Report

Name - Piyush Dubey

Roll no - 24NA10046

Model 1 - Logistic Regression

Logistic Regression is used for Binary Classification .
Logistic Regression starts with a Linear Equation

$$Z = \vec{x} . \vec{w} + b$$

Where w = [w_1 , w_2 , w_3 . . . .] and x = [x_1 , x_2 , x_3 . . . . .]
X = input feature
W = Weights
B = bias
Z = Raw Score

Sigmoid Function (Logistic Function)

To convert this raw score z into a **probability**, we apply the **sigmoid function**:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This Equation Squashes any real number into a value between 0 to 1 .
Now we can predict yhat equals one or zero using a threshold value for this sigmoid function The default threshold for this is 0.5 any value greater than 0.5 will predict one means true else false . We can also customize the threshold according to our need .

Loss Function (BinaryCrossentropy or Log Loss)

The loss function measures the error between predicted probability and true label:

$$L(y, \hat{y}) = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right]$$

y means actual label and yhat means predicted probability

Gradient Descent For Optimization

To find the best weights w, logistic regression uses **gradient descent**:

$$w_j := w_j - \alpha \frac{\partial J}{\partial w_j}$$

One Hot Encoding

One-Hot Encoding is a method used to convert **categorical variables** (like colors, names, labels) into a **numerical format** that can be used by machine learning models.

This converts each and every types of categorical values into a set of true and false arrays means if a categorical value has 4 types of categories it marks that category as true and all other three as false for that particular example

Model 2 - Neural Network

A **Neural Network** is a machine learning model inspired by the human brain, composed of layers of interconnected nodes ("neurons")

Tensorflow was used for the main implementation of neural network .
A neural network is mainly based on 3 things - 1 Input layer , 2 Hidden Layer , 3 Output Layer
And then each layer except for the input layer has a certain number of neuron which compute their weights and bias and then applies an activation function

Activation Function

Mainly two types of activation functions were used in this implementation
1 Relu (Rectified Linear Unit) :

$$\text{ReLU}(x) = \max(0, x)$$

2 Sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The Loss function used was same as that of logistic regression BinaryCrossentropy

BackPropagation (Gradient Descent)

We update weights using derivatives of the loss w.r.t. Weights:

$$w := w - \alpha \frac{\partial \text{Loss}}{\partial w}$$

All of the explanation of the code is given into the markdown cell of the notebook file

Comparing the Two Models

Comparing the models based on results of the test set

Model	Accuracy	Precision	Recall	F1 Score
Model 1	0.74	0.56	0.64	0.60
Model 2	0.75	0.59	0.62	0.60

From this table we can see that the performance of both model is nearly same .