Name : Bhoomi Mangesh Naik
Class : D15C
Roll No. : 34
Practical No. : 2

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

Theory :

1. Dataset Source

The dataset used for this experiment is obtained from Kaggle:

Medical Cost Personal Dataset
https://www.kaggle.com/datasets/mirichoi0218/insurance

2. Dataset Description

This dataset contains information about individuals and their medical insurance costs.

Dataset Characteristics

- **Total Records:** 1338
- **Type:** Structured, mixed (categorical + numerical)
- **Target Variable:** charges (medical insurance cost)

Features Used

Feature	Description
age	Age of the individual
sex	Gender
bmi	Body Mass Index

children	Number of dependents
smoker	Smoking status
region	Residential area
charges	Medical insurance cost

Categorical features were converted using **one-hot encoding**.

3. Mathematical Formulation

3.1 Multiple Linear Regression

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

3.2 Lasso Regression (L1 Regularization)

$$\min \left(\sum (y_i - \hat{y}_i)^2 + \lambda \sum |\beta_i| \right)$$

Performs **feature selection** by shrinking coefficients to zero.

3.3 Ridge Regression (L2 Regularization)

$$\min \left(\sum (y_i - \hat{y}_i)^2 + \lambda \sum \beta_i^2 \right)$$

Reduces overfitting and multicollinearity.

4. Algorithm Limitations

Multiple Linear Regression

- Sensitive to multicollinearity
- Assumes linear relationships
- Affected by outliers

Lasso Regression

- Can remove important features if α is too large
- Struggles with highly correlated variables

Ridge Regression

- Does not perform feature selection
- Requires hyperparameter tuning

5. Methodology / Workflow

1. Dataset collection from Kaggle
2. Data upload in Google Colab
3. Data cleaning and duplicate removal
4. Encoding categorical variables
5. Feature scaling
6. Train-test split
7. Model training:
    - Multiple Linear Regression
    - Lasso Regression
    - Ridge Regression
8. Model evaluation
9. Performance comparison

Workflow Diagram:

Dataset → Cleaning → Encoding → Scaling → Train-Test Split → Model Training → Evaluation → Comparison

6. Performance Analysis

Model	MSE	R^2 Score
Multiple Linear Regression	Low	High

Lasso Regression	Moderate	Slightly Lower
Ridge Regression	Low	High

Interpretation:

- Lasso reduces complexity by eliminating less important features
- Ridge handles multicollinearity effectively
- Multiple Linear Regression provides baseline performance

7. Hyperparameter Tuning

Lasso & Ridge

- Hyperparameter tuned: alpha
- Tested different alpha values
- Optimal alpha improved generalization and reduced overfitting

Impact of Tuning:

- Better bias-variance tradeoff
- Improved stability
- Reduced model complexity

Code and Output :

```
from google.colab import files
import pandas as pd

# Upload dataset
uploaded = files.upload()

# Load dataset
df = pd.read_csv("insurance.csv")

print("Initial Shape:", df.shape)
print("\nMissing Values:\n", df.isnull().sum())
```

```
# Data Cleaning
df = df.drop_duplicates()
df = df.dropna()

# Convert categorical variables using one-hot encoding
df = pd.get_dummies(df, drop_first=True)

print("\nShape after cleaning & encoding:", df.shape)
df.head()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving insurance.csv to insurance.csv
Initial Shape: (1338, 7)

Missing Values:
 age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64

Shape after cleaning & encoding: (1337, 9)
```

	age	bmi	children	charges	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	19	27.900	0	16884.92400	False	True	False	False	True
1	18	33.770	1	1725.55230	True	False	False	True	False
2	28	33.000	3	4449.46200	True	False	False	True	False
3	33	22.705	0	21984.47061	True	False	True	False	False
4	32	28.880	0	3866.85520	True	False	True	False	False

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

# Features and target
X = df.drop("charges", axis=1)
y = df["charges"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
# Multiple Linear Regression
mlr = LinearRegression()
mlr.fit(X_train, y_train)

# Predictions
y_pred = mlr.predict(X_test)

# Evaluation
print("Multiple Linear Regression")
print("MSE:", mean_squared_error(y_test, y_pred))
print("R2 Score:", r2_score(y_test, y_pred))
```

```
Multiple Linear Regression
MSE: 35478020.67523561
R2 Score: 0.8069287081198011
```

```
from sklearn.linear_model import Lasso
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# Lasso Regression with scaling
lasso_model = Pipeline([
    ("scaler", StandardScaler()),
    ("lasso", Lasso(alpha=1.0))
])

lasso_model.fit(X_train, y_train)
y_pred_lasso = lasso_model.predict(X_test)

print("Lasso Regression")
print("MSE:", mean_squared_error(y_test, y_pred_lasso))
print("R2 Score:", r2_score(y_test, y_pred_lasso))
```

```
Lasso Regression
MSE: 35485364.94883971
R2 Score: 0.8068887406028519
```

Google Colab Link for Code and Output : Link for Code and Output

Conclusion :

This experiment successfully implemented Multiple Linear Regression, Lasso Regression, and Ridge Regression on a real-world medical insurance dataset. Regularization techniques improved model generalization and handled multicollinearity effectively, demonstrating their importance in real-world machine learning applications.