MACHINE LEARNING

ASSIGNMENT 5

IIT2018178, Manav

PROBLEM STATEMENT

Using the dataset provided, we have to create three extra features and from there we have to develop a GDA model. Furthermore, convert the basic features to gaussian distribution using Box Muller transformation and create a GDA model and compare them.

INTRODUCTION

Discriminative models, also referred to as conditional models or backward models, are a class of supervised machine learning used for classification or regression. These distinguish decision boundaries by inferring knowledge from observed data. This is different to the idea of generative or forward models, and discriminative models make fewer assumptions about the underlying data distribution and rely more on data quality.
Finally, We say that algorithms that model p(y|x) directly from the training set are called discriminative algorithms. Once we learn the model p(y) and p(x|y) using training set, we

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

Where p(x) = p(x|y=1)p(y=1) + p(x|y=0)p(y=0)

If we are calculating p(y|x) in order to make a prediction we don't need p(x) as

$$\begin{aligned} argmax_y p(y \mid x) &= argmax_y \frac{p(x \mid y)p(y)}{p(x)} \\ &= argmax_y p(y \mid x)p(y) \end{aligned}$$

use Bayes Rule to derive the p(y|x) as

When we have a classification problem in which the input features are continuous random variable, we can use GDA, it's a generative learning algorithm in which we assume p(x|y) is distributed according to a multivariate normal distribution and p(y) is distributed according to Bernoulli. So the model is shown below.

$$p(y) = \phi^y(1-\phi)^{(1-y)}$$

$$p(x\,|\,y=0) = \frac{1}{(2\pi)^{n/2}|\sum|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu_0)^T \sum^{-1} (x-\mu_0))$$

$$p(x\,|\,y=1) = \frac{1}{(2\pi)^{n/2}|\sum|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu_1)^T \sum^{-1} (x-\mu_1))$$

Here the parameters of the model are ϕ, μ_0, μ_1 and $\sum$.
And n is the dimension of the density function
Note : While there are two separate mean vectors μ_0 and μ_1(each for one class), but the covariance matrix $\sum$ is common for all the classes.

The features to be calculated are

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

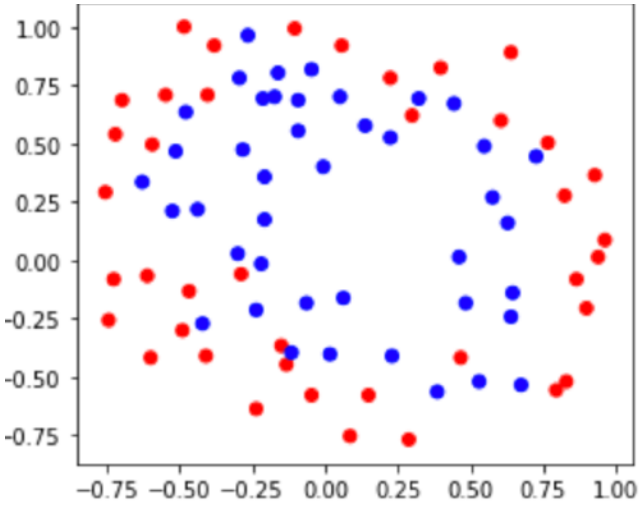
$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

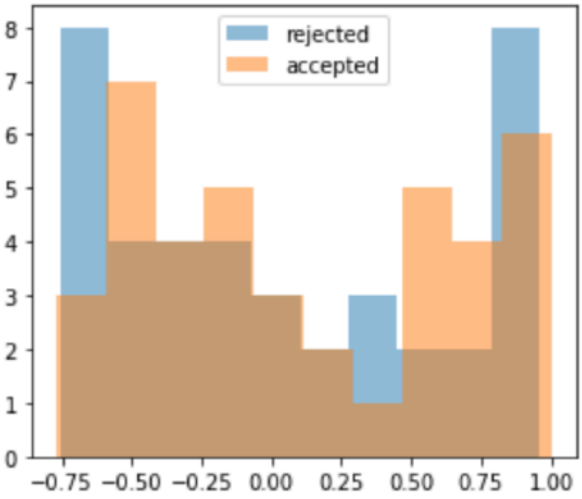
$$\sum = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T \quad \text{where } k = 1\{y^{(i)} = 1\}$$

