Experiment No: 5

Instructions:

1. Refer a new dataset for Experiments 5 to 9.
2. Dataset must have more than 1 lakh instances (Big Data).
3. Each group must upload a dataset, verify feature relevancy, and apply appropriate algorithms.
4. No repetition of datasets within or across batches.

Aim: To perform regression analysis using **Scipy** and **Scikit-learn**.

Problem Statement: a) Perform Logistic Regression to find the relationship between variables. b) Apply regression model techniques to predict data on the selected dataset.

Theory: Regression is a statistical method used in supervised machine learning to model the relationship between a **dependent (target)** variable and one or more **independent (predictor)** variables. The main objective is to predict continuous or discrete outcomes based on input features. In this experiment, we focus on two types of regression analysis:

---

1. Logistic Regression:

- Logistic Regression is a **classification technique** used to predict binary or categorical outcomes using a logistic function (sigmoid).

- It estimates the probability that a given input belongs to a particular category.

- Despite the name 'regression', it is used for **classification problems**.

Mathematical Function: σ(z)=11+e−zwhere z=b0+b1x1+...+bnxn\sigma(z) = \frac{1}{1 + e^{-z}} \quad \text{where } z = b_0 + b_1x_1 + ... + b_nx_n

- The output σ(z)\sigma(z) lies between 0 and 1 and represents the probability.

- A threshold (commonly 0.5) is applied to decide the class label.

Applications:

- Predicting churn, spam detection, medical diagnosis (e.g., whether a person has a disease or not).

---

2. Linear Regression (Scipy and Scikit-learn):

- Linear Regression is used to predict a **continuous numeric** value based on input features.

- It assumes a linear relationship between the dependent variable and the independent variables.

Equation: y=b0+b1x1+b2x2+...+bnxn+ϵy = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n + \epsilon Where:

- yy: Predicted value

- x1,x2,...,xnx_1, x_2, ..., x_n: Independent variables
- b0,b1,...,bnb_0, b_1, ..., b_n: Coefficients (model parameters)
- ϵ\epsilon: Error term

Types:

- **Simple Linear Regression**: Involves a single feature.
- **Multiple Linear Regression**: Involves multiple features.

Scipy vs. Scikit-learn:

- **Scipy**: Offers `linregress` function from `scipy.stats`, used for quick simple linear regression.
- **Scikit-learn**: Offers `LinearRegression()` and `LogisticRegression()` classes for more flexible modeling and performance evaluation.

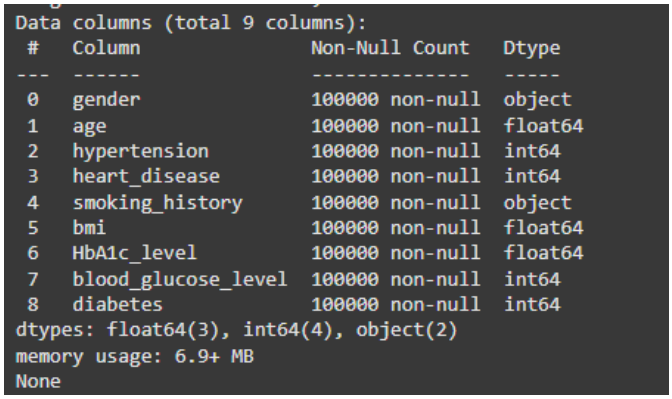
Steps (Implementation Summary):

1. Import necessary libraries (NumPy, pandas, matplotlib, seaborn, sklearn, scipy).
2. Load the large dataset (>1 lakh rows) and verify relevant features.
3. Perform preprocessing (null value handling, encoding categorical data, feature scaling).

4. Apply **Logistic Regression** using `sklearn.linear_model.LogisticRegression`:
    - Fit model and interpret coefficients.
    - Predict binary labels.
    - Evaluate with confusion matrix, accuracy, precision, recall, F1-score.
5. Apply **Linear Regression** using `scipy.stats.linregress` and `sklearn.linear_model.LinearRegression`:
    - Fit models.
    - Predict target variable.
    - Evaluate using metrics: Mean Squared Error (MSE), R2 score.
6. Visualize the regression line and residuals.

