Git_hub link:
https://github.com/10danish/ML_Assgn_2

**ML Assignment-2**
**Simple Logistic Regression**

Danish Imroz Khan
21051125

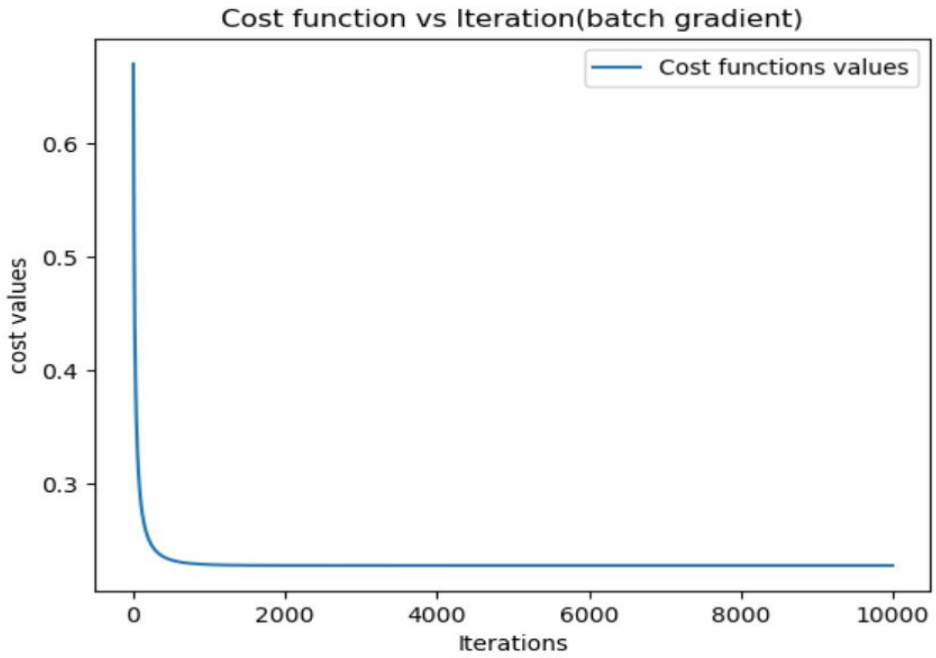
**Q1)Use logistic regression to find decision boundary For the given database. Set your learning rate to 0.1. What is the cost function value and learning parameter value after convergence?**

**Ans:**

```
Theta0: 0.4012529278579288, Theta1: 2.58854709562992, Theta2: -2.725587813835302
Final Cost: 0.22834144984473004
```

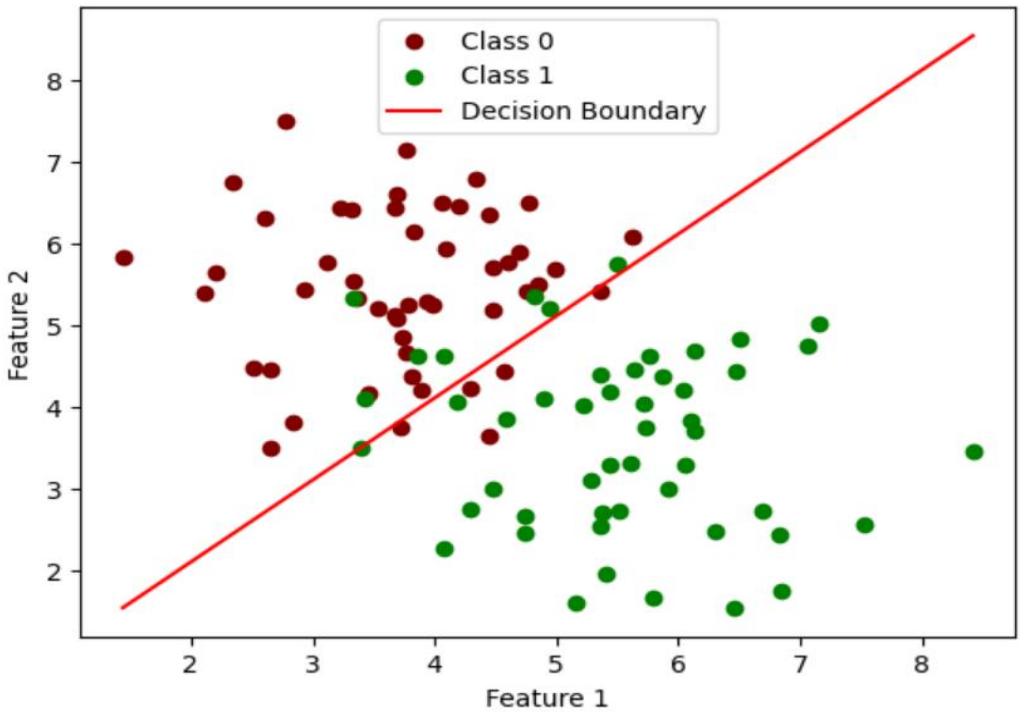
**Q2) Plot cost function v/s iteration graph for the model trained in question 1.**

