University of Tehran

College of Engineering

School of Electrical and Computer Engineering

Intelligent Systems Course
Computer Assignment #1

Under the Supervision of: Dr. Hosseini

Golmehr Khosrokhavar

Student ID: 810198507

Khosrokhavar.g@gmail.com

Jan 2023

Contents

Question 1: Binary logistic regression

It is suggested to write a summary of the process of solving the question, its purpose and its requirements in this section. In your report, write Persian texts using B Nazanin font and English texts using Times New Roman font.

Part A:

$$J(w,b)=\frac{1}{n}\sum_{i=1}^{n}\log\left\{1+exp\left(-y_i\left(b+x_i^T w\right)\right)\right\}$$

$$\mu_i(w,b)=\frac{1}{1+exp\left(-y_i\left(b+x_i^T w\right)\right)}$$

$$\nabla_w=\frac{\partial J}{\partial w}=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{1+exp\left(-y_i\left(b+x_i^T w\right)\right)}*(-y_i)\,exp\left(-y_i\left(b+x_i^T w\right)\right)$$

$$\nabla_b=\frac{\partial J}{\partial b}=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{1+exp\left(-y_i\left(b+x_i^T w\right)\right)}*(-y_i x_i^T)\,exp\left(-y_i\left(b+x_i^T w\right)\right)$$

Rewrite:

$$\boxed{\begin{aligned}\boldsymbol{\nabla_w}&=\frac{\boldsymbol{\partial J}}{\boldsymbol{\partial w}}=\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{\mu_i(w,b)}*\boldsymbol{(-y_i)}\,\boldsymbol{exp}\left(\boldsymbol{-y_i\left(b+x_i^T w\right)}\right)\\ \boldsymbol{\nabla_b}&=\frac{\boldsymbol{\partial J}}{\boldsymbol{\partial b}}=\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{\mu_i(w,b)}*\boldsymbol{(-y_i x_i^T)}\,\boldsymbol{exp}\left(\boldsymbol{-y_i\left(b+x_i^T w\right)}\right)\end{aligned}}$$

Part B & C:

Calculation of descending gradient and calculation of accuracy:

Considering the initial values of zero and Learning rate = [0.005, 0.01, 0.1, 1], the graph of changes of J in terms of the number of repetitions per training data is drawn below.

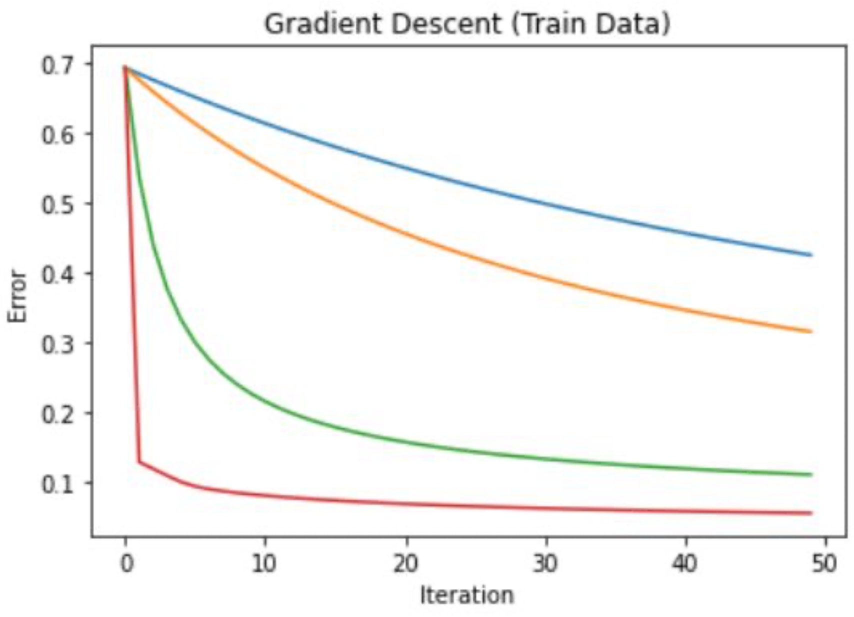
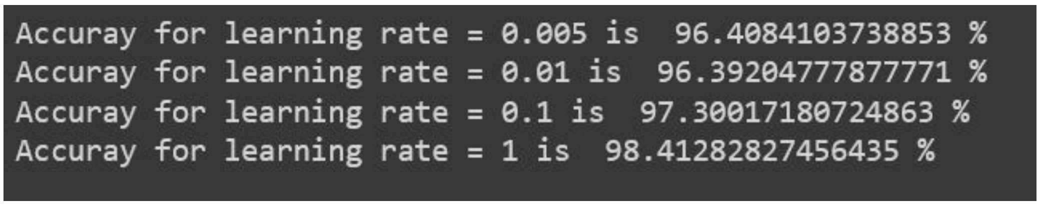


