COMP 551
Assignment 2

Richie Piyasirisilp, 260624968

Linear Classification and Nearest Neighbor Classification

Q1. Dataset DS1

The training, validation, and testing sets for this part were generated in the folder *dataGenerated* as *DS1_train.csv*, *DS1_valid.csv*, and *DS1_test.csv* respectively. Questions 2 and 3 use data from DS1.

Q2. GDA model using the maximum likelihood approach

2.1(a) GDA best-fit accuracy, precision, recall and F-measure achieved

```
Accuracy:  0.95
Precision:  0.9526184538653366
Recall:  0.9478908188585607
F-measure:  0.9502487562189055
```

2.1(b) Coefficients learnt

These coefficients were use to find the results in part 2.1(a). They were also used to compute the confusion matrix found below.

```
w0:  27.0063735387
w1:  [ 14.20973873  -8.39842924  -5.83927592  -3.27194781  -9.52374936
  -4.13029902  16.79636065 -23.57558641 -28.65705894   8.85520668
 -12.9757844  -12.10569073  15.52936256  12.90480411  -5.57306485
  12.88872263  29.0973488   -6.82594044  -0.86265384  -4.88902291]


Confusion Matrix:  [[382, 19], [21, 378]]
```

Q3. k-NN approach

3(a) Compare GDA and k-NN performance

Based on the results (found below in part 3(b)), it can be determined that the GDA approach was more accurate than the k-NN approach. This can be explained by the distribution of the data. Since the data followed a linear function, the GDA approach would more accurately separate the data points. If the data was more distributed and didn't follow a certain function, the k-NN approach could be more accurate since it is taking k nearest neighbors and classifying based on polling.

The following images show the accuracy achieved by k-NN from k=1 to k=30.

```
k: 1, Accuracy: 0.535
k: 2, Accuracy: 0.52625
k: 3, Accuracy: 0.52625
k: 4, Accuracy: 0.53375
k: 5, Accuracy: 0.535
k: 6, Accuracy: 0.54125
k: 7, Accuracy: 0.54
k: 8, Accuracy: 0.54375
k: 9, Accuracy: 0.54375
k: 10, Accuracy: 0.54375
k: 11, Accuracy: 0.54125
k: 12, Accuracy: 0.54
k: 13, Accuracy: 0.53875
k: 14, Accuracy: 0.54
k: 15, Accuracy: 0.54
```

```
k: 16, Accuracy: 0.54
k: 17, Accuracy: 0.5425
k: 18, Accuracy: 0.5425
k: 19, Accuracy: 0.54
k: 20, Accuracy: 0.53875
k: 21, Accuracy: 0.5375
k: 22, Accuracy: 0.535
k: 23, Accuracy: 0.53625
k: 24, Accuracy: 0.535
k: 25, Accuracy: 0.535
k: 26, Accuracy: 0.53625
k: 27, Accuracy: 0.54125
k: 28, Accuracy: 0.535
k: 29, Accuracy: 0.53625
k: 30, Accuracy: 0.5375
```

The accuracies for each k value were relatively similar. This can be explained once again by the distribution in that it followed a linear function so classifying based on nearest neighbors is not the most accurate method. However, if the data were distributed more randomly, we would see a rapid increase in accuracy early on as k increases then the accuracy would steadily decrease once k passes the optimal k value. This is because as k increases, the classification tends to overfit. In this case, the optimal k value was 8 and using the values from this, the following confusion matrix was computed.

```
Confusion Matrix:  [[314, 276], [89, 121]]
```

3(b) k-NN best-fit accuracy, precision, recall and F-measure achieved

```
Accuracy:  0.54375
Precision:  0.5322033898305085
Recall:  0.7791563275434243
F-measure:  0.6324269889224572
```

Q4. Dataset DS2

The training, validation, and testing sets for this part were generated in the folder *dataGenerated* as *DS2_train.csv*, *DS2_valid.csv*, and *DS2_test.csv* respectively. Question 5 uses data from DS2.

