COMP 551 - Assignment 2

John Wu - 260612056

February 11, 2018

Acknowledgement
This assignment was completed by myself, John Wu. However, some answers and methodologies were discussed with Frank Ye, Tiffany Wang, and Nabil Chowdhury.

1 Linear Classification and Nearest Neighbours Classification

1.1 Dataset 1 Generation

The generated datasets can be found under the generated_data folder with the files names DS1_train.csv and DS1_test.csv corresponding to the train and test classes respectively. The next two parts will be referring to tests run on these datasets.

1.2 LDA Model Using Maximum Likelihood Approach

Running the probabilistic LDA Model on the generated data set results in the following weights:

$w0 : 29.259797254602912$

$w1 : [15.394931289213986, -9.185646705696398, -5.928089971809898, -3.555251944730623, -10.344634497147682, -4.871682722537692, 18.2891256442551, -25.555524497556632, -31.21481606951889, 9.897539968625427, -14.158893375852987, -13.094090326307377, 16.553401531941205, 13.964014770335815, -5.956016064445741, 13.748578337520332, 31.633298731145825, -7.240627737286969, -0.832813528318292, -5.227616321886086]$

Using the weights above to predict the output on our test set, we mange to achieve the following metrics:

$FMeasure:$	0.9468267581475129
$Accuracy:$	0.948376353039134
$Precision:$	0.9419795221843004
$Recall:$	0.9517241379310345

1.3 k-NN Classifier Model Approach

Running the k-NN Classifier and trying different K values from 1 to 24, we achieve the following results:

K Value	Accuracy
1	0.4995836802664446
2	0.511240632805995
3	0.5187343880099917
4	0.5129059117402165
5	0.537052456286428
6	0.5328892589508742
7	0.5437135720233139
8	0.5611990008326395
9	0.5545378850957535
10	0.5503746877601998
11	0.5545378850957535
12	0.5487094088259784
13	0.5412156536219817
14	0.5453788509575354
15	0.5362198168193172
16	0.5587010824313072
17	0.5487094088259784
18	0.5503746877601998
19	0.5412156536219817
20	0.5495420482930891
21	0.5537052456286428
22	0.5587010824313072
23	0.556203164029975
24	0.5570358034970858

Based on the results, we see that we receive the best accuracy when $k = 8$. Calculating the metrics for our optimal value, we achieve the following metrics:

$$\begin{aligned} FMeasure: &\quad 0.4797630799605133 \\ Accuracy: &\quad 0.5611990008326395 \\ Precision: &\quad 0.5612009237875288 \\ Recall: &\quad 0.4189655172413793 \end{aligned}$$

As shown in the results, we see that the LDA model was significantly more accurate than the k-NN method. The reason for this is because our data is clustered together along one line with a single linear decision boundary, which means a linear model like LDA would be expected to perform better. However, if our data was spread out in different clusters throughout our domain space, then a k-NN classifier would probably perform better. In addition, most of the k values perform with about the same accuracy. This is once again due to the linear structure of our data.

1.4 Dataset 2 Generation

The generated datasets can be found under the generated_data folder with the files names DS2_train.csv and DS2_test.csv corresponding to the train and test classes respectively. The next part will be referring to tests run on these datasets.

1.5 LDA and k-NN Model With New Datasets

LDA Model Using Maximum Likelihood Approach

Running the probabilistic LDA Model on the generated data set results in the following weights:

$$w0 : -0.08382431667082715$$

$$\begin{aligned} w1 : [&-0.036121070473366684, 0.016843081807145055, 0.006689797742290029, \\ &-0.037607865056006745, 0.05312292616958406, 0.061027499793217886, \\ &-0.09028457452526403, 0.11140215822624577, -0.029867257675947535, \\ &-0.03358809136577283, 0.05297192396415256, 0.06418430227548107, \\ &0.0439929246741877, -0.026038907549131053, -0.028747123378560294, \\ &-0.010108810575986421, 0.01577340396191533, -0.03080936471329309, \\ &-0.04549991069995683, -0.003958605805103335] \end{aligned}$$

Using the weights above to predict the output on our test set, we mange to achieve the following metrics:

$FMeasure:$	0.5185185185185185
$Accuracy:$	0.49294605809128633
$Precision:$	0.4845360824742268
$Recall:$	0.5576271186440678

k-NN Classifier Model Approach

Running the k-NN Classifier and trying different K values from 1 to 24, we achieve the following results:

K Value	Accuracy
1	0.5228215767634855
2	0.5203319502074689
3	0.5219917012448133
4	0.5161825726141079
5	0.5443983402489626
6	0.5369294605809128
7	0.5427385892116182
8	0.5302904564315353
9	0.5203319502074689
10	0.5170124481327801
11	0.5236514522821577
12	0.5269709543568465
13	0.5385892116182572
14	0.5236514522821577
15	0.5261410788381743
16	0.5261410788381743
17	0.5170124481327801
18	0.5203319502074689
19	0.5195020746887967
20	0.5161825726141079
21	0.5195020746887967
22	0.5170124481327801
23	0.5311203319502075
24	0.5145228215767634

Based on the results, we see that we receive the best accuracy when $k = 5$. Calculating the metrics for our optimal value, we achieve the following metrics:

$$\begin{aligned} FMeasure: &\quad 0.5367088607594936 \\ Accuracy: &\quad 0.5443983402489626 \\ Precision: &\quad 0.534453781512605 \\ Recall: &\quad 0.5389830508474577 \end{aligned}$$

As shown in the results, we see that this time around both models performed very poorly by only predicting the right output about half the time.

1.6 Comparing Datasets

When comparing the performance of our probabilistic LDA Model on each of the two datasets, we can see that it only gives us an accurate prediction for dataset 1. This is because dataset 2 consists of 3 different Gaussian distributions, which deviates our data from being a linear model. Although the data isn't clustered in a linear curve in dataset two, the classification is still being determined by one decision boundary and not in random clusters. This is why we notice that the k-NN model still performs equally as bad on both datasets.