Figure 1-1

The percentage of correctness and accuracy of this implementation for different learning rates will be as below:



```
Accuray for learning rate = 0.005 is  96.4084103738853 %
Accuray for learning rate = 0.01 is  96.39204777877771 %
Accuray for learning rate = 0.1 is  97.30017180724863 %
Accuray for learning rate = 1 is  98.41282827456435 %
```

Figure 2-1

The graph of changes of J according to the number of repetitions per test data is drawn below:

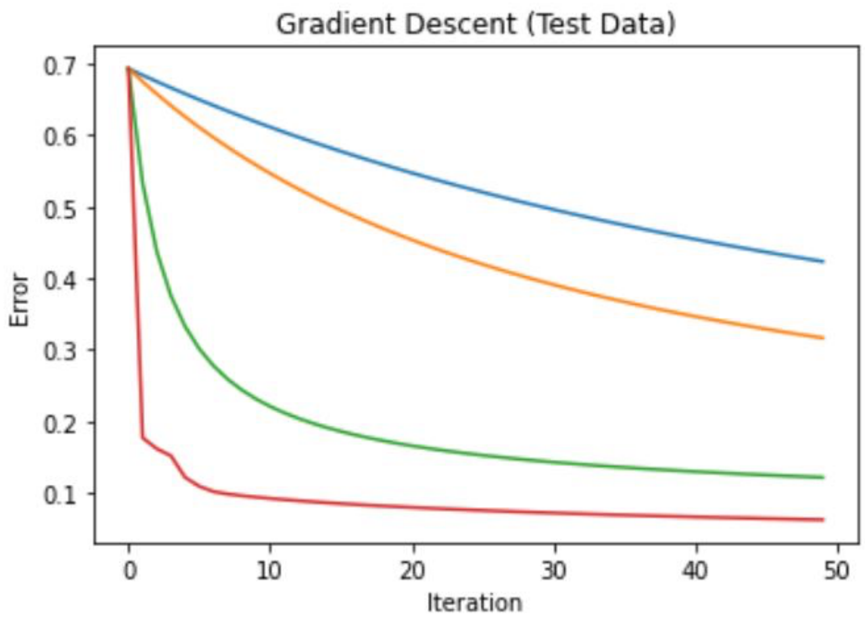
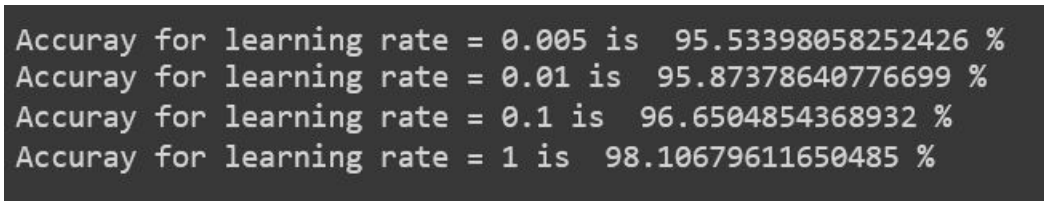


Figure 3-1

The percentage of correctness and accuracy of this implementation for different learning rates will be as follows:



```
Accuray for learning rate = 0.005 is  95.53398058252426 %
Accuray for learning rate = 0.01 is  95.87378640776699 %
Accuray for learning rate = 0.1 is  96.6504854368932 %
Accuray for learning rate = 1 is  98.10679611650485 %
```

Figure 4-1

Part C: Stochastic Gradient Descent

At first, for batch size = 1, we will have:

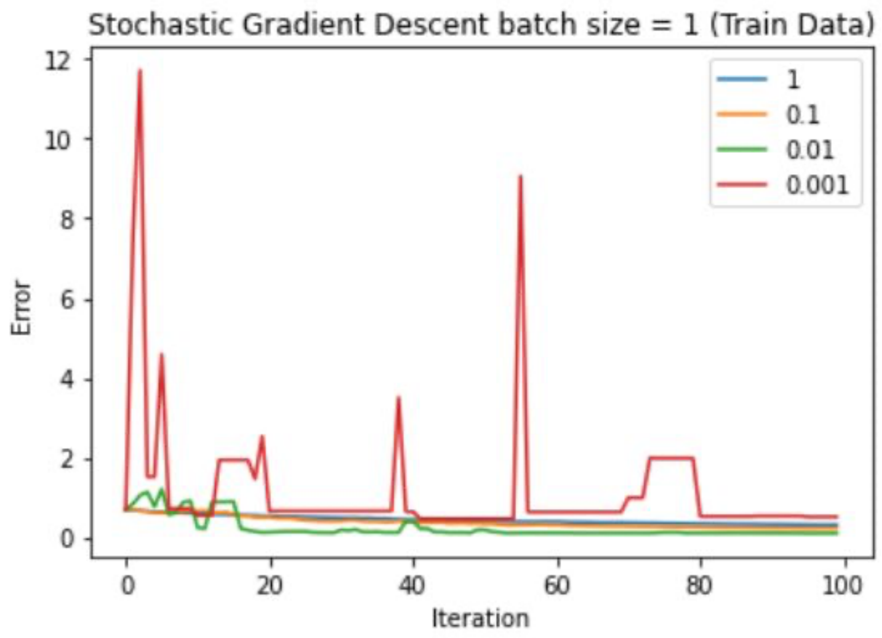
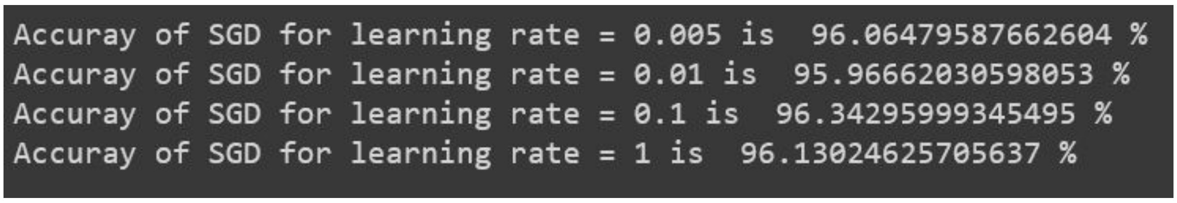


Figure 5-1

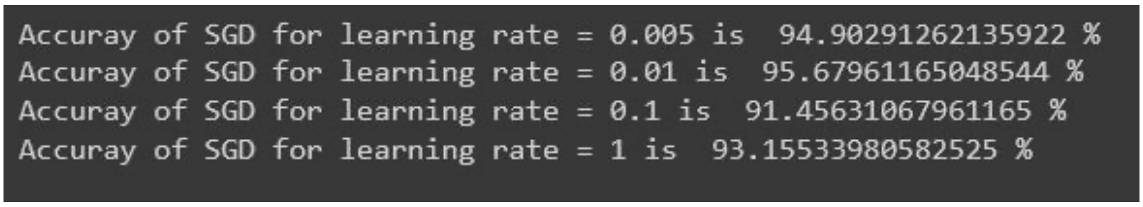


```
Accuray of SGD for learning rate = 0.005 is  96.06479587662604 %
Accuray of SGD for learning rate = 0.01 is  95.96662030598053 %
Accuray of SGD for learning rate = 0.1 is  96.34295999345495 %
Accuray of SGD for learning rate = 1 is  96.13024625705637 %
```

Figure 6-1

Then we repeat the same steps for the test data.

