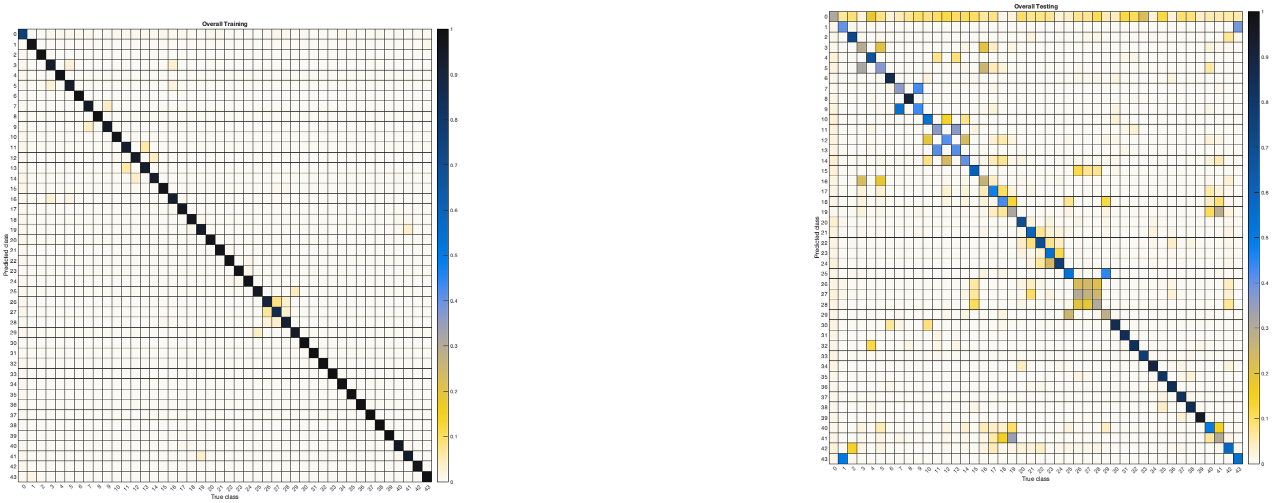

Figure 13: Decision Tree Confusion Matrices (a)Train, (b)Test

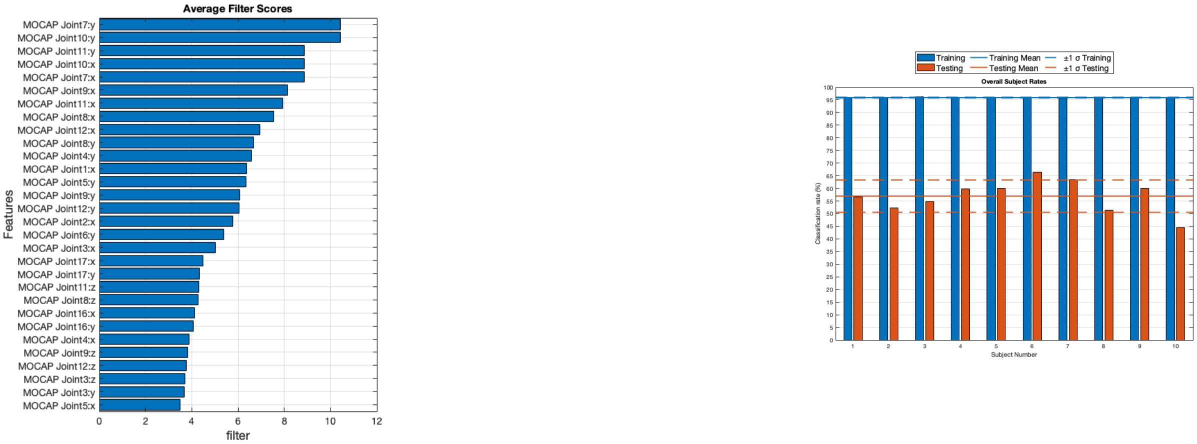

Figure 14: Decision Tree Top filters and Training and test rates by class

Problem 1 continued on next page. . .

The above images are for Decision Tree model for both classification model and wrapping model, with thresholding set on wrapping conditions. As we see, the training rates are very high(Overall training accuracy 95.89 percent), but very low test rates(Overall testing accuracy 56.94 percent), which suggests the Decision tree isn't a good method for classifications involving filtering and wrapping.
The same is reflected in the Confusion matrices and in the subject-wise rates.

References

[1] Seungkyu Lee. *Supervised Feature Selection.* 2008. URL: `http://vision.cse.psu.edu/seminars/talks/2008/feature/feature_sel.pdf` (visited on 02/13/2022).

[2] Yanxi Liu et al. "Facial Asymmetry Quantification for Expression Invariant Human Identification". In: (Aug. 2002). DOI: `10.1109/AFGR.2002.1004156`.