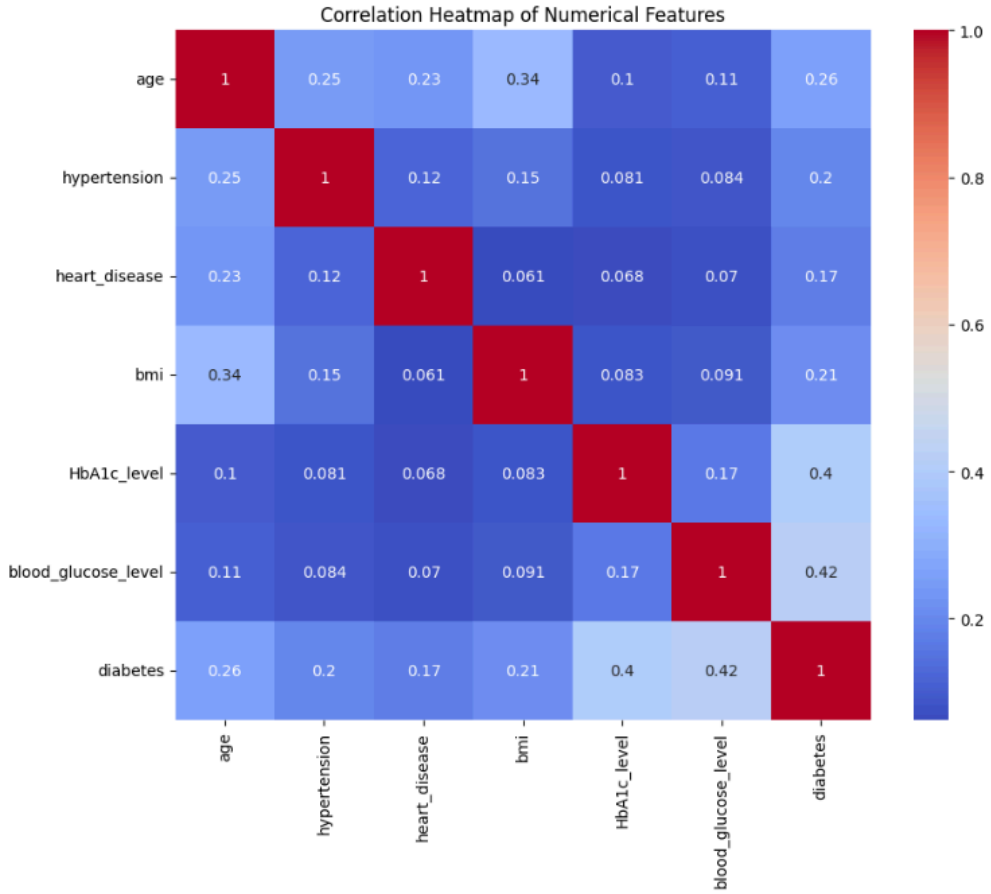
Space for Screenshots:

- Dataset preview and description



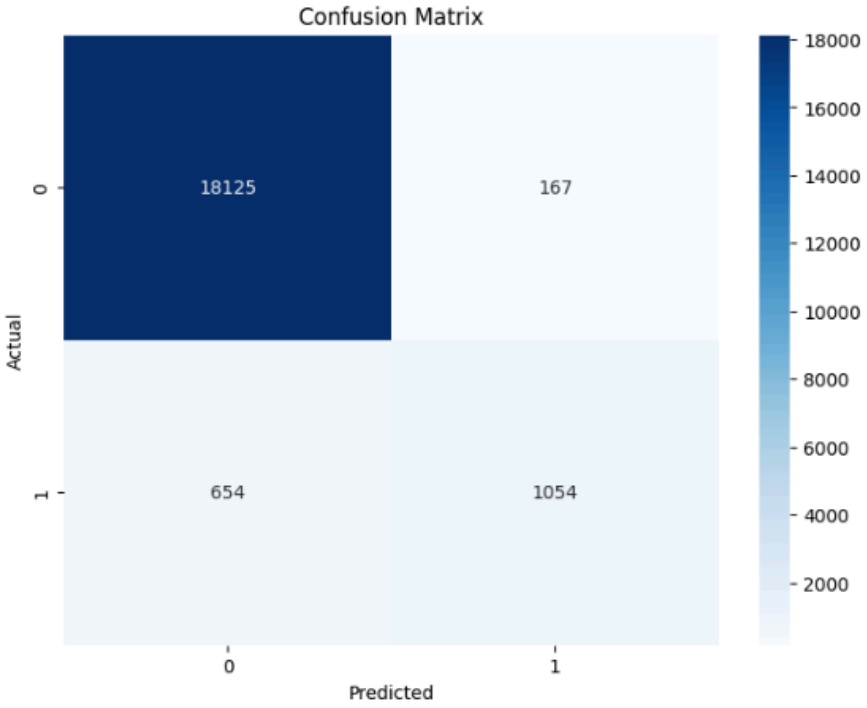
```
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100000 non-null  object
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64
 3   heart_disease        100000 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64
 8   diabetes             100000 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
None
```