The percentage of correctness and accuracy of the answers of the test data are as follows:



```
Accuray of SGD for learning rate = 0.005 is  94.90291262135922 %
Accuray of SGD for learning rate = 0.01 is  95.67961165048544 %
Accuray of SGD for learning rate = 0.1 is  91.45631067961165 %
Accuray of SGD for learning rate = 1 is  93.15533980582525 %
```

Figure 7-1

Next, we repeat the steps for batch size = 100:

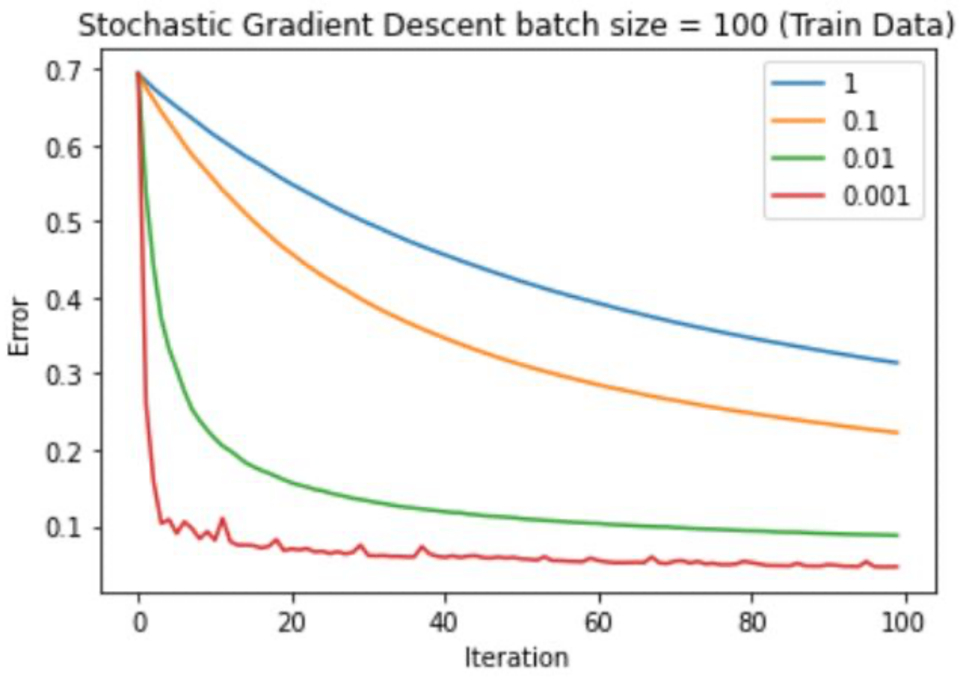
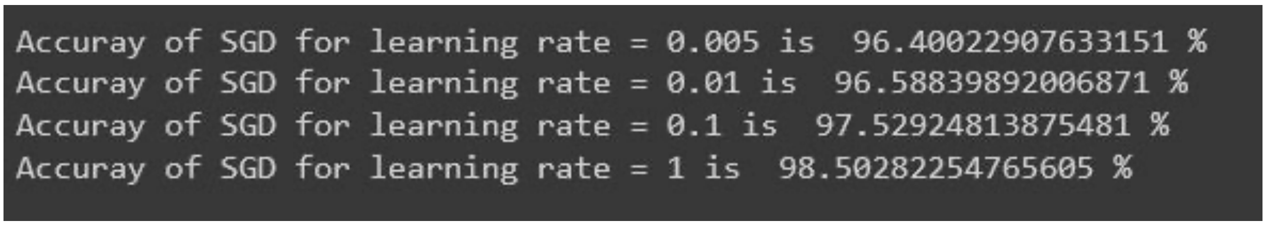


Figure 8-1



```
Accuray of SGD for learning rate = 0.005 is  96.40022907633151 %
Accuray of SGD for learning rate = 0.01 is  96.58839892006871 %
Accuray of SGD for learning rate = 0.1 is  97.52924813875481 %
Accuray of SGD for learning rate = 1 is  98.50282254765605 %
```

Figure 9-1

Question 2: Optimization in non-convex functions

Part A: Newton's method (analytical)

$$f(x) = 2x_1^2 + 2x_2^2 - 17x_2\cos(0.2\pi x_1) - x_1x_2$$

- $\frac{\partial f}{\partial x_1} = 4x_1 + 17x_2 * 0.2\pi * \sin(0.2\pi x_1) - x_2 = 4x_1 + 3.4\pi x_2\sin(0.2\pi x_1) - x_2$
- $\frac{\partial f}{\partial x_2} = 4x_2 - 17\cos(0.2\pi x_1) - x_1$
- $\frac{\partial^2 f}{\partial x_1^2} = 4 - 3.4\pi x_2 * 0.2\pi\cos(0.2\pi x_1) = 4 - 0.68\pi^2 x_2\cos(0.2\pi x_1)$
- $\frac{\partial^2 f}{\partial x_2^2} = 4$

