Haley Hauptfeld, Treksh Marwaha
Professor Yousefi
CS4342
15 December 2019

Term Project Report

1: Abstract

This report focuses on the Action Classification Problem for CS4342 final project. The data is a series of experimental sessions during which multiple subjects received a cue and performed an action (denoted by classes 2-5). Class 1 corresponds to times the subject was instructed to sit passively. For each action, there are 24 features (column 2 to 24). We implement 8 types of models with KK-10 in our efforts to get a good classifier. Furthermore, we analyse data using visualization, and then analyse the strengths and weaknesses of the algorithms we use to narrow down one which would be good for this particular dataset.

2: Methodology

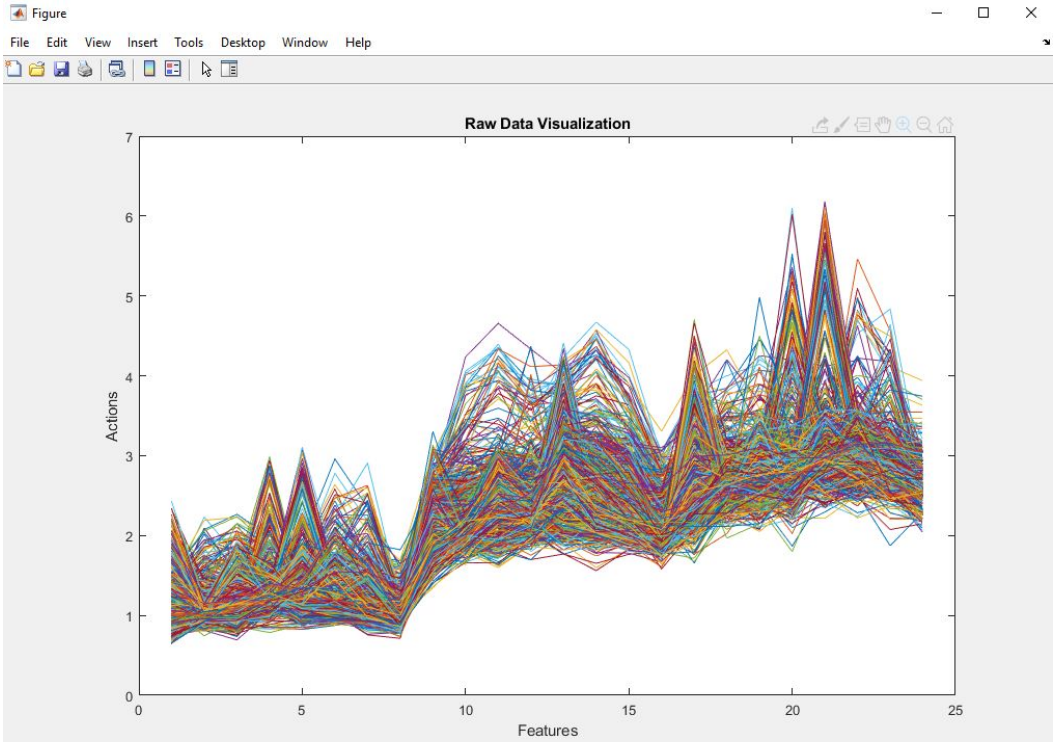
We used Matlab to produce our classifier models. Within Matlab, we visualized the data by using the 'plot' command for the training set of data, as well as the testing set of data. To produce our classifier models, we used the Classification Learner application. We put the raw training data into the learner and ran it through eight different models: Naive Bayes, K-Nearest Neighbors, Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Bagged Trees, Subspace Discriminant, and Boosted Trees. Afterwards, we exported each of these models, and made new predictions for the testing data set with the following Matlab command:

```
yfit = trainedDataModel.predictFcn(testingDataSet);
```

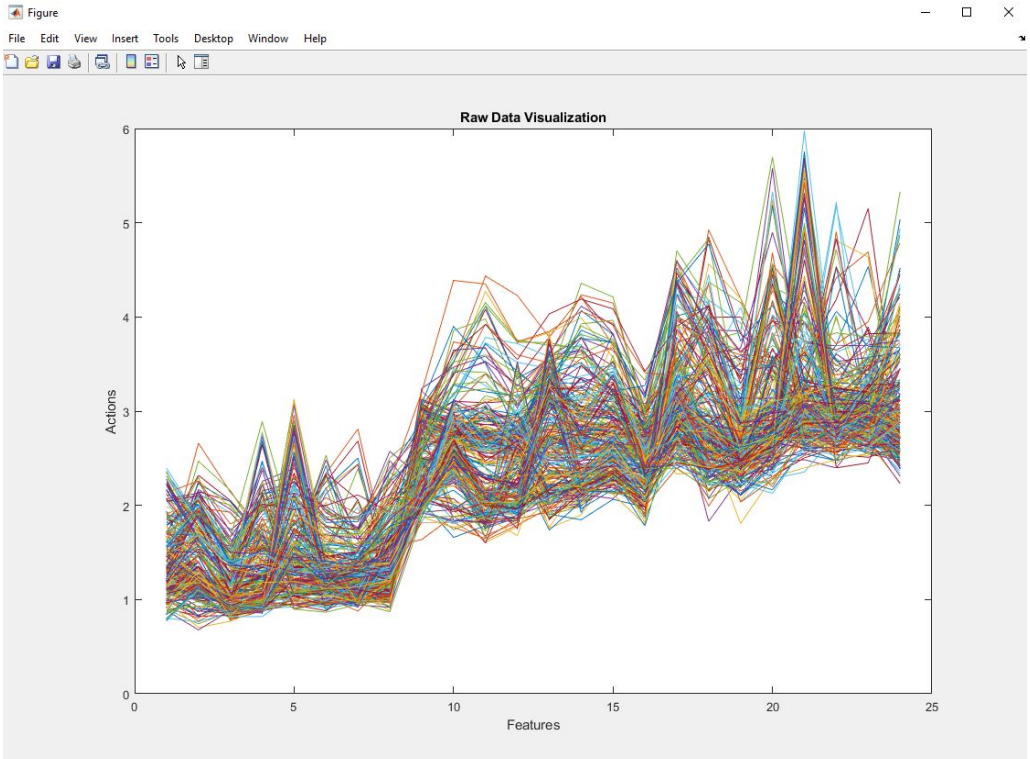
Our results are displayed below in section 4 of this paper. We included the scatter plot, the accuracy, and the confusion matrix of each model so we can determine which of these models produces the best results.