Q5. GDA and k-NN approach

5.1(a) GDA best-fit accuracy, precision, recall and F-measure achieved

```
Accuracy:  0.5429166666666667
Precision:  0.5574324324324325
Recall:  0.5352798053527981
F-measure:  0.546131568059578
```

5.1(b) Coefficients learnt

These coefficients were use to find the results in part 5.1(a). They were also used to compute the confusion matrix found below.

```
w0:  -0.147775446626
w1:  [-0.00629905 -0.04532273 -0.0781147   0.05489176  0.11819182 -0.02588029
 -0.05579255 -0.01832064  0.06520449  0.03776821  0.07484969  0.04302516
  0.0017316   0.01996336 -0.02421451  0.01356156  0.01791875 -0.0222463
  0.0027015  -0.04252262]


Confusion Matrix:  [[660, 524], [573, 643]]
```

5.2 Compare GDA and k-NN performance

Based on the results (found below in part 5.3), it can be determined that the GDA approach is still slightly more accurate than the k-NN approach. However, the difference is minor since both approaches tend to predict the wrong output about half the time. This is due to the fact that DS2 is comprised of 3 Gaussian distributions, thereby making the data non-linear. The GDA approach will therefore no longer perform as accurately.

The following images show the accuracy achieved by k-NN from k=1 to k=30.

```
k: 1, Accuracy: 0.535
k: 2, Accuracy: 0.52625
k: 3, Accuracy: 0.52625
k: 4, Accuracy: 0.53375
k: 5, Accuracy: 0.535
k: 6, Accuracy: 0.54125
k: 7, Accuracy: 0.54
k: 8, Accuracy: 0.54375
k: 9, Accuracy: 0.54375
k: 10, Accuracy: 0.54375
k: 11, Accuracy: 0.54125
k: 12, Accuracy: 0.54
k: 13, Accuracy: 0.53875
k: 14, Accuracy: 0.54
k: 15, Accuracy: 0.54
```

```
k: 16, Accuracy: 0.54
k: 17, Accuracy: 0.5425
k: 18, Accuracy: 0.5425
k: 19, Accuracy: 0.54
k: 20, Accuracy: 0.53875
k: 21, Accuracy: 0.5375
k: 22, Accuracy: 0.535
k: 23, Accuracy: 0.53625
k: 24, Accuracy: 0.535
k: 25, Accuracy: 0.535
k: 26, Accuracy: 0.53625
k: 27, Accuracy: 0.54125
k: 28, Accuracy: 0.535
k: 29, Accuracy: 0.53625
k: 30, Accuracy: 0.5375
```

Once again, the accuracies for each k value were relatively similar. This can be explained by the distribution in that it followed a linear function so classifying based on nearest neighbors is not the most accurate method. In this case, the optimal k value was 9 and using the values from this, the following confusion matrix was computed.

```
Confusion Matrix:  [[781, 807], [452, 360]]
```

5.3 k-NN best-fit accuracy, precision, recall and F-measure achieved

```
Accuracy:  0.47541666666666665
Precision:  0.49181360201511337
Recall:  0.6334144363341444
F-measure:  0.553704360155973
```

Q6. Compare classifier performance on DS1 and DS2

As mentioned earlier, the GDA performance sees a drastic drop in accuracy when moving from DS1 to DS2. This is due to DS1 only using 1 Gaussian distribution, making it easy for the GDA approach to compute. Once the dataset DS2 used 3 Gaussian distributions, the GDA could no longer predict with high accuracy since the data distribution was no longer linear.

The k-NN approach for both datasets performed poorly. In DS1, this was due to the fact that the data was distributed linearly but k-NN is best suited for more random distributions. In DS2, the approach performed poorly due to only having 1 decision boundary. Even though the data was more distributed this time, the data points were mixed well and weren't suited for binary classification.