# Experiment 1

### Aim

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

## 1. Dataset Source

The dataset used in this experiment is a Student Performance Dataset, containing academic and behavioral attributes of students.
https://github.com/tanaya2005/MLDL-lab/blob/main/Experiment-%201/expanded_student_performance.csv

## 2. Dataset Description

The dataset contains academic performance data of students.

### - Dataset Characteristics

- Type: Tabular (Numerical)
- Domain: Education / Academic Performance
- Target Variable: Final_Score
- Size: Medium-sized dataset
- Data Quality: Contains missing (null) values

### - Feature Description

| Feature Name | Description |
|---|---|
| Hours_Studied | Number of hours studied |
| Attendance | Attendance percentage |
| Assignment_Score | Assignment marks |
| Midterm_Score | Midterm exam marks |
| Final_Score | Final exam marks (Target) |

### - Handling Missing Values

Since null values were present in my dataset:

- Numerical columns were filled using **mean imputation**
- This **prevents loss of data** and maintains distribution balance.

## 3. Mathematical Formulation of the Algorithm

- **Mean**

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- **Standard Deviation**

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

- **Linear Regression**

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

**Where:**

- $y$ = Final Score
- $\beta_0$ = Intercept
- $\beta_i$ = Coefficients

**Loss Function (MSE):**

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

- **Logistic Regression**

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

**(Sigmoid function converts output to probability between 0 and 1)**

## 4. Algorithm Limitations

➢ Linear Regression
- Assumes linear relationship
- Sensitive to outliers
- Performs poorly if data is non-linear

➢ Logistic Regression
- Only suitable for binary classification
- Sensitive to multicollinearity
- Requires sufficient data for stable decision boundary

➢ General EDA Limitations
- Visualization interpretation may be subjective
- Correlation does not imply causation

## 5. Methodology / Workflow

Step 1: Data Loading: Load dataset using Pandas.

Step 2: Data Cleaning

- Check missing values

Step 3: Statistical Analysis

- Compute mean, median, std deviation

Step 4: Label Creation

Create "Pass" category

Step 6: Visualization

- Scatter plot
- Heatmap

Step 7: Model Training

- Linear Regression
- Logistic Regression

Step 8: Evaluation

- $R^2$ Score
- Accuracy
- Confusion Matrix

## 6. Performance Analysis

➢ Linear Regression
  - $R^2$ score shows proportion of variance explained.
    Higher $R^2$ → Better model fit.

➢ Logistic Regression
  - Accuracy measures classification correctness.
  - Confusion matrix shows TP, TN, FP, FN.

➢ Observations**:**
  - Hours studied strongly influences final score.
  - Assignment and midterm scores have high correlation.
  - Logistic model performs well in classifying Pass/Fail.

## 7. Hyperparameter Tuning

To optimize performance, the following hyperparameters can be tuned:

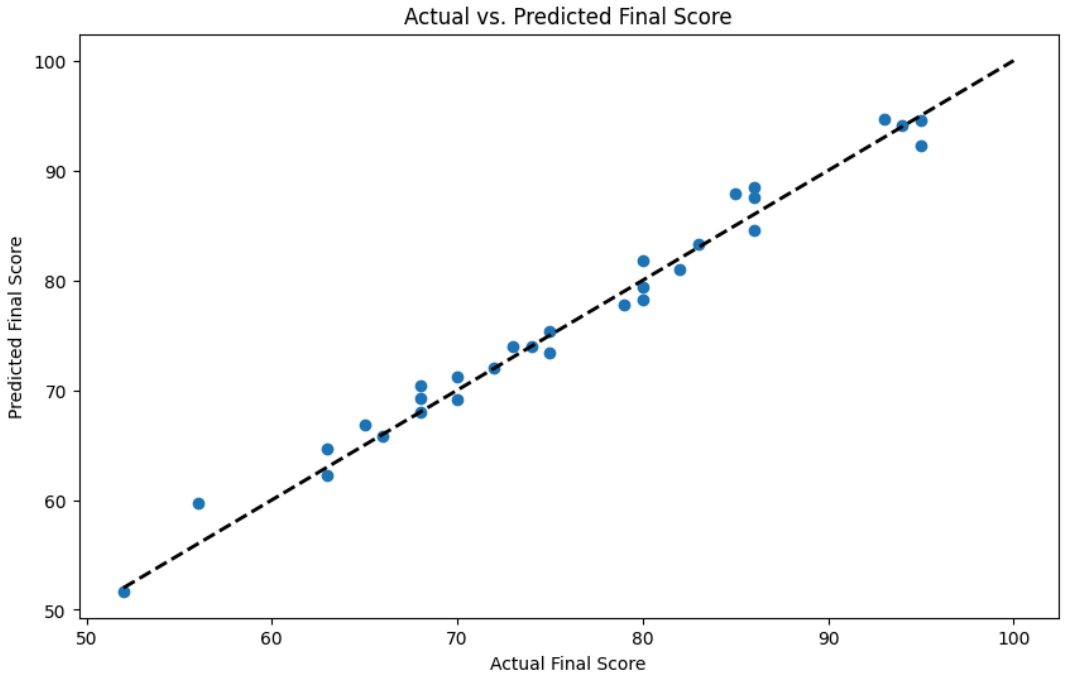
- Learning Rate ($\alpha$): Controls the step size during gradient descent. Too large leads to divergence; too small leads to slow convergence.
- Regularization (L1/L2):
- Lasso (L1): Encourages sparsity (feature selection), potentially removing less relevant features like 'Attendance' if it doesn't contribute significantly.
- Ridge (L2): Penalizes large coefficients to prevent overfitting on the expanded dataset.
- Max Iterations: Ensures the solver converges within a reasonable time frame.

**Code:**

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read_csv('/content/expanded_student_performance.csv')

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression, LogisticRegression

from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix, accuracy_score,
classification_report

y = df['Final_Score']

X = df.drop('Final_Score', axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

lin_model = LinearRegression()

lin_model.fit(X_train, y_train)

y_pred = lin_model.predict(X_test)

r2_score(y_test, y_pred)

plt.figure(figsize=(10, 6))

plt.scatter(y_test, y_pred)

plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)

plt.xlabel('Actual Final Score')

plt.ylabel('Predicted Final Score')

