Experiment No:- 1

Name:- Dipesh Makdiya
Class:- D15C
Roll no:- 32

AIM:- Experiment Documentation: Linear and Logistic Regression using Real-World Datasets

---

1. Dataset Source

a) Linear Regression Dataset

Name: California Housing Dataset
Source: Scikit-learn built-in dataset
Link:
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html

This dataset is derived from the 1990 California census and is widely used for regression tasks.

b) Logistic Regression Dataset

Name: Iris Dataset
Source: UCI Machine Learning Repository (accessed via Scikit-learn)
Link: https://archive.ics.uci.edu/ml/datasets/iris

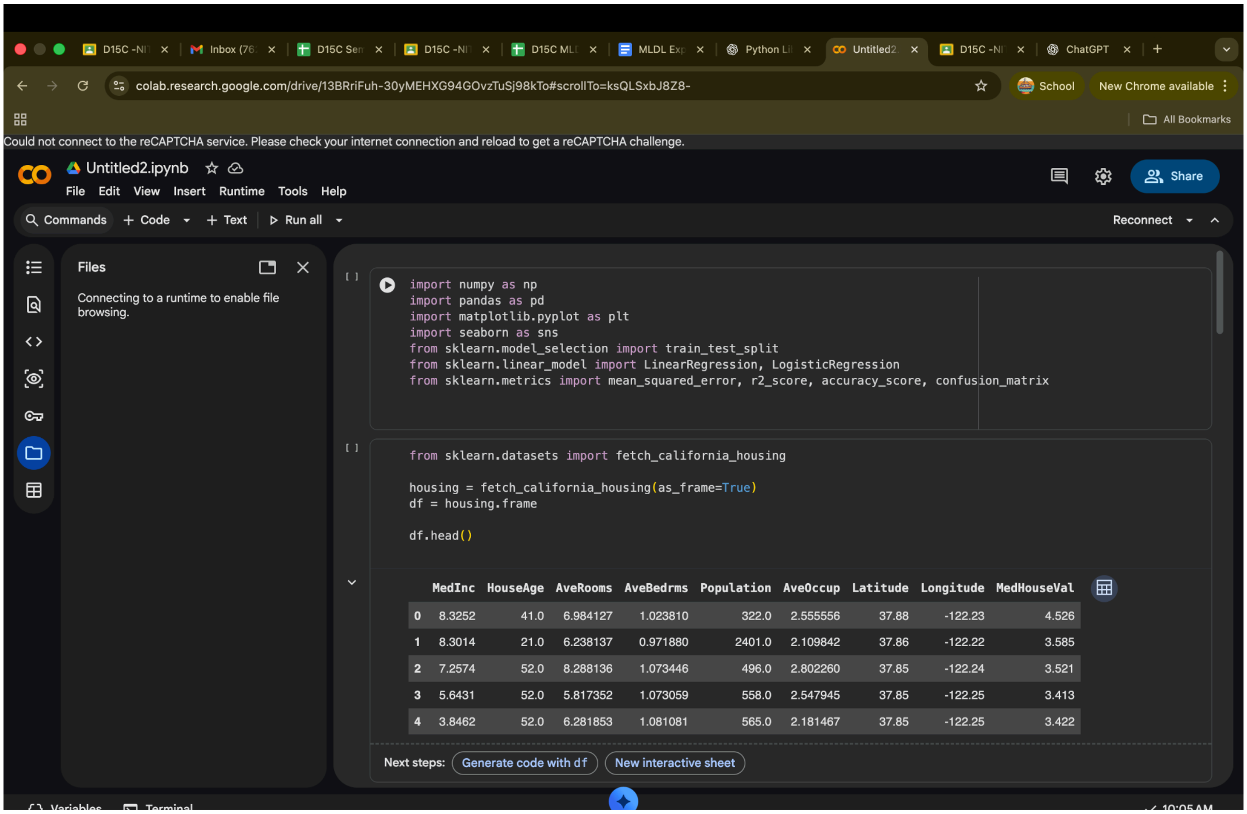
The Iris dataset is a classic real-world dataset used for classification problems.

