COMP 551 - A2 - Camilo Garcia #260657037

1. Generalized Discriminant Analysis

We generate 2 classes (c_1 positive, c_2 negative), each with 20 features and 2000 samples. Both classes follow a Gaussian distribution with different means for each feature (μ_{1-20}) but the same covariance matrix Σ. We form testing and training datasets by taking a 70/30 ratio of each class and adding it to **DS1_train.csv** and **DS1_test.csv** respectively. These files can be found under the `./Output` directory.

2. Maximum Likelihood Approach

The likelihood of a datapoint $x^{(n)}$, $t^{(n)}$ is given the following equation, from which we aim to maximize the parameters μ_1, μ_2 and Σ:

$$[\pi \mathcal{N}(x^{(n)} \mid \mu_1, \Sigma)]^{t^{(n)}} [(1-\pi)\mathcal{N}(x^{(n)} \mid \mu_2, \Sigma)]^{1-t^{(n)}}$$

We obtain the following metrics by classifying the samples in **DS1_test** with the parameters maximized with the samples in **DS1_train**:

```
Confusion Matrix: {'TP': 560, 'FP': 35, 'FN': 40, 'TN': 565}
Accuracy:         0.9375
Precision:        0.9412
Recall:           0.9333
F_1:              0.9372
```

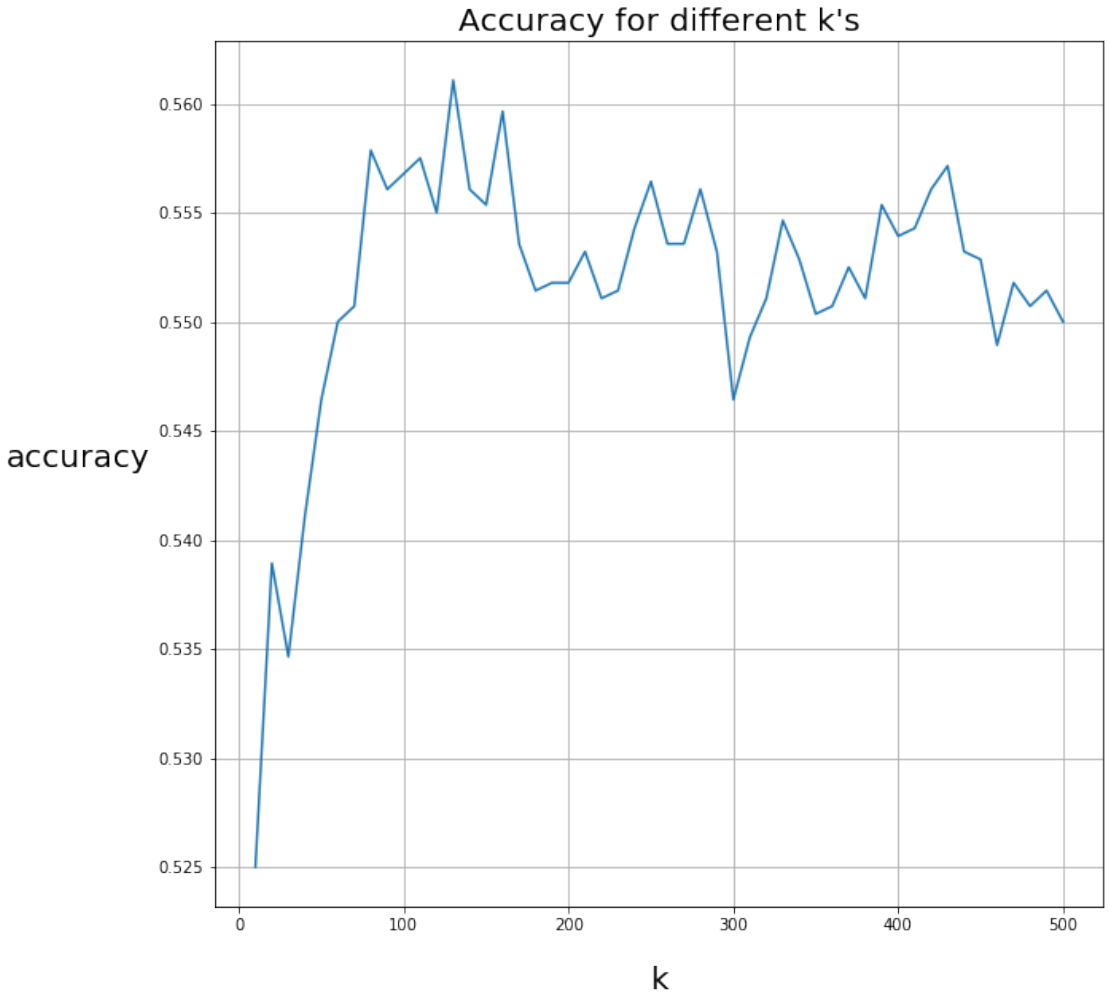
The learned coefficients are:

```
w_0: -26.5150
W:
[-14.0095   8.1975   5.1954   2.7242   9.3934   4.0421 -15.9463  23.5596
  28.1587  -9.0513  12.6924  11.4783 -14.8929 -12.5811   5.6651 -12.7657
 -28.0339   6.8136   0.4087   4.9846]
```

