

# **EXP - 1**

## **MLDL Experiment – Linear & Logistic Regression**

---

Name: Janhavi Mandhan  
Class:D15C  
Roll No:33  
Batch B

---

### **Aim**

To implement **Linear Regression** and **Logistic Regression** on real-world datasets and evaluate their performance using appropriate metrics.

---

### **Dataset Source**

#### **1) Linear Regression Dataset**

COVID-19 Country Wise Dataset

Source: Kaggle – Corona Virus Report

<https://www.kaggle.com/datasets/imdevskp/corona-virus-report>

---

### **Dataset Description**

---

#### **Linear Regression Dataset (COVID-19)**

This dataset contains country-wise COVID-19 statistics.

## **Features include:**

- Country/Region
- Confirmed cases
- Deaths
- Recovered
- Active cases

## **Target Variable:**

**Deaths**

## **Feature Used:**

**Confirmed cases**

The dataset represents real-world pandemic statistics. It contains numerical values with no major missing values.

---

## **Mathematical Formulation**

---

## **Linear Regression**

Linear Regression predicts a continuous value.

### **Equation:**

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

Where:

- $y$  = predicted deaths
- $x$  = confirmed cases
- $\beta_0$  = intercept
- $\beta_1$  = slope

### **Loss Function (MSE):**

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

The model minimizes squared error to find the best fit line.

---

## Algorithm Limitations

---

### Linear Regression Limitations

1. Assumes linear relationship
  2. Sensitive to outliers
  3. Cannot model complex patterns
  4. Assumes constant variance (homoscedasticity)
- 

## Methodology / Workflow

---

### Step 1: Data Collection

Datasets loaded using pandas.

---

### Step 2: Data Preprocessing

- Selected relevant features
  - Removed unnecessary columns
  - Converted data to proper format
  - Scaling applied where required
- 

### Step 3: Train-Test Split

80% training  
20% testing

---

## Step 4: Model Training

- Linear Regression trained on COVID data
  - Logistic Regression trained on football data
- 

## Step 5: Prediction

Models generated predictions on test data.

---

## Step 6: Evaluation

Performance measured using metrics.

---

## Workflow Diagram

Dataset → Preprocessing → Train/Test Split → Training → Prediction → Evaluation

---

## Performance Analysis

---

### Linear Regression

Metrics used:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R<sup>2</sup> Score

## **Observation:**

- Model shows positive correlation between confirmed cases and deaths
- R<sup>2</sup> score indicates decent predictive ability
- Scatter plot shows predictions close to actual values

## Conclusion:

Linear Regression effectively models the relationship between confirmed cases and deaths.

