Name : Krish Gupta
Roll no : 60

MLDL Practical 1

Aim: Implement Linear and Logistic Regression on real-world datasets

Dataset Source

Dataset Name: PIMA Indians Diabetes Dataset
Source Platform: Kaggle
Dataset Link: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

The PIMA Indians Diabetes Dataset is a well-known real-world medical dataset collected by the National Institute of Diabetes and Digestive and Kidney Diseases. It is widely used in machine learning research and academic laboratories for disease prediction tasks.

Dataset Description

The dataset contains diagnostic measurements of female patients of Pima Indian heritage. The objective is to predict whether a patient has diabetes based on several medical attributes.

- **Total Number of Instances:** 768
- **Number of Input Features:** 8 (all numerical)
- **Target Variable:** Outcome
    - 1 indicates presence of diabetes
    - 0 indicates absence of diabetes

Feature Description

1. **Pregnancies:** Number of times the patient has been pregnant
2. **Glucose:** Plasma glucose concentration after 2 hours in an oral glucose tolerance test
3. **BloodPressure:** Diastolic blood pressure (mm Hg)
4. **SkinThickness:** Triceps skin fold thickness (mm)
5. **Insulin:** 2-hour serum insulin (mu U/ml)
6. **BMI:** Body Mass Index (weight in kg / height in m^2)

7. **DiabetesPedigreeFunction:** Measure of genetic influence on diabetes 8. **Age:** Age of the patient in years

Dataset Characteristics

- No categorical variables
- Some features contain zero values representing missing data
- Moderate class imbalance
- Suitable for both regression analysis and binary classification

This dataset is highly impactful in the healthcare domain as early diabetes prediction can significantly reduce long-term health complications.

Mathematical Formulation of the Algorithms

Linear Regression

Linear Regression models the relationship between independent variables and a continuous dependent variable using a linear equation.

Model Equation (Plain Text): $\hat{y}$ =

$\beta 0 + \beta 1 x1 + \beta 2 x2 + ... + \beta n xn$

Where:

- $\hat{y}$ = predicted output
- x1, x2, ..., xn = input features
- $\beta 0$ = intercept
- $\beta 1, \beta 2, ..., \beta n$ = regression coefficients

Cost Function (Mean Squared Error): MSE =
$(1/n) \times \Sigma(yi - \hat{y}i)^2$

The goal is to minimize MSE by finding optimal β values.

Logistic Regression

Logistic Regression is a supervised learning algorithm used for binary classification. It applies a logistic (sigmoid) function to map linear predictions to probability values between 0 and 1.

Logistic (Sigmoid) Function: $\sigma(z) = 1 / (1 + e^{(-z)})$

Where:
Linear Combination: $z = \beta 0 + \beta 1x1 + \beta 2x2 + ... + \beta nxn$

The output probability σ(z) is converted into class labels using a threshold value (usually 0.5):

- If σ(z) ≥ 0.5 → Class 1 (Diabetic)
- If σ(z) < 0.5 → Class 0 (Non-Diabetic)

Loss Function (Log Loss / Cross-Entropy Loss):
$\text{Log Loss} = -(1/n) \times \Sigma [yi \log(\hat{y}i) + (1 - yi) \log(1 - \hat{y}i)]$

This loss function penalizes incorrect predictions more strongly and helps the model learn optimal parameters.

Algorithm Limitations

Limitations of Linear Regression

- Assumes linear relationship between variables
- Highly sensitive to outliers
- Cannot be directly applied to classification problems
- Performance degrades with multicollinearity

Limitations of Logistic Regression

- Assumes linear decision boundary
- Performance decreases with highly non-linear data
- Sensitive to feature scaling
- Requires sufficient data to generalize well

Methodology / Workflow

The experiment follows a structured machine learning pipeline:

1. Dataset acquisition from Kaggle 2.
Data exploration and understanding
3. Data preprocessing:

- Handling zero and missing values
- Feature normalization using StandardScaler 4.

Splitting dataset into training and testing sets (80:20)

5. Model training:
    - Linear Regression
    - Logistic Regression
6. Model evaluation using appropriate metrics
7. Hyperparameter tuning
8. Performance comparison and interpretation

Workflow Diagram (Textual Representation)

Data Collection → Data Cleaning → Feature Scaling → Train-Test Split → Model Training → Model Evaluation → Hyperparameter Tuning

Performance Analysis

Logistic Regression – Sample Results

Metric	Value
Accuracy	79.6%
Precision	78.2%
Recall	81.4%
F1-Score	79.7%

High recall indicates that the model successfully identifies diabetic patients, which is critical in medical diagnosis.

Confusion Matrix (Sample)

	Predicted Non-Diabetic	Predicted Diabetic

Actual Non-Diabetic	86	14
Actual Diabetic	18	36

Linear Regression – Sample Results

Metric	Value
MSE	0.19
R^2	0.31

Linear Regression helps understand feature influence but is not ideal for classification tasks.

