

DATA, STATS ASSIGNMENT 07 [Option B]

I have used 3 different classification models:

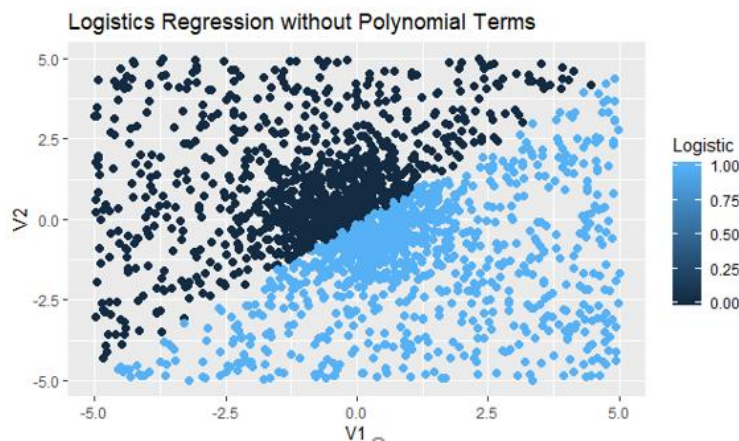
- Logistics Regression without Polynomial Terms
- Logistics Regression with Polynomial Terms
- Naive Bayes Classifier

Out of the 3 models, only 2 models can plot values in circular figure which are : Logistics Regression with Polynomial terms and Naive Bayes Classifier.

MODEL 1: Logistics Regression without Polynomical terms

1. Loaded the circle.arff data from <http://abel.ischool.illinois.edu/data/circle.arff>
2. Installed the ggplot library.
3. Setting the current working Directory
4. Reading the data from circle.arff file as CSV file by making all the "@" character as comments and without header as it is not given in the data file.
5. Fitting Logistics model without using Polynomial terms **x and y**. This means the points V1 and V2 are inside the circle and therefore, V3 = 1.
6. Predicting the probability of having a circle.
7. Making 0.5 or more as a cutoff rate for having circle to get the accurate circle plot.
8. Plotting the expected output using ggplot library as per logistics regression.

In this model, we are using logistic regression without using polynomial terms the plotting of points is getting divided above and below of a straight line instead of a circle. As show below:



Code:

```
library(ggplot2)
#Changing the working Directory
setwd("C:\\Users\\admin\\Desktop\\Data Stats\\Assignment 07")

# reading the data
circle = read.csv("circle.arff", header=FALSE, comment.char = "@")

# Logistics Regression without Polynomical terms

# Fitting Logistics model
```

```
logistic_model = glm(V3~V2+V1, family = "binomial", data = circle)

# Predicting the probability of having a circle
p =predict(logistic_model, circle, type = "response")

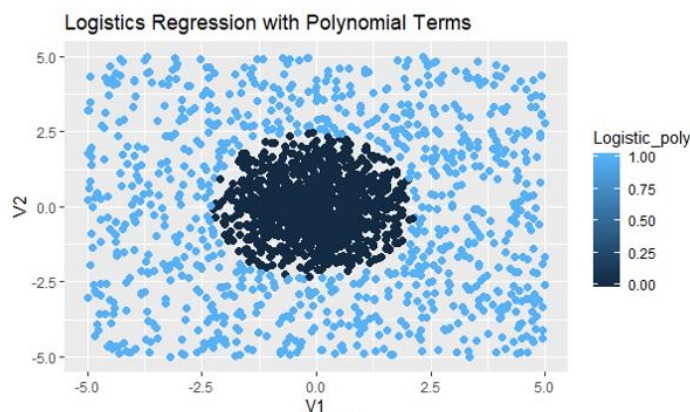
#Making 0.5 or more as a cutoff rate for having circle
Logistic = ifelse(p >= 0.5, 1, 0)

#plotting the expected output as per logistics regression
ggplot(circle, aes(x = V1, y =V2, color = Logistic)) + geom_point(size = 2) + ggtitle("Logistics
Regression without Polynomial Terms")
```

MODEL 2: Logistics Regression with Polynomial terms

1. Fitting Logistics model with using Polynomial terms **x and y**. This means the points V1 and V2 are at (outside) the circle and therefore, V3 = 0.
2. Predicting the probability of having a circle.
3. Making 0.5 or more as a cutoff rate for having circle to get the accurate circle plot.
4. Plotting the expected output using **ggplot** library as per logistics regression.

