Name: Nikhil Suresh
Email: nsuresh@clemson.edu

Midterm Project

Logistic Regression

Model

In this part of the project I have developed a Logistic Regression classifier for the given dataset. There are 3 parts to the model:

1. Without adding any new features.
2. Testing by adding all the features together.
3. Testing by adding one new feature at a time.

The below table talks about the accuracy of each of the model that was tested by altering the features:

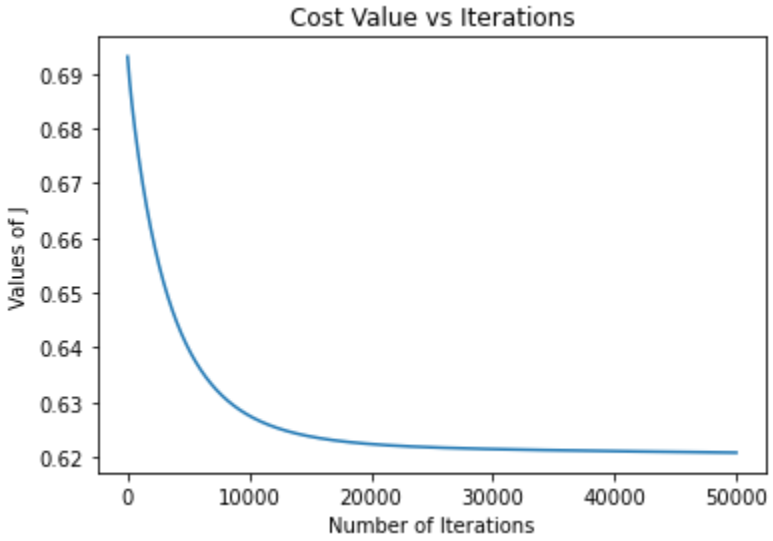
Features								Accuarcy %	Iterations
						x1	x2	51.51	10,000
					x1	x1 ** 2	x2	75.75	50,000
					x1	x2	x2 ** 2	66.66	50,000
				x1	x2	x1 ** 2	x2 ** 2	78.78	50,000
					x1	x2	x1.x2	33.33	100,000
					x1	x2	x1.x2 ** 2	39.39	100,000
					x1	x2	x1 ** 2.x2	45.45	100,000
					x1	x2	x1 ** 2. x2 ** 2	63.63	1,000,000
			x1	x2	x1 ** 2	x2 ** 2	x1 ** 2 * x2 ** 2	78.78	50,000
x1	x2	x1 ** 2	x2 ** 2	x1.x2 ** 2	x1 ** 2.x2	x1 ** 2 * x2 ** 2	x1.x2	72.72	10,000,000

We can see that the row highlighted in red with the features:

x1,x2, x1^2, x2^2 has the highest accuracy among all the tested models.

The model with the features:
x1,x2, x1^2, x2^2, x1^2.x2^2 has similar values but due to the presence of an extra feature the compute time is higher.

Initial Values
alpha = 0.01
weights = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]]
J = 0.6931



Max iterations = 50,000

Final values
alpha = 0.01
weights = [0.6127617 , 0.85311389, -1.60197502, -2.01023484, -2.0881704, -0.11393043]
J = 0.6208

Cost function value for test: [[0.55354638]]

Below is the performance metrics for the model:
Confusion Matrix

	Actual 1	Actual 0
Predicted 1	14	4
Predicted 0	3	12

```
Accuracy: 0.79
Precision: 0.78
Recall: 0.82
f1_score: 0.8
```

This model performs much better than the other models that we had tested during the process of training.
The model has an accuracy of 0.79 or 79%, and also displays good values across other performance metrics.

Note: I have attached a .html version of the Jupyter Notebook for reference.