Alaa Ayach A20317680

Problem Statement

In this assignment I am trying to implement Generative learning for both Gaussian and Naive Bayes algorithms as discriminant functions. I changed parameters such as data, number of dimensions, and number of classes.
I analyzed the differences comparing the different results based on confusion Matrix.

Proposed Solution

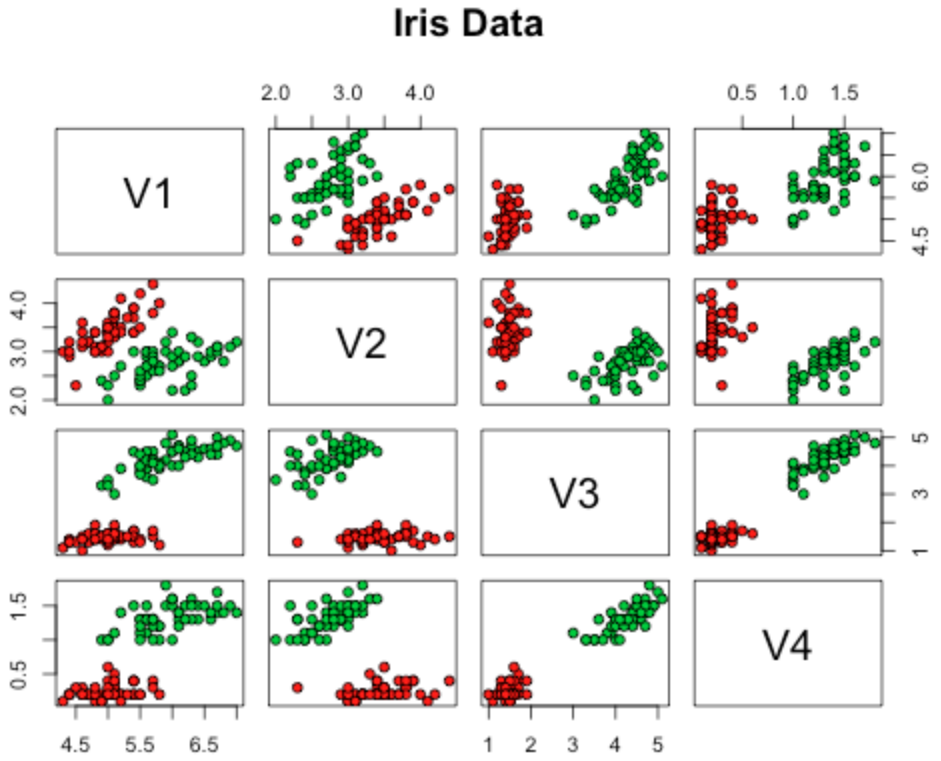
I implemented all functions from scratch using only matrix manipulation given by R. I used aggregate function to compute sums.
I used confusion Matrix to come up with the error.

Implementation Details

I encapsulate all the work in a modular way using files and parameters. Those files took the parameters that I talked about above. used some package like cvTools to help in cross validation, ROC measures and comparing existent classification functions

Results and Discussion

2-class GDA - Loading and plotting Iris data (2 classes kept / dataset uploaded)
Example from Iris dataset



1D GDA (iris 1d.R)

I used here the first feature of the data along with the first two classes "Iris-setosa" and "Iris-versicolor", and I got the following results for test data (20 examples out of 100, 80 examples to train)

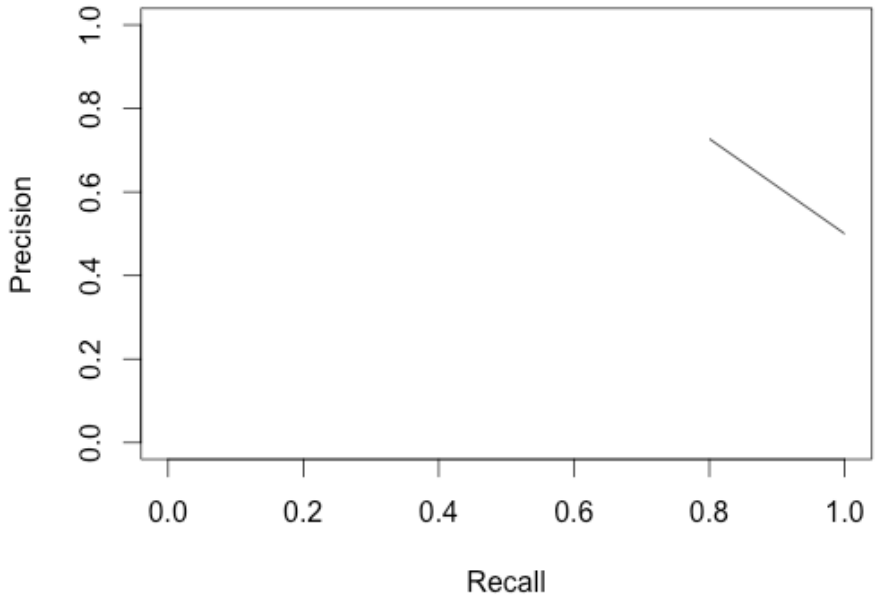
Confusion matrix:

yhat	Iris-setosa	Iris-versicolor
Iris-setosa	7	2
Iris-versicolor	3	8

We can say it isn't very bad results (it tends that the first feature play a good role), however this result could be better

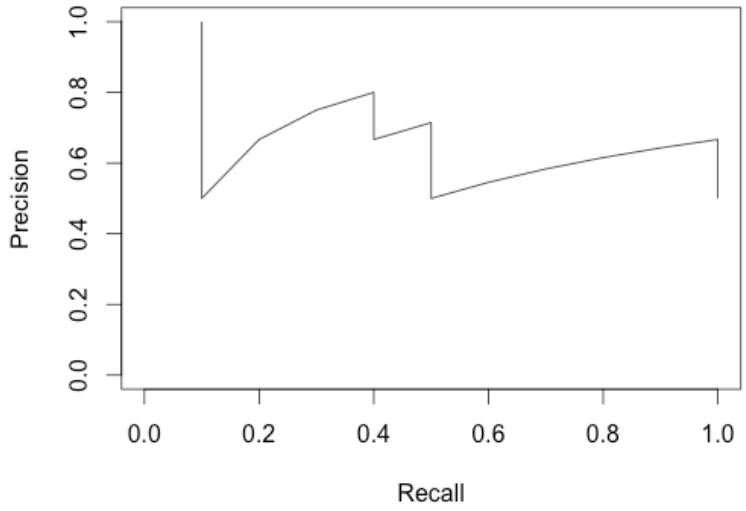
ROC Curve and computing all measures (Measures.R)

I used here the package ROCR and got these results:

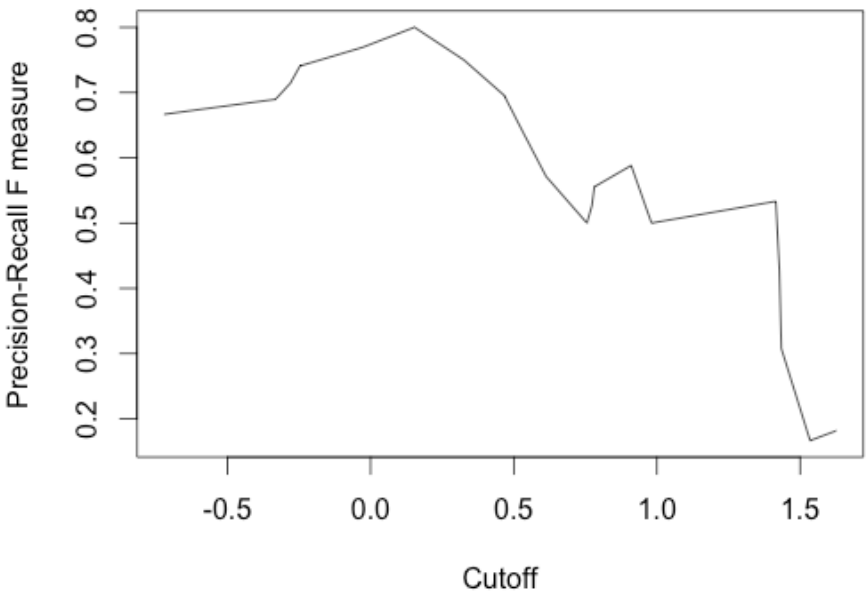


We notice that the curve is good but this is for this small testing set

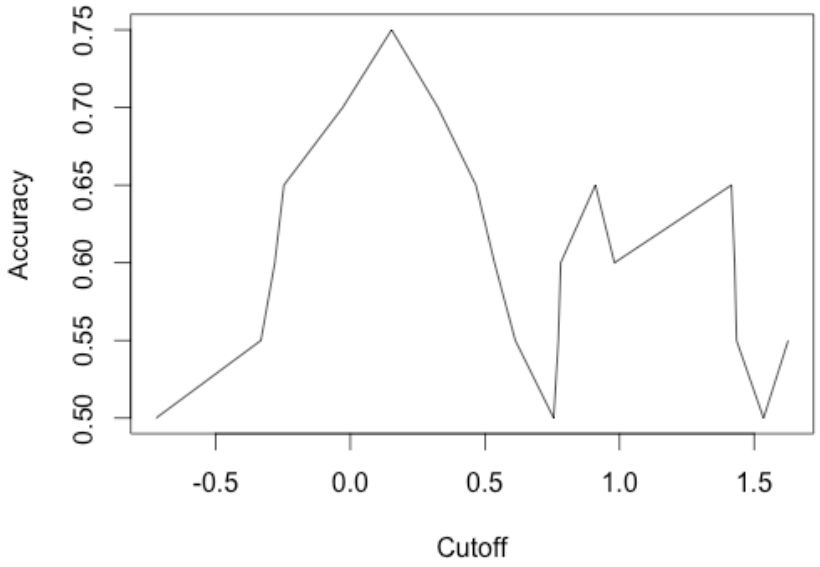
We can see that when we put some noise:



F-Measure



accuracy



>auc
[1] 0.7
> rmse
[1] 1.104314

We notice that model now isn't doing very good

nD 2-class GDA (iris Gaussian nD 2-class.R)

I used here the four features of the data along with the first two classes "Iris-setosa" and "Iris-versicolor", and I got the following results for test data (20 examples out of 100, 80 examples to train)

Confusion matrix:

yhat	Iris-setosa	Iris-versicolor
Iris-setosa	10	0
Iris-versicolor	0	10

Here we had perfect results with no error, that is because the iris data is well distributed as for Gaussian, there wasn't cross validation, and because the dataset isn't that big

nD k-class GDA (iris Gaussian nD k-class.R)

I used here the four features of the data along with the three classes "Iris-setosa" and "Iris-versicolor", and I got the following results for test data (20 examples out of 100, 80 examples to train)

Confusion matrix:

yhat	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	10	0	0
Iris-versicolor	0	10	0

As the test data didn't include the third class, here we had perfect results with no error, however we should perform here cross validation to see all examples and get the real error

Cross validation(cross validation.R)

I used here the package cvTools for cross validation and I got the average results:

```
     [,1] [,2] [,3]
[1,]    9    1    1
[2,]    1    7    0
[3,]    0    2    9
```

We see that it did really good as for three classes

Naive Bayes discriminant Analysis

I used here CNAE-9 with 9-category (classes but used only 2). here is a link to the dataset http://archive.ics.uci.edu/ml/machine-learning-databases/00233/CNAE-9.data
where the first column is the class

nD k-class NB (Naive Bayes Bernoulli.R)

I used here only Bernoulli attributes omitting the others

Confusion matrix:

```
   1  2
1 12 0
2  0  11
```

Here because choosing only data that has Bernoulli, the results were good for classes 1 and 2

nD k-class NB (Naive Bayes Binomioal.R)

I used here all attributes but I picked two classes as mentioned in the assignment

Confusion matrix:

```
   1  2
1 12 0
2  0  11
```

We noticed the same results here, however with Binomioal features the results are getting much better for k-class

Estimating equations for Naive Bayes

Likelihood will begin the same except for using Binomioal function:

$l(\boldsymbol{\alpha}) = log(\Pi\ \Pi\ (P\ x)\ (\boldsymbol{\alpha}|y=1)^{x}\ (1-\boldsymbol{\alpha}|y=1)^{1-x}\ )$

$= \Sigma\ \Sigma\ log(...$

with the same steps we will find

$\boldsymbol{\alpha} = \Sigma\ 1(y=1) * x / m + \Sigma\ 1(y=2) * x / m + ... + \Sigma\ 1(y=maxFreq) * x / m$