**ASSIGNMENT 3**

**CLL788**

**Submitted by:**

**Mohd Zaki**

**2019CEZ8233**

**Department of Civil Engineering**

Table of Contents

[Solution to Question 1 2](#_Toc36854553)

[1) Visualizing training data (Data1.xlsx) 2](#_Toc36854554)

[Solution to Question 2 2](#_Toc36854555)

[2.) Training data with decision boundary obtained by running vanilla version of SVM 2](#_Toc36854556)

[Solution to Question 3 2](#_Toc36854557)

[3.) Modified Optimization Problem 3](#_Toc36854558)

[Solution to Question 4 4](#_Toc36854559)

[4.) Naïve Bayes Classification 4](#_Toc36854560)

[Solution to Question 5 6](#_Toc36854561)

[5.) K-Means classification 6](#_Toc36854562)

Table of Figures

[**Figure 1 – Visualizing training dataset** 2](#_Toc36854569)

[**Figure 2 - SVM Result** 2](#_Toc36854570)

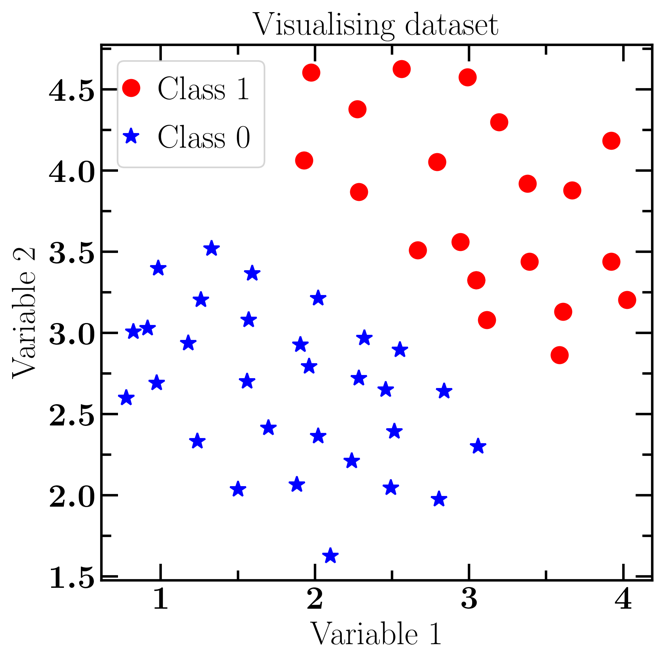
[**Figure 3a – Comparing effect of C** 3](#_Toc36854571)

[**Figure 3b – Comparing effect of C ≤ 1** 3](#_Toc36854572)

[**Figure 3c – Comparing effect of C ≤ 1** 3](#_Toc36854573)

# Solution to Question 1

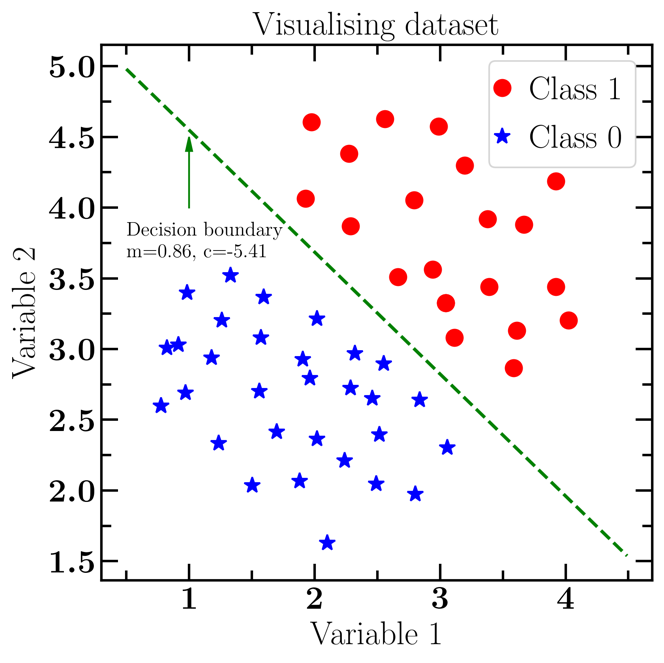
## 1) Visualizing training data (Data1.xlsx)



