Programming Assignment: Logistic Regression

October 30, 2024

Deadline: 06, Nov 2024, 2:59PM

Introduction

You are provided with a dataset containing information on `Gender`, `Age`, `EstimatedSalary`, and `Purchased` (target variable indicating purchase decision as 0 or 1). Your task is to predict whether a customer will purchase a product based on the given features using Logistic Regression. You are required to implement the following functions from scratch, without using any built-in logistic regression functions.

Download the dataset from: `https://drive.google.com/drive/folders/1bKRknh6X-Pque_LmfZTVgpmD--mULiUD?usp=sharing`

Tasks

1. **Binary Cross-Entropy Loss**
   Implement a function, `binary_cross_entropy_loss(Y_true, Y_pred)`, to compute the Binary Cross-Entropy Loss between the true labels Y_{true} and the predicted probabilities Y_{pred}.

2. **Sigmoid Function**
   Implement the sigmoid function, `sigmoid(z)`, which maps any input value z to a probability between 0 and 1. This will be used as the activation function for the logistic regression model.

3. **Feature Normalization**
   Create a function, `normalize_features(X)`, that normalizes the features `Age` and `EstimatedSalary` to have zero mean and unit variance. Convert `Gender` to a binary variable (e.g., 0 for female and 1 for male).

4. **Fit Logistic Regression Model**
   Implement `fit_logistic_regression(X, y, learning_rate, num_epochs=10)`. This function should:

   - Initialize weights and bias to small random values.

   - Use gradient descent to update weights and bias for 10 epochs. Consider learning rate = 0.1
   - Return the optimized weights and bias.

5. **Predict Function**
   Implement `predict(X, weights, bias, threshold=0.5)` to return binary predictions (0 or 1) based on the logistic regression model's outputs.

6. **Evaluation Metrics: Accuracy, Precision, Recall, and F1-Score**
   Write functions to compute accuracy, precision, recall, and F1-score for model evaluation. Use these metrics to evaluate the performance of your model on the test data.

7. **ROC Curve and AUC**
   Implement a function `plot_roc_curve(Y_true, Y_scores)` to plot the ROC curve using Matplotlib. Compute the Area Under the Curve (AUC) and display it on the plot.

8. **Confusion Matrix**
   Implement a function to display the confusion matrix for model predictions on test data. Visualize it as a heatmap.

Instructions

- **Allowed Libraries**: You may use `Numpy`, `Pandas`, `random` and `Matplotlib`. Using other library is prohibited. You may use only a few functions of *sklearn* which are already imported in the sample code.
- **Submission Requirements**: Submit a single Jupyter notebook containing all code, comments, plots, and explanations.
- **Data Handling**: The dataset is already split; use the provided training and test sets.