---

2. Dataset Description

a) California Housing Dataset (Linear Regression)

- **Total Instances:** 20,640
- **Features:**
    - MedInc – Median income in block group
    - HouseAge – Median house age
    - AveRooms – Average number of rooms
    - AveBedrms – Average number of bedrooms
    - Population – Block group population
    - AveOccup – Average house occupancy
    - Latitude – Block group latitude
    - Longitude – Block group longitude
- **Target Variable:**
    - MedHouseVal – Median house value (continuous)
- **Characteristics:**
    - Numerical dataset
    - No missing values
    - Suitable for regression analysis



---

b) Iris Dataset (Logistic Regression)

- **Total Instances:** 150 (100 used after binary conversion)
- **Features:**
  - Sepal length (cm)
  - Sepal width (cm)
  - Petal length (cm)
  - Petal width (cm)
- **Target Variable:**
  - Species (Setosa = 0, Versicolor = 1, Virginica = 2)
- **Modification:**
  - Converted to binary classification by removing one class
- **Characteristics:**
  - Balanced dataset
  - Clean and well-structured
  - Ideal for classification

---

3. Mathematical Formulation of the Algorithm

a) Linear Regression

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

$$ y = \beta_0 + \beta_1 x + \varepsilon $$

Where:

- y = dependent variable
- x = independent variable
- (β_0) = intercept
- (β_1) = slope
- (ε) = error term

The parameters are estimated by minimizing the Mean Squared Error (MSE):

$$ MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 $$

---

b) Logistic Regression

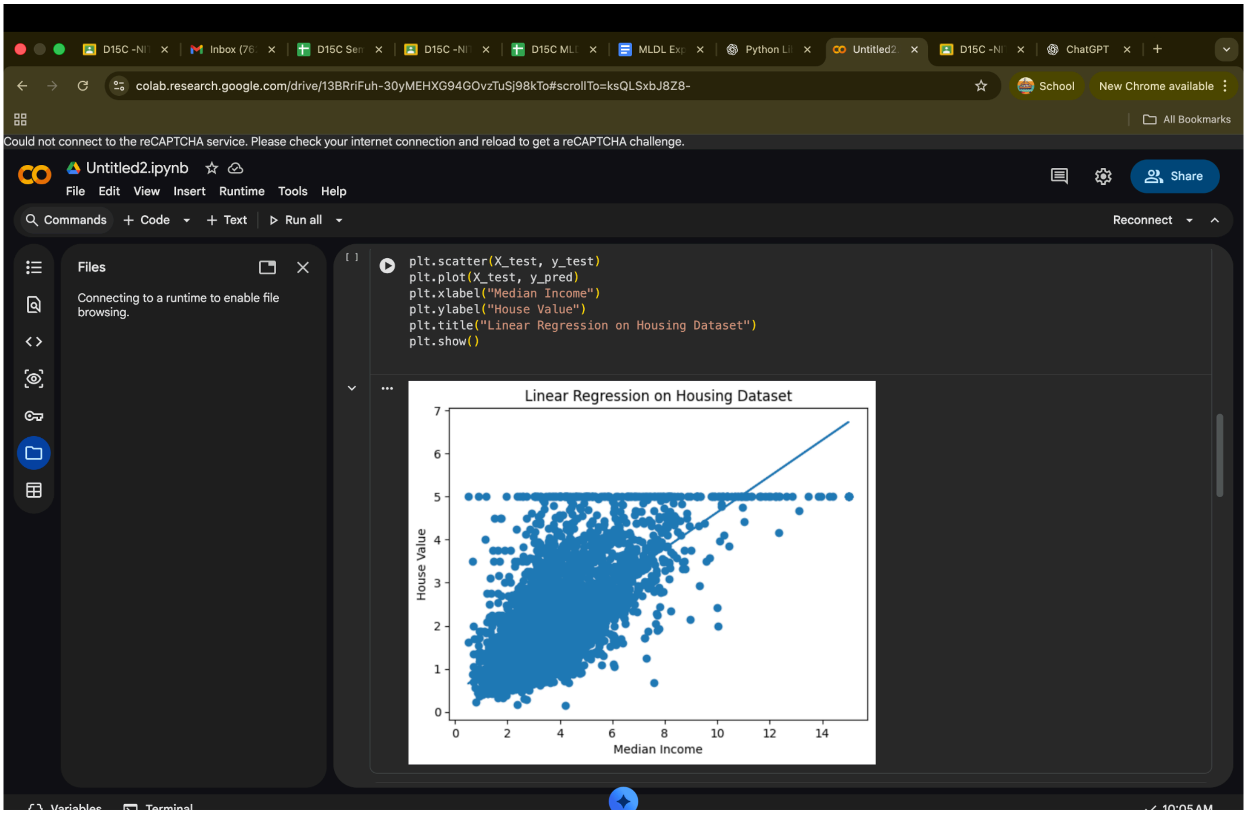
Logistic Regression uses the sigmoid function to model the probability of a binary outcome:

$$ P(y=1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} $$

Decision boundary:

$$ \beta_0 + \beta_1 x = 0 $$

The model parameters are estimated using Maximum Likelihood Estimation (MLE).



