CS 584

MACHINE LEARNING

ASSIGNMENT 2

by

Saurabh Katkar

A20320336

Code created using Rstudio version 0.98.1062

Generative Learning

Introduction

Generative classifiers model the joint probability distribution **p(x,y)** of the inputs **x** and the labels **y** and then use the Bayes' rule to calculate **P(y|x)** and pick the most likely label.
Generative model can be thus used to generate new samples given the joint distribution of the inputs. Generative algorithms model **P(x,y)** which can be tranformed into **P(y|x)** by applying Bayes rule and then used for classification. However, the distribution **P(x,y)** can also be used for other purposes., for example to *generate* likely **(x,y)** pairs.

The Iris Dataset (Continuous Dataset)

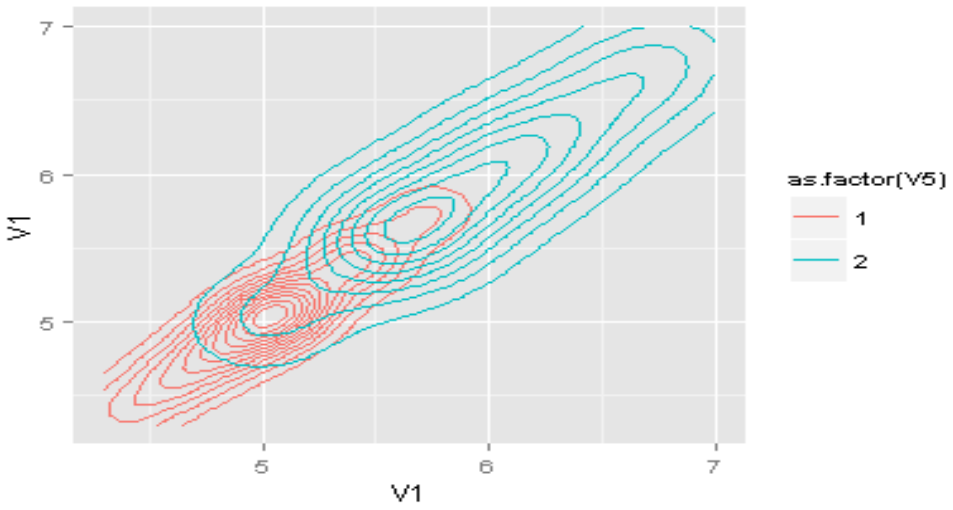
The ***Iris* flower data set** is a multivariate data set consisting of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals. Using these continuous features classification can be made by developing a generative model of the species each flower in the dataset belongs to.

1. Using a 1-D, 2 Class Iris Dataset

1.1 Loading the dataset:

The dataset is loaded into **R** using the function **read.table()**. We can subsequently print the contents of the dataset and plot them on the graph using **print()** and **plot()** respectively.
We then split the dataset so as to include only one feature of the dataset mapping two classes.

Thus we obtain the following plot for the dataset;



1.2 Estimating model parameters, generating membership function and deriving the discriminant for the model

We compute the mean(μ) and variance(σ) for the set of features for deriving the membership function. The membership function for 1-Dimensional features will be computed using Gaussian Discriminant Analysis (GDA)

Gaussian Discriminant Analysis (GDA) 1-D:

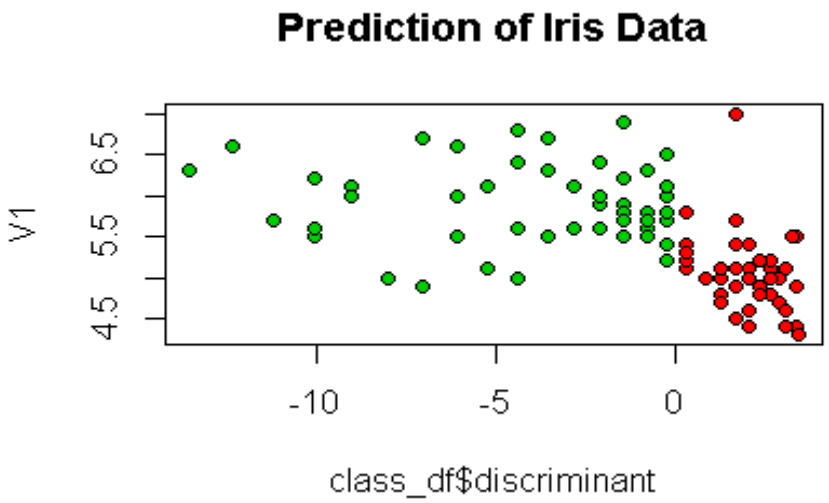
$$P(X|y=j) = \frac{1}{sqrt(2 * pi * \sigma^2)} * \exp\left(-\frac{(x-\mu)^2}{2} * \sigma^2\right)$$

Class membership function lets us know how much a variable **x** belongs to class **y**. Applying GDA gives us two values for the class membership function using which we can determine the discriminant as follows:

$$d(x) = g_1(x) - g_2(x)$$

Using the resulting value we can generate a decision boundary - a boundary that will classify the data into two predicted classes.

