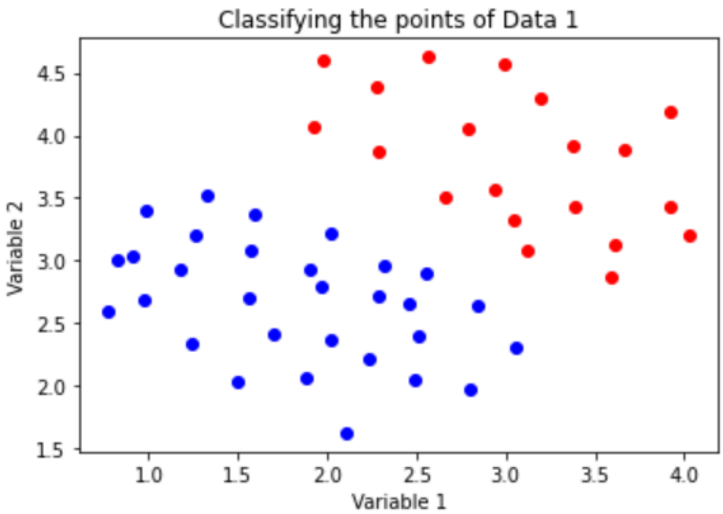
CLL 788 Assignment 3 Report

Aditi Singh
2017CH10188

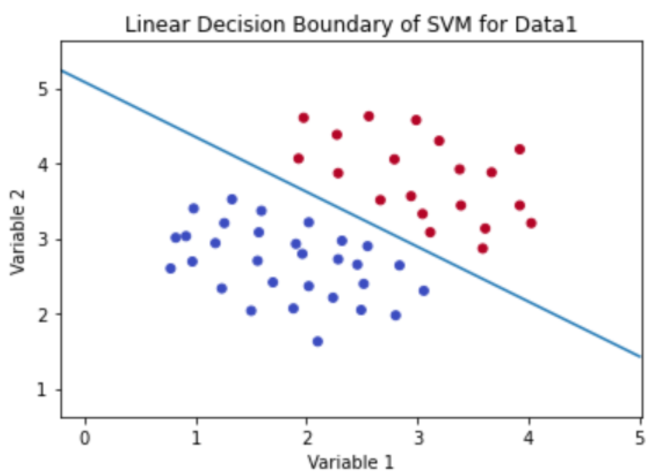
Q1. Plot the training data (Data1.xlsx) to get an idea of the data distribution. Plot the points with variable 1 on x-axis and variable 2 on y-axis. Now color the coordinates/points of class -1 with blue and class 1 with red. Report your visual observations.



Red Points denote the class 1 while blue shows class -1, from the plot it is clearly evident that the points are linearly separable.

Q2. Apply SVM on training data (Data1.xlsx) to find the decision boundary. Plot training data along with decision boundaries.

Used the following parameters: **lambda = 0.001, alpha = 0.001**
Values of weights and biases obtained after fitting the Data 1 in SVM code:
weight_0 : 1.78629621
weight_1: 2.44439214
bias: -12.421999999998555
For checking accuracy, I had split up my data into test (30%) and training (70%) and fit the training data and predicted test data with accuracy of **100% (15/15)** labels correctly predicted.
Using the above weights and biases the decision boundary plotted separating the classes is:



As one can clearly see that the linear boundary separates the dataset with classes with the best possible margin thus my SVM Code is working fine.

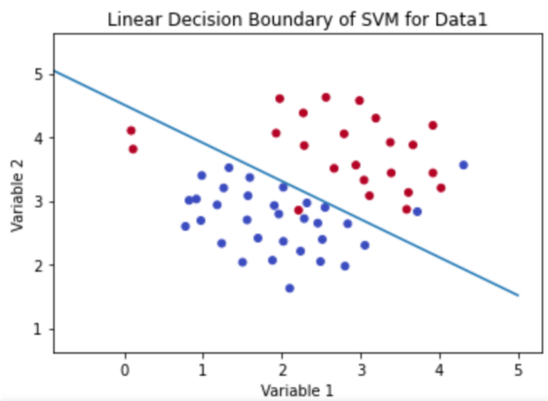
Q3. Now apply SVM with "modified optimization problem" on Data2.xlsx and try out different values of C and report your observations along with plots of the decision boundary.