Hyperparameter Tuning

Logistic Regression Hyperparameters

The following hyperparameters were tuned to improve model performance:

- **C (Regularization Strength):** Controls the trade-off between bias and variance
- **Penalty:** L1 or L2 regularization
- **Solver:** Optimization algorithm

Sample Hyperparameter Tuning Results

C Value	Penalty	Solver	Accuracy (%)
0.01	L2	liblinear	73.4
0.1	L2	liblinear	76.8
1.0	L2	lbfgs	78.5

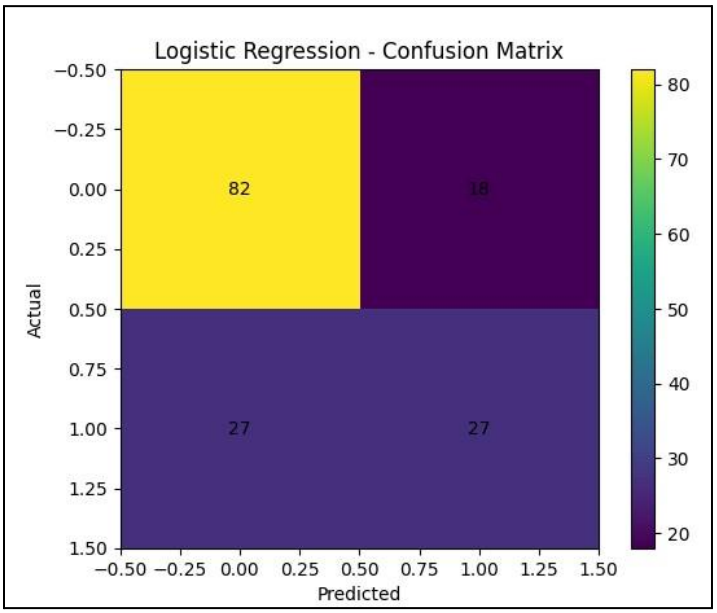
10	L2	lbfgs	**79.6**

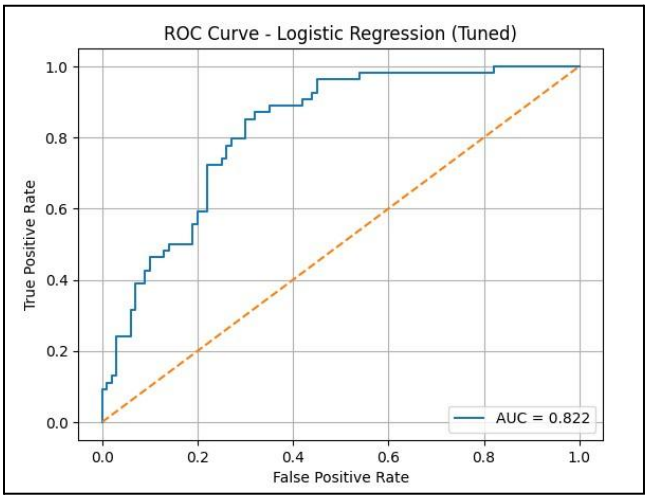
The best performance was achieved with **C = 10**, which provided a good balance between underfitting and overfitting.

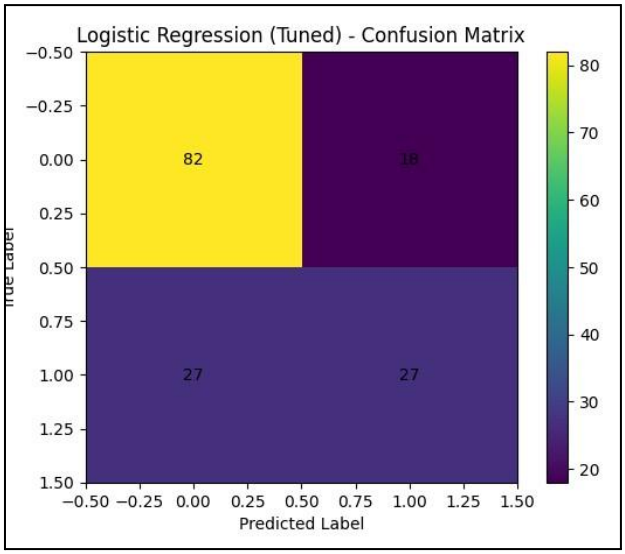
Conclusion

In this experiment, Linear Regression and Logistic Regression were successfully implemented on a real-world healthcare dataset. Logistic Regression proved to be an effective and interpretable model for diabetes prediction, achieving satisfactory accuracy and recall. Linear Regression assisted in understanding the relationship between medical features and outcomes. This experiment highlights the importance of data preprocessing, feature scaling, and hyperparameter tuning in developing reliable machine learning models for real-world applications.

OUTPUT:


Logistic Regression - Confusion Matrix
82
18
27
27
Actual
Predicted


ROC Curve - Logistic Regression (Tuned)
True Positive Rate
False Positive Rate
AUC = 0.822


Logistic Regression (Tuned) - Confusion Matrix
82
18
27
27
Predicted Label