We start from the point (0,0) and according to Newton's method we will have:

$$x_{1_0} = 0\,,\ x_{2_0} = 0$$

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

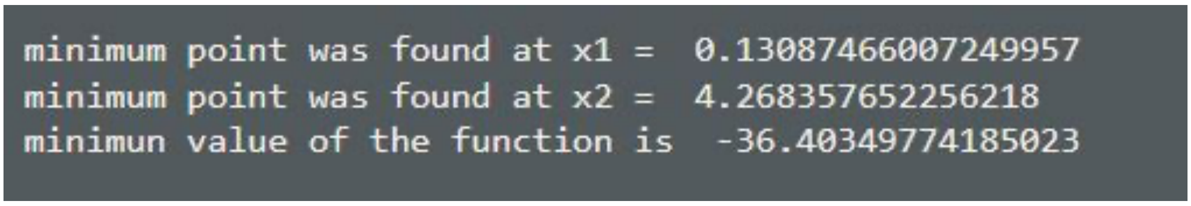
$$x_{1_1} = x_{1_0} - \frac{f'(x_{1_0})}{f''(x_{1_0})} = 0 - \frac{f'(0)}{f''(0)} = 0$$

$$x_{2_1} = x_{2_0} - \frac{f'(x_{2_0})}{f''(x_{2_0})} = 0 - \frac{f'(0)}{f''(0)} = 0 - \frac{-17}{4} = 4.25$$

Part B: Newton's method (simulation)

In this section, with the help of Newton's method and genetic algorithm, we want to find the minimum point for the given function.

The coordinates of the minimum point are found and the value of the function at that point is determined in the following figure:



```
minimum point was found at x1 =  0.13087466007249957
minimum point was found at x2 =  4.268357652256218
minimun value of the function is  -36.40349774185023
```

Figure 2-1

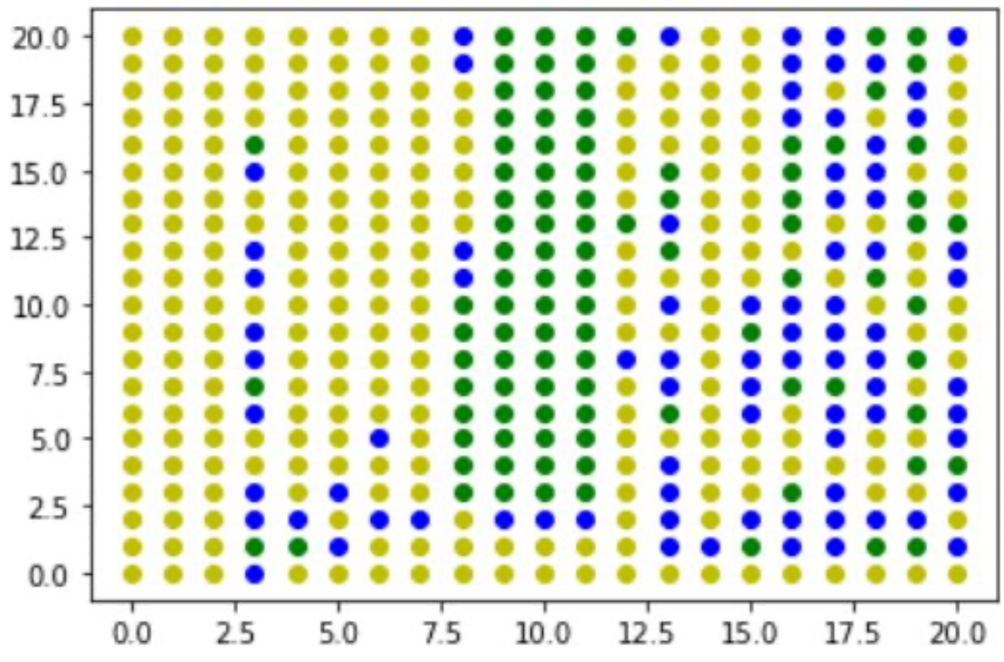
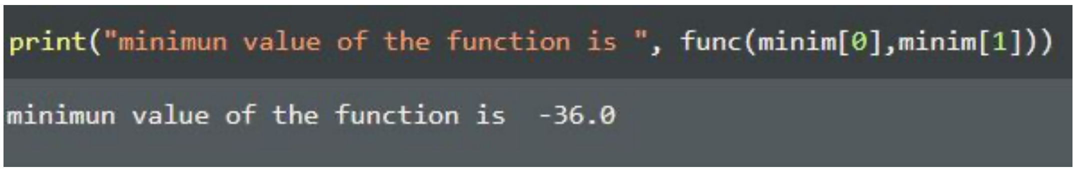