plt.title('Actual vs. Predicted Final Score')
```

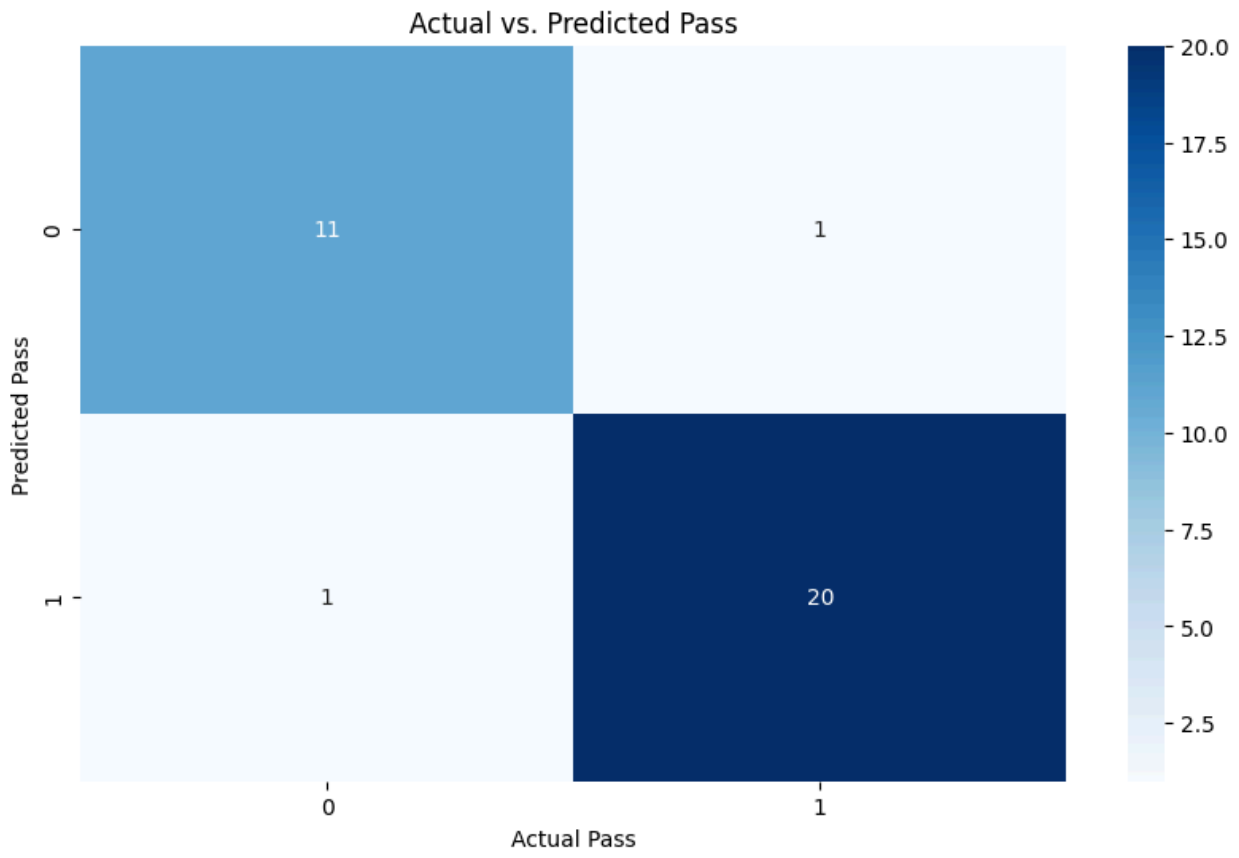
```
df['Pass'] = (df['Final_Score'] >= 70).astype(int)

X_log = df.drop(['Final_Score', 'Pass'], axis=1)

y_log = df['Pass']

X_train_log, X_test_log, y_train_log, y_test_log = train_test_split(X_log, y_log, test_size=0.33,
random_state=21)

log_model = LogisticRegression()

log_model.fit(X_train_log, y_train_log)

y_pred_log = log_model.predict(X_test_log)

print(accuracy_score(y_test_log, y_pred_log))

print(confusion_matrix(y_test_log, y_pred_log))

print(classification_report(y_test_log, y_pred_log))

plt.figure(figsize=(10, 6))

sns.heatmap(confusion_matrix(y_test_log, y_pred_log), annot=True, fmt='d', cmap='Blues')

plt.xlabel('Actual Pass')

plt.ylabel('Predicted Pass')

plt.title('Actual vs. Predicted Pass')
```

**Output:**




```
print(accuracy_score(y_test_log, y_pred_log))
print(confusion_matrix(y_test_log, y_pred_log))
print(classification_report(y_test_log, y_pred_log))
```

```
0.9393939393939394
[[11  1]
 [ 1 20]]
              precision    recall  f1-score   support

           0       0.92      0.92      0.92        12
           1       0.95      0.95      0.95        21

    accuracy                           0.94        33
   macro avg       0.93      0.93      0.93        33
weighted avg       0.94      0.94      0.94        33
```




## Conclusion

This experiment successfully implemented Linear and Logistic Regression on the expanded_student_performance.csv dataset. Linear Regression proved effective for estimating continuous student scores, while Logistic Regression provided robust binary classification for determining pass/fail outcomes. The use of evaluation metrics like MSE and Accuracy validated the reliability of these fundamental machine learning algorithms in analyzing academic performance data.