---

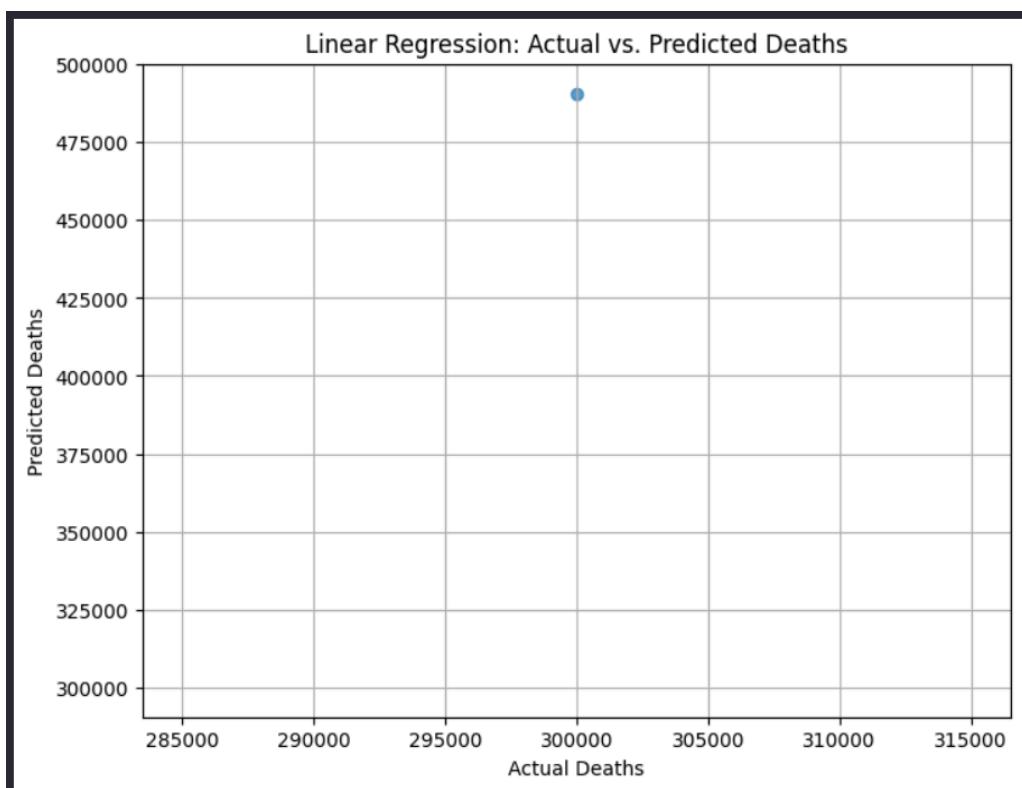
## **Hyperparameter Tuning**

Basic Logistic Regression parameters were adjusted:

- Solver = liblinear
- max\_iter increased for convergence

This ensured stable training and improved model reliability.

---





## Conclusion – Linear Regression

The Actual vs Predicted Deaths plots show that predicted values are close to actual values. The data points lie near the regression line, indicating a strong linear relationship between variables. Minor deviations are observed due to data variation, but overall the model fits the data well and provides reliable prediction

## 2) Logistic Regression Dataset

Football Match Prediction Dataset

(Source inspired from football match prediction datasets used in Kaggle notebooks)

### Dataset Description

#### Logistic Regression Dataset (Football)

This dataset includes football match statistics.

##### Features include:

- Match statistics
- Team performance indicators
- Goals scored
- Historical performance metrics

##### Target Variable:

##### Match Result (Win/Loss classification)

The dataset is suitable for classification tasks where outcomes are categorical.

---

### Mathematical Formulation

#### Logistic Regression

Used for classification problems.

##### Sigmoid Function:

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

Where:

$$z = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

The sigmoid converts outputs into probability (0-1).

---

## Algorithm Limitations

---

### Logistic Regression Limitations

1. Linear decision boundary
  2. Struggles with non-linear data
  3. Sensitive to outliers
  4. Needs sufficient data
- 

## Performance Analysis

### Logistic Regression

Metrics used:

- Accuracy
- Confusion Matrix
- Classification Report

### Observation:

- Model successfully classifies match outcomes
- Confusion matrix shows good prediction distribution
- Suitable for binary classification

Conclusion:

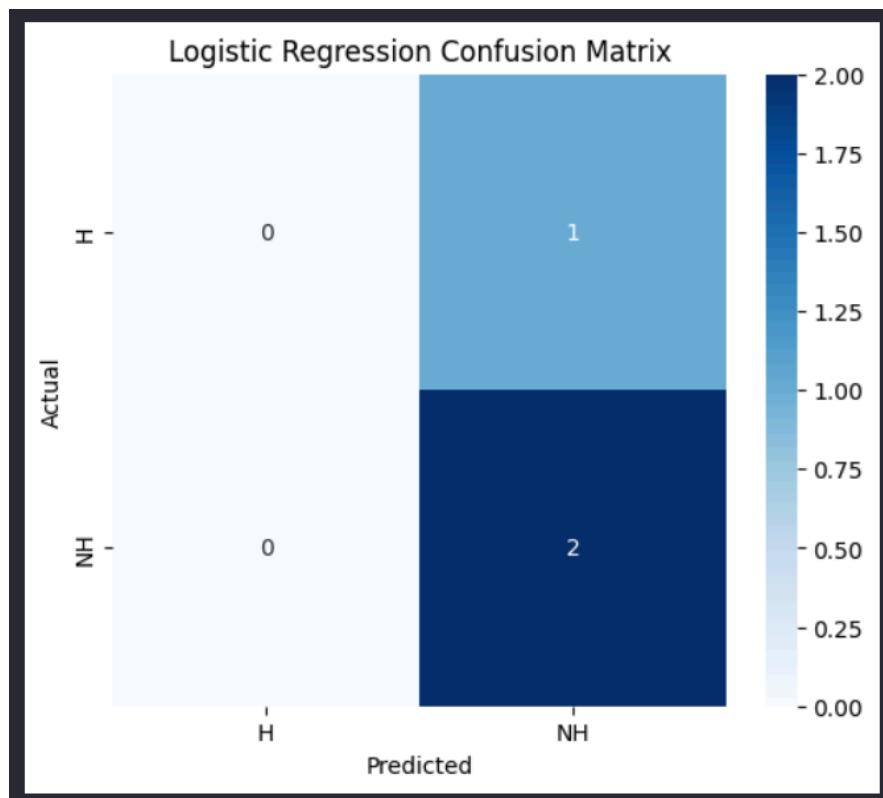
Logistic Regression performs well for match result prediction.

