EXPERIMENT 2

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

1.Dataset Source: Custom dataset : Link

2. Dataset Description

The dataset used in this experiment is obtained from Fitbit activity records. It contains timestamp-based physical activity data.

Independent Variables (Features)

- Steps
- Distance
- Active Minutes

Dependent Variable (Target)

- Calories burned

Each record corresponds to a specific timestamp and contains numerical values. The dataset is continuous in nature and suitable for regression analysis, as calories burned depend on physical activity metrics.

3. Mathematical Formulation of the Algorithm

Multiple Linear Regression

The prediction equation is:

$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

Where:
x_1 = Steps
x_2 = Distance
x_3 = Active Minutes

The objective is to minimize Mean Squared Error:

$MSE = (1/n) \Sigma (y - \hat{y})^2$

Ridge Regression (L2 Regularization)

Ridge modifies the cost function as:

Cost = MSE + λ Σ β²

It reduces the magnitude of coefficients to prevent overfitting.

Lasso Regression (L1 Regularization)

Lasso modifies the cost function as:

Cost = MSE + λ Σ |β|

It shrinks coefficients and may reduce some to zero.

4. Algorithm Limitations

- Assumes a linear relationship between features and target.
- Sensitive to outliers.
- Not suitable for complex non-linear patterns.
- Regularization may shrink important coefficients if not properly chosen.

5. Methodology / Workflow

1. Loaded Fitbit CSV files.
2. Selected timestamp and numeric columns.
3. Merged datasets using timestamp.
4. Removed missing values.
5. Selected features (Steps, Distance, Active Minutes) and target (Calories).
6. Split data into training and testing sets (80–20).
7. Trained Linear, Ridge, and Lasso regression models.
8. Evaluated models using MSE and R² score.

Workflow:

Data Loading → Data Cleaning → Merging → Feature Selection → Train-Test Split → Model Training → Evaluation

6. Performance Analysis

Model performance was evaluated using:

- Mean Squared Error (MSE)
- R^2 Score

All three models produced similar results due to strong linear correlation between activity metrics and calories burned. Ridge and Lasso slightly reduced coefficient values compared to Linear Regression.

7. Hyperparameter Tuning

In this experiment, a fixed alpha value was used for Ridge and Lasso regression. No multiple hyperparameter values were tested. The models were trained using the default regularization settings to compare their behavior with standard Linear Regression.

Code:

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, r2_score

def load_and_clean(file_path, new_column_name):
    df = pd.read_csv(file_path)


    if "timestamp" not in df.columns:
        raise Exception(f"'timestamp' column not found in {file_path}")

    numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns

    if len(numeric_cols) == 0:
        raise Exception(f"No numeric column found in {file_path}")

    value_column = numeric_cols[0]

    df = df[["timestamp", value_column]]
    df = df.rename(columns={value_column: new_column_name})

    df["timestamp"] = pd.to_datetime(df["timestamp"])

    return df

steps = load_and_clean("steps.csv", "steps")
distance = load_and_clean("distance.csv", "distance")
active = load_and_clean("active_minutes.csv", "active_minutes")
calories = load_and_clean("calories.csv", "calories")


data = steps.merge(distance, on="timestamp") \
            .merge(active, on="timestamp") \
            .merge(calories, on="timestamp")

data = data.dropna()
```

```python
print("Merged Data Sample:")
print(data.head())


X = data[["steps", "distance", "active_minutes"]]
y = data["calories"]


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


linear_model = LinearRegression()
ridge_model = Ridge(alpha=1.0)
lasso_model = Lasso(alpha=0.1)

linear_model.fit(X_train, y_train)
ridge_model.fit(X_train, y_train)
lasso_model.fit(X_train, y_train)


y_pred_linear = linear_model.predict(X_test)
y_pred_ridge = ridge_model.predict(X_test)
y_pred_lasso = lasso_model.predict(X_test)


def evaluate(name, y_test, y_pred):
    print(f"\n{name}")
    print("MSE:", mean_squared_error(y_test, y_pred))
    print("R2 Score:", r2_score(y_test, y_pred))

evaluate("Linear Regression", y_test, y_pred_linear)
evaluate("Ridge Regression", y_test, y_pred_ridge)
evaluate("Lasso Regression", y_test, y_pred_lasso)
```

```
print("\nCoefficient Comparison")
print("Features:", X.columns.tolist())
print("Linear:", linear_model.coef_)
print("Ridge :", ridge_model.coef_)
print("Lasso :", lasso_model.coef_)

plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred_linear, label="Linear")
plt.scatter(y_test, y_pred_ridge, label="Ridge")
plt.scatter(y_test, y_pred_lasso, label="Lasso")

plt.xlabel("Actual Calories")
plt.ylabel("Predicted Calories")
plt.title("Regression Model Comparison")
plt.legend()
plt.show()
```

Output: