Report
INF8245E (Fall 2021) : Machine Learning - Assignment #2
Amine EL AMERI - Matricule: 2164634

1 Linear Classification and Nearest Neighbor Classification

1- The dataset generated

```
array([[-1.18474138, -1.25977689, -1.68983557, ..., -0.69545796,
        -4.07552171, -1.        ],
       [-1.09434916, -1.03537902, -1.60875295, ..., -0.81211933,
        -2.25372176,  1.        ],
       [ 3.65984402,  1.96751785,  2.64469733, ...,  4.12154688,
         0.38132273, -1.        ],
       ...,
       [ 6.02915369,  6.07205315,  6.62761002, ...,  3.23147505,
         1.17855292, -1.        ],
       [ 1.13816372,  2.33211841,  1.21921264, ...,  1.49713828,
         2.82223798,  1.        ],
       [-1.66702404, -0.27993517, -1.38812707, ...,  0.25300242,
         0.07598672,  1.        ]])
```

DS1_train

```
array([[ 5.98984839,  4.76813199,  2.88593891, ...,  5.19998197,
         5.0860626 , -1.        ],
       [-4.28356112, -4.51576241, -7.15470186, ..., -3.82864207,
        -4.38581016,  1.        ],
       [-2.54773639, -1.05379814, -1.93188084, ..., -2.98969601,
        -3.26757111, -1.        ],
       ...,
       [-0.19998895,  1.40402531, -0.09539388, ...,  0.85164452,
         2.99462177, -1.        ],
       [ 0.66778423, -0.05088119, -0.32423634, ..., -1.34344541,
         1.11190834,  1.        ],
       [ 0.53657233,  5.01523639,  0.31698755, ...,  5.04172497,
         0.75359688,  1.        ]])
```

DS1_valid

```
array([[-1.79319912, -0.29260561,  0.09119591, ..., -0.73849547,
         0.44440939, -1.        ],
       [ 3.81344793,  3.48378856,  1.98957064, ...,  3.44278662,
         2.78416397, -1.        ],
       [ 6.94530346,  7.33914055,  5.68141271, ...,  6.96933248,
         5.78252231, -1.        ],
       ...,
       [ 2.73875604,  2.08698449,  2.19047547, ...,  1.13408044,
         3.75425017, -1.        ],
       [ 3.0814123 ,  3.29340666,  4.62522582, ...,  3.01136463,
         2.67734137,  1.        ],
       [ 4.33252433,  4.80155692,  3.79469316, ...,  0.73122148,
         6.41135556, -1.        ]])
```

DS1_test

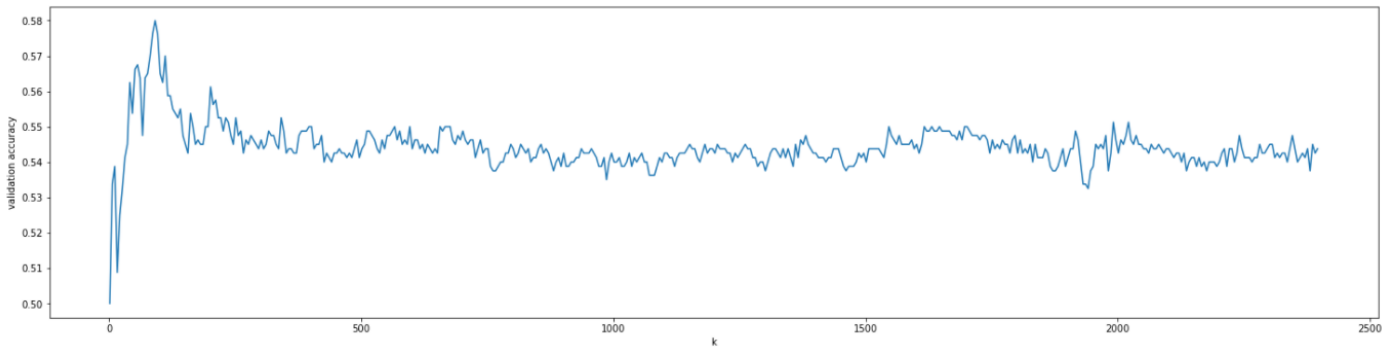
2- a) The best fit accuracy (number of correctly classified instances / number of instances) my classifier achieved on the testing set is 0.87 (it varies of course when the data is regenerated).