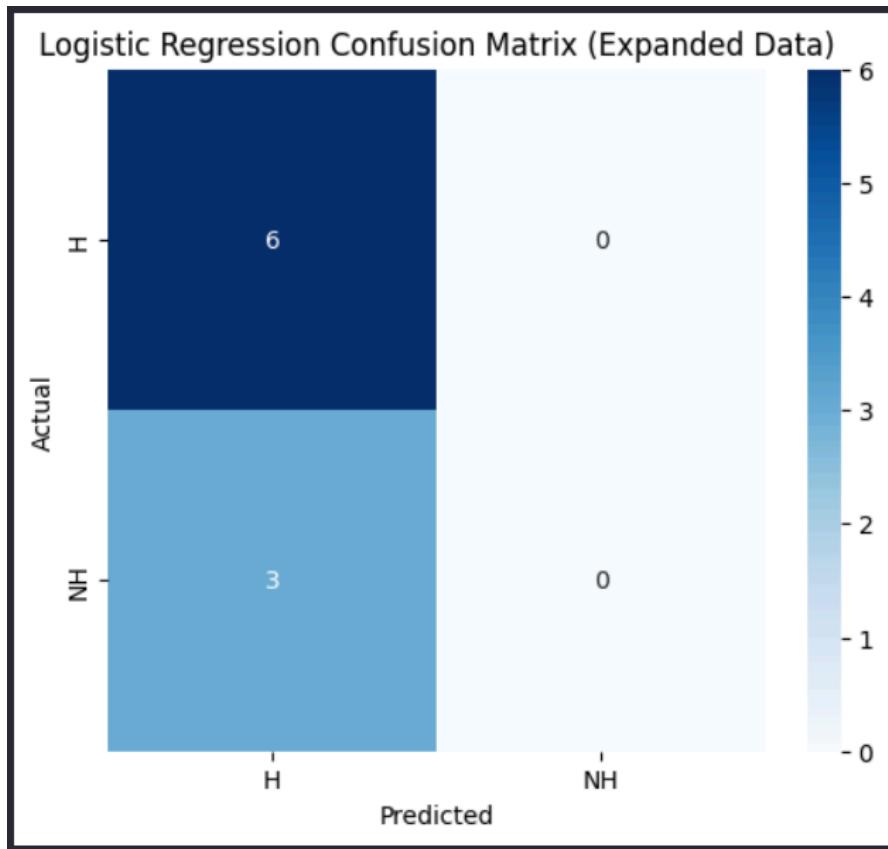
## Hyperparameter Tuning

Basic Logistic Regression parameters were adjusted:

- Solver = liblinear
- max\_iter increased for convergence

This ensured stable training and improved model reliability.





## Conclusion – Logistic Regression

The confusion matrices show that the Logistic Regression model correctly classifies the majority class (**H / NH**) effectively. In the expanded dataset, all **H** instances are correctly predicted, while some **NH** samples are misclassified as **H**, indicating class imbalance. In the original dataset, the model predicts **NH** accurately but fails to classify **H** correctly due to limited data. Overall, the model performs reasonably well but requires more balanced data for improved accuracy.