1 Task 15

Using the results produced on Task 9 we implement a Naive Bayes Classifier. This classifier is based on the assumption of independence between different features. In our example, where different features are actually different pixel colors the features themselves should be considered dependent. Therefore we expect the results of this classifier to be worse than those of a classifier that considers features to be dependent, such as sklearn's Gaussian Proccess Classifier.

In comparison to the already implemented sklearn naive bayes classifier, our implementetation lacks the functionality of refiting new data incrementally. Specifically the corresponding sklearn classifier uses a function called partial fit, in order to refit the model faster, in case additional data are provided.

Another interesting feature of the sklearn implementetation, which we decided to include as well, is the class attribute ϵ. This attribute allows for the homogenization of the variances between different classes and features. It has been empirically shown that this smoothing process improves classifier perforamnce.

In our specific case where the dataset contains images, a number of pixels of a digit are unaltered between different samples. In other words, some of the dataset's features have zero variance, leading to possible warnings of division by zero, during the prediction process. For this reason it is essential to introduce a smoothing parameter,in order to tackle this problem.

We now present the accuracy score of our classifier for different values of ϵ. Our models have been trained and tested on the same dataset.
After the optimal value for our hyperparameter is selected, we evaluate our model on the test set. Firstly, we use the same value as the sklearn classifier, in order to compare our results. The accuracy score for sklearn's implementation is 0.7587. Subsequentlly we increase the factor as shown below:

- $\epsilon = 10^{-9}$ Custom Naive Bayes score = 0.7587
- $\epsilon = 10^{-7}$ Custom Naive Bayes score = 0.7767
- $\epsilon = 10^{-5}$ Custom Naive Bayes score = 0.7907
- $\epsilon = 10^{-3}$ Custom Naive Bayes score = 0.8132
- $\epsilon = 10^{-1}$ Custom Naive Bayes score = 0.8512

To begin with, we observe that our classifier produces identical results with sklearn's classifier, when using the same ϵ. Adiitionally, it is apparent that performance increases with ϵ. Therefore for our model we select $\epsilon = 0.1$ In general increasing ϵ might lead to overhomogenization, which in turn could hide substantial information from our model. In the next task we will observe in further detail the effects of overhomogenization on our model's performance. Speciffically, what happens to a model's accuracy when the variances of different features are falsely considered identical.

Finally we evaluate our model's generalized score on the test set, for the selected hyperparameter. Custom Naive Bayes generalized score = 0.8156

2 Task 16

On the following task we repeat the process of classifying the test data and calculating the accuracy score. The difference here lies on the fact that we consider the variances of different features to be unitary. As said before, this is equivalent to completely homogenizing feauture variances. Therefore we expect the model fitted here, to have poorer perforamnce than the one fitted on task 15, as computed variances are indeed different.

In this case of course, the hyperameter ϵ is redundant, as the variances are already identical. Therefore we just have to compute the accuracy score on the test set. Indeed we calculate the generalized score for the Naive Bayes classifier with unit var to be 0.8127 As we expected this model's score is lower than the previous ones' since the hypothesis of unitary variances is false.

Interestingly enough, in the special case that class prior probabilities are equal, this model is equivalent to the EuclideanDistanceClassifier.