b) The parameters learned by the GDA model using the maximum likelihood approach:

W = [1.00318231, -0.63243116, -0.30788267, -0.07091545, -0.6686976, -0.21366324, 1.00220101, -1.70526693, -1.85761444, 0.68129428, -0.9252942, -0.6908877, 0.79004166, 0.83846184, -0.53899563, 0.90781958, 1.81052028, -0.36112187, 0.10760423, -0.25101916]

W_0 = 1.7870661809861543

3- a) Plotting the validation accuracy of KNN for every value of k from 1 to 2500, we can see that the accuracy decreases when k increases, and the **best accuracy is achieved for k=51**. That's because k impacts the flexibility of the model, the greater k, the less flexible the model becomes, and in our case, k=51 gives the best flexibility compromise (in particular for this dataset it seems that not a lot of datapoints from different classes are close, that's why we don't need a big value of k).



But in general we can see that all the accuracies achieved with all the k values are less than the accuracy achieved by the GDA model. GDA is a lot better than KNN for this dataset.

b) The best fit accuracy achieved by KNN (k=51) on the testing set is 0.5625

4- The dataset generated

```
array([[-0.13723776,  1.14566834, -2.59681829, ..., -0.53109083,
        -2.63397847, -1.        ],
       [ 1.35996124,  0.65391606,  1.28062151, ...,  1.7834053 ,
        -0.06798311, -1.        ],
       [-3.10769651,  0.05350656, -3.03487239, ..., -1.34588222,
        -2.50358485,  1.        ],
       ...,
       [ 1.29515548,  1.38524482,  3.66518758, ...,  3.81775292,
         2.08592284, -1.        ],
       [ 2.48154701,  0.88172262,  3.17160777, ...,  3.03984112,
         2.45318498, -1.        ],
       [ 1.86051024,  0.46225624,  1.27925999, ...,  2.76911413,
         0.56669328, -1.        ]])
```

DS2_train

```
array([[ 2.03709848, -0.06121663,  0.30889375, ..., -0.84941019,
         0.90399334, -1.        ],
       [ 2.09777903,  1.33837939,  2.86846669, ...,  2.45890452,
         3.9450927 , -1.        ],
       [ 0.74453978,  2.18760103, -0.50396128, ...,  2.66255753,
         0.1363702 ,  1.        ],
       ...,
       [ 2.51690014,  3.02842591,  0.21759233, ...,  3.89684687,
         2.6137049 , -1.        ],
       [ 1.79276592,  1.7610349 ,  1.02873235, ..., -0.78794208,
         0.39749907, -1.        ],
       [ 5.5535489 ,  1.75148102,  2.50453376, ...,  2.09442206,
         3.39385649,  1.        ]])
```

DS2_valid

```
array([[-0.83964566,  1.43541254,  0.85243957, ...,  2.15118375,
        -0.02453872,  1.        ],
       [ 1.15463881,  1.41042037,  1.7672248 , ...,  1.71551145,
         1.39668391, -1.        ],
       [ 5.31058189,  3.97520276,  4.32566247, ...,  3.68684566,
         3.49625691, -1.        ],
       ...,
       [ 1.18698605,  1.98766511,  0.58303353, ...,  3.02768271,
         2.88876061, -1.        ],
       [ 3.93635488,  2.8812073 ,  2.11920465, ...,  4.81926972,
         1.97494471, -1.        ],
       [ 7.31143438,  3.89488135,  0.92733213, ...,  2.24242187,
         1.39979699,  1.        ]])
```

DS2_test

5- 1)

