Experiment 1: Implementation of Linear and Logistic Regression on Real-World Datasets

1. Database Source

2. Dataset Description : The experiment utilizes the Bank Marketing Dataset (bank.csv).

- **Size:** The dataset contains **11,162 records** and **17 columns**.
- **Features:**
  - **Numerical:** age, balance, day, duration, campaign, pdays, previous.
  - **Categorical:** job, marital, education, default, housing, loan, contact, month, poutcome.
- **Target Variables:**
  - **Linear Regression:** balance (Continuous numerical).
  - **Logistic Regression:** deposit (Categorical: "yes", "no").
- **Characteristics:** The data represents marketing campaigns (phone calls) of a bank. It is a mix of demographic data and campaign-specific metrics. The target variable deposit is relatively balanced in this specific version of the dataset.

3. Mathematical Formulation :

A. Linear Regression

It models the target as a linear combination of features:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon$$

The objective is to minimize the **Residual Sum of Squares (RSS)**:

$$RSS = \sum_{j=1}^{m} (y_j - \hat{y}_j)^2$$

B. Logistic Regression

It models the probability of a binary outcome using the **Logit link function**:

$$P(y=1) = \frac{1}{1+e^{-z}} \text{ where } z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

The parameters are estimated using **Maximum Likelihood Estimation (MLE)** to maximize the likelihood that the observed classes are predicted by the model.

4. Algorithm Limitations :

- **Linear Regression:**
  - **Linearity Assumption:** Fails if the relationship between features and target is non-linear.
  - **Outlier Sensitivity:** Highly susceptible to outliers (common in "balance" data).
  - **Multicollinearity:** Performance degrades if independent variables are highly correlated.
- **Logistic Regression:**
  - **Binary Restriction:** Standard version only handles two classes.
  - **Linear Boundary:** Assumes a linear decision boundary; struggles with complex, overlapping data clusters.
  - **Large Sample Requirement:** Requires a relatively large sample size for MLE to converge reliably.

5. Methodology / Workflow :

1. **Data Ingestion:** Load bank.csv using Pandas.
2. **Preprocessing:** * Handling categorical variables via **Label Encoding**.
   - Feature Selection: Choosing numerical predictors for Linear Regression.
3. **Data Splitting:** 80% Training and 20% Testing sets using train_test_split.
4. **Model Training:**
   - Instantiating LinearRegression() and LogisticRegression(max_iter=1000).
   - Fitting models on training data.
5. **Evaluation:** Calculating metrics on the test set.

6. Performance Analysis :

- **Logistic Regression (Classification):**
  - **Accuracy:** ~79%
  - **Interpretation:** The model is effective at predicting subscriptions. The F1-score (~0.77) shows balanced performance between precision and recall.

- **Linear Regression (Regression):**
  - R^2 **Score:** 0.0135
  - **RMSE:** 3539.20
  - **Interpretation:** The model performed poorly for predicting balance. An R^2 near zero indicates the features used do not linearly explain the variation in customer balances.

7. Hyperparameter Tuning :

For Logistic Regression, the **Inverse Regularization Strength (C)** was tuned:

- **Process:** Compared $C = 0.1$, $C = 1.0$, and $C = 10$.
- **Impact:**
  - $C = 0.1$(Stronger Regularization) reduced overfitting but slightly lowered accuracy.
  - $C = 1.0$ (Default) provided the best balance for this dataset.
  - Increasing max_iter to 1000 was necessary to ensure the optimization algorithm reached convergence given the number of categorical features.

Conclusion: Logistic Regression is suitable for this dataset's classification task, while Linear Regression requires more complex feature engineering or non-linear approaches to predict financial balances.