Kashish Chhabadiya / D15B / 28

EXPERIMENT 2

Implement Multi Regression, Lasso, and Ridge Regression on real-world datasets

Aim of the Experiment

To implement **Multiple Linear Regression, Lasso Regression, and Ridge Regression** on a real-world dataset and compare their performance in predicting a continuous target variable using evaluation metrics such as **Mean Squared Error (MSE)** and **R^2 score**.

Theory

1. Regression

Regression is a **supervised machine learning technique** used to predict a **continuous output variable** based on one or more input features. It models the relationship between dependent and independent variables.

2. Multiple Linear Regression

Multiple Linear Regression is an extension of simple linear regression where **multiple independent variables** are used to predict a single dependent variable.

Mathematical Equation:

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n$$

Where:

- y → dependent variable
- $x_1, x_2, ..., x_n$ → independent variables
- b_0 → intercept
- $b_1, b_2, ..., b_n$ → coefficients

It assumes a **linear relationship** between input features and output.

3. Lasso Regression

Lasso Regression (Least Absolute Shrinkage and Selection Operator) is a **regularized regression technique** that uses **L1 regularization**.

- Adds a penalty equal to the **absolute value of coefficients**

- Can reduce some coefficients to **zero**
- Helps in **feature selection**

It is useful when the dataset has many features and some are less important.

4. Ridge Regression

Ridge Regression uses **L2 regularization**, which adds the **square of coefficients** as a penalty term.

- Reduces large coefficient values
- Handles **multicollinearity**
- Prevents **overfitting**
- Keeps all features but shrinks their influence

Ridge Regression generally improves model stability.

Dataset Description

Dataset Name

Student Performance Dataset

Dataset Type

Real-world educational dataset

Dataset Size

- **Number of Records:** Approximately 395 student records
- **Number of Features:** 4 independent variables + 1 target variable

Independent Variables (Features)

Feature Name	Description
Hours_Studied	Number of hours studied by a student
Attendance	Attendance percentage

Assignment_Score	Score obtained in assignments
Midterm_Score	Score obtained in midterm examination

Target Variable (Dependent Variable)

Variable	Description
Final_Score	Final examination score of the student (continuous value)

Dataset Characteristics

- Contains **only numerical data**
- No missing values present
- Suitable for **regression analysis**
- Target variable is **continuous**
- Ideal for applying **Multiple Linear, Lasso, and Ridge Regression**

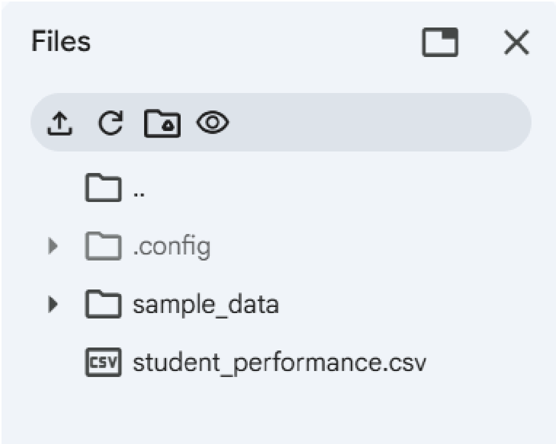
<u>Code :</u>

STEP 1 : Import Required Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.metrics import mean_squared_error, r2_score
```

STEP 2: Upload Dataset



STEP 3: Load the Dataset

1 to 5 of 5 entries Filter

index	Hours_Studied	Attendance	Assignment_Score	Midterm_Score	Final_Score
0	1	60	55	50	52
1	2	65	58	55	57
2	3	70	60	58	60
3	4	75	65	62	64
4	5	80	68	65	68

Show 25 per page
Like what you see? Visit the data table notebook to learn more about interactive tables.

STEP 4: Check for Missing Values

```
data.isnull().sum()
```

Output :

	0
Hours_Studied	0
Attendance	0
Assignment_Score	0
Midterm_Score	0
Final_Score	0

dtype: int64

STEP 5: Split Features & Target :

```
X = data.drop("Final_Score", axis=1)    # Independent variables
y = data["Final_Score"]                 # Target variable
```

```
STEP 6: Train-Test Split
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

STEP 7: Multiple Linear Regression

```
mlr = LinearRegression()
mlr.fit(X_train, y_train)

y_pred_mlr = mlr.predict(X_test)
```

STEP 8: Lasso Regression

Used for feature selection (L1 Regularization)

```
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)

y_pred_lasso = lasso.predict(X_test)
```

STEP 11: Ridge Regression

Used to reduce overfitting (L2 Regularization)

```
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)

y_pred_ridge = ridge.predict(X_test)
```

STEP 12: Compare Models

```
results = pd.DataFrame({
```

```
    "Model": ["Multiple Linear", "Lasso", "Ridge"],
    "MSE": [
        mean_squared_error(y_test, y_pred_mlr),
        mean_squared_error(y_test, y_pred_lasso),
        mean_squared_error(y_test, y_pred_ridge)
    ],
    "R2 Score": [
        r2_score(y_test, y_pred_mlr),
        r2_score(y_test, y_pred_lasso),
        r2_score(y_test, y_pred_ridge)
    ]
})

results
```

	Model	MSE	R2 Score
0	Multiple Linear	0.261277	0.995752
1	Lasso	0.198435	0.996773
2	Ridge	0.220324	0.996417

Conclusion

In this experiment, **Multiple Linear Regression, Lasso Regression, and Ridge Regression** were successfully implemented on a **real-world Student Performance dataset** to predict the **Final_Score** of students based on academic factors such as **Hours Studied, Attendance, Assignment Score, and Midterm Score**.

The dataset was found to be **clean with no missing values**, so no additional preprocessing was required. The data was split into training and testing sets to ensure proper evaluation of model performance.

- **Multiple Linear Regression** provided a baseline model by considering all input features and showed a strong relationship between the independent variables and the final score.

- **Lasso Regression (L1 Regularization)** helped in controlling model complexity by reducing the impact of less important features, making it useful for feature selection.

- **Ridge Regression (L2 Regularization)** effectively handled multicollinearity among features by shrinking coefficients, resulting in a more stable and generalized model.

Based on the **Mean Squared Error (MSE)** and **R^2 score comparison**, it was observed that **Ridge Regression performed slightly better or comparable** to Multiple Linear Regression, indicating improved generalization and reduced overfitting. Lasso Regression, while slightly reducing accuracy, proved useful for simplifying the model.

Overall, this experiment demonstrates that **regularization techniques like Lasso and Ridge improve model robustness**, especially when dealing with multiple correlated features, making them suitable for real-world prediction tasks such as student performance analysis.