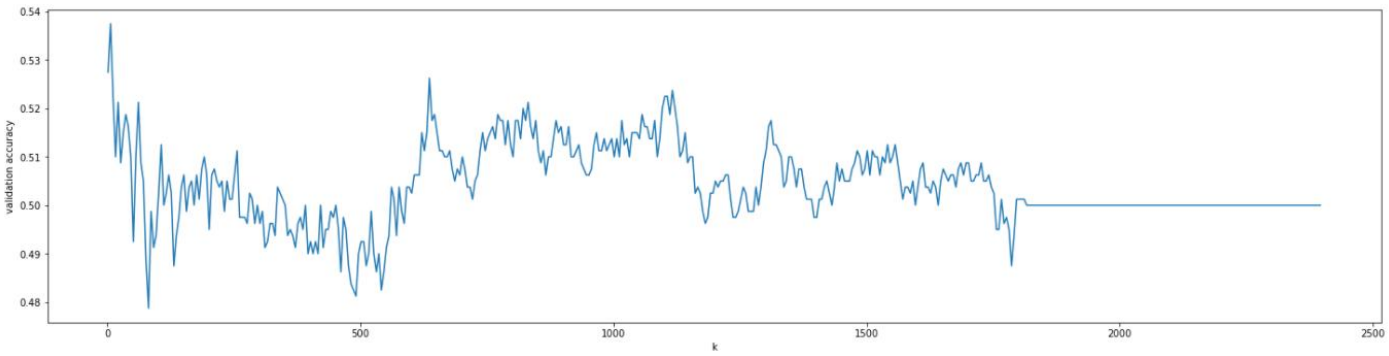
a) The best fit accuracy (number of correctly classified instances / number of instances) my classifier achieved on the testing set is 0.51625 (it varies of course when the data is regenerated).

b) The parameters learned by the GDA model using the maximum likelihood approach:

W = [0.22655101, 0.01416807, 0.04487186, 0.01264898, -0.35058712, 0.00850368, 0.30804466, -0.20957603 -0.25992249, -0.11718761, -0.3028806, -0.16935253, -0.26262091, -0.01955056, 0.38374957, -0.03190767, 0.05291551, -0.01642507, 0.25561316, -0.09770199]

W_0 = 0.6058740228202765

2) Plotting the validation accuracy of KNN for every value of k from 1 to 2500, we can see that the accuracy decreases when k increases until 500, then starts increasing to achieve its maximum at k=1125 (**best accuracy is achieved for k=1125).** That's because in this new dataset DS2 contrary to DS1, there are a lot of datapoints from different classes that are close, and so a small k which only takes into account few number of neighbors is not sufficient to achieve a good performance.



For this dataset the performances of GDA and KNN are almost equivalent.

3) The best fit accuracy achieved by KNN (k=1125) on the testing set is 0.5312

6- We saw that the performance of GDA was very good for DS1 achieving 0.8 accuracy while it was not very good on DS2, achieving only 0.5 accuracy, still this was better than KNN who achieved no more than 0.5 accuracy on both datasets. For DS1 GDA is better than KNN while for DS2 GDA and KNN are almost equivalent.

2 MNIST Handwritten Digits Classification

1)

a) In the Gaussian Naïve Bayes (GNB) model, features are conditionally independent given the class label, and the likelihoods are Gaussian:

$$p(x_i|C_k) = \frac{1}{\sqrt{2\pi}\sigma_{ik}} \exp\left(\frac{-(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}\right)$$

with

$$\mu_{ik} = \frac{\sum_{n=1}^{N} \mathbf{1}[t^{(n)} = k].\, x_i^{(n)}}{\sum_{n=1}^{N} \mathbf{1}[t^{(n)} = k]}$$

and

$$\sigma_{ik}^2 = \frac{\sum_{n=1}^{N} \mathbf{1}[t^{(n)} = k].\, (x_i^{(n)} - \mu_{ik})^2}{\sum_{n=1}^{N} \mathbf{1}[t^{(n)} = k]}$$

b) After splitting the data to train, valid and test, flattening and normalizing it, implementing the GNB algorithm with the equations above, and adding smoothing so that the standard deviation is not equal to 0. I've found an accuracy of 0.8139 on the validation set and an accuracy of 0.8111 on the testing set.

2)

a)

K=5	K=50	K=500	K=1300
Validation accuracy = 0.9712	Validation accuracy = 0.9518	Validation accuracy = 0.9047	Validation accuracy = 0.8579

We can see that best value of k for KNN is a small value **k=5,** and that's because in the images of the MNIST dataset we care more about the few closest pixels to our pixel of interest, because a lot of the pixels that are further away don't help at all in the classification (black background pixels).

b) The best fit accuracy achieved by the KNN classifier (with k = 5) on the testing set is : 0.9664

3) KNN performs better than GNB because its accuracy (0.9664) is greater than GNB's one (0.8111) and that's because GNB's assumptions : conditionally independent features and gaussian likelihood, do not apply very well in the MNIST dataset.