The high accuracy, precision and recall achieved by this Probabilistic LDA (or Gaussian Distribution Analysis GDA) is to be expected, as its strongest assumption is that the distributions are Gaussian and they share the same covariance matrix in addition to the **i.i.d** assumption (Independent and Identically Distributed samples).

3. kNN

Note that because the distance between each point depends on the unit of measurement, normalization of the features is strongly recommended for kNN.
kNN's inductive bias is that a sample for a given class will be closest (Euclidean distance) to only other samples of the same class. Unfortunately, the Gaussian distribution of both classes overlaps considerably. Hence, we can't expect kNN with any value of k to perform as well as the previous GDA.



As observed in the graph, the model can't reach an accuracy better than around 0.57. We will use k = 200 to predict the testing samples:

```
k = 200
Confusion Matrix: {'TP': 300, 'FP': 221, 'FN': 300, 'TN': 379}
Accuracy:         0.5658
Precision:        0.5758
Recall:           0.5000
F_1:              0.5352
```

Notice Precision is relatively better than recall. Indicating that a little more than half of the samples classified as positive were indeed positive. Recall is coincidentally exactly .5 which means that of all the positive samples exactly half of them were correctly classified (i.e. 1000) .

4. Multivariate Gaussian

We generate 3 Gaussian distributions for 2 classes (c_1 positive, c_2 negative), each with 20 features and 2000 samples. Each pair of Gaussian distributions has different means for each feature ($\mu_{i,1-20}$)

but the same covariance matrix Σ. We form a single dataset of 2000 samples of each class by taking sampling from the 3 Gaussians wih probabilities [0.1, 0.42, 0.48]. We form testing and training datasets by taking 70/30 ratio of each class and adding it to **DS2_train.csv** and **DS2_test.csv** respectively. These files can be found under the `./Output` directory.