Figure 2-2

Part C: meta-heuristic method



```
print("minimun value of the function is ", func(minim[0],minim[1]))
```

```
minimun value of the function is  -36.0
```

Figure 2-3

As can be seen from the image above, the minimum value of this function calculated by the genetic algorithm method is close to the answer of part b with an acceptable approximation.

Question 3: Support vector machine

Part A: Analytical

1- Consider the following formulation for the support vector machine classification:

$$\boxed{\begin{array}{c} \textit{Minimize}\ \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ \textit{subject to}\ \ \xi_i = \max\big(0, 1 - y_i(W^T x_i + b)\big) \end{array}}$$

What result can be obtained if the value of ξ_i becomes zero for a sample?

To minimize the margin, we must satisfy the following condition:

$$\boxed{y_i(W^T x_i + b) \geq\ +1}$$

which is equivalent to the following condition:

$$\boxed{1 - y_i(W^T x_i + b) \geq\ 0}$$

Practically, ξ_i becoming zero means that when calculating

$\max\big(0, 1 - y_i(W^T x_i + b)\big)$, the value of the term $1 - y_i(W^T x_i + b)$ was less than zero Is. This means that the above condition is not fulfilled.

This means that some data values are placed between the marginal lines H_1 and H_2

Suppose we have three support vectors $x_1 = (1,0),\ x_2(2,1),\ x_3 = (2,2)$