The plot for the data classification based on our prediction of classes is as follows:



1.3 Computing Mean Square Error using Cross Validation, Confusion Matrix, Accuracy, Precision, Recall and F-measure

Cross Validation:

Applying Cross Validation gives the following value for Mean Square Error;

CV	
MSE	**0.122**

Confusion Matrix:

Confusion Matrix is a specific table layout that allows visualization of the performance of an algorithm. Each column in the matrix represents a predicted class and each row represents the actual class. Thus the matrix helps us gauge the predicted data with the real data.

The derived confusion matrix in our case is;

Confusion Matrix		
	1	**2**
1	**50**	**1**
2	**0**	**49**

Accuracy, Precision, Recall and F-measure:

These are the different performance assessment parameters that help us determine the quality of our classification algorithm. A well behaved model should produce a balanced matrix and have consist percent correctness numbers for accuracy, recall, precision and an F measure.

The Accuracy, Precision, Recall and F-measure for our 1-D, 2 Class dataset is;

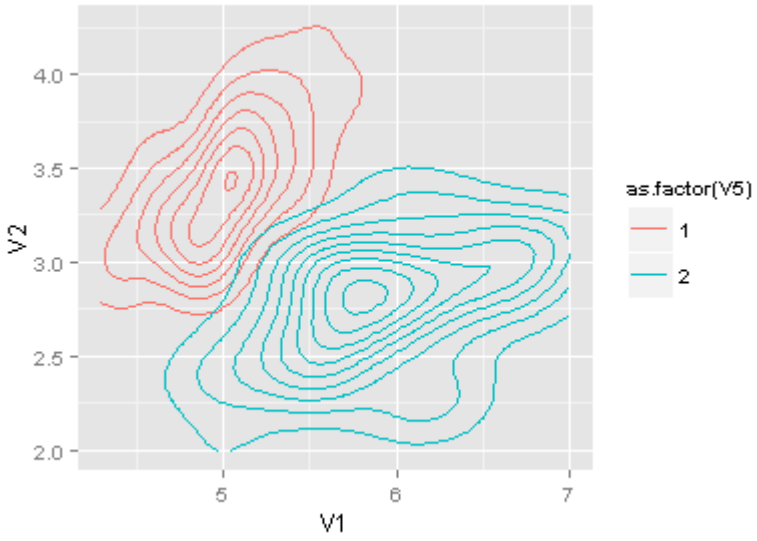
Species	**Accuracy**	**Precision**	**Recall**	**F-measure**
Setosa	**0.98**	**0.9804**	**1**	**0.99099**
Versicolor		**1**	**0.98**	**0.989899**

2. Using a n-D, 2 Class Iris Dataset

2.1 Loading the dataset:

The dataset is loaded into **R** using the function **read.table()**. We can subsequently print the contents of the dataset and plot them on the graph using **print()** and **plot()** respectively.
We then split the dataset so as to include only one feature of the dataset mapping two classes.

Thus we obtain the following plot for the dataset;



2.2 Estimating model parameters, generating membership function and deriving the discriminant for the model

We compute the mean(μ) and covariance(Σ) matrix for the set of features for deriving the membership function.
The membership function for n-Dimensional features will be an exponential function of the **Mahalanobis' Distance.**

