# CW1 Report on Supervised Machine Learning

## 1. The content of the Dataset

The dataset contains images of different letters organized in both training and testing. The training data is a 124800x784 matrix, containing overall 124800 images of handwritten letters, and each of them is represented by 28x28 pixel image reshaped to a 1x784 vector. There are 26 features classes, each of them represents a letter, which are created by the algorithms using the test and trained data and organize them into separate classes. The number of observations in the algorithms are 124800. In order to study the data from the file, the dataset of images is saved into variable and the dataset for labels is saved into variables, which are two kinds one for the trained data and one for test data.

## 2. The model training and evaluation methods used

For the model training three different classification algorithms are used, in order to train and compare them implemented in Matlab. The first classifier used in this coursework is the K-Nearest neighbor (KNN), which supports multiclasses, the prediction speed is slow for cubic and medium for others and the memory usage is medium, also the interpretability is hard. This algorithm uses the function fitcknn(), which train a numeric matrix that contain the trained images and array of vectors that contain the labels, also it predicts the number of nearest neighbors of the data. From the documentations this algorithm should perform the best.

Another classifier used in this exercise is naïve Bayes model, which is relatively faster than the previous model and uses less memory, but the performance, based on the documentation should be medium. As function it uses fitcnb() which takes as parameters the trained images in double format the trained labels and default parameters for “Distribution Names” and “mn”.

Decision Three model is used for the last classifier, using the algorithm fitctree(). This is the fastest model, which takes the less memory and the interpretability is easy, but the performance is worse than the other two models.

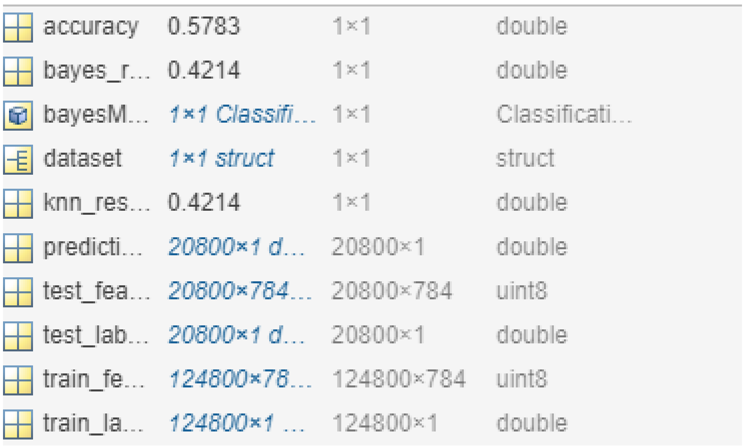
For the evaluation of the model accuracy and resub error variables are used in order to compare the models. Predict function is used to get the label of any new set of observation and tested with the same training data using this function. The accuracy is to get the number of correctly predicted labels as fraction of the total number of observations and the resub error is used in the misclassification error, that is the proportion of misclassified observations on the training set. The function resubLoss() is used to compare the performance. Also for each model confusion matrix shows the correctly classified observations using the training set confusionchart().

## 3. The selected model and criteria

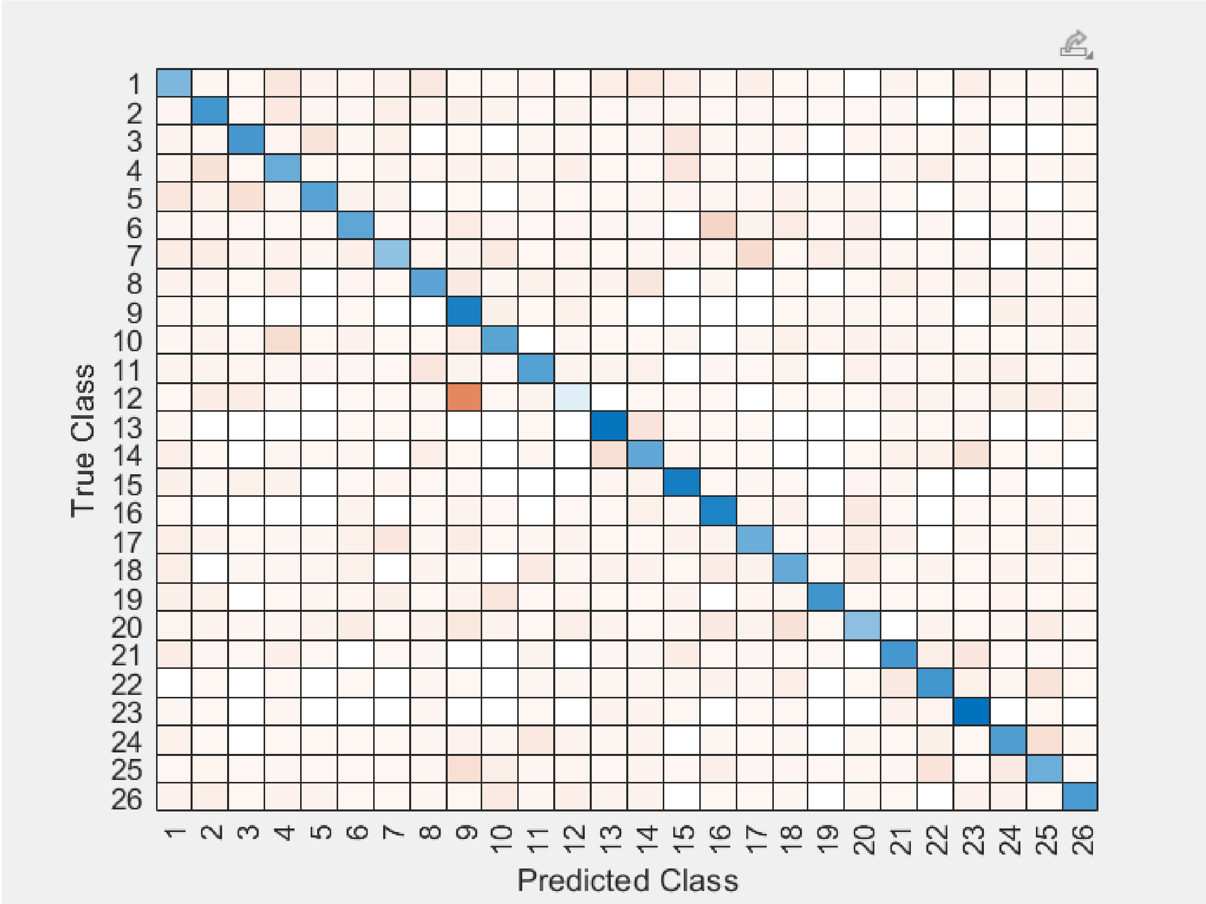
In my view the model, that performs the best is the K-Nearest neighbor (KNN), which gives accuracy of 0.8582 and misclassification error of 0. The drawback is the speed of implementation which in my case was 15 minutes.

accuracy 0.8582

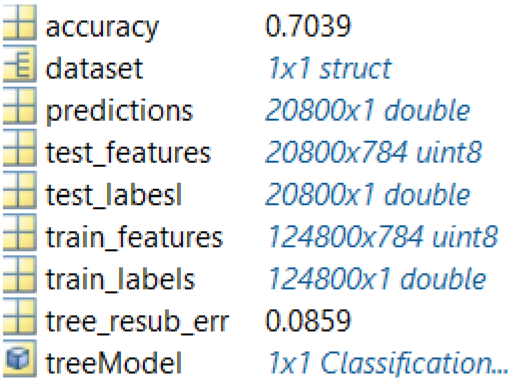
The accuracy performance of the naïve Bayes model is 0.5783 and the errors are 0.4214. The speed of this model is relatively fast around 4-5 minutes.



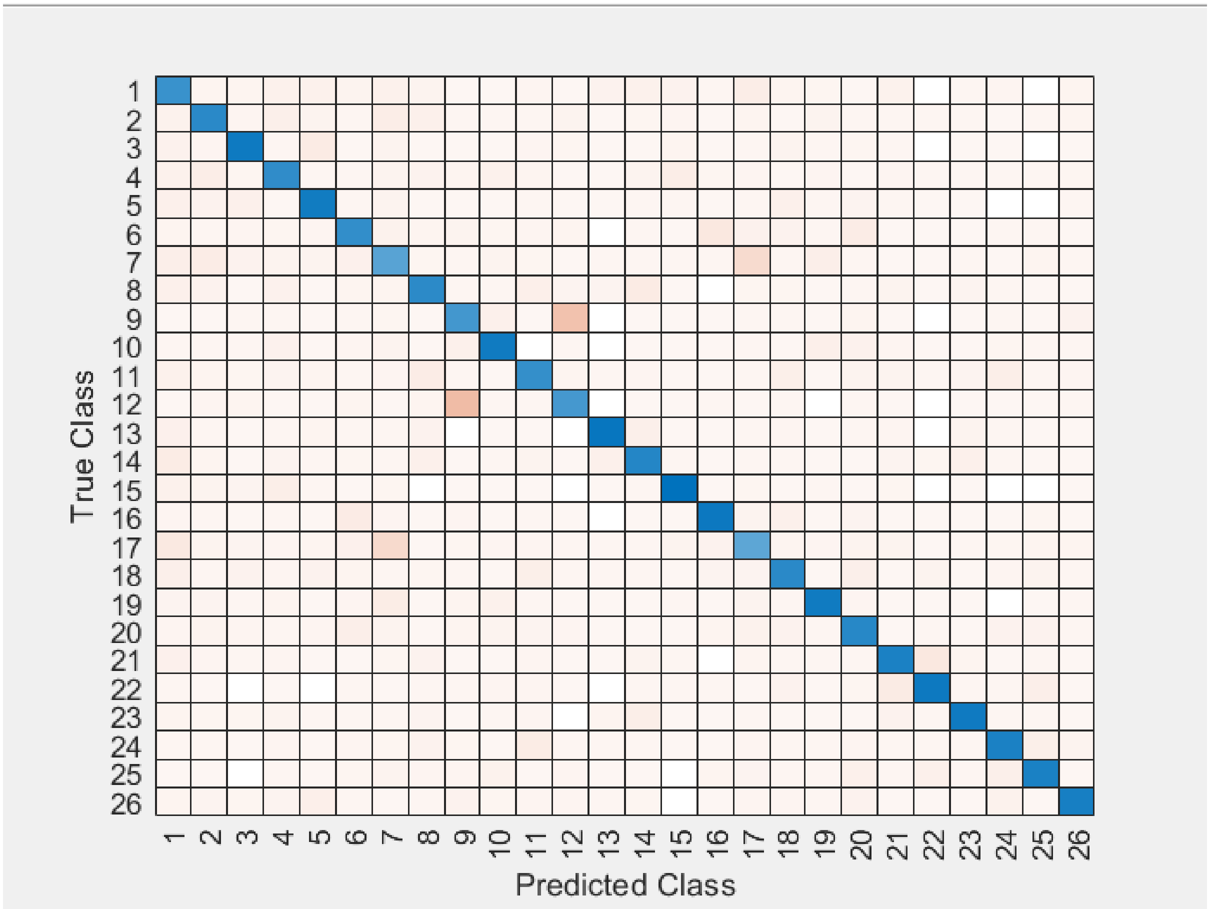
| | | | |
|---|---|---|---|
| accuracy | 0.5783 | 1×1 | double |
| bayes_r… | 0.4214 | 1×1 | double |
| bayesM… | *1×1 Classifi…* | 1×1 | Classificati… |
| dataset | *1×1 struct* | 1×1 | struct |
| knn_res… | 0.4214 | 1×1 | double |
| predicti… | *20800×1 d…* | 20800×1 | double |
| test_fea… | *20800×784…* | 20800×784 | uint8 |
| test_lab… | *20800×1 d…* | 20800×1 | double |
| train_fe… | *124800×78…* | 124800×784 | uint8 |
| train_la… | *124800×1 …* | 124800×1 | double |



The speed of the last model is very fast around 1-minute giving accuracy of 0.7039 and errors of 0.0859.



| | |
|---|---|
| accuracy | 0.7039 |
| dataset | 1x1 struct |
| predictions | 20800x1 double |
| test_features | 20800x784 uint8 |
| test_labesl | 20800x1 double |
| train_features | 124800x784 uint8 |
| train_labels | 124800x1 double |
| tree_resub_err | 0.0859 |
| treeModel | 1x1 Classification... |


True Class
Predicted Class