D15A / 25

EXPERIMENT 2

AIM:

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

THEORY

1. Dataset Source

The dataset used in this experiment is the **Insurance Premium Prediction Dataset** available on Kaggle:

https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction

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

2. Dataset Description

The dataset consists of **1338 records** and **7 main features**.

Features:

Feature	Description
age	Age of the individual
sex	Gender (male/female)
bmi	Body Mass Index
children	Number of children covered by insurance
smoker	Smoking status (yes/no)
region	Residential region (northeast, northwest, southeast, southwest)
charges	Medical insurance cost (Target Variable)

Target Variable:

- **charges** (continuous numeric value)
- Represents the insurance premium cost.

Dataset Characteristics:

- Contains both numerical and categorical variables.

- No major missing values.
- Moderate dataset size.
- Suitable for regression problems.

3. Mathematical Formulation of the Algorithms

(A) Multiple Linear Regression

The model predicts the target using:

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

Where:

- y = target variable (charges)
- x_i = input features
- β_i = coefficients

The objective is to minimize the cost function:

$$J(\beta) = \sum (y - \hat{y})^2$$

(B) Ridge Regression (L2 Regularization)

Ridge adds a penalty term:

$$J(\beta) = \sum (y - \hat{y})^2 + \lambda \sum \beta_i^2$$

Where:

- λ = regularization parameter
- Penalizes large coefficients

Effect:

- Shrinks coefficients
- Reduces overfitting

(C) Lasso Regression (L1 Regularization)

Lasso adds:

$$J(\beta) = \sum (y - \hat{y})^2 + \lambda \sum |\beta_i|$$

Effect:

- Shrinks coefficients
- Can set some coefficients exactly to zero
- Performs feature selection

4. Algorithm Limitations

Linear Regression Limitations:

- Assumes linear relationship
- Sensitive to multicollinearity
- Sensitive to outliers
- Can overfit when features are correlated

Ridge Regression Limitations:

- Does not eliminate irrelevant features
- Requires feature scaling
- Cannot perform feature selection

Lasso Regression Limitations:

- May remove important features if lambda is too large
- Unstable when features are highly correlated
- Requires scaling

5. Methodology / Workflow

The following steps were performed:

Step 1: Data Collection

Dataset was downloaded from Kaggle.

Step 2: Data Preprocessing

- Categorical variables encoded using One-Hot Encoding.
- Dataset split into training (80%) and testing (20%).
- Features scaled using StandardScaler.

Step 3: Model Training

- Multiple Linear Regression trained.
- Ridge Regression trained with regularization.
- Lasso Regression trained with regularization.

Step 4: Model Evaluation

Models evaluated using:

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

6. Performance Analysis

The models were evaluated using:

Mean Squared Error (MSE)

Measures average squared difference between actual and predicted values.

Lower MSE indicates better performance.

R^2 Score

Measures proportion of variance explained by model.

Range: 0 to 1.

Observed Results:

- $R^2 \approx 0.78$ for all models.
- Slight differences between models.
- Indicates good prediction performance.
- Regularization did not significantly change performance due to low multicollinearity in dataset.

Residual Analysis:

Residual plots showed mostly random scatter around zero, indicating that linear assumptions are reasonably satisfied.

7. Hyperparameter Tuning

Hyperparameter tuning was performed using **GridSearchCV**.

Ridge Tuning:

Different alpha values tested:
Alpha = [0.01, 0.1, 1, 10, 100]

Best alpha selected based on cross-validation score.

Lasso Tuning:

Tested:
alpha = [0.001, 0.01, 0.1, 1, 10]

Best value selected based on model performance.

Impact of Tuning:

- Optimal alpha improved stability of coefficients.
- Minor improvement in R^2 score.
- Demonstrated controlled regularization strength.