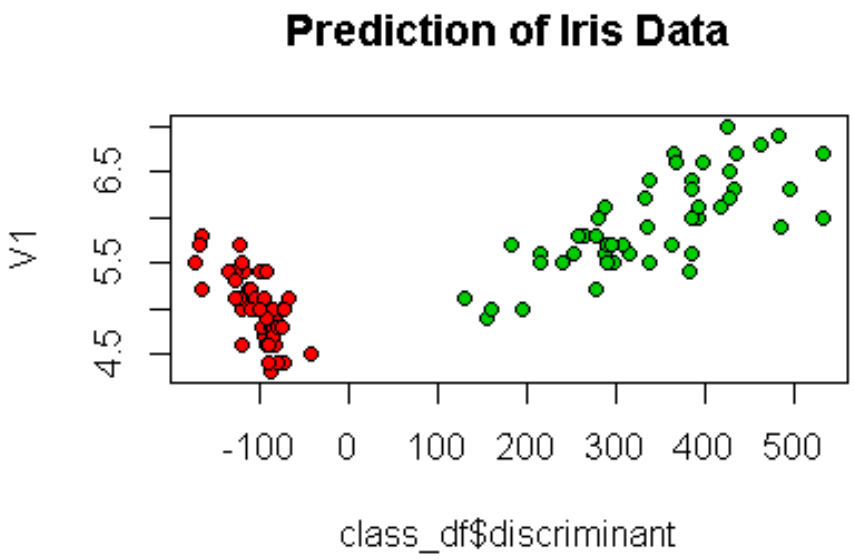
Gaussian Discriminant Analysis (GDA) n-D:

$$gi(x) = \frac{1}{sqrt(2 * pi)^n * sqrt(abs(\Sigma))} * \exp\left(-\frac{1}{2} * (x-\mu)^t * \Sigma^{-1} * (x-\mu)\right)$$

Applying GDA gives us two values for the class membership function using which we can determine the discriminant as follows:

$$d(x) = g_1(x) - g_2(x)$$

Based on the derived decision boundary, the plot for the data classification based on our prediction of classes is as follows:



2.3 Computing Mean Square Error using Cross Validation, Confusion Matrix, Accuracy, Precision, Recall and F-measure

Cross Validation:

Applying Cross Validation gives the following value for Mean Square Error;

Cross Validation	
MSE	**0.0105**

Confusion Matrix:

The derived confusion matrix in our case is;

Confusion Matrix		
	1	**2**
1	**50**	**0**
2	**0**	**50**

Accuracy, Precision, Recall and F-measure:

The Accuracy, Precision, Recall and F-measure for our n-D, 2 Class dataset is;

Species	**Accuracy**	**Precision**	**Recall**	**F-measure**
Setosa	**1**	**1**	**1**	**1**
Versicolor		**1**	**1**	**1**

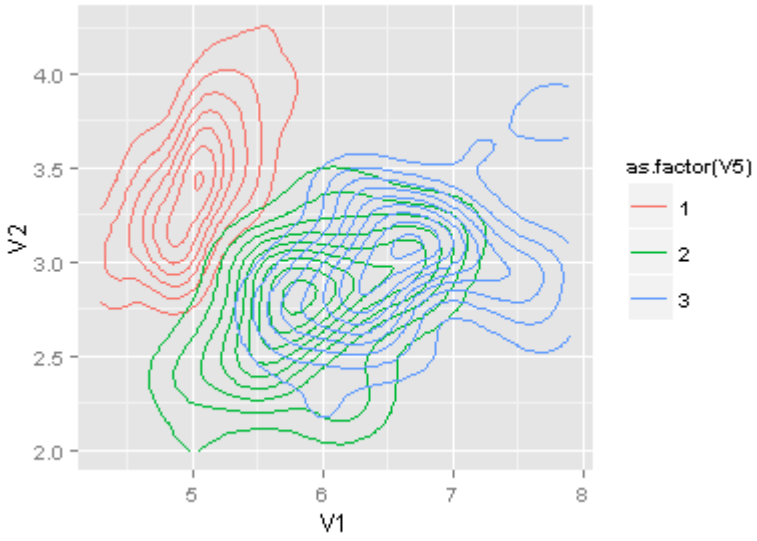
Precision Recall Curve

3. Using a n-D, 3 Class Iris Dataset

3.1 Loading the dataset:

The dataset is loaded into **R** using the function **read.table()**. We can subsequently print the contents of the dataset and plot them on the graph using **print()** and **plot()** respectively.
We then split the dataset so as to include only one feature of the dataset mapping two classes.

