ASSIGNMENT 2: LINEAR REGRESSION AND RIDGE REGRESSION

LAB REPORT

Gohil Happy

Roll No: 21IM30006

26th Aug 2023

Contents

ABSTRACT

This is report for MLFA lab assignment 3 by 21IM30006

1 Methodology

2 Experiment 1

Exploratory Data Analysis (EDA):
I conducted an EDA by creating a correlation graph and visually mapping the species related to the target column "Species."

Dataset Splitting:
To prepare the dataset, I developed a function to split it into training, validation, and test sets. This process ensured randomness and maintained data integrity.

Data Standard Scaling:
Data scaling was performed using a function that applied standard scaling methods to the dataset.

Softmax Function:
To enable multiclass classification, I implemented the softmax function, which transforms raw class scores into probabilities.

Parameter Initialization:
For initializing the parameters (weights) of the multiclass logistic regression model, I developed a function that considered the number of features and classes.

Bias Feature Addition:
An additional function was created to add a bias feature (intercept term) to the feature matrix.

Prediction:
To make predictions, I utilized the model parameters and the softmax function to predict class probabilities across multiple classes.

Gradient Descent:
Gradient descent was implemented to optimize the model parameters in multiclass logistic regression. This iterative process updated the weights to minimize the loss function, considering multiple classes. The outcome included the loss history over epochs and the updated weights.

Algorithm 1 Multiple-Class Logistic Regression Model

Input: Training data: X (feature matrix), y (target labels), where y is one-hot encoded Learning rate: α Number of classes: K Number of features: N Number of epochs: T Batch size: B Initialize weights: $W \in \mathbb{R}^{N \times K}$ t in 1 to T i in 1 to N (mini-batch sampling) Select a mini-batch of size B from X and corresponding one-hot encoded y Add a bias feature to X Compute the logits: $Z = X \cdot W$ Compute softmax probabilities for each example: $P = \text{softmax}(Z)$ Compute the gradient of the loss: $\nabla L = \frac{1}{B} X^T \cdot (P - y)$ Update weights: $W = W - \alpha \cdot \nabla L$ **Output:** Trained weights: W

2.1 Output values

Learning Rate (lr)	Accuracy
0.05	0.766667
0.10	0.766667
1×10e5	0.7533334
1×10e4	0.73333
1×10e3	0.73333
1×10e2	0.733333

Table 1: Optimal Learning Rates and Accuracies

3 Experiment 2

For this experiment, the training set was segregated into three distinct sets, each corresponding to one of the classes: 'setosa' (0), 'versicolor' (1), and 'virginica' (2). The weights stored for every epoch, using the optimal learning rate of 0.05, were accessed. For each of these three sets, the mean class probabilities were calculated for each epoch and plotted together in a single graph, as shown in Figure 2.

