# ML Assignment - 2

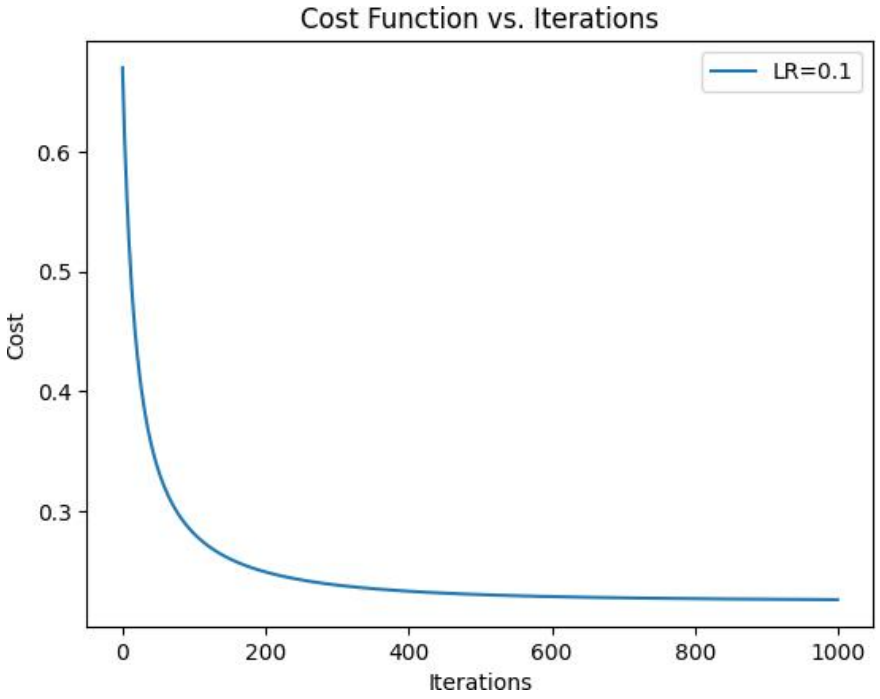
**Name - JIT DAS**
**Roll - 2206095**
**Section - IT-06**

**1. 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 : The cost function value in learning rate 0.1 is **0.2261**.
The learning parameter value after convergence is **[[ 0.38193936]**
**[ 2.35449296]**
**[-2.53913828]]**

**2. Plot cost function v/s iteration graph for the model trained in question?**

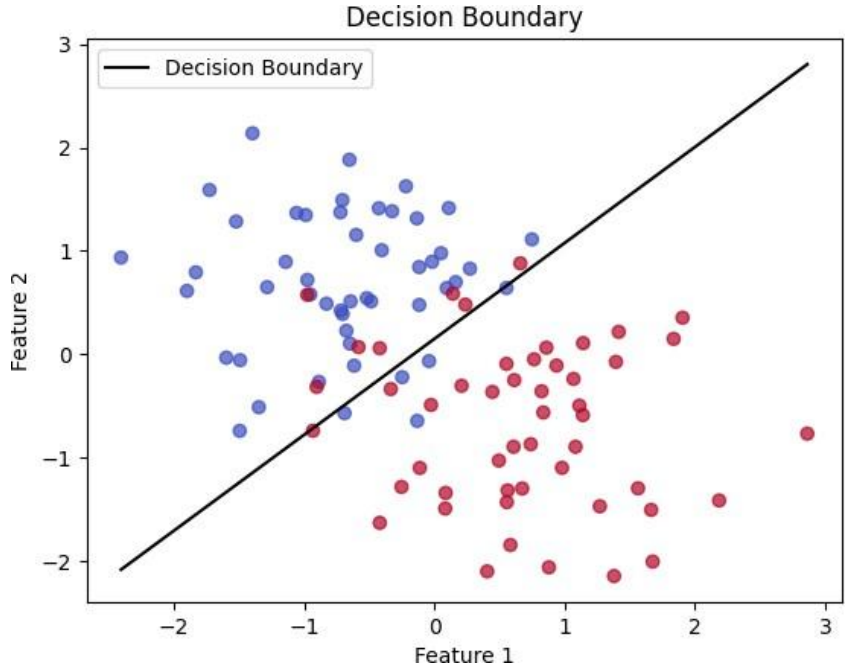
**Ans :** The graph shows the cost function vs. iterations for logistic regression with a learning rate of 0.1 :