Thus we obtain the following plot for the dataset;



3.2 Estimating model parameters, generating membership function and deriving the discriminant for the model

We compute the mean(μ) and covariance(Σ) matrix for the set of features for deriving the membership function.
The membership function for n-Dimensional features will be an exponential function of the **Mahalanobis' Distance.**

Gaussian Discriminant Analysis (GDA) n-D:

$$gi(x) = \frac{1}{sqrt(2 * pi)^n * sqrt(abs(\Sigma))} * \exp\left(-\frac{1}{2} * (x-\mu)^t * \Sigma^{-1} * (x-\mu)\right)$$

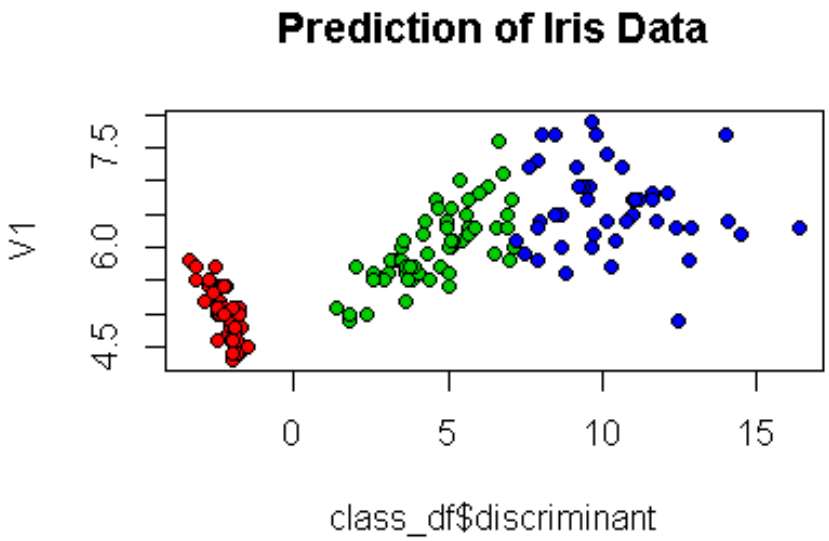
Applying GDA gives us three values for the class membership function using which we have to derive two decision boundaries. This can be done as follows:

$$d_1(x) = g_1(x) - g_2(x)$$

$$d_2(x) = g_2(x) - g_3(x)$$

$$d_3(x) = g_1(x) - g_3(x)$$

For decision boundary, we take the maximum value returned by each equation within the span of the real classes in the dataset.
Based on the derived decision boundary, the plot for the data classification based on our prediction of classes is as follows:



3.3 Computing Mean Square Error using Cross Validation, Confusion Matrix, Accuracy, Precision, Recall and F-measure

Cross Validation:

Applying Cross Validation gives the following value for Mean Square Error;

CV	
MSE	0.0105

Confusion Matrix:

The derived confusion matrix in our case is;

Confusion Matrix			
	1	2	3
1	50	0	0
2	0	50	6
3	0	0	44

Accuracy, Precision, Recall and F-measure:

The Accuracy, Precision, Recall and F-measure for our n-D, 2 Class dataset is;

Species	Accuracy	Precision	Recall	F-measure
Setosa	0.92	1	1	1
Versicolor		0.8929	1	0.9434
Virginica		1	0.88	0.9362

The Spambase Dataset (Discrete Dataset)

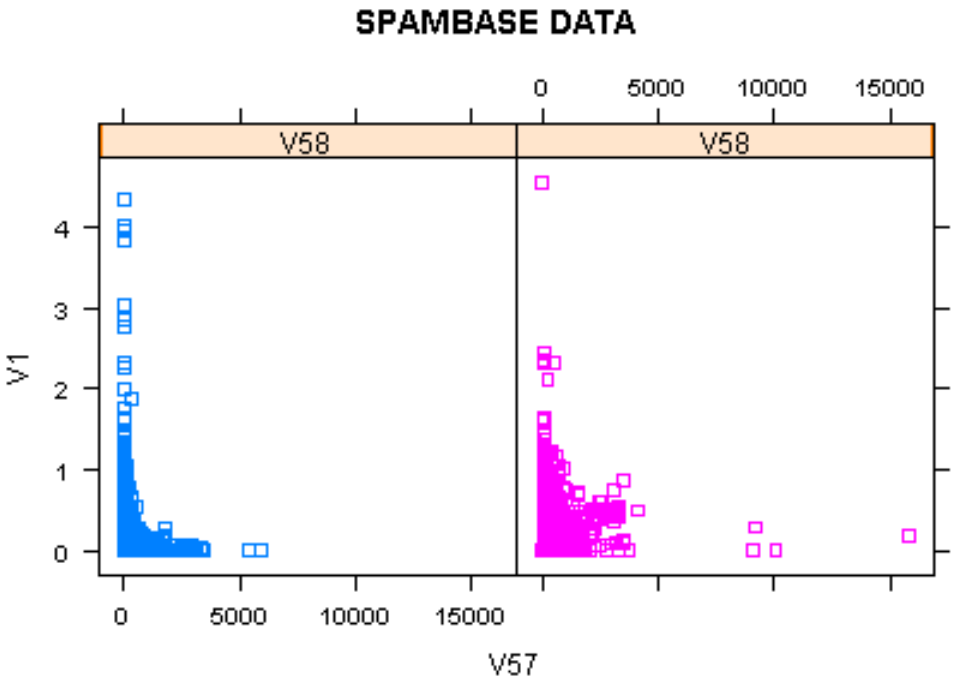
This dataset is based on the classification of the mails received as to whether they are spam(1) or not spam(0). The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail. Most of the attributes indicate whether a particular word or character was frequently occurring in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.

