Abhijay Gupta
Student ID: 260708548

Comp 551 Assignment 2

P.S. Most of the code in both of my Jupyter notebooks is the same since questions 4 and 5 were a repeat of everything in questions 1,2 and 3. I didn't put them in the same notebook to avoid messiness of states since I intended to use the same variable names.

Problem 1

>>Submitting datasets generated with the code<<

Assignment2_260708548_DS1_test.csv
Assignment2_260708548_DS1_train.csv
Assignment2_260708548_DS1_valid.csv

Problem 2

1.(a) Performance metrics for GDA (DS1)

```
Test Accuracy:        0.96
Test Precision:       0.96
Test Recall:               0.96
Test F1 Measure:           0.96
```

>>(Also contained in the file "Assignment2_260708548_2_1_a.txt")<<

To show that the above values are the same by coincidence and not by an incorrect implementation, I am submitting the file –

>>Assignment2_260708548_2_1_a_DS1_test_predictions.csv<<

which contains the predictions made for the test set derived from DS1. Each entry in the file denotes the probability of that test example belonging to the Positive

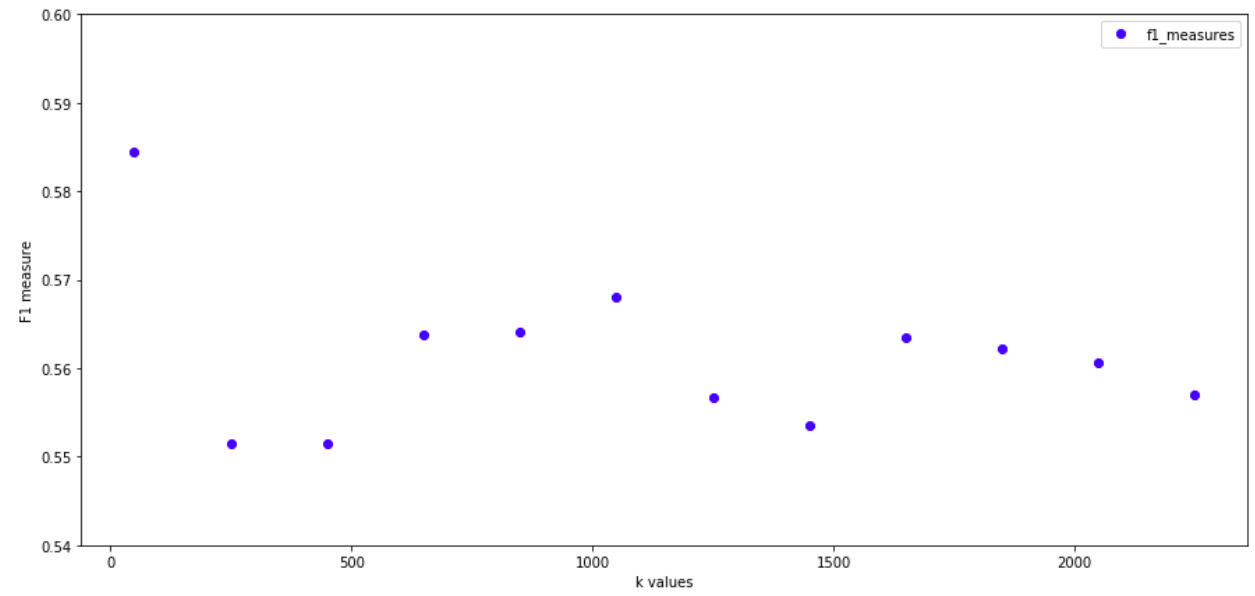
class (or class 1). If the probability is high, there is more indication for belonging to the Positive class and if the probability is low, then there is more indication for belonging to the Negative class (or class 0).

1.(b) Coefficients learnt

>>Submitting the coefficients and parameters used to learn those coefficients<<
>>(contained in the file “Assignment2_260708548_2_1_b.txt”)<<

Problem 3

(a)



Above graph shows the F1 measures obtained for different k (number of nearest neighbors) values. I’ve also submitted the file which contains the input data for the above graph.
>>(“Assignment2_260708548_3_a_f1_measure_for_different_k.csv”)<<

The best F1 measure along with its associated k-value –

```
Best F1 measure:      0.5844980940279542
Associated k-value:         50
```

Therefore, the k-value chosen to use with the test set is **50.**

The k-NN classifier here performs worse than GDA. There are **no** values of k (at least amongst the ones considered here **specifically**) for which k-NN performs better than GDA.

k-NN simply uses an average of the classifications of the neighbors of a point to decide the classification of that point. GDA uses a maximum likelihood procedure wherein various parameters are maximized to get the best classification possible. It takes into account prior probabilities, means and covariances. This provides a more comprehensive analysis and evaluation of the input data and thus leads to a model with higher/better performance metrics.