Maximum Likelihood Approach

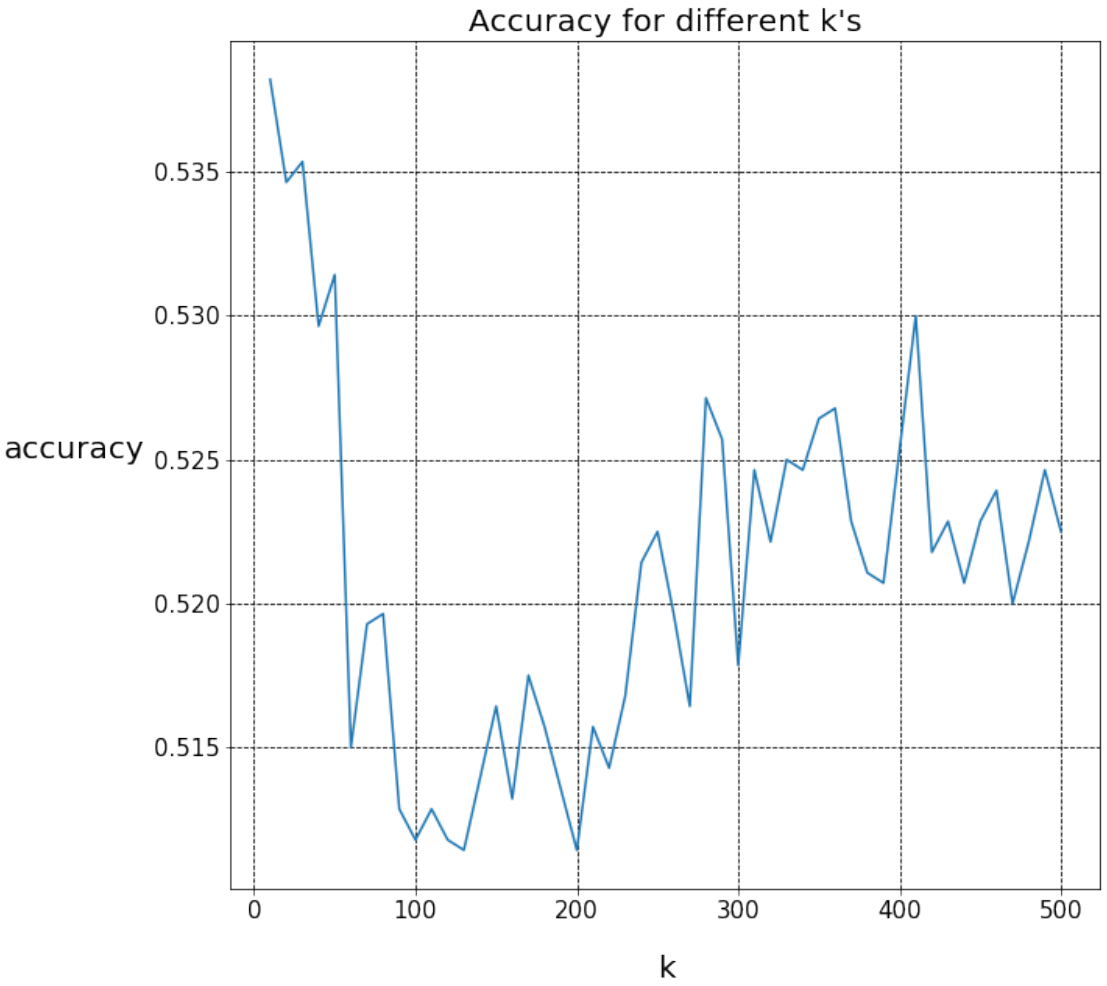
```
Confusion Matrix: {'TP': 303, 'FP': 324, 'FN': 297, 'TN': 276}
Accuracy:         0.4825
Precision:        0.4833
Recall:           0.5050
F_1:              0.4939
```

The learnt coefficients are:

```
w_0: 0.0629
W:
[ 0.0184  0.0173 -0.0316 -0.0452 -0.0372 -0.0287  0.0303 -0.02   -0.0098
 -0.0278 -0.063   0.0275  0.06    0.0291 -0.0149  0.0095 -0.0122  0.0275
  0.0614 -0.047 ]
```

Note that all the learned parameters have a much smaller weight (in the magnitude of 1e-2) as opposed to those learned when the 2 classes came from a single Gaussian distribution (in the magnitude of 1e+1).

kNN



As observed in the graph, the model can't reach an accuracy better than around 0.57. We will use k = 70 to predict the testing samples:

```
k = 70
Confusion Matrix: {'TP': 376, 'FP': 282, 'FN': 224, 'TN': 318}
Accuracy:         0.5783
Precision:        0.5714
Recall:           0.6267
F_1:              0.5978
```

Notice that the change in accuracy is negligible over a the span of 50 k's between [1,500]. Hence the model's parameter is relatively insensitive to the clustering of both classes.

6. Multivariate Gaussian

By relaxing the i.i.d assumption, hence allowing non independent identically distributed samples (Mixing 3 different Gaussian Distributions with different covariance matrices in different ratios),

it is to be expected that the Gaussian Discriminant Analysis will perform badly. Because the resulting function is quadratic with respect to x, a Quadratic Discriminant would perform better. The drastic change in the pondering given to the parameters can also be explained by the fact that the model itself cannot represent well the underlying mix of distributions. No parameter can efficiently predict the target class and as such no parameter is given a large magnitude. This is the opposite of the results over DS1 where each parameter was very representative of the target's class.

While kNN performed badly because of its uninformative inductive bias over DS1, it's metrics showed negligible change over DS2. This can be explained by the fact that kNN does not rely on shared covariance but rather on Euclidean distance. So mixing the data's distributions from a set of 3 different Gaussian's doesn't have an direct impact on its inductive bias.