In this model, we are using logistic regression with using polynomial terms the plotting of points is getting spread-out in centre but in oval shape and not circular shape. As show below:



Code to be continued:

```
#Fitting the Logistics Regression
logistic_model = glm(V3~V2+V1+ I(V1^2)+ I(V2^2), family = "binomial", data = circle)

# Predicting the probability of having a circle
p =predict(logistic_model, circle, type = "response")

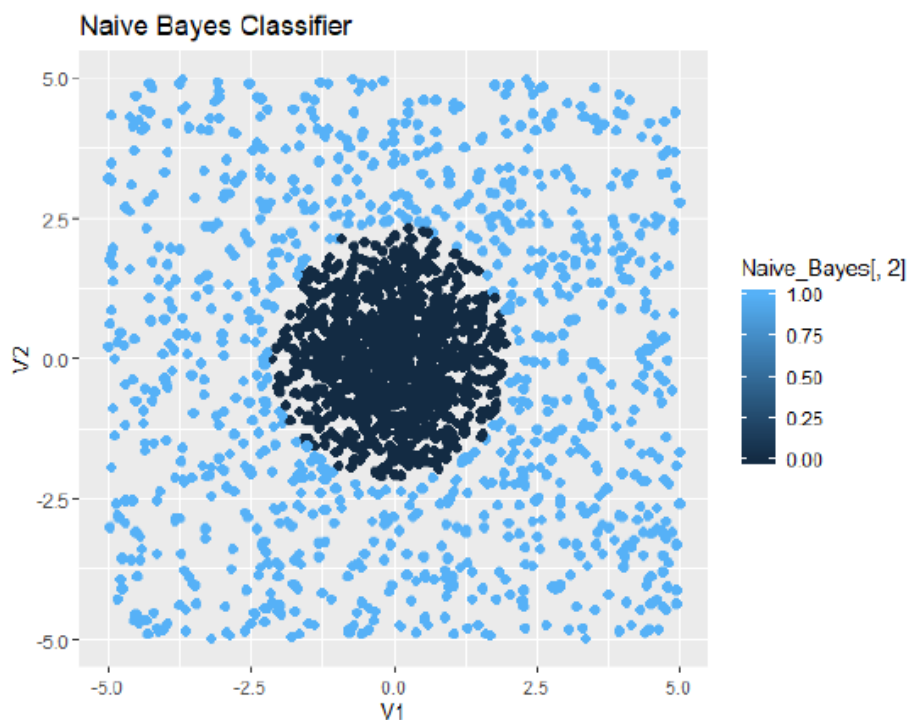
#Making 0.5 or more as a cutoff rate for having circle
Logistic_poly = ifelse(p >= 0.5, 1, 0)

#plotting the expected output as per logistics regression with polynomial terms
ggplot(circle, aes(x = V1, y =V2, color = Logistic_poly)) +
  geom_point(size = 2) +
  ggtitle("Logistics Regression with Polynomial Terms")
```

MODEL 3: Naive Bayes Classifier

1. Fitting the naive bayes classifier
2. Predicting the probability of having a circle
3. Making 0.5 or more as a cutoff rate for having circle
4. Plotting the expected output as per Naive Bayes Classifier

Here, we can see the plotting in a kind of oval shape but not an exact circular shape as compared to above to models.



Code to be continued:

```
library(e1071)
circle_nb = naiveBayes(V3 ~V1+V2, data=circle)

# Predicting the probability of having a circle
nb_p = predict(circle_nb, circle, type = "raw")

#Making 0.5 or more as a cutoff rate for having circle
Naive_Bayes = ifelse(nb_p >= 0.5, 1, 0)

#plotting the expected output as per Naive Bayes Classifier
ggplot(circle, aes(x = V1, y = V2, color = Naive_Bayes[,2])) +
  geom_point(size = 2) +
  ggtitle("Naive Bayes Classifier")
```