4. Using a n-D, 2 Class Spambase Dataset

4.1 Loading the dataset:

The dataset is loaded into **R** using the function **read.table()**. We can subsequently print the contents of the dataset and plot them on the graph using **print()** and **plot()** respectively.
We then split the dataset so as to include only one feature of the dataset mapping two classes.

Thus we obtain the following plot for the dataset;



4.2 Estimating model parameters, generating membership function and deriving the discriminant for the model

In a multivariate distribution if we have discrete random variables, we use **Multinomial Distribution** instead of a Gaussian Distribution.

Thus for discrete random variables;

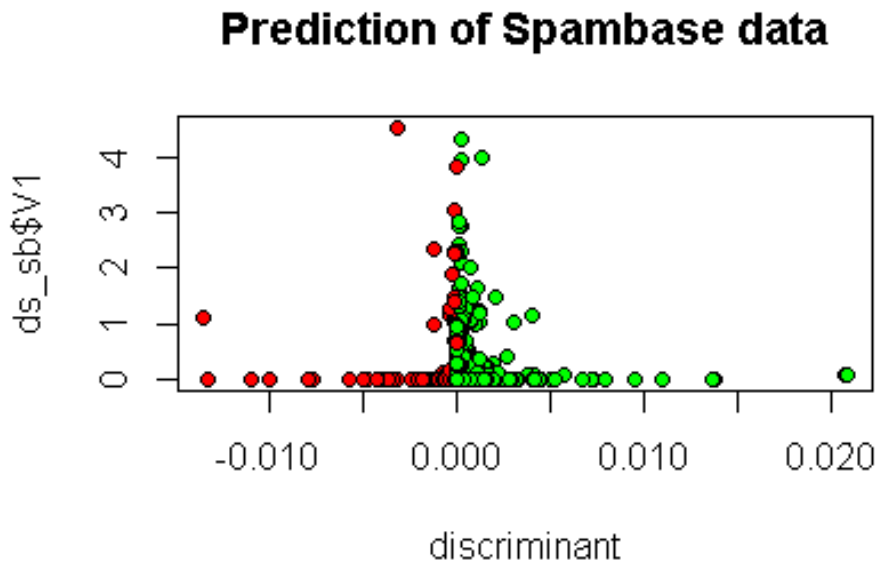
$$f(k) = \binom{n}{k} p^k * (i-p)^{(n-k)}$$

Class membership function lets us know how much a variable **x** belongs to class **y**. Applying GDA gives us two values for the class membership function using which we can determine the discriminant analyisis as follows:

$$d(x) = g_1(x) - g_2(x)$$

Using the resulting value we can generate a decision boundary - a boundary that will classify the data into two predicted classes.

The plot for the data classification based on our prediction of classes is as follows:



4.3 Computing Mean Square Error using Cross Validation, Confusion Matrix, Accuracy, Precision, Recall and F-measure

Cross Validation:

Applying Cross Validation gives the following value for Mean Square Error;

CV	
MSE	0.226

Confusion Matrix:

The derived confusion matrix in our case is;

Confusion Matrix		
	1	2
1	1011	1777
2	237	1576

Accuracy, Precision, Recall and F-measure:

The Accuracy, Precision, Recall and F-measure for our n-D, 2 Class dataset is;