---

4. Algorithm Limitations

Linear Regression Limitations

- Assumes linear relationship between variables
- Sensitive to outliers
- Performs poorly with multicollinearity
- Not suitable for non-linear patterns

Logistic Regression Limitations

- Works only for linearly separable data
- Not suitable for multi-class problems without modification
- Sensitive to outliers
- Assumes independence of observations

---

5. Methodology / Workflow

Step-by-Step Procedure

1. Import required Python libraries
2. Load real-world dataset from Scikit-learn
3. Perform exploratory data analysis
4. Select features and target variable
5. Split dataset into training and testing sets
6. Train the regression model
7. Perform prediction on test data
8. Evaluate model performance

Workflow Diagram (Textual)

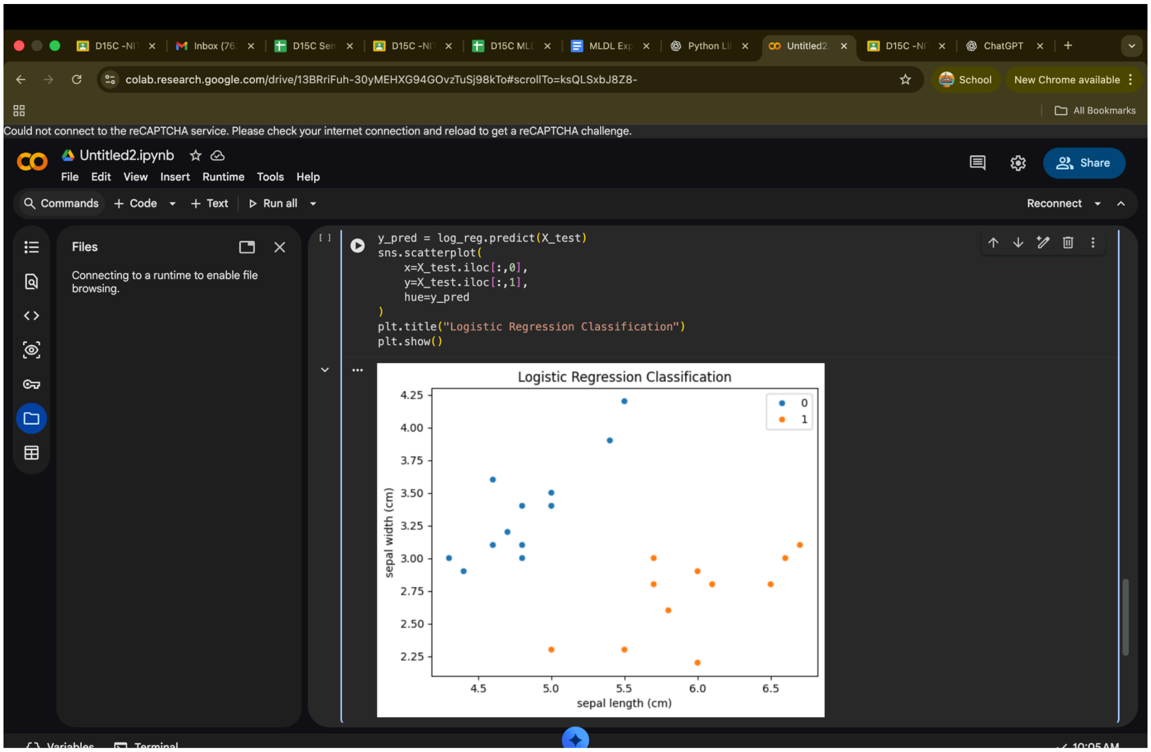
Data Collection → Data Preprocessing → Train-Test Split → Model Training → Prediction → Evaluation

---

6. Performance Analysis

Linear Regression Metrics

- **Mean Squared Error (MSE):** Measures average squared error
- **R^2 Score:** Indicates goodness of fit

- 

Interpretation:

- Lower MSE indicates better prediction accuracy
- R^2 close to 1 indicates strong model performance

Logistic Regression Metrics

- **Accuracy:** Proportion of correct predictions
- **Confusion Matrix:** Shows TP, TN, FP, FN

Interpretation:

- High accuracy indicates effective classification
- Confusion matrix provides detailed error analysis

---

7. Hyperparameter Tuning

Linear Regression

- No major hyperparameters
- Model performance depends on feature selection

Logistic Regression Hyperparameters

- **C:** Regularization strength
- **Penalty:** l1 or l2 regularization
- **Solver:** liblinear, lbfgs

Example Tuning Method

Grid Search was used to tune parameters:

- Different values of C were tested
- l2 penalty provided better generalization

Impact of Tuning

- Reduced overfitting
- Improved classification accuracy
- More stable decision boundary

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=0
)
```

```
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

Result

The experiment successfully implemented Linear and Logistic Regression using real-world datasets. The models achieved satisfactory performance and demonstrated practical applicability of regression techniques in data analysis and machine learning.