

COMP 551 - Applied Machine Learning

Assignment 2

Tiffany Wang - 260684152

February 11, 2018

Acknowledgement

This assignment was fully completely by myself, Tiffany Wang. However, I discussed answers and methodologies with Daniel Lim, John Wu, Frank Ye and Nabil Chowdhury.

Question 1

I randomly generated 2000 class 0 and class 1 examples using the `numpy.random.multivariate_normal` function. Then I randomly selected 70% samples from both classes to build the training set, and used the remaining 30% of the data for the testing set.

The training and testing datasets are saved separately as *DS1_test0.csv*, *DS1_train0.csv* for class 0, and *DS1_test1.csv*, *DS1_train1.csv* for class 1. *DS1_test.csv* and *DS1_train.csv* hold the total datasets.

Question 2

It is important to mention that the weights and the accuracy measures differ from run to run, as the datasets are randomly generated.

The weights have the model are found using the following models:

$$w_0 = \log(p(y_0)) - \log(p(y_1)) - \frac{1}{2}\mu_0^T \Sigma^{-1} \mu_0 + \frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1$$

$$w_1 = x^T \Sigma^{-1} (\mu_0 - \mu_1)$$

The following are the results obtained from the LDA model:

$$\begin{aligned} w_0 &= 27.8476 \\ &\quad [14.708 \quad -8.830 \quad -5.522 \quad -2.621 \quad -10.030 \\ w_1 &= \begin{matrix} -4.185 & 16.792 & -25.039 & -29.697 & 9.557 \\ -13.382 & -12.230 & 15.742 & 13.102 & -5.926 \\ 13.480 & 29.5497 & -6.974 & 0.014 & -5.3335 \end{matrix} \end{aligned}$$

$$\text{Accuracy} = 95.64\%$$

$$\text{Precision} = 95.55\%$$

$$\text{Recall} = 95.70\%$$

$$\text{F-measure} = 0.9562$$

Question 3

In this section, I used the k -NN model to train the model. The approach is to find the output k nearest points to the test input and classify the test input with the highest probably

class from the k nearest points.

Steps of the algorithm: (S1) Calculate euclidean distance from test input to every single training example.

(S2) Select the points closest to the test input (lowest euclidean distance).

(S3) Calculate the mean of the training example ground truth outputs.

(S4) Set prediction_output = 0 if mean \leq 0.5

else prediction_output = 1

The k -NN model was trained using k from 1 to 50. Although the higher the k , the more accurate the value, the variation in accuracy from $k = 1$ to $k = 50$ was minimal, all between 51% and 57%. Therefore, I would say that there was not a particular k that performed better. In fact, the performance of k -NN is close to random guessing (50%), which shows that the model is not suited for this specific problem.

K	Accuracy	25	0.552632
1	0.513158	26	0.549342
2	0.512336	27	0.547697
3	0.54523	28	0.550987
4	0.523026	29	0.539474
5	0.521382	30	0.546053
6	0.524671	31	0.556743
7	0.528783	32	0.553454
8	0.537829	33	0.554276
9	0.537007	34	0.560033
10	0.53125	35	0.555921
11	0.525493	36	0.560033
12	0.516447	37	0.553454
13	0.527961	38	0.558388
14	0.532895	39	0.5625
15	0.527138	40	0.5625
16	0.524671	41	0.5625
17	0.544408	42	0.5625
18	0.533717	43	0.560855
19	0.537007	44	0.570724
20	0.523849	45	0.560855
21	0.532072	46	0.570724
22	0.541941	47	0.573191
23	0.546053	48	0.568257
24	0.539474	49	0.571546

The *best* accuracy was obtained with $k = 47$.

Accuracy = 57.32%

Precision = 63.64%

Recall = 56.29%

F-measure = 0.5973

The sample set, generated from a single Gaussian multivariate distribution, is linear. Hence, as expected, the LDA model performed a lot better than k -NN. The accuracy was almost double, it was 50.1% better, to be exact.

Question 4

Essentially, the examples were created in the same way as in Question 1 using the numpy random multivariate normal generator. However, knowing that the mixture probability of the datasets 1, 2, 3 is (10%, 42%, 48%), I built the datasets with:

$$\begin{aligned} \text{training_set} &= (10\% \cdot 2000 \cdot \text{Dataset1} + 42\% \cdot 2000 \cdot \text{Dataset3} + 48\% \cdot 2000 \cdot \text{Dataset3}) \cdot 70\% \\ \text{testing_set} &= (10\% \cdot 2000 \cdot \text{Dataset1} + 42\% \cdot 2000 \cdot \text{Dataset3} + 48\% \cdot 2000 \cdot \text{Dataset3}) \cdot 30\% \end{aligned}$$

The training and testing datasets are saved separately as *DS2_test0.csv*, *DS2_train0.csv* for class 0, and *DS2_test1.csv*, *DS2_train1.csv* for class 1. *DS2_test.csv* and *DS2_train.csv* hold the total datasets.

Question 5

- LDA model:

$$\begin{aligned} w_0 &= -0.0279 \\ &\quad [0.0402 \quad -0.0017 \quad 0.052 \quad -0.0063 \quad -0.025 \\ w_1 &= \quad 00315 \quad -0.052 \quad 0.0254 \quad -0.0221 \quad 0.0189 \\ &\quad 0.0522 \quad 0.00537 \quad 0.0072 \quad -0.0037 \quad -0.0084 \\ &\quad 0.0633 \quad -0.0032 \quad -0.02449 \quad -0.0776 \quad 0.0060] \end{aligned}$$

$$\text{Accuracy} = 50.12\%$$

$$\text{Precision} = 48.88\%$$

$$\text{Recall} = 61.115\%$$

$$\text{F-measure} = 0.543$$

- k -NN model

K	Accuracy	25	0.51234
1	0.490213	26	0.502128
2	0.498723	27	0.502128
3	0.480851	28	0.502128
4	0.50383	29	0.49617
5	0.48	30	0.490213
6	0.495319	31	0.493617
7	0.48766	32	0.492766
8	0.465532	33	0.494468
9	0.481702	34	0.473191
10	0.479149	35	0.493617
11	0.479149	36	0.47234
12	0.482553	37	0.491064
13	0.492766	38	0.483404
14	0.493617	39	0.493617
15	0.481702	40	0.481702
16	0.480851	41	0.492766
17	0.488511	42	0.480851
18	0.491915	43	0.490213
19	0.497872	44	0.490213
20	0.494468	45	0.491064
21	0.50383	46	0.485106
22	0.502128	47	0.479149
23	0.505532	48	0.480851
24	0.500426	49	0.492766

The *best* accuracy obtained with $k = 25$.

Accuracy = 51.23%

Precision = 55.71%

Recall = 49.68%

F-measure = 0.5253

Since the sample is a mixture of three different multivariate distributions, it lost its linearity. The performance of LDA decreased from 95.64% to 50.12%.

However, in the case of the k -NN model, the prediction results are the similar.

Question 6

On the first hand, the accuracy of the LDA model is changes drastically between DS1 and DS2. As mentioned earlier, DS1 is linear, so the predictions using the LDA model is expectedly precise. However, as the DS2 loses linearity with the mixture of three different Gaussian distributions, the performance of the LDA model drops to around 50%, which is

close to random guessing. I would like to conclude that the LDA model is powerful for the prediction of linear models.

On the other hand, the k -NN model have similar performances for both DS1 and DS2. As k -NN does not make any assumptions on the dataset, its results are independent of the linearity of the datasets.