**Figure 1 – Visualizing training dataset**

# Solution to Question 2

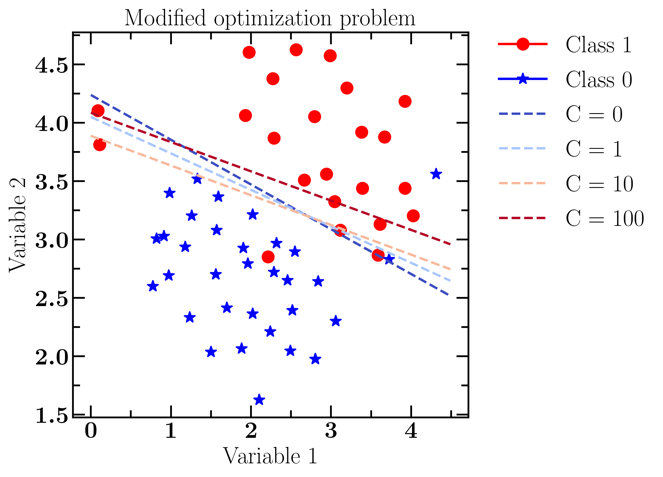
## 2.) Training data with decision boundary obtained by running vanilla version of SVM



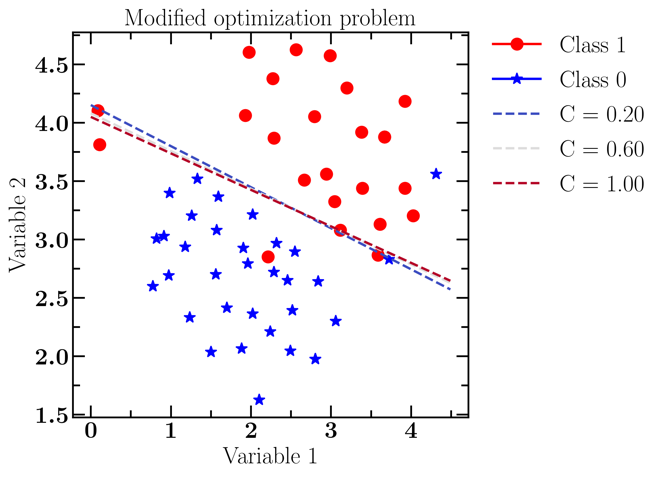
**Figure 2 - SVM Result**

# Solution to Question 3

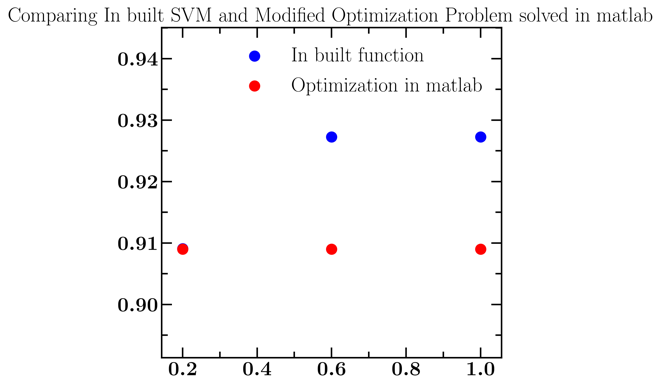
## 3.) Modified Optimization Problem



**Figure 3a – Comparing effect of C**



**Figure 3b – Comparing effect of C ≤ 1**



**Figure 3c – Comparing effect of C ≤ 1**

It has been observed from Figure 3a and 3b that number of misclassified points decreases when we enforce the optimization part strongly i.e. use **large C value.**

Also, we made an attempt in comparing our own optimization code and inbuilt SVM, the results hardly have a difference of 0.01 in accuracy which is equal to total correctly classified points/total number of points.

# Solution to Question 4

## 4.) Naïve Bayes Classification

Pre-requisites:

|  |  |
| --- | --- |
| Probability | Value |
| P(stolen = yes) | 0.5 |
| P(stolen = no) | 0.5 |
| P(red | stolen = yes) | 0.6 |
| P(red | stolen = no) | 0.4 |
| P(SUV | stolen = yes) | 0.2 |
| P(SUV | stolen = no) | 0.6 |
| P(domestic | stolen = yes) | 0.4 |
| P(domestic | stolen = no) | 0.6 |

**Hence,**

**P(stolen = yes | (Color = red, Type = SUV, Origin = Domestic)**

**= P(red | stolen=yes) x P(SUV | stolen=yes) x P(domestic | stolen=yes) x P(stolen=yes)**

**=**

**= 0.6 x 0.2 x 0.4 x 0.5**

**= 0.024**

**P(stolen = no | (Color = red, Type = SUV, Origin = Domestic)**

**= P(red | stolen=no) x P(SUV | stolen=no) x P(domestic | stolen=no) x P(stolen=no)**

**=**

**= 0.4 x 0.6 x 0.6 x 0.5**

**= 0.072**

By normalizing the above quantities to sum to one,

The conditional probability that the target value is no, given the observed attribute values =

The conditional probability that the target value is yes, given the observed attribute values =

Since, conditional probability of target value as No > conditional probability of target value as Yes

Thus, our example gets classified as ’NO’ or

The STOLEN status for Red Domestic SUV will be highly YES.