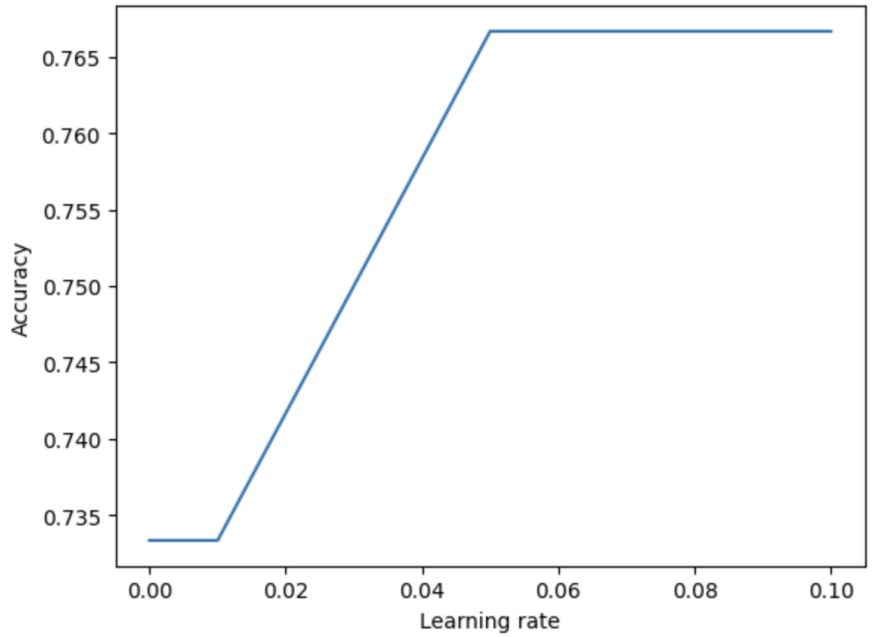

Figure 1: Lr vs Accuracy value

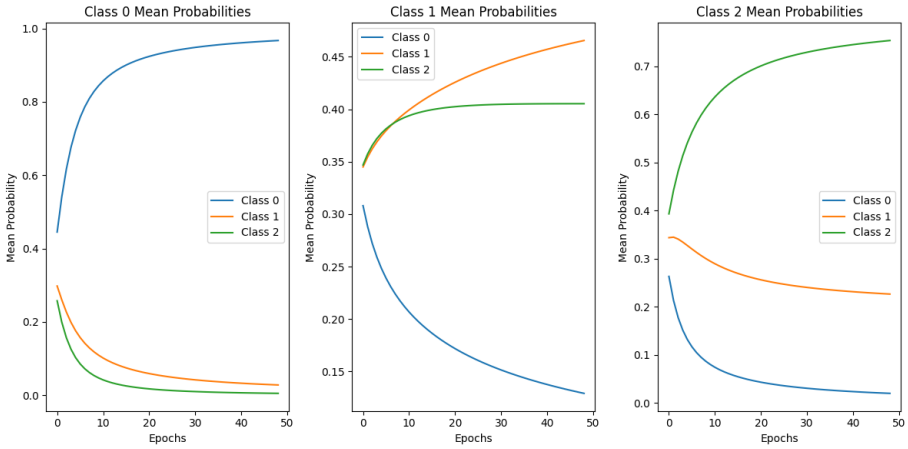

Figure 2: Mean Class Probabilities for Three Sets

4 Experiment 3

In this experiment, I collected classification reports that include details on precision, recall, and F1-scores for individual classes. Additionally, I acquired confusion matrices for both the validation and test datasets. Below, I provide summaries of the classification reports and their corresponding confusion matrices.

Table 2: Classification Report for Test Set

Class	Precision	Recall	F1-Score
0	0.875	0.9333	0.9032
1	1.0	0.8182	0.9
2	0.8182	0.8696	0.8438

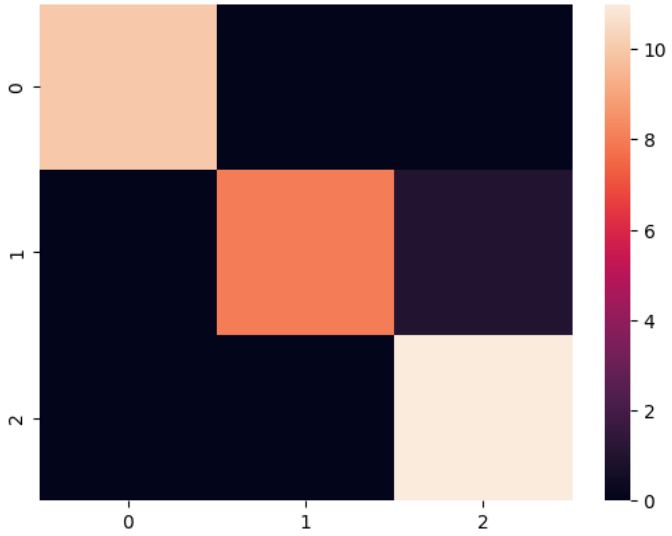

Figure 3: Confusion Matrix for test Set