(b) Performance metrics for k-NN (DS1)

```
Test Accuracy:          0.55875
Test Precision:         0.5560859188544153
Test Recall:                 0.5825
Test F1 Measure:             0.5689865689865691
```

>>(Also contained in the file “Assignment2_260708548_3_b.txt”)<<

Problem 4

>>Submitting datasets generated with the code<<

Assignment2_260708548_DS2_test.csv
Assignment2_260708548_DS2_train.csv
Assignment2_260708548_DS2_valid.csv

Problem 5

1.(a) Performance metrics for GDA (DS2)

```
Test Accuracy:          0.4975
```

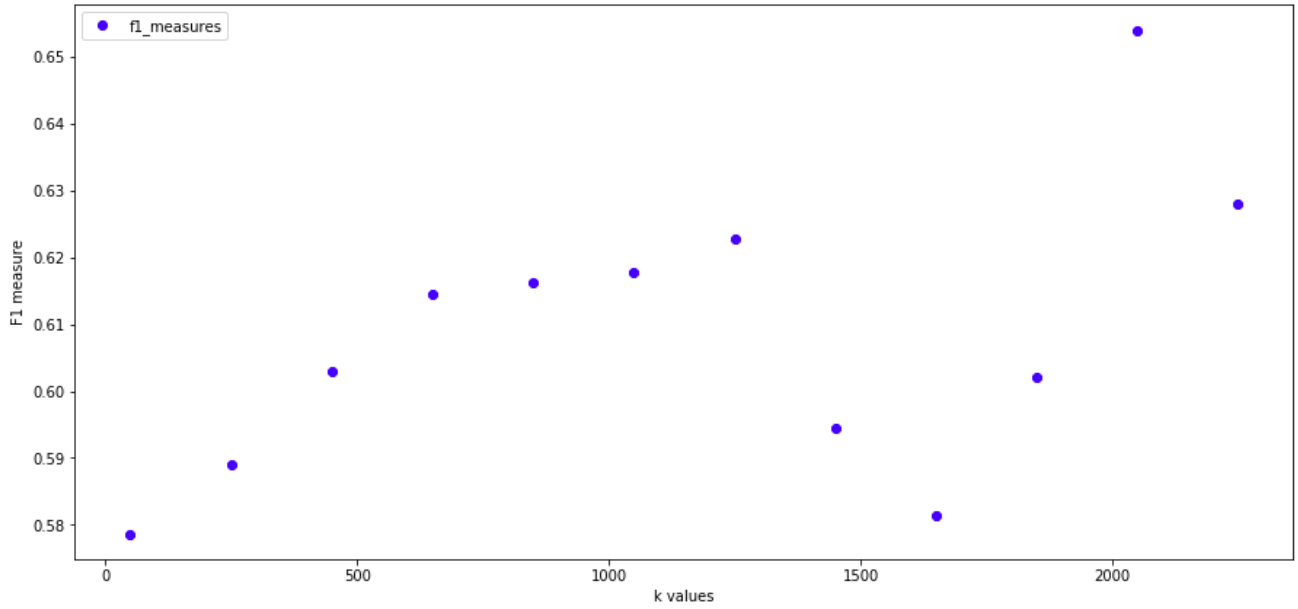
```
Test Precision:        0.49759615384615385
Test Recall:                 0.5175
Test F1 Measure:       0.5073529411764706
```

>>(Also contained in the file “Assignment2_260708548_5_1_a.txt”)<<

1.(b) Coefficients learnt

>>Submitting the coefficients and parameters used to learn those coefficients<<
>>(contained in the file “Assignment2_260708548_5_1_b.txt”)<<

2.



Above graph shows the F1 measures obtained for different k (number of nearest neighbors) values. I’ve also submitted the file which contains the input data for the above graph.
>>(“Assignment2_260708548_5_2_DS2_f1_measure_for_different_k.csv”)<<

The best F1 measure along with its associated k-value –

```
Best F1 measure:           0.6540198735320687
Associated k-value:        2050
```

Therefore, the k-value chosen to use with the test set is **2050.**

Comparing the F1 measures for k-NN and GDA, we can see that k-NN gives a better performance than GDA for **all** values of k that were considered in the solution presented **above**.
DS2 was constructed by mixing three different Gaussians. It isn't composed of the qualities of a single Gaussian. The GDA fails to perform better here due to the mixing of the qualities of the different Gaussians. k-NN is a simple approach which only looks at its neighbors and in this scenario that turns out to be advantageous in classifying the examples.

3. Performance metrics for k-NN (DS2)

```
Test Accuracy:          0.50125
Test Precision:         0.5006993006993007
Test Recall:                 0.895
Test F1 Measure:             0.6421524663677131
```

>>(Also contained in the file "Assignment2_260708548_5_3.txt")<<

Problem 6

The performance metrics for k-NN over DS1 and DS2 vary a little bit but the metrics for GDA differ quite a lot over the two datasets.

With DS1, which is based off of a single Gaussian, GDA does extremely well with its performance metrics. However, when used for DS2, the performance drops significantly. I cite the reasons mentioned above in 5.2.

For k-NN, the baseline of the F1 measure for the different values of k increases in DS2 when compared to DS1. The minimum F1 measure for k-NN in DS1 was ~0.55 while in DS2 it was ~0.58. The recall for k-NN in DS2 is much higher than for DS1, with a value of 0.895.

The metrics among the two classifiers are much more similar in DS2 than in DS1. k-NN takes the lead in all metrics in DS2 and performs better than GDA while the reverse is true for DS1. So, as stated before, introducing multiple Gaussians affected GDA greatly while k-NN was not affected as much as GDA.