3: Data Visualization

Training Data:

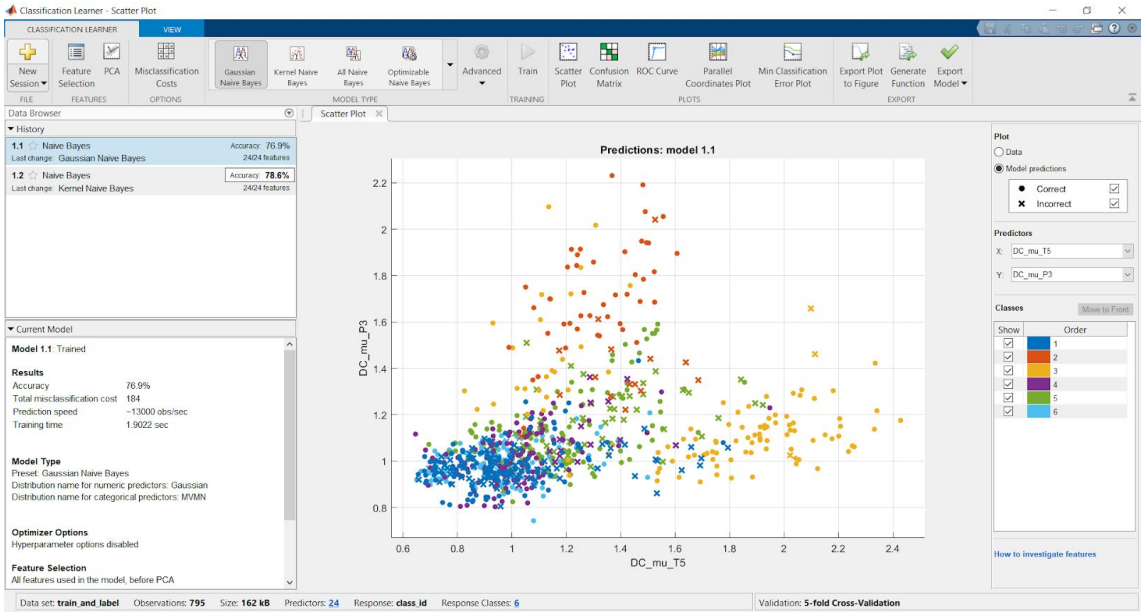


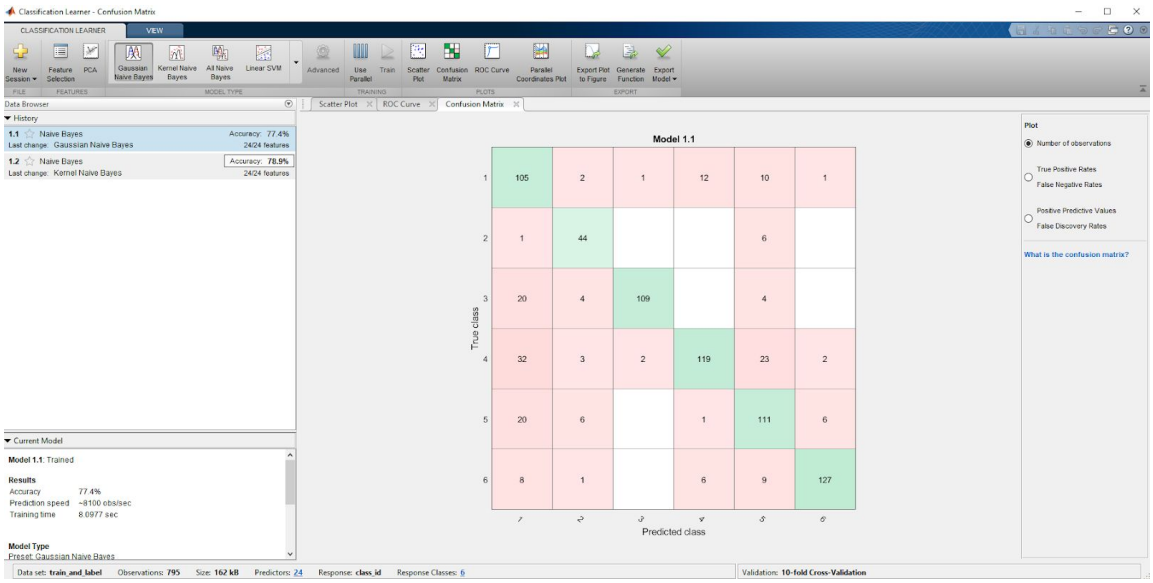
Testing Data:



4: Data Cleaning

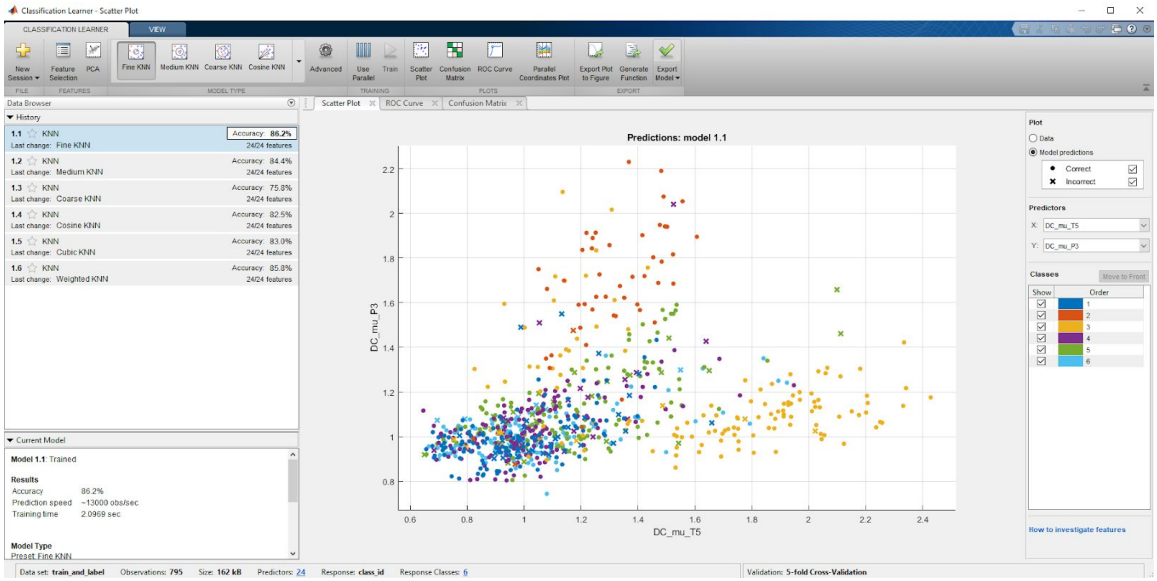
Classifier Model 1: Naive Bayes

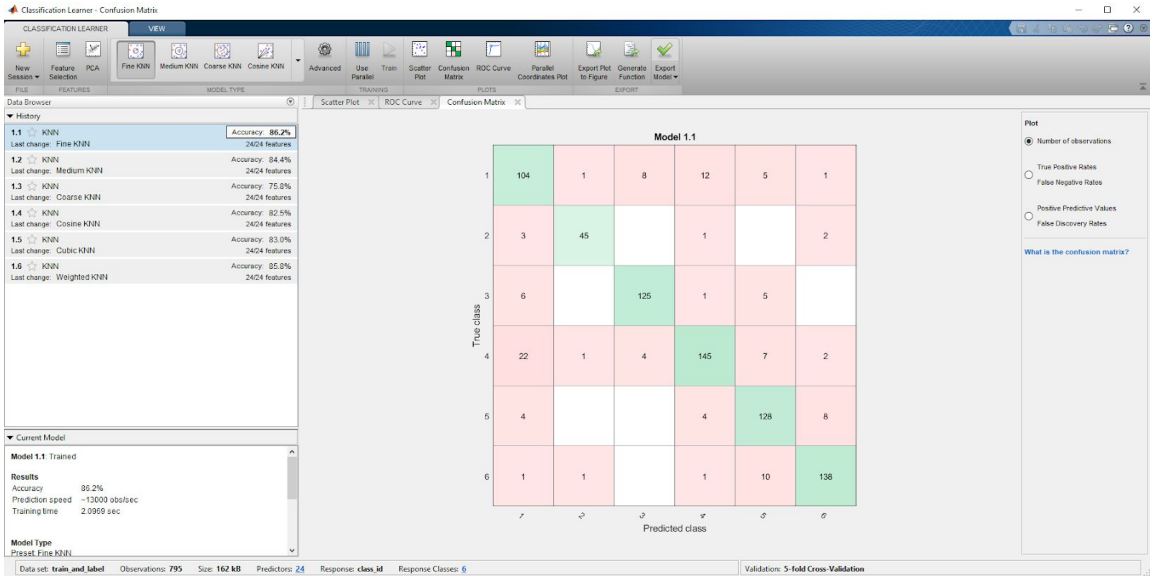




Accuracy: 78.9%

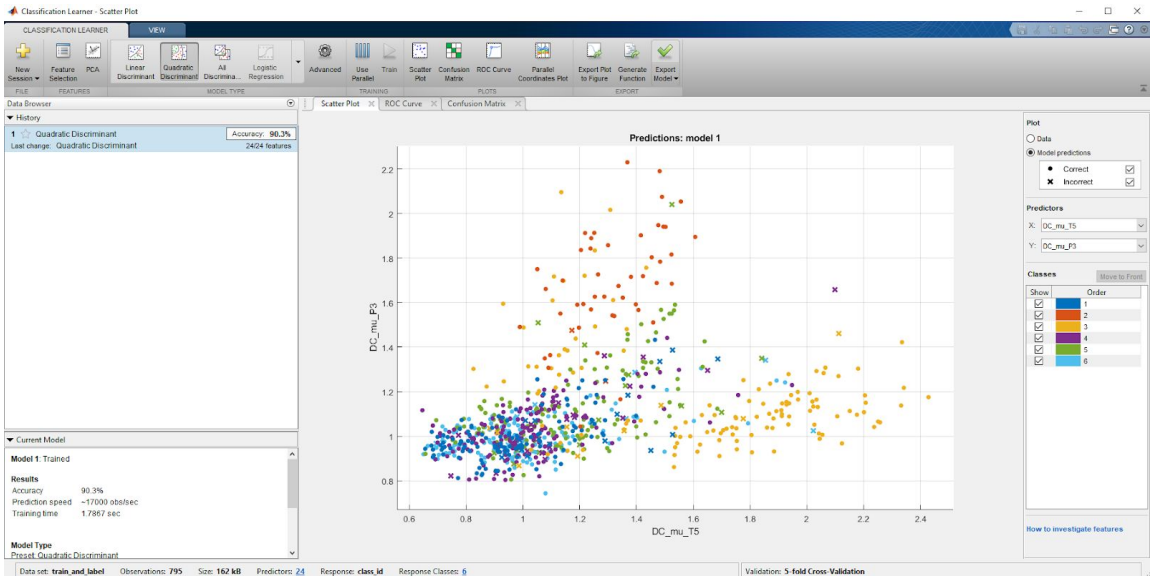
Classifier Model 2: K-Nearest Neighbors

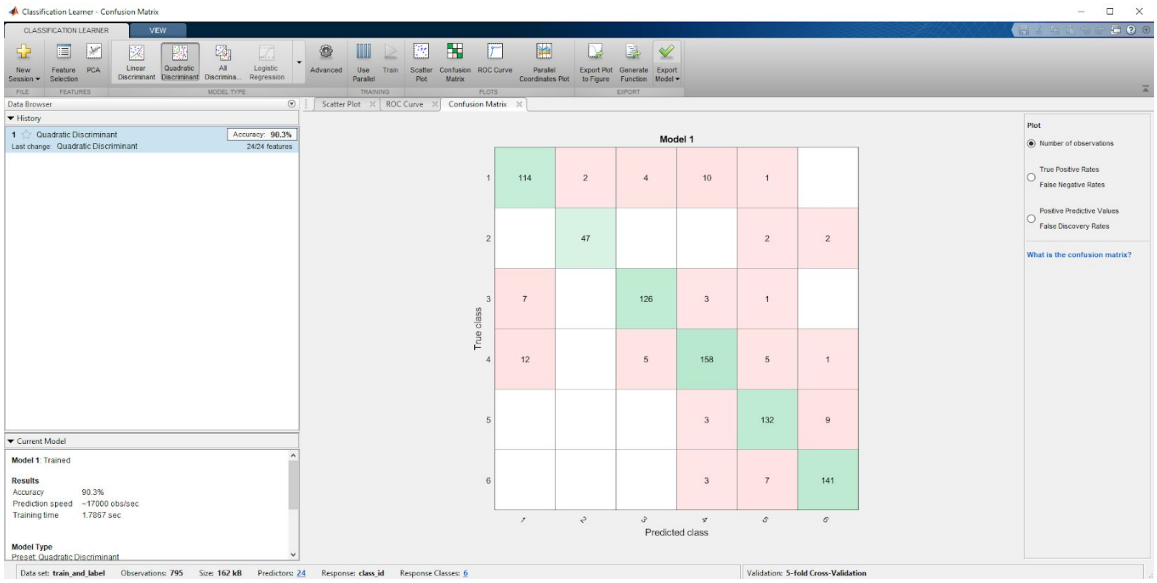




Accuracy: 86.2%

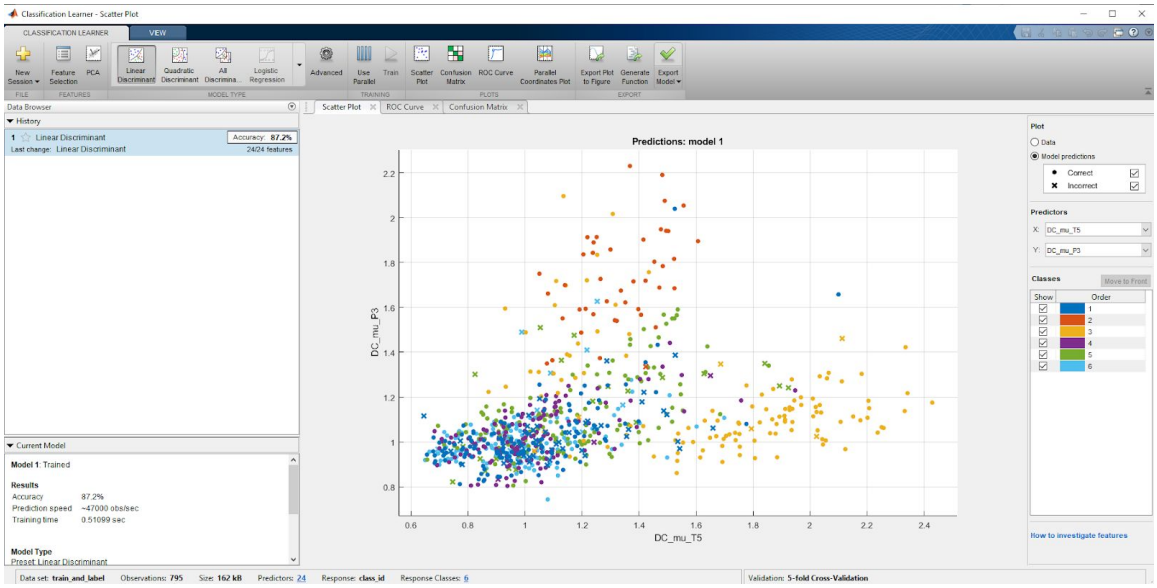
Classifier Model 3: Quadratic Discriminant Analysis (QDA)

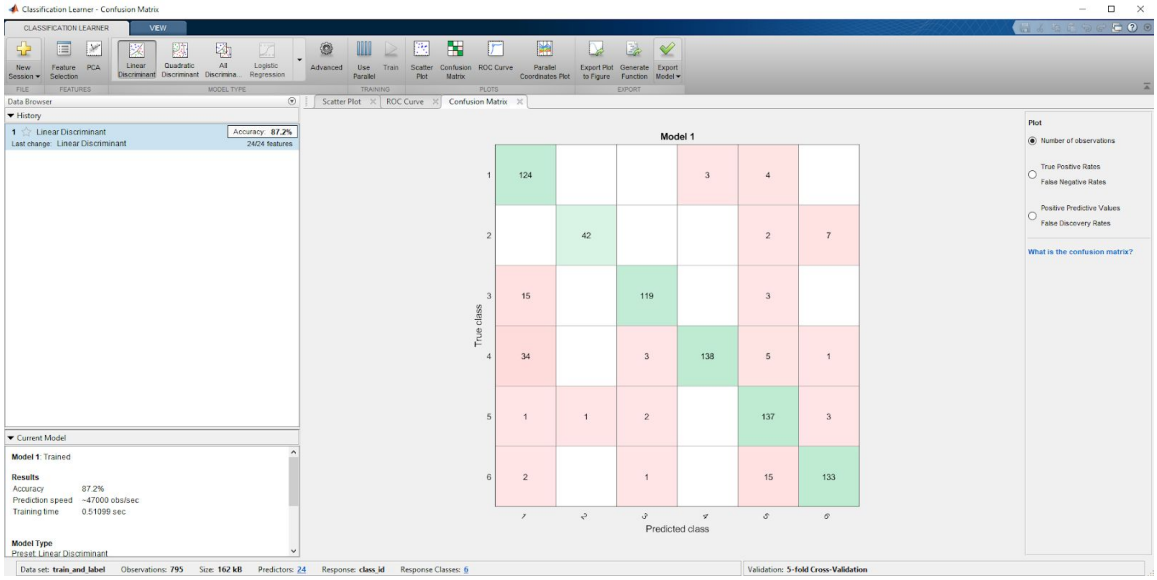




Accuracy: 90.3%

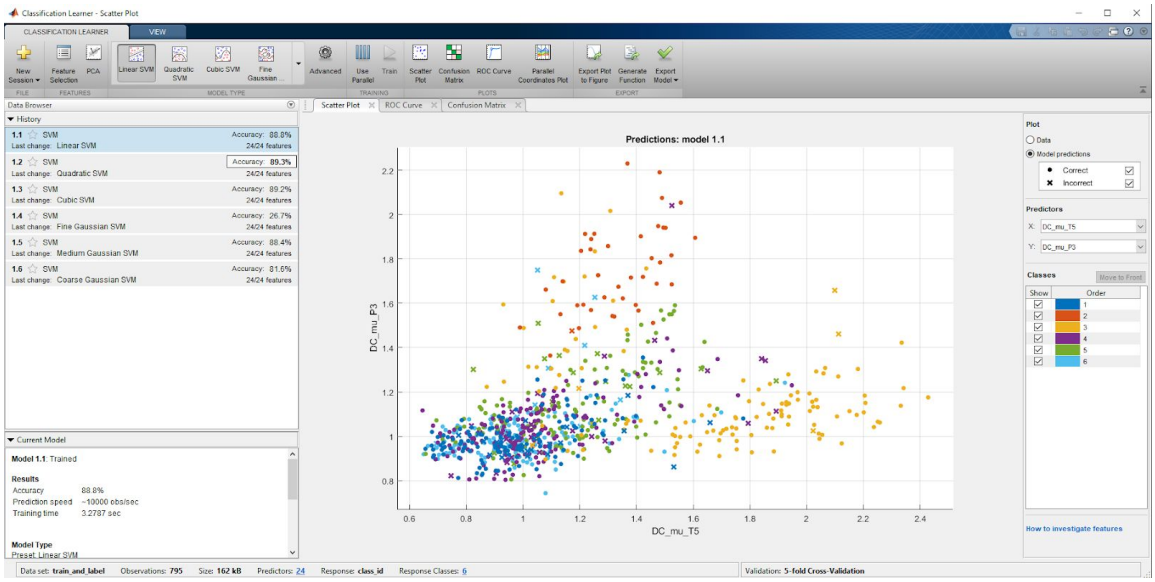
Classifier Model 4: Linear Discriminant Analysis (LDA)

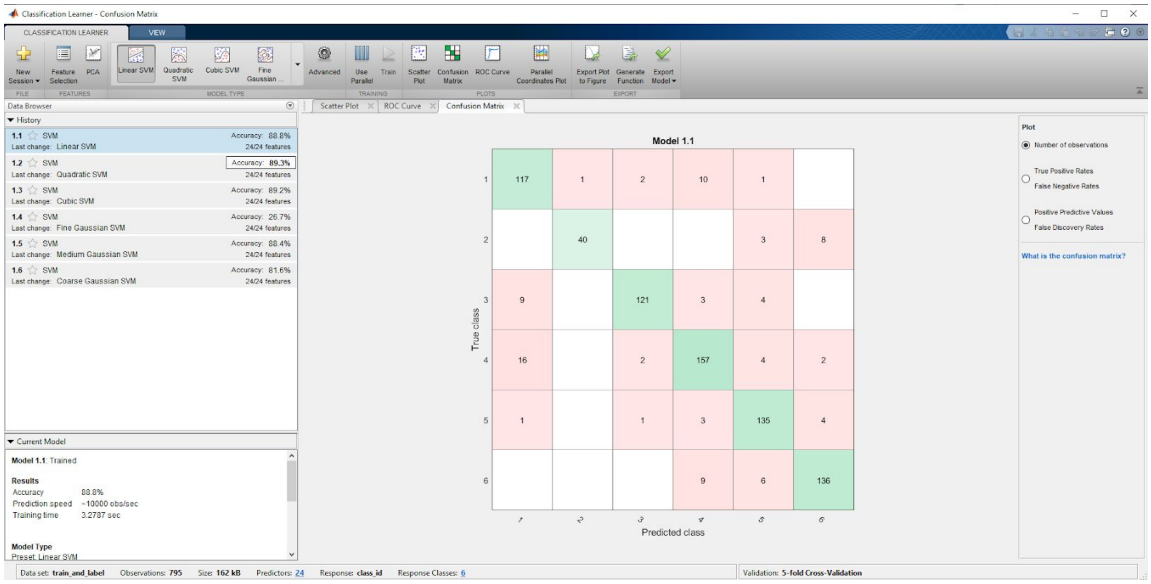




Accuracy: 87.2%

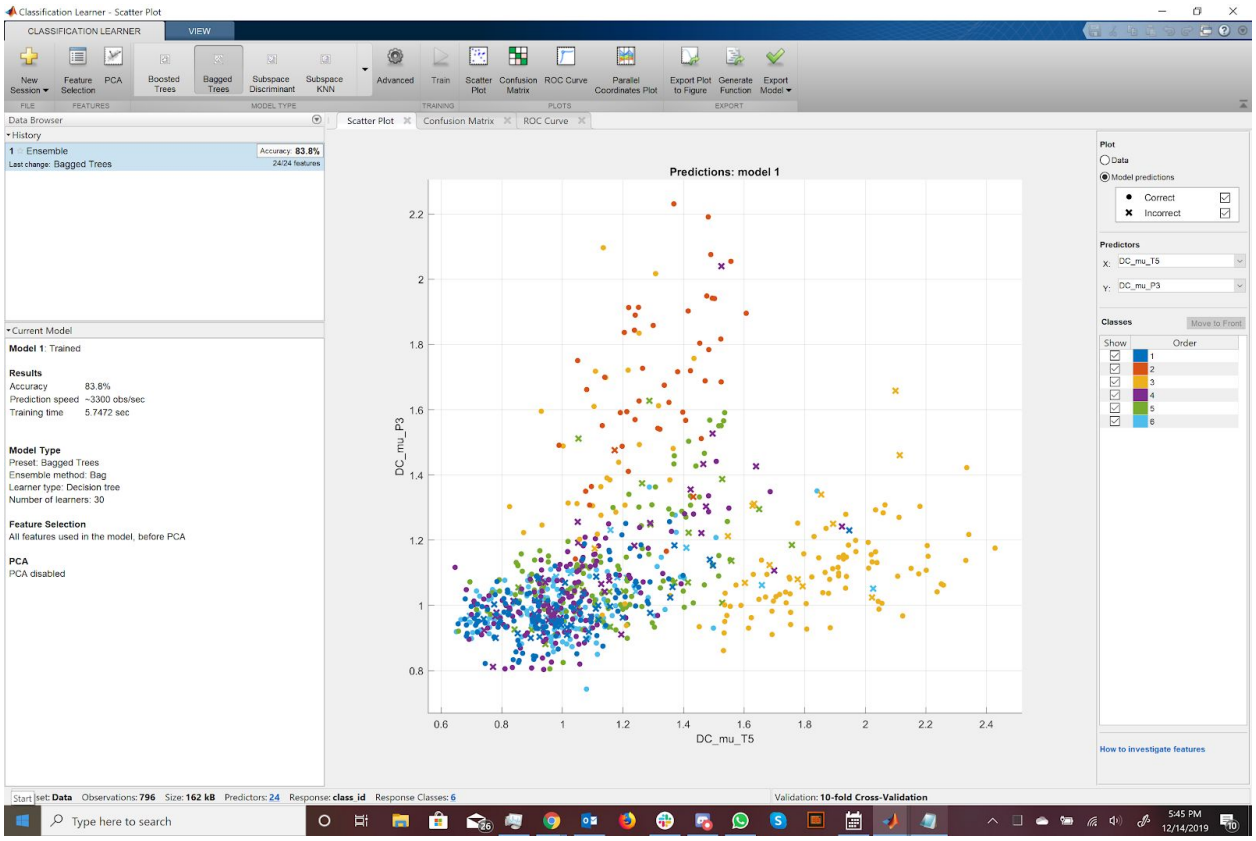
Classifier Model 5: Support Vector Machines

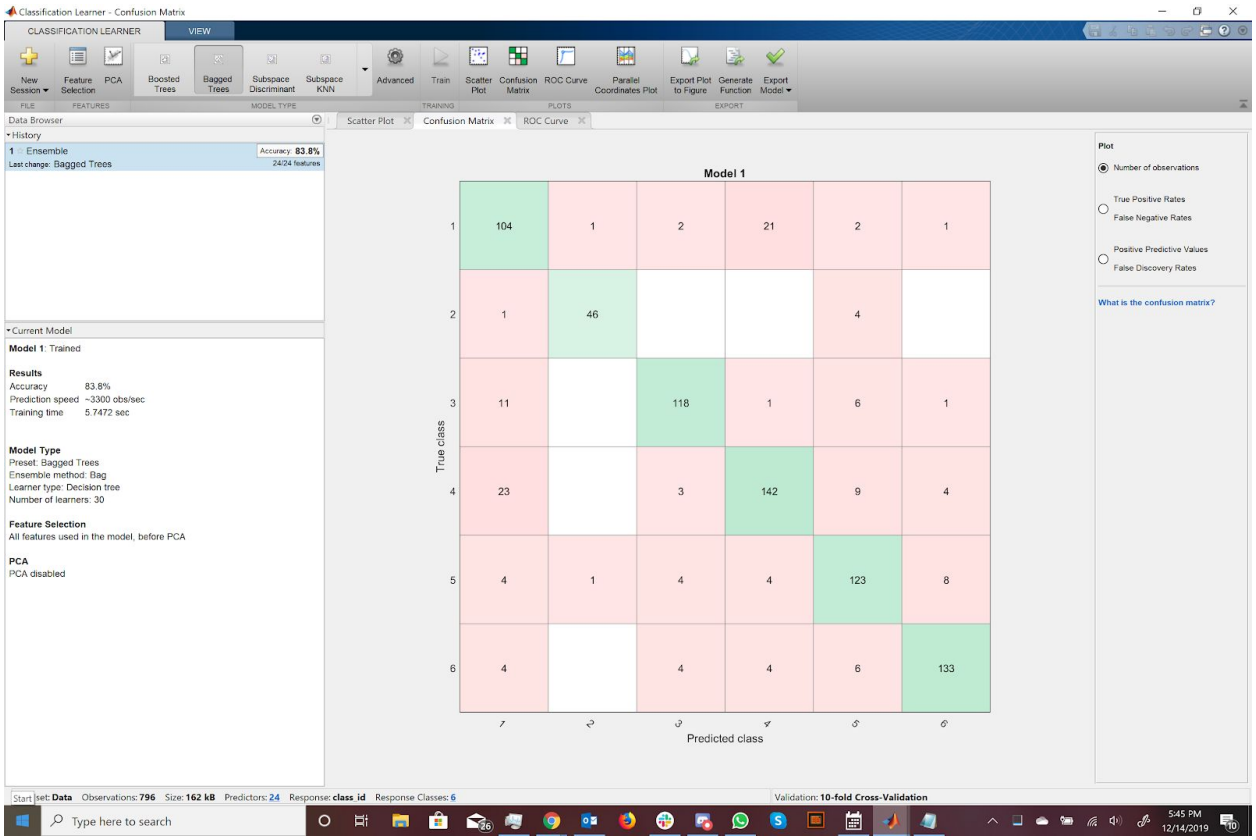




Accuracy: 89.3%

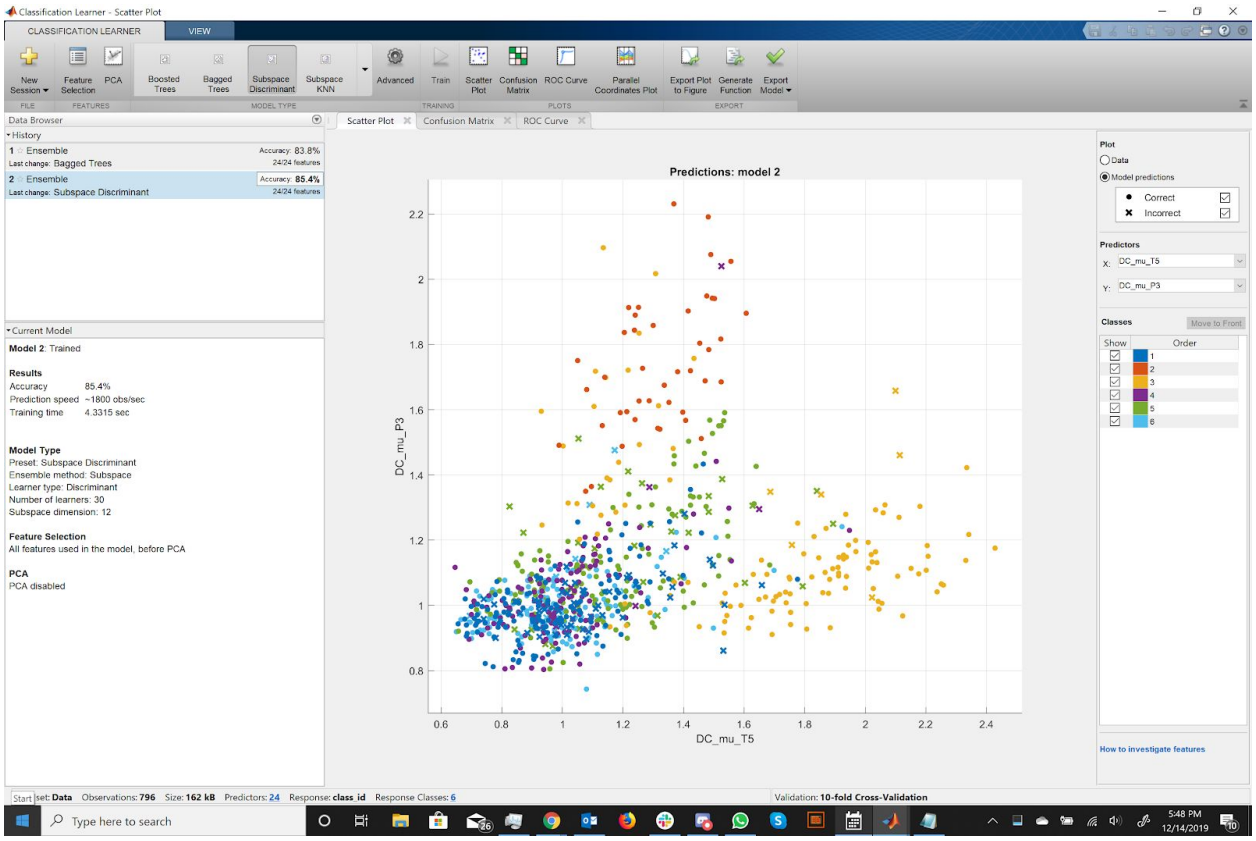
Classifier Model 6: Bagged Trees

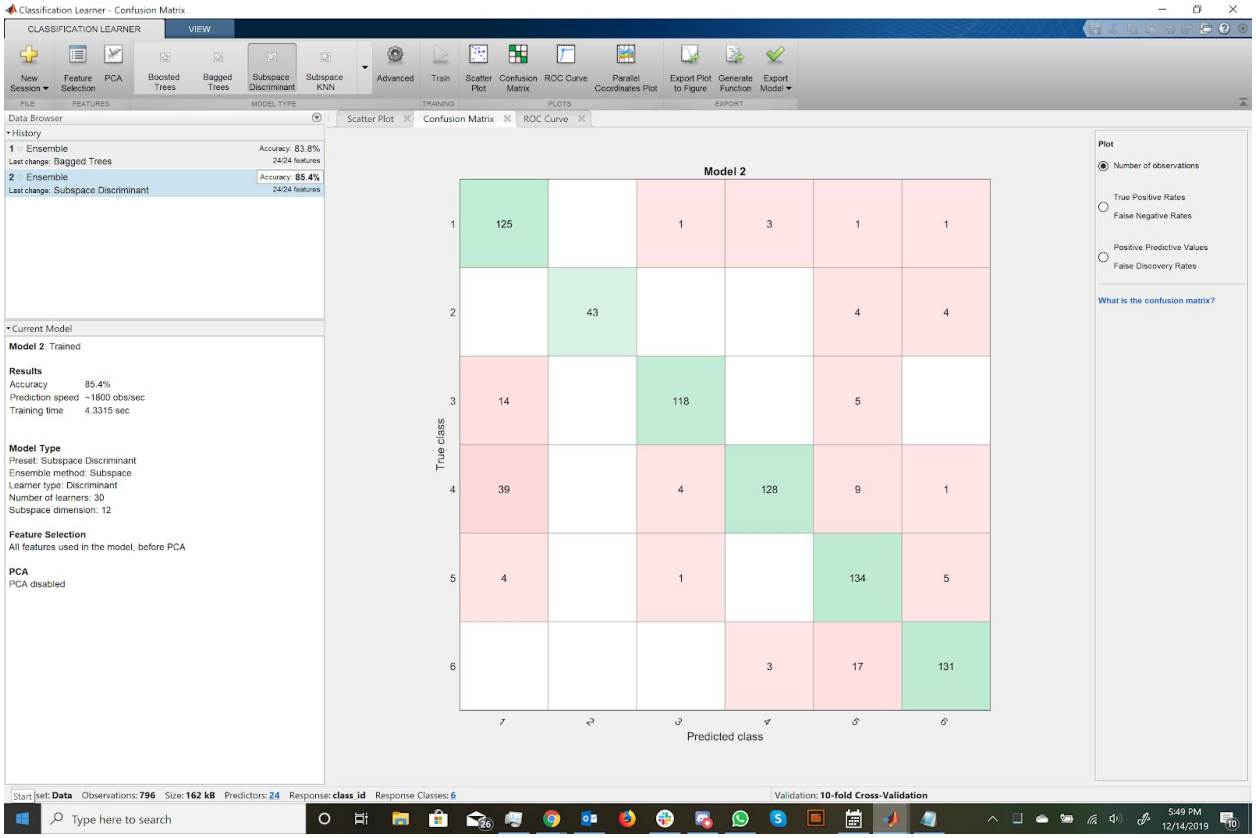




Accuracy: 83.8%

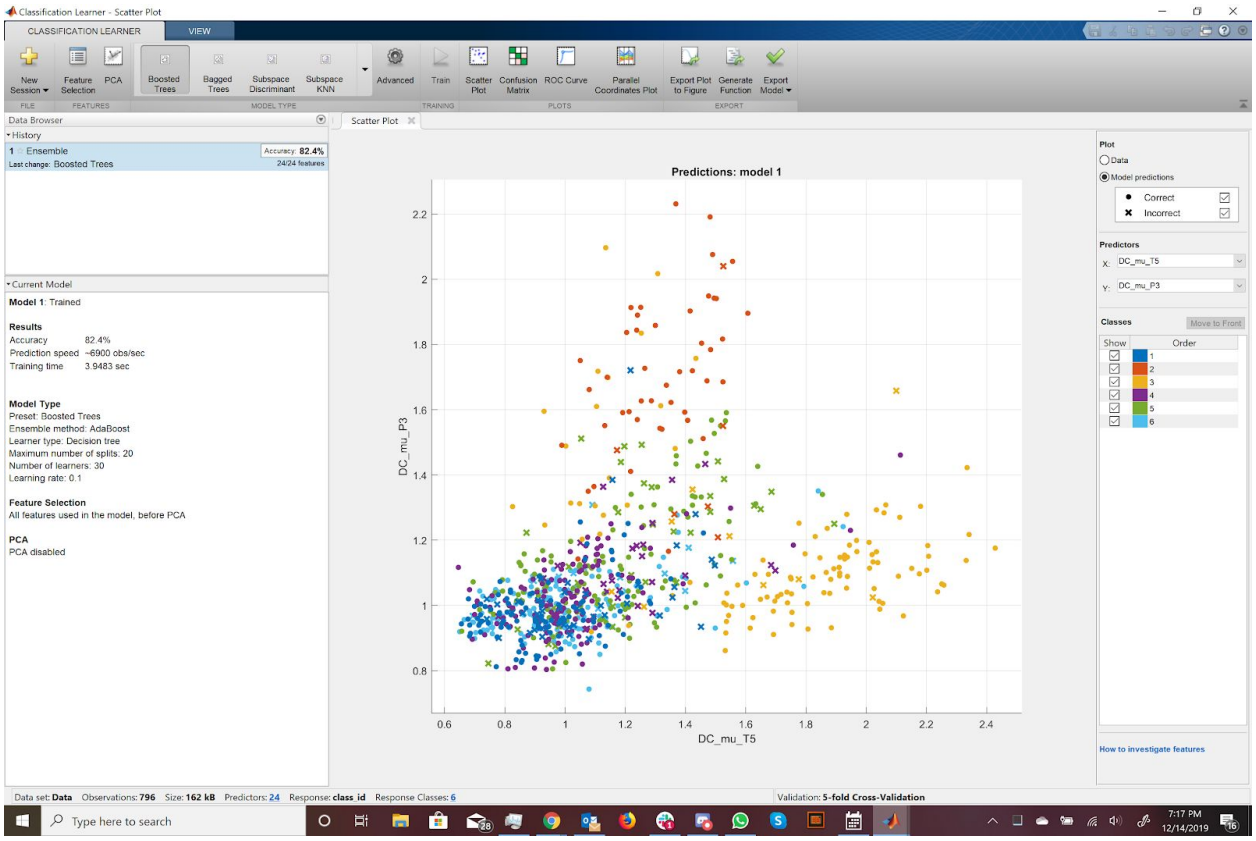
Classifier Model 7: Subspace Discriminant

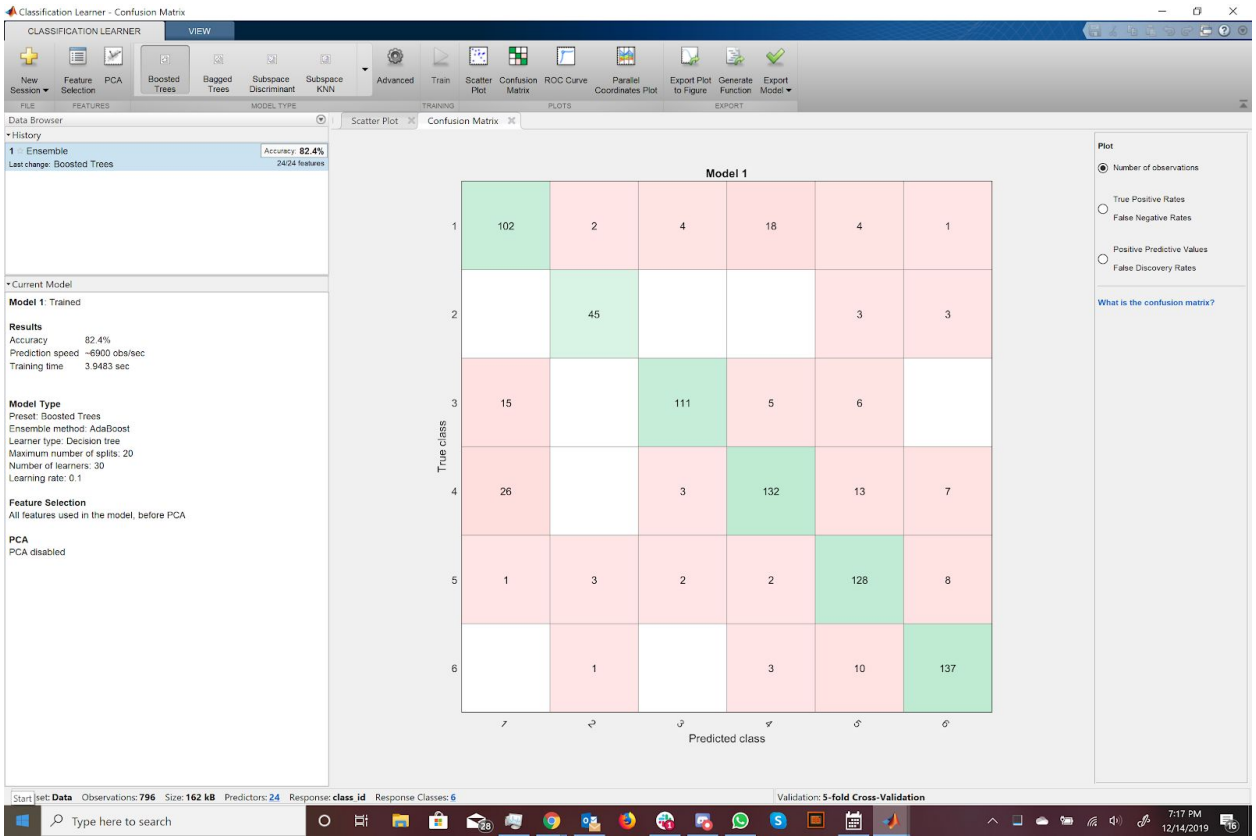




Accuracy: 85.4%

Classifier Model 8: Boosted Trees





5: Model Assessment

The first classifier model, Naive Bayes, has an accuracy of 78.9%. Naive Bayes predicts a new observation by looking up the class probabilities in a probability table that is based off of feature values. Some advantages of using the Naive Bayes model are that it scales data very well, and the algorithm is clean and straightforward. However, a large disadvantage of Naive Bayes is that it assumes conditional independence, meaning that it assumes all input features are independent of each other. This does not always occur with data in the real world.