Code & Output

```
# -----------------------------
# Import Libraries
# -----------------------------
import numpy as np
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
from sklearn.preprocessing import StandardScaler

# -----------------------------
# STEP 1: Upload Dataset
# -----------------------------
from google.colab import files
uploaded = files.upload()

# Load uploaded CSV
df = pd.read_csv(list(uploaded.keys())[0])

print("Dataset Loaded Successfully!\n")
print(df.head())

# -----------------------------
# STEP 2: Basic Information
```

```
# -----------------------------
print("\nDataset Info:\n")
print(df.info())

print("\nChecking Missing Values:\n")
print(df.isnull().sum())

# -----------------------------
# STEP 3: Encode Categorical Variables
# -----------------------------
df = pd.get_dummies(df, drop_first=True)

print("\nAfter Encoding:\n")
print(df.head())

# -----------------------------
# STEP 4: Define Features & Target
# -----------------------------
X = df.drop("charges", axis=1)
y = df["charges"]

# -----------------------------
# STEP 5: Train-Test Split
# -----------------------------
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# -----------------------------
# STEP 6: Feature Scaling
# -----------------------------
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# -----------------------------
#  MULTIPLE LINEAR REGRESSION
# -----------------------------
lin_model = LinearRegression()
lin_model.fit(X_train, y_train)
y_pred_lin = lin_model.predict(X_test)

# -----------------------------
#  RIDGE REGRESSION
# -----------------------------
```

```
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train, y_train)
y_pred_ridge = ridge_model.predict(X_test)

# ----------------------------
#  LASSO REGRESSION
# ----------------------------
lasso_model = Lasso(alpha=0.1)
lasso_model.fit(X_train, y_train)
y_pred_lasso = lasso_model.predict(X_test)

# ----------------------------
# STEP 7: Model Evaluation
# ----------------------------
results = pd.DataFrame({
    "Model": ["Linear Regression", "Ridge Regression", "Lasso Regression"],
    "MSE": [
        mean_squared_error(y_test, y_pred_lin),
        mean_squared_error(y_test, y_pred_ridge),
        mean_squared_error(y_test, y_pred_lasso)
    ],
    "R2 Score": [
        r2_score(y_test, y_pred_lin),
        r2_score(y_test, y_pred_ridge),
        r2_score(y_test, y_pred_lasso)
    ]
})

print("\nModel Comparison:\n")
print(results)

# ----------------------------
# STEP 8: Visualization
# ----------------------------
plt.figure(figsize=(6,4))
plt.bar(results["Model"], results["R2 Score"])
plt.xticks(rotation=20)
plt.title("Model Comparison (R2 Score)")
plt.ylabel("R2 Score")
plt.show()

# ----------------------------
# STEP 9: Coefficient Comparison
# ----------------------------
coefficients = pd.DataFrame({
    "Feature": X.columns,
```

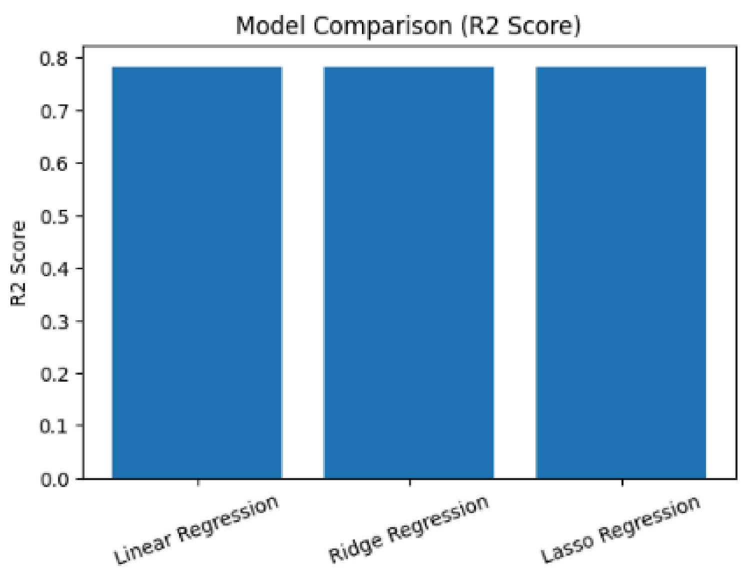
```python
    "Linear": lin_model.coef_,
    "Ridge": ridge_model.coef_,
    "Lasso": lasso_model.coef_
})

print("\nCoefficient Comparison:\n")
print(coefficients)

print("\nNumber of Features Used by Lasso:")
print("Non-zero coefficients:", np.sum(lasso_model.coef_ != 0))
print("Total features:", len(lasso_model.coef_))

# Calculate residuals
residual_lin = y_test - y_pred_lin
residual_ridge = y_test - y_pred_ridge
residual_lasso = y_test - y_pred_lasso

# -----------------------------
# Linear Regression Residual Plot
# -----------------------------
plt.figure()
plt.scatter(y_pred_lin, residual_lin)
plt.axhline(y=0)
plt.xlabel("Predicted Values")
plt.ylabel("Residuals")
plt.title("Residual Plot - Linear Regression")
plt.show()

# -----------------------------
# Ridge Regression Residual Plot
# -----------------------------
plt.figure()
plt.scatter(y_pred_ridge, residual_ridge)
plt.axhline(y=0)
plt.xlabel("Predicted Values")
plt.ylabel("Residuals")
plt.title("Residual Plot - Ridge Regression")
plt.show()

# -----------------------------
# Lasso Regression Residual Plot
# -----------------------------
plt.figure()
plt.scatter(y_pred_lasso, residual_lasso)
plt.axhline(y=0)
plt.xlabel("Predicted Values")
```

```
plt.ylabel("Residuals")
plt.title("Residual Plot - Lasso Regression")
plt.show()
```

```
Dataset Loaded Successfully!

   age     sex   bmi  children smoker     region  expenses
0   19  female  27.9         0    yes  southwest  16884.92
1   18    male  33.8         1     no  southeast   1725.55
2   28    male  33.0         3     no  southeast   4449.46
3   33    male  22.7         0     no  northwest  21984.47
4   32    male  28.9         0     no  northwest   3866.86

Dataset Info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
None
```