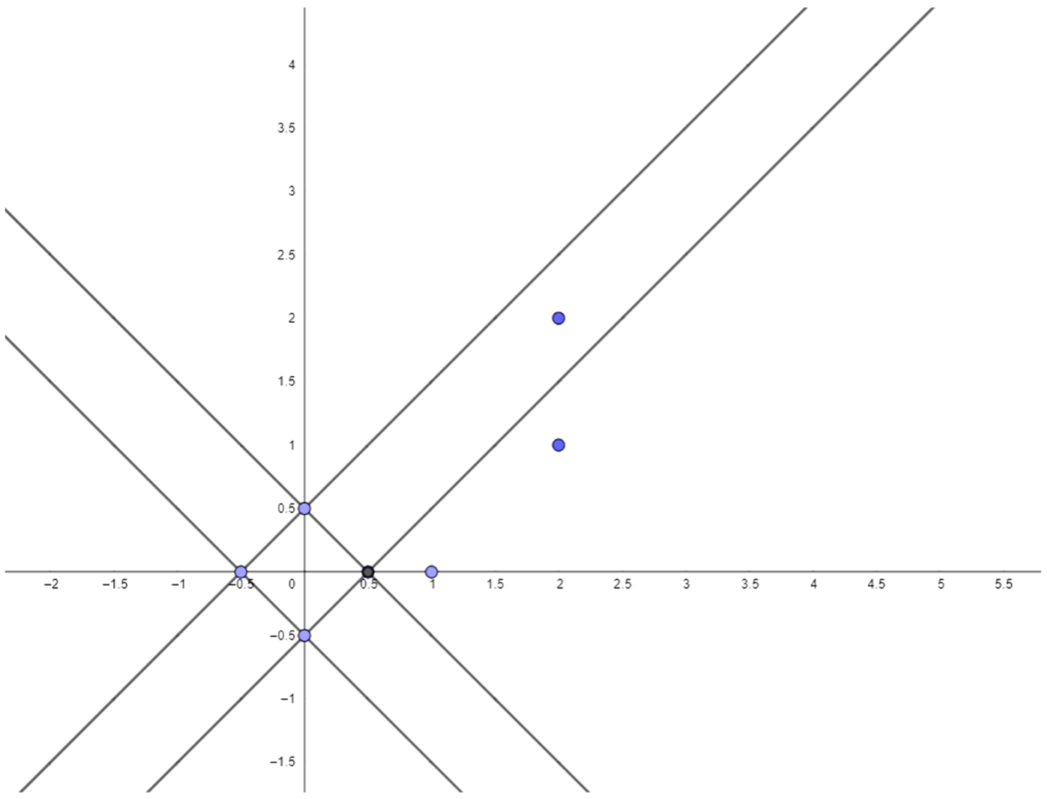


Figure 2-4

As it is clear from the above coordinates, the optimal decision matrix for separating the positive and negative classes from each other is the third case.

The line $y = x - 0.5$ conveniently separates positive and negative class points from each other.

In the case of other lines, all the defects fall on one side of them, and in Saman, there is not even a proper margin on both sides of the dividing line.

Part B: Implementation

In this part, we are going to classify a data set with the help of support vector machine.

The data distribution will initially be as follows:

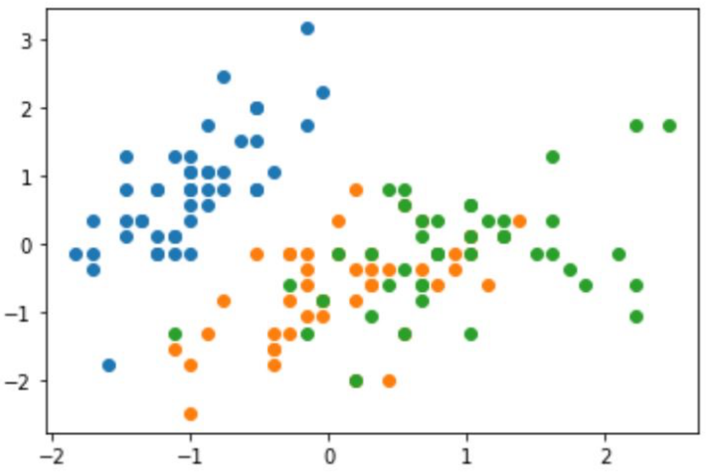
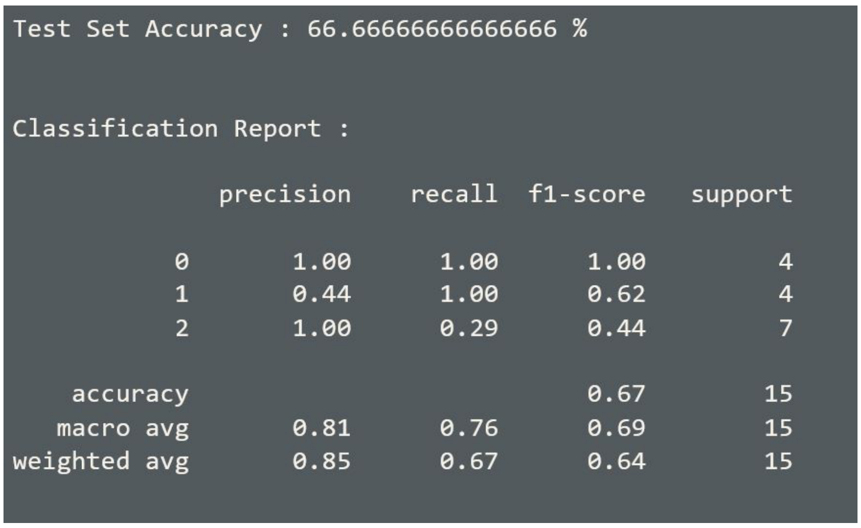


Figure 2-5

Then by implementing the support vector machine using method one in comparison with the others, we define different areas. The accuracy and details of this classification are as follows:



```
Test Set Accuracy : 66.66666666666666 %


Classification Report :

              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       0.44      1.00      0.62         4
           2       1.00      0.29      0.44         7

    accuracy                           0.67        15
   macro avg       0.81      0.76      0.69        15
weighted avg       0.85      0.67      0.64        15
```

Figure 2-6

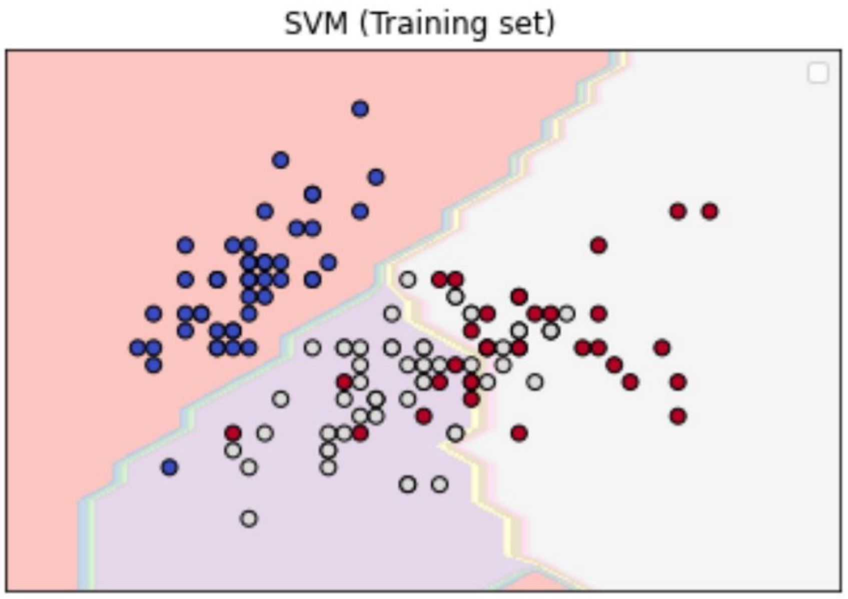


Figure 2-7

The confusion matrix will be as shown below:

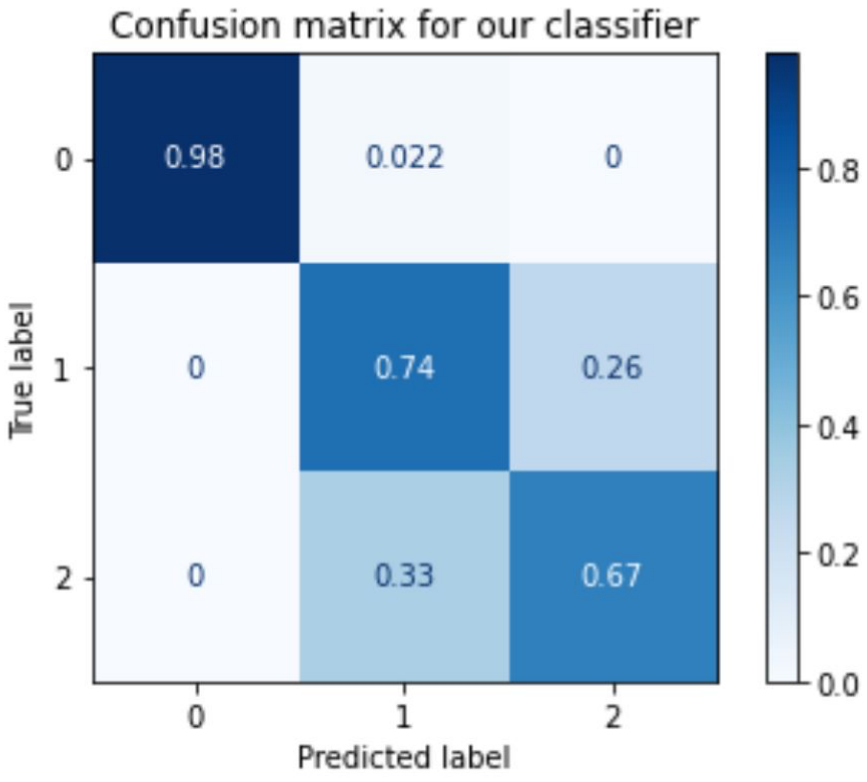


Figure 2-8

References:

- https://scikit-learn.org/stable/modules/sgd.html
- https://towardsdatascience.com/complete-step-by-step-genetic-algorithm-from-scratch-for-global-optimization-6fee5c55dd3b
- https://towardsdatascience.com/an-overview-of-the-gradient-descent-algorithm-8645c9e4de1e