- Feature verification and correlation heatmap

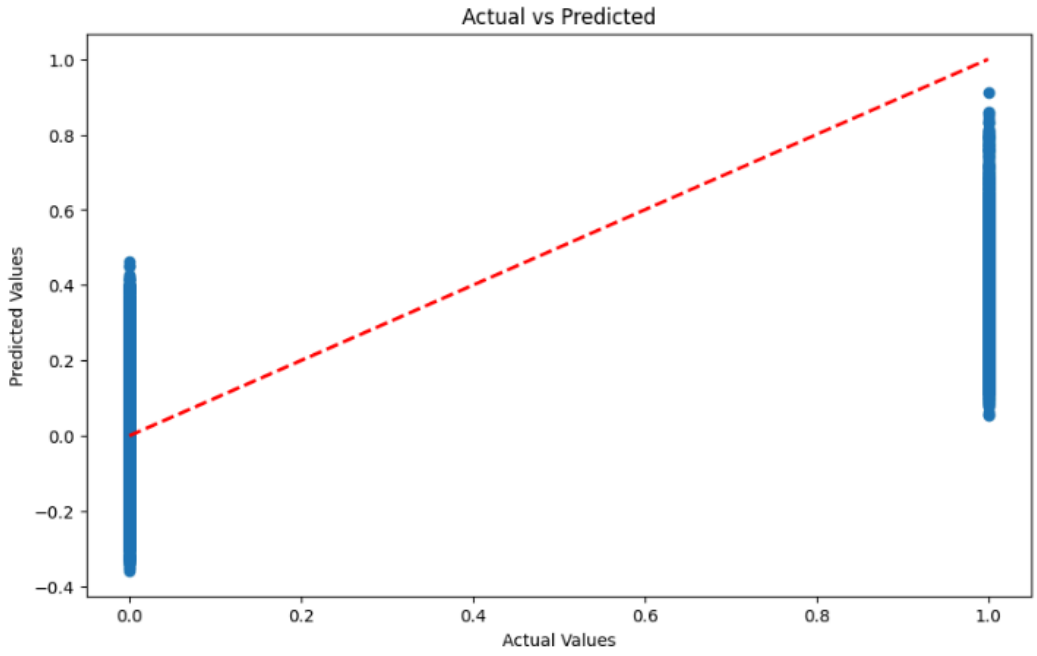


- Logistic regression coefficients and classification report

```
Logistic Regression Results:
Accuracy: 0.95895

Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     18292
           1       0.86      0.62      0.72      1708

    accuracy                           0.96     20000
   macro avg       0.91      0.80      0.85     20000
weighted avg       0.96      0.96      0.96     20000
```



- Linear regression line and predicted vs actual plot



- Model performance

```
SciPy Statistical Analysis:
age:
t-statistic: 84.44765154890852
p-value: 0.0
Statistically significant

bmi:
t-statistic: 69.39822715441193
p-value: 0.0
Statistically significant

HbA1c_level:
t-statistic: 138.28308350581383
p-value: 0.0
Statistically significant

blood_glucose_level:
t-statistic: 146.1610562407839
p-value: 0.0
Statistically significant
```

Conclusion: This experiment explored both logistic and linear regression models using Scipy and Scikit-learn on a large-scale dataset. Logistic regression effectively classified binary outcomes by modeling the relationship between input features and probabilities, while linear regression was used to predict continuous values. Through evaluation metrics and visualizations, we gained insights into the model performance and the importance of feature selection and data preprocessing. This analysis demonstrates how regression techniques help in predictive modeling and decision-making processes in real-world applications.