

# Week5 - Logistic Regression from Scratch and Naive bayes classification

DS3010: Introduction to Machine Learning Lab

Timing: 02:00 PM to 04:45 PM

Max Marks: 5

## Instructions

1. Submit one .ipynb file containing all answers. The name should be [student\_name]\_[rollno]\_[lab].ipynb
2. Write the questions in separate text blocks before the answers.
3. Outputs for all sub-questions should be given and the code should be executable.
4. Write justifications for your choices where needed.
5. Ensure that all plots include clear labels and legends for better interpretation.
6. Use of generative AI tools (such as ChatGPT, Gemini, etc.) is strictly prohibited. Any submission found to contain AI-generated or plagiarized content will receive a score of zero, and disciplinary action.

---

## 1. Implementing Logistic Regression from Scratch

### Dataset

Heart Failure Prediction Dataset

#### a. Data Preprocessing (0.5 marks)

1. Load the dataset. Examine it for missing values and check the data types.
2. Encode all categorical variables (`Sex`, `ChestPainType`, `RestingECG`, `ExerciseAngina`, `ST_Slope`) into numerical values using `LabelEncoder`.
3. **Feature Scaling:** Standardize the entire feature set using `StandardScaler`.
4. Split the scaled data into a training set (80%) and a test set (20%). Use `random_state=42`.
5. Reshape values of `y_train` and `y_test` to `(-1,1)`

## b. Core Implementation (From Scratch) (2 marks)

You must implement the logistic regression using only `numpy`:

1. `sigmoid(z)`: Computes the sigmoid function of input `z`.

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

2. `initialize_parameters(dim)`: Returns initialized weights (a zero vector of shape `(dim, 1)`).
3. Loss function of logistic regression is

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^N \log(1 + e^{-y_i w^T x_i}) + \lambda \|w\|_2^2.$$

Find gradient with respect to  $W$

4. write `optimize(w, X, y, lambda, num_iterations, learning_rate)`: Runs the gradient descent optimization loop, updating parameters and storing the cost history. Returns the parameters,
5. `predict(w, X)`: Returns binary predictions (0 or 1) for the input data `X` using the learned parameters.

$$\hat{y} = \begin{cases} 1 & \text{if } \sigma(w^T x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

## c. Training and Evaluation (0.5 marks)

1. Train your model on the training set using `num_iterations=2000` and `learning_rate=0.1`.
2. **Plot the cost function value against the number of iterations.** Analyze the plot. Has the algorithm converged?
3. Use your trained (`w`) to make predictions on the training and test sets.
4. Calculate and report the **accuracy** for both sets.

## 2. Naive Bayes Classification with Scikit-Learn

### Dataset

Use the same Heart Failure Prediction Dataset

#### a. Data Preprocessing (0.25 marks)

- **Do NOT** perform feature scaling.
- **Handle Categorical Variables Differently:** Instead of `LabelEncoding`, use `OneHotEncoder` for the categorical features.
- Split the data into training and test sets (80/20, `random_state=42`).

#### b. Model Training and Evaluation (Using Scikit-Learn) (1.5 marks)

1. **Train a GaussianNB Model:** Import `GaussianNB` from `sklearn.naive_bayes`. Train it on your preprocessed training data.
2. **Make Predictions:** Use the trained model to predict classes for the test set.
3. **Evaluate the Model:** Calculate and print the **accuracy, precision, recall, and F1-score**. Generate a confusion matrix.
4. **Interpret Predictions:** Use the `predict_proba()` method to examine the predicted probabilities for the first 10 samples in the test set. How confident is the model in its predictions?

#### c. Analysis and Comparison (0.25 marks)

1. Compare the test accuracy of your Naive Bayes model with the accuracy of the Logistic Regression model . Which performed better on this dataset?
2. The "naive" in Naive Bayes comes from the assumption of feature independence. Looking at the features in this dataset (e.g., age, cholesterol, max heart rate), is this assumption perfectly true? Why does the model often work well in practice despite this violation?