- The cost starts high (~0.7) and rapidly decreases in the early iterations.
- Around 200-300 iterations, the cost stabilizes, indicating convergence.
- After 600+ iterations, the cost remains almost constant, meaning further training does not significantly improve the model.

**3. Plot the given dataset on a graph, use different colours for different classes and also show the decision boundary you obtained in question 1. Do not use scatter plot.**

**Ans :** The graph shows the decision boundary for logistic regression :

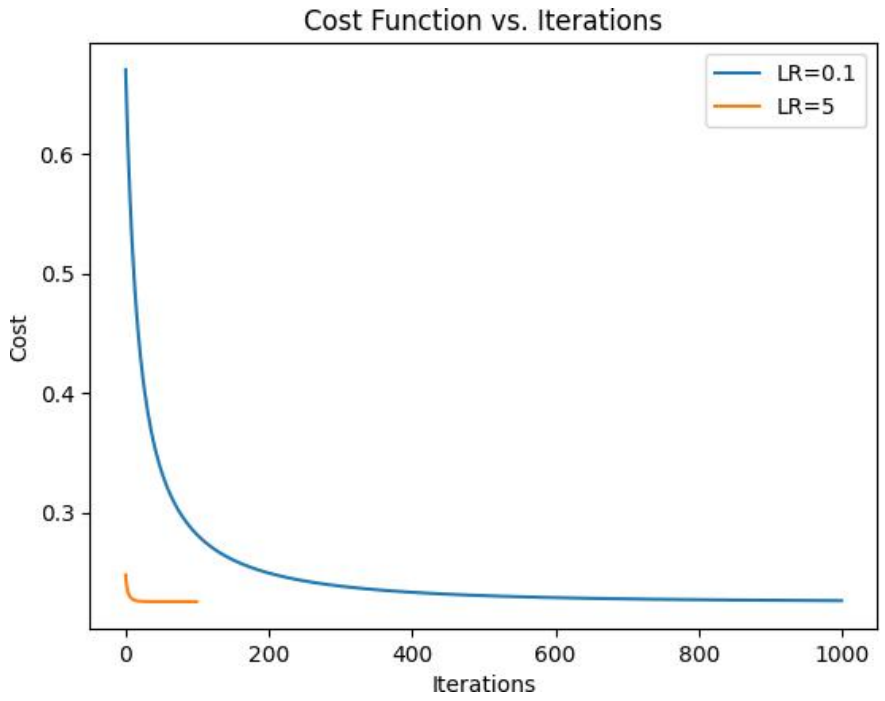



- The blue points represent one class, while the red points represent the other.
- The black line is the decision boundary, which separates the two classes based on the logistic regression model.
- Some points are misclassified (i.e., a few red points are on the blue side and vice versa), but overall, the boundary effectively separates the two groups.

**4. 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 :** The graph compares the cost function vs. iterations for two different learning rates: 0.1 (blue) and 5 (orange) :

- For LR = 0.1 (blue), the cost function decreases smoothly and gradually converges.
- For LR = 5 (orange), the cost function starts much lower and converges very quickly within a few iterations.
- However, a very high learning rate (like 5) may risk overshooting the optimal solution or causing instability.




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

**Ans :**

- Confusion Matrix **: [ [44 5]**
  **[ 7 43] ]**
- Accuracy **: 0.8788**
- Precision **: 0.8958**
- Recall **: 0.8600**
- F1-score **: 0.8776**

The model performs well with a high accuracy and F1-score. Some false positives (5) and false negatives (7) exist, but the model is generally reliable.

**Github link :** https://github.com/JITDAS-HUB/Logistic-Regression