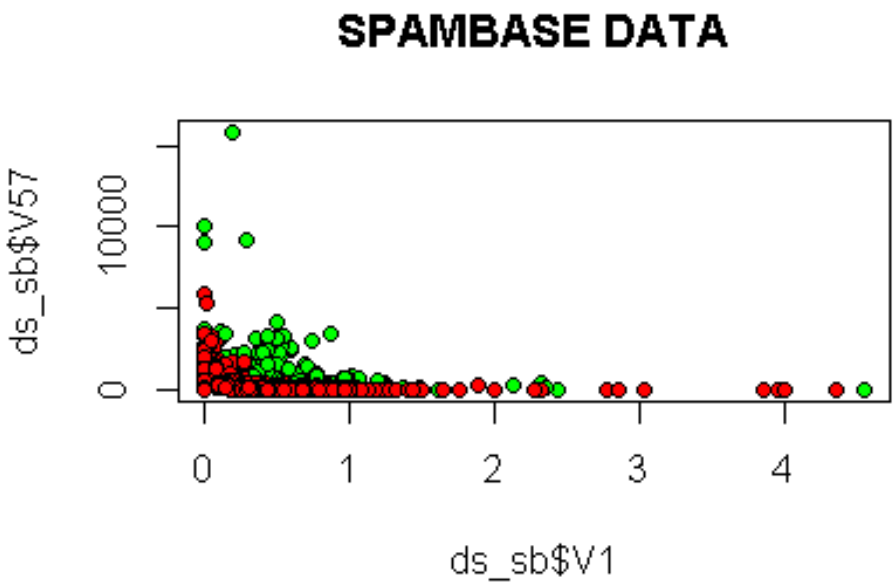
Type	Accuracy	Precision	Recall	F-measure
Spam	0.1245	0.3626	0.8101	0.51
Not Spam		0.8692	0.47	0.61

5. Using a n-D, n Class Spambase Dataset

5.1 Loading the dataset:

The dataset is loaded into **R** using the function **read.table()**. We can subsequently print the contents of the dataset and plot them on the graph using **print()** and **plot()** respectively.
We then split the dataset so as to include only one feature of the dataset mapping two classes.

Thus we obtain the following plot for the dataset;



5.2 Using Naïve-Bayes assumption and estimating model parameters, generating membership function and deriving the discriminant for the model

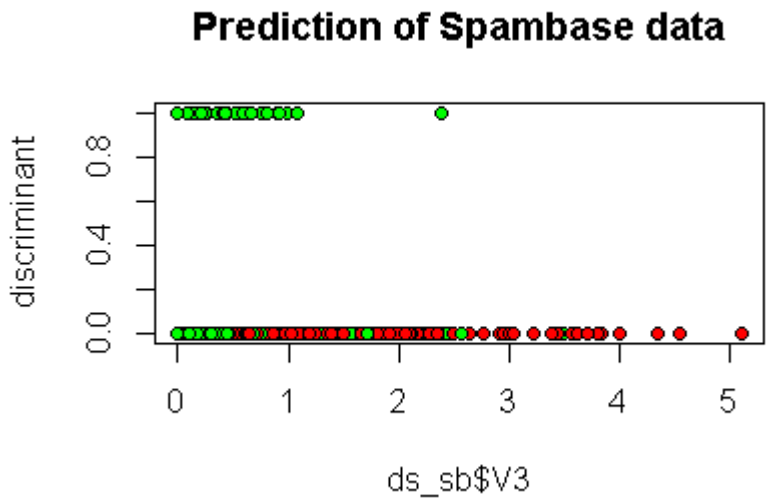
In a multivariate distribution if we have discrete random variables, we use **Multinomial Distribution** instead of a Gaussian Distribution.

We make use of the **Naïve-Bayes** assumption to compute the discriminant function;

Thus for discrete random variables;

$$f(k) = \binom{n}{k} p^k * (i-p)^{(n-k)}$$

Applying Naïve-Bayes assumption for discrimant analysis, we get the following classification plot;



5.3 Computing Mean Square Error using Cross Validation, Confusion Matrix, Accuracy, Precision, Recall and F-measure

Cross Validation:

Applying Cross Validation gives the following value for Mean Square Error;

CV	
MSE	0.45

Confusion Matrix:

The derived confusion matrix in our case is;

Confusion Matrix		
	1	2
1	1636	1152
2	817	996

Accuracy, Precision, Recall and F-measure:

The Accuracy, Precision, Recall and F-measure for our n-D, 2 Class dataset is;

Type	Accuracy	Precision	Recall	F-measure
Spam	0.1441	0.5868	0.667	0.6243
Not Spam		0.5493	0.4637	0.501