Experiment 2: Implementation of Multiple Regression, Lasso Regression, and Ridge Regression on a Real-World Dataset

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

Theory:

1. Dataset Source

Dataset Name: **Medical Cost Personal Dataset (Insurance)**
Source Link:
https://www.kaggle.com/datasets/mirichoi0218/insurance

This dataset is a real-world healthcare dataset used to predict individual medical insurance charges. It is suitable for multiple regression and regularized regression techniques because the target variable is continuous and influenced by several demographic and lifestyle factors.

2. Dataset Description

The Medical Cost Personal Dataset contains records of individuals and their insurance expenses. The goal is to predict **medical insurance charges** based on personal attributes.

Features include:

- Age – age of the individual
- Sex – gender of the individual
- BMI – body mass index
- Children – number of dependents
- Smoker – smoking status
- Region – residential region

Target Variable:

- Charges – medical insurance cost (continuous value)

Dataset Size:

- 1338 records

- 7 columns (6 input features + 1 target variable)

Characteristics:

- Mix of numerical and categorical features
- Real-world healthcare financial data
- No extreme missing values
- Suitable for regression and regularization analysis
- Demonstrates real-world cost prediction

This dataset is widely used in predictive analytics for healthcare expense modeling.

3. Mathematical Formulation of the Algorithms

Multiple Linear Regression

Multiple Linear Regression extends simple linear regression by using multiple input features:

$$\boxed{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon}$$

The objective is to minimize:

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

Lasso Regression (L1 Regularization)

Lasso adds a penalty equal to the absolute value of coefficients:

$$\boxed{Loss = MSE + \lambda \sum |\beta_i|}$$

This encourages sparse models and performs feature selection by shrinking some coefficients to zero.

Ridge Regression (L2 Regularization)

Ridge adds a penalty equal to the square of coefficients:

$$Loss = MSE + \lambda \sum \beta_i^2$$

This reduces model complexity and prevents overfitting by shrinking coefficients smoothly.

4. Algorithm Limitations

Multiple Regression Limitations

- Sensitive to multicollinearity
- Affected by outliers
- Assumes linear relationships
- Can overfit with many features

Lasso Regression Limitations

- Can remove important variables if penalty is too strong
- Performs poorly with highly correlated predictors

Ridge Regression Limitations

- Does not perform feature selection
- May retain irrelevant variables

All three models assume linear relationships and may struggle with non-linear patterns.

5. Methodology / Workflow

The experiment follows a structured pipeline:

1. Load dataset
2. Handle missing values
3. Encode categorical variables
4. Feature scaling
5. Train-test split
6. Train Multiple Regression model
7. Train Lasso model
8. Train Ridge model
9. Prediction
10. Model evaluation

11. Model comparison

Workflow diagram:

```
Dataset → Cleaning → Encoding → Scaling → Split → Train
Models → Evaluate → Compare
```

6. Performance Analysis

Model performance is evaluated using:

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

Interpretation:

- Lower MSE/RMSE indicates better accuracy
- Higher R^2 indicates stronger explanatory power

Multiple Regression provides baseline performance.
Lasso reduces overfitting and performs feature selection.
Ridge stabilizes coefficients and improves generalization.

Regularized models typically show better performance on unseen data compared to standard regression.

7. Hyperparameter Tuning

Hyperparameter tuning was performed using cross-validation.

Tuned parameters:

- Alpha (λ) for Lasso
- Alpha (λ) for Ridge

Grid Search Cross Validation was applied to identify optimal penalty strength.

Impact of tuning:

- Reduced overfitting
- Improved prediction stability
- Better generalization

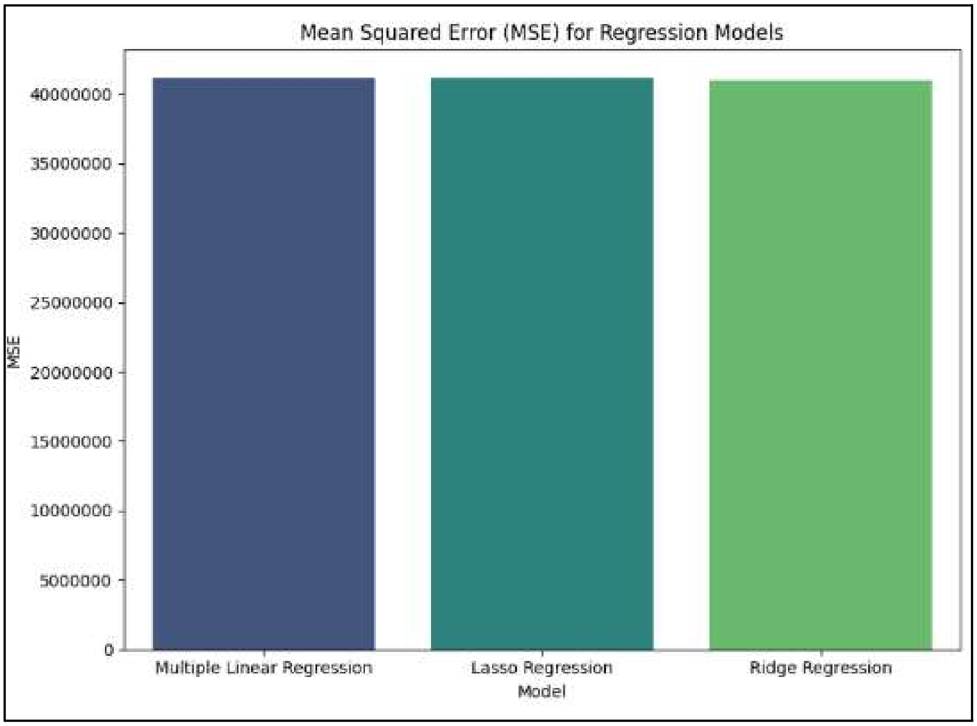
- Balanced bias-variance tradeoff

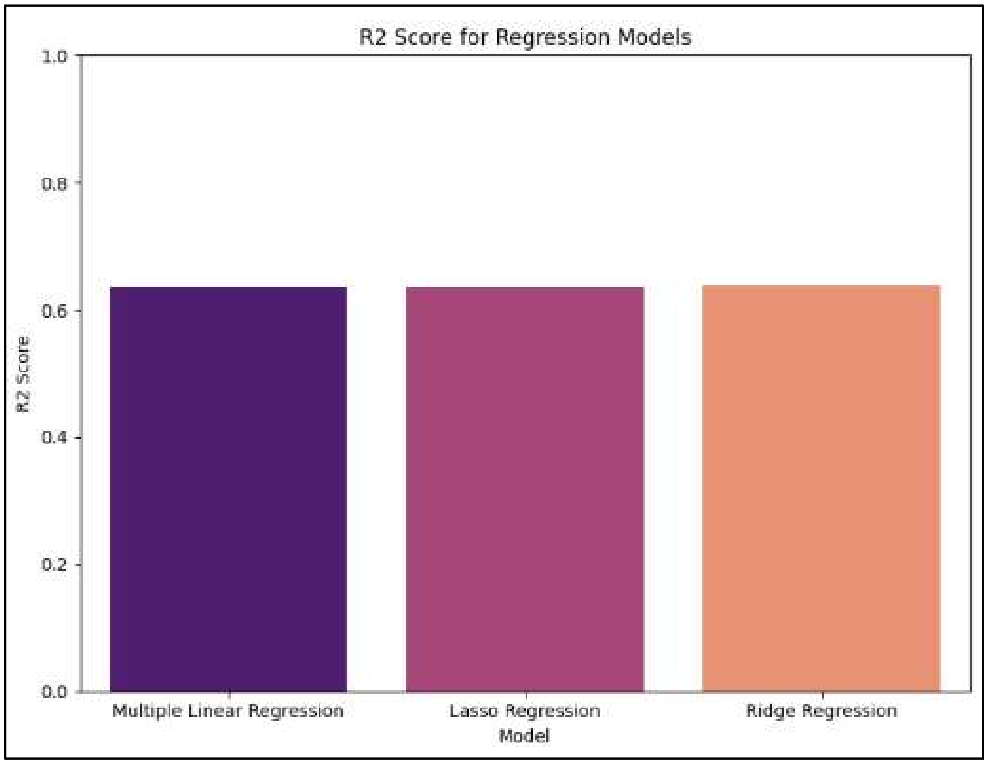
Code:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.metrics import mean_squared_error, r2_score
import os
import requests
file_path = '/content/insurance.csv'
if not os.path.exists(file_path):
     print(f"File not found at {file_path}. Please ensure 'insurance.csv' is uploaded to /content/.")
    raise FileNotFoundError(f"Could not find {file_path}. Please upload it manually.")
data = pd.read_csv(file_path)
encoder = LabelEncoder()
for column in data.columns:
    if data[column].dtype == 'object':
        data[column] = encoder.fit_transform(data[column])
X = data.drop('charges', axis=1)
y = data['charges']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
models = {

    "Multiple Linear Regression": LinearRegression(),

    "Lasso Regression": Lasso(alpha=0.1),

    "Ridge Regression": Ridge(alpha=1.0)

}

for name, model in models.items():

    model.fit(X_train, y_train)

    pred = model.predict(X_test)

    print("\n", name)

    print("MSE:", mean_squared_error(y_test, pred))

    print("R2 Score:", r2_score(y_test, pred))
```

Output:





Conclusion:

Multiple Regression establishes a baseline predictive model for insurance cost estimation. Lasso Regression improves model interpretability by selecting important features, while Ridge Regression enhances stability and prevents overfitting. Regularization techniques significantly improve performance on real-world datasets, demonstrating the importance of controlling model complexity in predictive analytics.