# Solution to Question 5

## 5.) K-Means classification

Iteration-1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Var1 | Var2 | Mean=x1 | Mean=y1 | Mean=x2 | Mean=y2 | D1 | D2 | CLASS |
| -1.54 | 2.29 | 0.5 | 1.5 | -4.5 | -5 | 2.19 | 7.87 | 1 |
| -0.44 | 2.34 | 0.5 | 1.5 | -4.5 | -5 | 1.26 | 8.39 | 1 |
| 0.03 | 0.41 | 0.5 | 1.5 | -4.5 | -5 | 1.19 | 7.06 | 1 |
| 1.2 | 1.87 | 0.5 | 1.5 | -4.5 | -5 | 0.79 | 8.93 | 1 |
| 0.65 | 2.39 | 0.5 | 1.5 | -4.5 | -5 | 0.90 | 9.01 | 1 |
| -4.67 | -4.8 | 0.5 | 1.5 | -4.5 | -5 | 8.15 | 0.26 | 2 |
| -3.37 | -5.41 | 0.5 | 1.5 | -4.5 | -5 | 7.92 | 1.20 | 2 |
| -3.93 | -4.64 | 0.5 | 1.5 | -4.5 | -5 | 7.57 | 0.67 | 2 |
| -4.78 | -4.96 | 0.5 | 1.5 | -4.5 | -5 | 8.34 | 0.28 | 2 |
| -4.12 | -5.36 | 0.5 | 1.5 | -4.5 | -5 | 8.27 | 0.52 | 2 |

Iteration-2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Var1 | Var2 | Mean=x1 | Mean=y1 | Mean=x2 | Mean=y2 | D1 | D2 | CLASS |
| -1.54 | 2.29 | -0.02 | 1.86 | -4.17 | -5.03 | 1.58 | 7.78 | 1 |
| -0.44 | 2.34 | -0.02 | 1.86 | -4.17 | -5.03 | 0.64 | 8.27 | 1 |
| 0.03 | 0.41 | -0.02 | 1.86 | -4.17 | -5.03 | 1.45 | 6.88 | 1 |
| 1.2 | 1.87 | -0.02 | 1.86 | -4.17 | -5.03 | 1.22 | 8.75 | 1 |
| 0.65 | 2.39 | -0.02 | 1.86 | -4.17 | -5.03 | 0.85 | 8.85 | 1 |
| -4.67 | -4.8 | -0.02 | 1.86 | -4.17 | -5.03 | 8.12 | 0.55 | 2 |
| -3.37 | -5.41 | -0.02 | 1.86 | -4.17 | -5.03 | 8.00 | 0.89 | 2 |
| -3.93 | -4.64 | -0.02 | 1.86 | -4.17 | -5.03 | 7.59 | 0.46 | 2 |
| -4.78 | -4.96 | -0.02 | 1.86 | -4.17 | -5.03 | 8.32 | 0.61 | 2 |
| -4.12 | -5.36 | -0.02 | 1.86 | -4.17 | -5.03 | 8.30 | 0.33 | 2 |

Iteration-3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Var1 | Var2 | Mean=x1 | Mean=y1 | Mean=x2 | Mean=y2 | D1 | D2 | CLASS |
| -1.54 | 2.29 | -0.02 | 1.86 | -4.17 | -5.03 | 1.58 | 7.78 | 1 |
| -0.44 | 2.34 | -0.02 | 1.86 | -4.17 | -5.03 | 0.64 | 8.27 | 1 |
| 0.03 | 0.41 | -0.02 | 1.86 | -4.17 | -5.03 | 1.45 | 6.88 | 1 |
| 1.2 | 1.87 | -0.02 | 1.86 | -4.17 | -5.03 | 1.22 | 8.75 | 1 |
| 0.65 | 2.39 | -0.02 | 1.86 | -4.17 | -5.03 | 0.85 | 8.85 | 1 |
| -4.67 | -4.8 | -0.02 | 1.86 | -4.17 | -5.03 | 8.12 | 0.55 | 2 |
| -3.37 | -5.41 | -0.02 | 1.86 | -4.17 | -5.03 | 8.00 | 0.89 | 2 |
| -3.93 | -4.64 | -0.02 | 1.86 | -4.17 | -5.03 | 7.59 | 0.46 | 2 |
| -4.78 | -4.96 | -0.02 | 1.86 | -4.17 | -5.03 | 8.32 | 0.61 | 2 |
| -4.12 | -5.36 | -0.02 | 1.86 | -4.17 | -5.03 | 8.30 | 0.33 | 2 |

Center of Cluster 1 = (-0.02,1.86)

Center of Cluster 2 = (-4.17,-5.03)