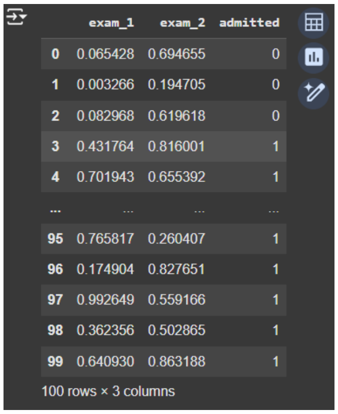
Logistic Regression Assignment Report

1. Loading Data and Preprocessing

Objective: Load the dataset and prepare it for logistic regression analysis.
Steps:
1. Load the data using pandas.
2. Normalize the features 'exam_1' and 'exam_2' to ensure they fall within the range [0, 1] using min-max normalization.
3. Store the original min and max values of the features for consistent normalization of test points.



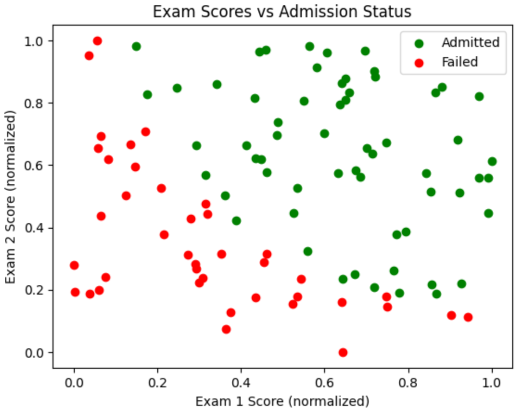
	exam_1	exam_2	admitted
0	0.065428	0.694655	0
1	0.003266	0.194705	0
2	0.082968	0.619618	0
3	0.431764	0.816001	1
4	0.701943	0.655392	1
...	...	...	...
95	0.765817	0.260407	1
96	0.174904	0.827651	1
97	0.992649	0.559166	1
98	0.362356	0.502865	1
99	0.640930	0.863188	1

100 rows × 3 columns

2. Visualization

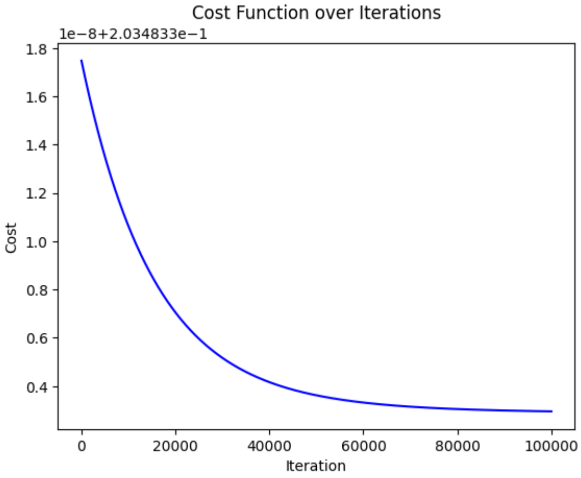
Objective: Visualize the dataset to understand the relationship between the exam scores and admission status.
Steps:
1. Separate the data points into two groups based on admission status (admitted or failed).
2. Plot 'exam_1' and 'exam_2' scores on a scatter plot, using different colors to represent the admitted and failed students.

Achieved Results:
The plot shows two groups of students, visualizing the correlation between exam scores and admission status.



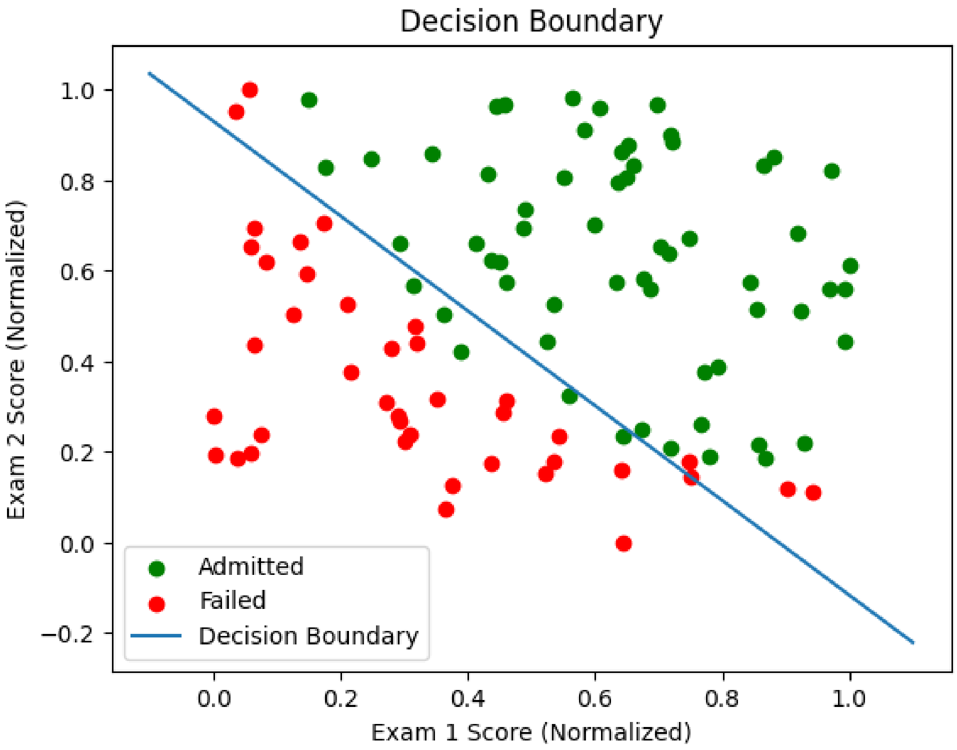
3. Logistic Regression Implementation from Scratch

Objective: Implement logistic regression from scratch to classify students based on their exam scores.
Steps:
1. Define the sigmoid function, which is the activation function for logistic regression.
2. Define the cost function to measure the model's performance and check how well it fits the data.
3. Implement gradient descent to minimize the cost function by iteratively adjusting the model parameters (theta).
4. Visualize the cost reduction over iterations and plot the decision boundary.



Achieved Results:
Training Accuracy: 89.00%
Student with scores (55, 70) - Predicted: 1, Expected: 1
Student with scores (40, 60) - Predicted: 0, Expected: 0

Decision Boundary Plot: Plot the decision boundary created by the logistic regression model.



4. Logistic Regression Using Library

Objective: Use scikit-learn to perform logistic regression and compare the results with the custom implementation.

Steps:

1. Initialize and fit the logistic regression model with 'exam_1' and 'exam_2' features.
2. Evaluate the model's accuracy on the training data.
3. Use the trained model to make predictions on specific test points and compare them with the expected results.

Achieved Results:

```
Training Accuracy using scikit-learn: 93.00%
Student with scores (55, 70) - Predicted: 1, Expected: 1
Student with scores (40, 60) - Predicted: 0, Expected: 0
```