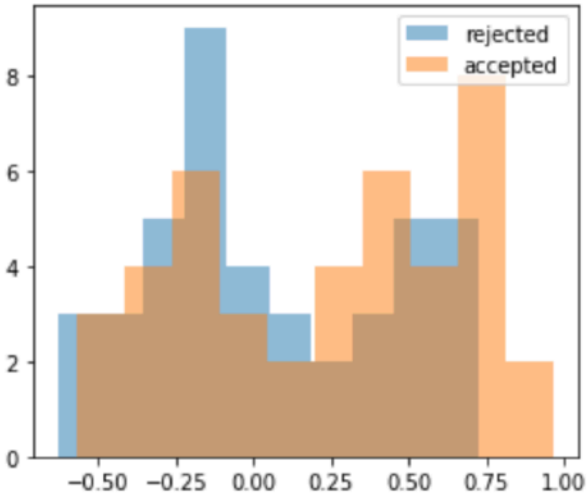
PERFORMANCE ANALYSIS OF PART 1

Here, The three features added are $(\text{Original\_feature\_1})^2$, $(\text{Original\_feature\_2})^2$ and the product of the original features.

When we use scatter plot while plotting original features we see that the features are rather spread out.







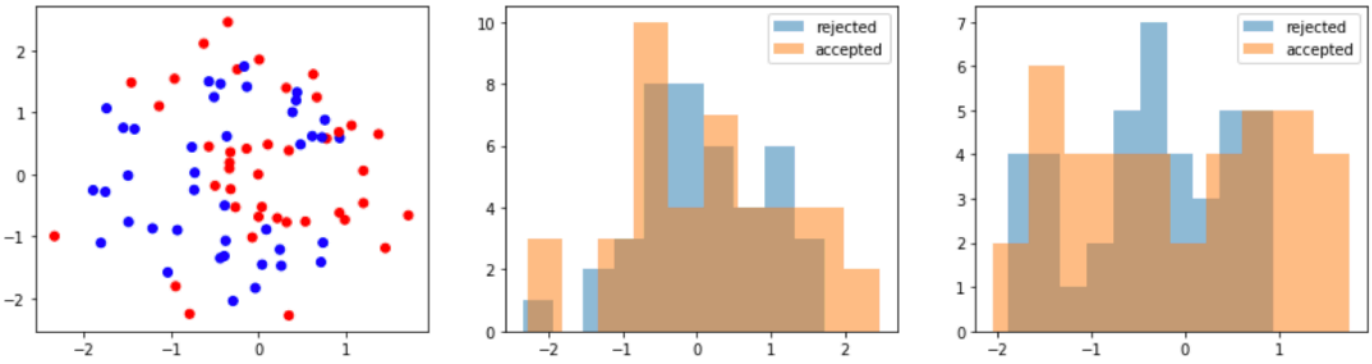
Next, we see the involvement of each of the original features against our output.

To increase accuracy we shuffle our dataset. The seed value used for this is 30. The accuracy at the end achieved is :

```
Accuracy on test Set 80.55555555555556
```

PERFORMANCE ANALYSIS OF PART 2

Here the features added are the same. When we use scatter plot and histogram for plotting our dataset we see that the data is less scattered. It is easy to make out that it is gaussian data.



Finally to increase accuracy we shuffle data and set the seed value to 1739. The accuracy obtained is

```
Accuracy of testing set 86.11111111111111
```

RESULT

As the results show that after box muller transformation that accuracy of the test set increases to 86.1% from 80.5%. We see that this model is perfect for small datasets with to have a high accuracy. However, if the transformation is not perfect then the performance in this model will vary greatly.