**Ans:**



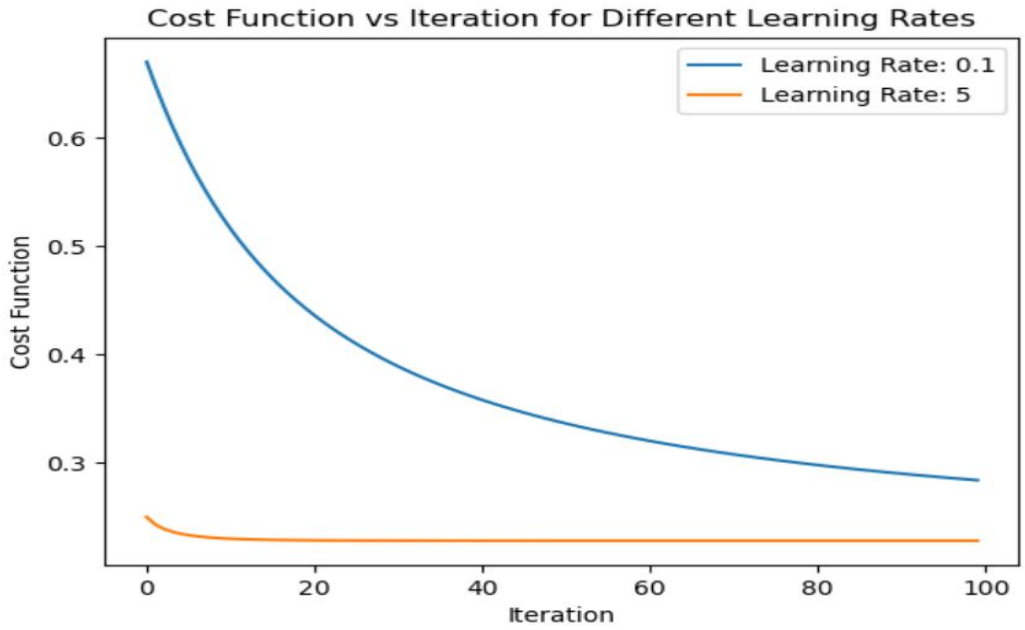

**Q3)Plot the given dataset on a graph, use different colours for different classes and also show the decision boundary you obtained in question 1.**

**Ans:**



**Q4)** **Train your model with a learning rate of 0.1 and 5. Plot the cost-function v/s iteration curve for both learning rates on the same graph. For this task, only train your model for 100 iterations.**

**Ans:**




**Q5)** **Find the confusion matrix for your training dataset. Using the confusion matrix to calculate the accuracy, precision, recall, F1-score.**

**Ans: For alpha=0.1**

```
predictions = (sigmoid(th0 + th1 * X1 + th2 * X2) >= 0.5).astype(int)

# Calculate confusion matrix
true_negatives = np.sum((predictions == 0) & (Y == 0))
false_positives = np.sum((predictions == 1) & (Y == 0))
false_negatives = np.sum((predictions == 0) & (Y == 1))
true_positives = np.sum((predictions == 1) & (Y == 1))

conf_matrix = np.array([[true_negatives, false_positives],
                        [false_negatives, true_positives]])

# Calculate metrics
accuracy = (true_positives + true_negatives) / len(Y)
precision = true_positives / (true_positives + false_positives)
recall = true_positives / (true_positives + false_negatives)
f1 = 2 * (precision * recall) / (precision + recall)

# Print results
print("Confusion Matrix:")
print(conf_matrix)
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```

```
Confusion Matrix:
[[45  5]
 [ 7 43]]
Accuracy: 0.88
Precision: 0.8958333333333334
Recall: 0.86
F1-score: 0.8775510204081632
```