The C values used were : **[0.05, 0.1, 1, 2, 5, 10, 20, 100]**
For testing the accuracy, I split up the data set into test (25%) and train (75%), the following accuracy for test set and weights & biases were obtained :

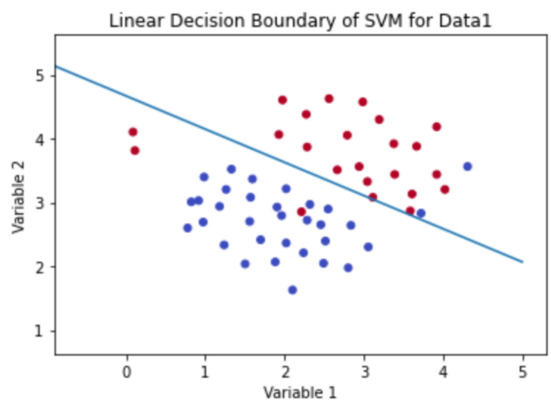
	C	b	weight[0]	weight[1]	accuracy of Test Set
0	0.05	-3.523260	0.467325	0.781775	0.470588
1	0.10	-4.839949	0.539149	1.036869	0.647059
2	1.00	-8.069433	0.723772	1.948405	0.882353
3	2.00	-8.909258	0.724835	2.183307	0.941176
4	5.00	-10.601917	0.742771	3.117461	0.941176
5	10.00	-10.718513	0.730520	3.173460	0.941176
6	20.00	-10.874339	0.740234	3.215659	0.941176
7	100.00	-12.881877	0.957892	3.643562	0.941176

Below are the Linear Decision boundary for Optimised SVM with varying C values:

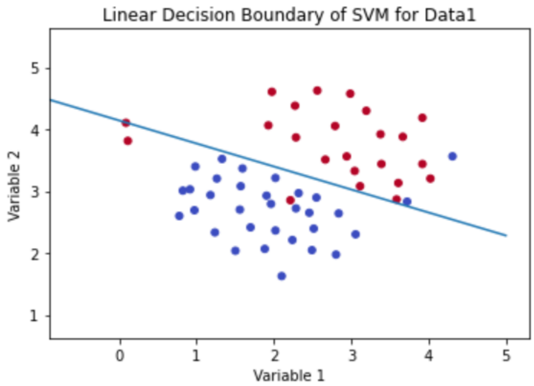
The decision Boundary for C = 0.05



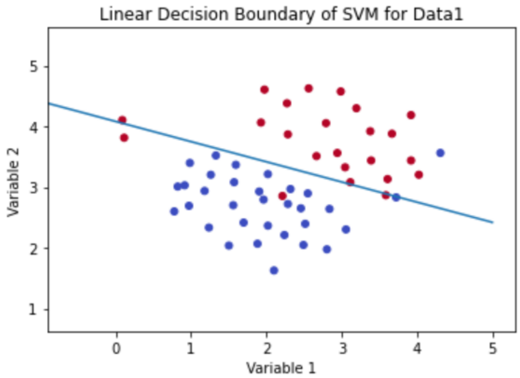
The decision Boundary for C = 0.1



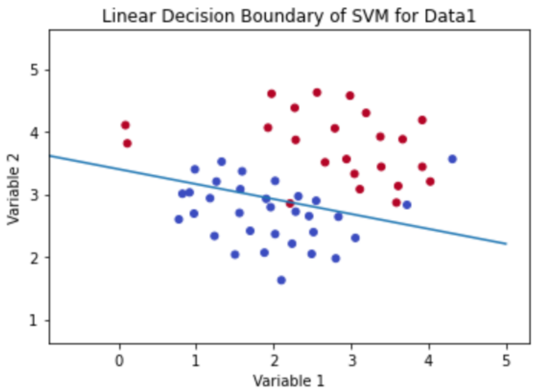
```
The decision Boundary for C = 1
```



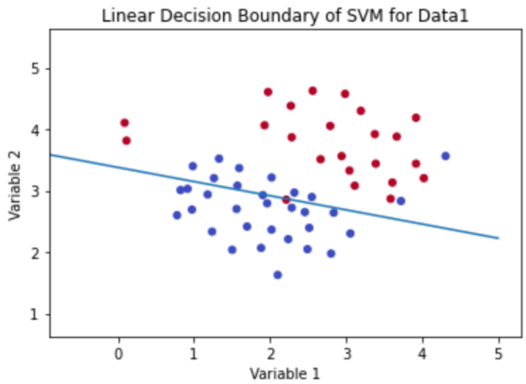

```
The decision Boundary for C = 2
```



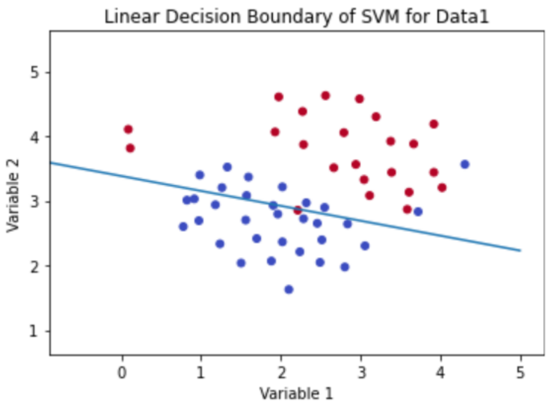

```
The decision Boundary for C = 5
```



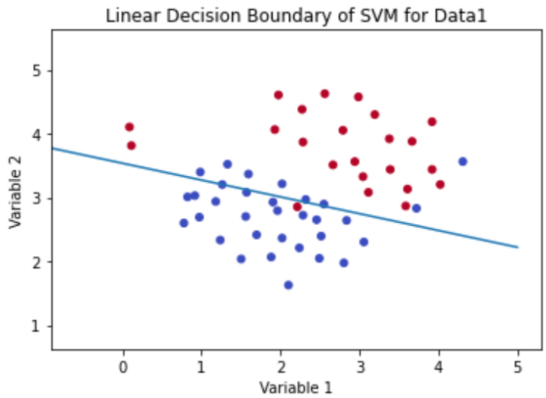

```
The decision Boundary for C = 10
```




```
The decision Boundary for C = 20
```




```
The decision Boundary for C = 100
```



Accuracy is low for C=0.05 and 0.1 but from then on the accuracy improves too, accuracy for **C=2,5,10,20,100 is around 95%** now from **decision boundary** plots we can see that for **C=0.05,0.1,1,2** plots are separating the classes in most efficient manner thus on combining both the aspects (accuracy and decision boundary) one can see the best results are with **C=2**

Q4. Classify Red Domestic SUV using Naïve Bayes classifier manually

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

We have to classify Red SUV Domestic
Yes(1): P(Stolen=Yes)*P(Red|Yes)*P(Domestic|Yes)*P(SUV|Yes)
No(0): P(Stolen=No)*P(Red|No)*P(Domestic|No)*P(SUV|No)
P(Stolen=Yes) =5/10=0.5
P(Stolen=No) =5/10=0.5

YES	NO
P(Red) =3/5 =0.6	P(Red) = 2/5 =0.4
P(Domestic)= 2/5 =0.4	P(Domestic)= 3/5 =0.6
P(SUV)= 1/4 =0.25	P(SUV)= 3/4 =0.75
P(X|Yes) = 0.5*0.6*0.4*0.25 = 0.03	P(X|No) = 0.5*0.4*0.6*0.75 = 0.09
P(Yes|Red SUV Domestic) = (0.03)/(0.03+0.09) = **0.25**	P(No|Red SUV Domestic) = (0.09)/(0.03+0.09) = **0.75**

Thus the chances are that a Red SUV Domestic car will **not be stolen** according to Naive Bayes Algorithm. As **P(No|Red SUV Domestic) = 0.75**