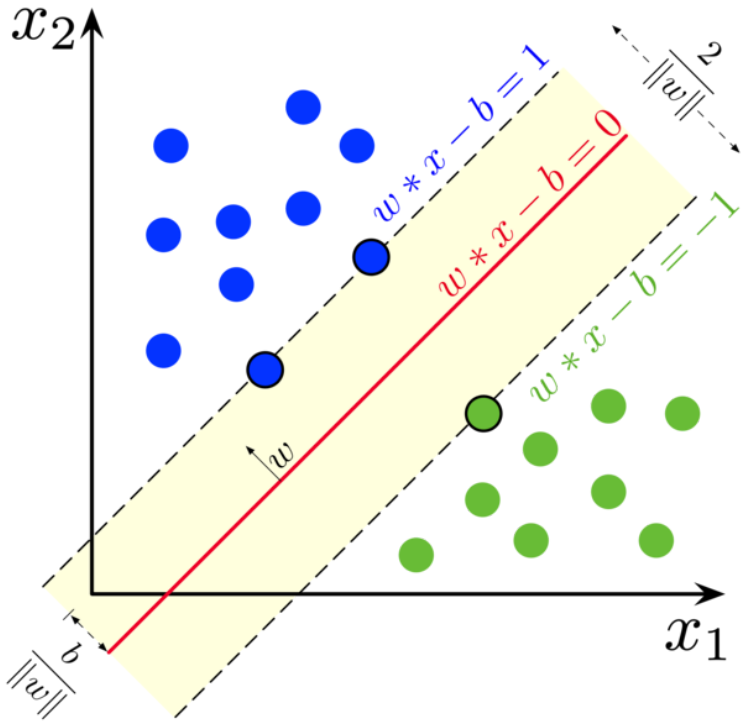
The second classifier model, K-Nearest Neighbors (KNN), has an accuracy of 86.2%. KNN looks at the K points in the training set that are nearest to the test input, counts how many members of each class are in this set, and returns that fraction as an estimate. Some advantages of using KNN is that the training is completed very quickly. However, some disadvantages of

KNN is that it requires a large storage space, it is sensitive to noise, and the actual testing is slow.

The third classifier model, Quadratic Discriminant Analysis (QDA), has an accuracy of 90.3%. QDA is an extension of the model, LDA, which is described in the next paragraph. The unique part of QDA is that each class uses it own estimate of variance or covariance if there are multiple input variables. QDA is used when individual classes exhibiting distinct covariances is prior knowledge. Therefore, it's useful for multi-class problems. However, it is not useful for dimensionality reduction.

The fourth classifier model, Linear Discriminant Analysis (LDA), has an accuracy of 87.2%. LDA makes predictions by estimating the probability that a new set of inputs belongs to each class. The class with the highest probability is the output class. An advantage of LDA is that it can perform supervised dimensionality reduction by putting the input data in a linear subspace that maximizes separation between classes. However, a disadvantage of LDA is that it only works for data with multiple classes.

The fifth classifier, Support Vector Machines, has an accuracy of 89.3%. We used a linear SVM, one which makes a simple linear separation between the classes, using the linear kernel. This is the easiest SVM to interpret. The advantages of SVM are that they scale relatively well to high dimensional data and the SVM models have generalization in practice as such the risk of overfitting is less. A few disadvantages of SVM are that it takes a long time to train with large datasets. In addition its difficult to understand and interpret the final model, variable weights and individual impact.



A Linear SVM

Our sixth classifier, Bagged Trees, are bootstrapped aggregate ensemble of fine trees, slow and memory intensive for large data sets, our model has an accuracy of 83.8%. Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithms that have high variance. A few algorithms that have a high variance are decision trees, like classification and regression trees. Decision trees are sensitive to the specific data on which they are trained. If the training data is changed the resulting decision tree can be quite different and in turn the predictions can be quite different. Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees. Generally this method has a few advantages. First, it handles higher dimensionality data very well. Secondly, it can handle missing values and maintains accuracy for missing data. But as a downside since the final prediction is based on the mean predictions from subset trees, it won't give precise values for the regression model.

Our seventh classifier, subspace discriminant, has an accuracy of 85.4%. A subspace discriminant is an ensemble of discriminant classifiers using the random subspace algorithm. Its

advantages are that its good for many predictors, relatively fast for fitting and prediction and low on memory usage. But as a downside its accuracy values varies largely based on the data.

Our eighth and final classifier, boosted trees, had an accuracy of 82.4%. Boosted trees create an ensemble of medium decision trees using the AdaBoost algorithm, compared to bagging, boosted trees take relatively little time and memory but need more ensemble members. Boosting is another ensemble technique to create a collection of predictors. In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. In other words, we fit consecutive trees and at every tick, the aim is to solve for net-error from the previous tree. When an input is misclassified by a hypothesis, its weight is increased so that the next hypothesis is more likely to classify it correctly. By combining the whole set at the end converts weak learners into better performing model. A few advantages of this model are that it supports different loss functions, in addition it works well with interactions. Its disadvantage is that its prone to overfitting and requires careful tuning of the parameters.

Our best result was achieved using the QDA model giving us the following result.

	precision	recall	f1-score
1	0.60	0.62	0.61
2	1.00	0.37	0.54
3	0.52	0.78	0.63
4	0.65	0.96	0.77
5	0.31	0.35	0.33
6	1.00	0.48	0.65
avg / total	0.68	0.59	0.59

6: Sources

https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623
https://www.mathworks.com/company/newsletters/articles/using-machine-learning-to-predict-epileptic-seizures-from-eeg-data.html
https://www.researchgate.net/publication/258285203_A_survey_on_Data_Mining_approaches_for_Healthcare/figures
https://elitedatascience.com/machine-learning-algorithms
https://scikit-learn.org/stable/modules/lda_qda.html
https://doc.lagout.org/science/Artificial%20Intelligence/Machine%20learning/Machine%20Learning_%20A%20Probabilistic%20Perspective%20%5BMurphy%202012-08-24%5D.pdf