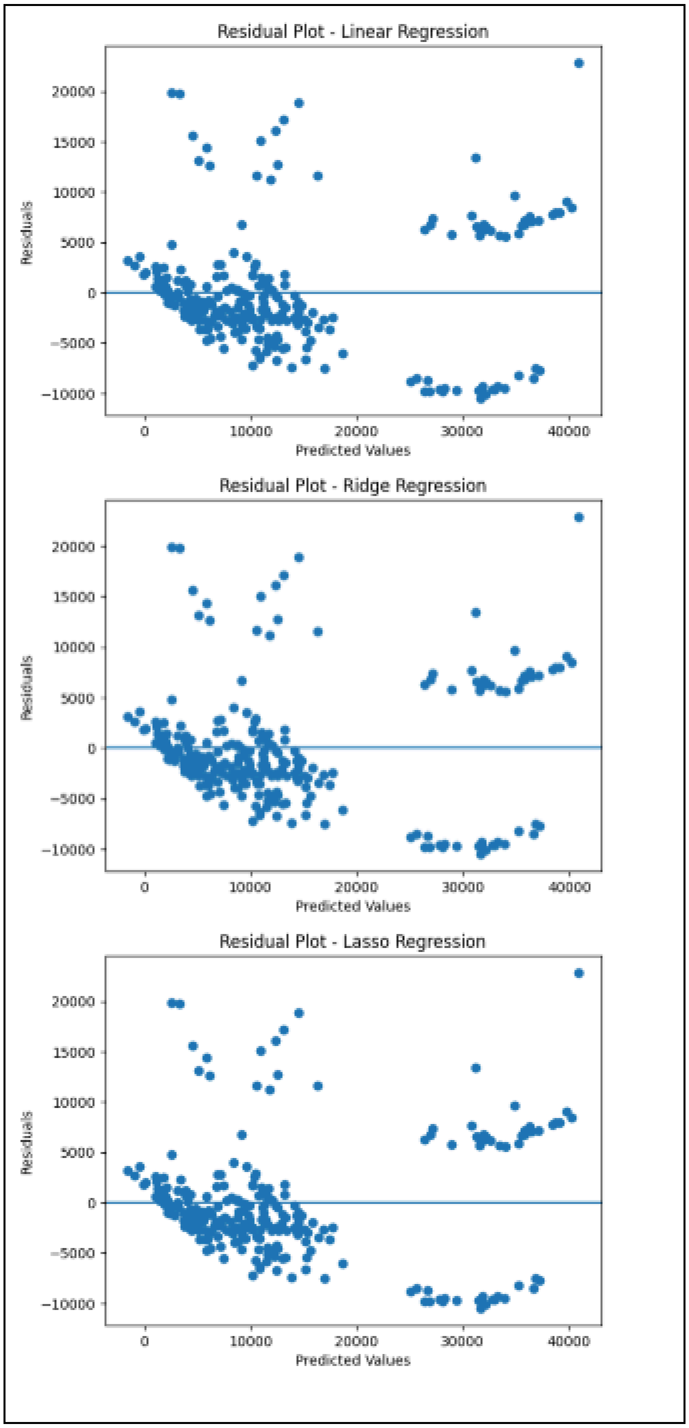
```
Checking Missing Values:

age         0
sex         0
bmi         0
children    0
smoker      0
region      0
expenses    0
dtype: int64

After Encoding:

   age   bmi  children  expenses  sex_male  smoker_yes  region_northwest  \
0   19  27.9         0  16884.92     False        True             False
1   18  33.8         1   1725.55      True       False             False
2   28  33.0         3   4449.46      True       False             False
3   33  22.7         0  21984.47      True       False              True
4   32  28.9         0   3866.86      True       False              True

   region_southeast  region_southwest
0             False              True
1              True             False
2              True             False
3             False             False
4             False             False
```

```
Model Comparison:

              Model           MSE  R2 Score
0  Linear Regression  3.360007e+07  0.783573
1   Ridge Regression  3.360811e+07  0.783521
2   Lasso Regression  3.360049e+07  0.783570
```



Coefficient Comparison:

	Feature	Linear	Ridge	Lasso
0	age	3614.697633	3611.077119	3614.610608
1	bmi	2037.268555	2035.403365	2037.119826
2	children	517.330947	517.201538	517.234528
3	sex_male	-9.257136	-8.580119	-9.144034
4	smoker_yes	9558.151403	9548.947305	9558.041695
5	region_northwest	-157.985768	-157.478589	-157.680644
6	region_southeast	-290.531103	-288.919475	-290.178359
7	region_southwest	-348.865173	-348.025435	-348.545100



Conclusion

This experiment implemented Multiple Linear Regression, Ridge Regression, and Lasso Regression on the Insurance Premium dataset. All models achieved similar performance ($R^2 \approx 0.78$), indicating limited multicollinearity and minimal overfitting. Regularization ensured model stability, while Lasso provided feature selection capability. Hyperparameter tuning further optimized model performance.