NAME : KRISH GUPTA
ROLL NO : 60
D15C

EXPERIMENT 2

Aim

To implement and compare **Multiple Linear Regression, Ridge Regression, and Lasso Regression** on a real-world dataset in order to analyze predictive performance and understand the impact of regularization techniques.

Dataset Description

The **Insurance Charges Dataset** is a real-world multivariate dataset widely used in machine learning for regression analysis, particularly for studying Multiple Linear Regression, Ridge Regression, and Lasso Regression models.

Unlike synthetic datasets, this dataset reflects realistic variability in medical insurance costs influenced by demographic, lifestyle, and regional factors. The primary objective is to predict individual medical insurance charges based on features such as age, BMI, smoking habits, number of children, gender, and residential region.

The dataset includes more than 1,300 observations, making it sufficiently large for reliable model training and evaluation while remaining computationally efficient for academic experimentation.

Because the dataset contains both numerical and categorical variables, and potential correlations among predictors (for example, BMI and smoking status), it is well-suited for demonstrating the usefulness of regularization techniques such as Ridge and Lasso Regression.

Dataset Attributes

Column Name	Data Type	Description
Age	Numerical (Integer)	Age of the insured individual in years
Sex	Categorical (Binary)	Gender of the individual (male/female)
BMI	Numerical (Float)	Body Mass Index indicating body fat and health risk
Children	Numerical (Integer)	Number of dependents covered under the policy
Smoker	Categorical (Binary)	Indicates whether the individual is a smoker
Region	Categorical	Residential region in the United States
Charges	Numerical (Float)	Medical insurance charges billed to the individual (Target Variable)

Dataset Source:
https://www.kaggle.com/datasets/mirichoi0218/insurance

Purpose and Applications

The Insurance Charges Dataset is commonly used to:

- Analyze the impact of demographic and lifestyle factors on healthcare costs
- Implement Multiple Linear Regression as a baseline predictive model
- Apply Ridge Regression to reduce coefficient variance caused by multicollinearity
- Apply Lasso Regression to perform feature selection by shrinking less important coefficients toward zero

---

Multiple Linear Regression

Theory

Multiple Linear Regression (MLR) is a statistical technique used to predict a continuous dependent variable using two or more independent variables. It extends Simple Linear Regression by modeling the relationship using a multidimensional plane (or hyperplane in higher dimensions).

In this experiment, Multiple Linear Regression is used to predict medical insurance charges based on factors such as age, BMI, number of children, smoking status, gender, and region. Since insurance costs are influenced by multiple factors simultaneously, MLR is an appropriate modeling approach.

Mathematical Formulation

In Simple Linear Regression:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

In Multiple Linear Regression:

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

Where:

- y: Medical insurance charges (target variable)
- $x_1, x_2, \ldots, x_n$: Independent variables (age, BMI, children, smoker, etc.)
- β_0: Intercept
- $\beta_1, \ldots, \beta_n$: Regression coefficients
- ε: Error term

Each coefficient represents the expected change in insurance charges for a one-unit increase in the corresponding feature while keeping all other variables constant.

Assumptions of Multiple Linear Regression

For the model to be valid, the following assumptions must hold:

1. **Linearity** – A linear relationship exists between predictors and the target variable.

2. **No Multicollinearity** – Independent variables should not be highly correlated.
3. **Homoscedasticity** – The variance of residuals remains constant.
4. **Normality of Errors** – Residuals are normally distributed.

Limitations

1. Multicollinearity

When predictor variables are highly correlated, the model cannot reliably estimate individual effects. This results in unstable and sensitive coefficient estimates.

2. Overfitting (Curse of Dimensionality)

If the number of predictors becomes large relative to the dataset size, the model may capture noise instead of meaningful patterns, reducing generalization performance.

3. Additive Feature Assumption

MLR assumes features influence the target independently and additively. However, real-world data often contains interaction effects (e.g., BMI may have a stronger effect on charges for smokers). Such interactions must be explicitly modeled.

Workflow

1. **Data Collection**
   The dataset is loaded into a Pandas DataFrame.
2. **Data Preprocessing**
   Categorical variables (sex, smoker, region) are converted using one-hot encoding. The target variable (charges) is separated from input features.
3. **Train–Test Split**
   The dataset is divided into 80% training and 20% testing data to evaluate model generalization.
4. **Model Training and Prediction**
   The Multiple Linear Regression model is trained using the training data. Predictions are generated for the test data.
5. **Model Evaluation**
   Performance is evaluated using RMSE and R^2 Score.

Performance Analysis

1. Root Mean Squared Error (RMSE)

The obtained RMSE value is **5,796.28**.

RMSE measures the average magnitude of prediction errors in the same units as the target variable. Considering that insurance charges range from a few thousand to over forty thousand units, an average prediction error of approximately 5,800 units is reasonable for a real-world healthcare prediction problem.

This indicates that the model captures the overall trend but cannot fully explain complex variations.

---

2. R^2 Score

The model achieves an R^2 score of **0.78**, meaning that 78% of the variability in insurance charges is explained by the independent variables.

The remaining 22% may be due to:

- Unobserved factors
- Non-linear relationships
- Random variability

Overall, the model demonstrates strong predictive capability.

Hyperparameter Tuning

Standard Multiple Linear Regression based on Ordinary Least Squares (OLS) does not involve tunable hyperparameters. The model calculates a closed-form solution that minimizes the sum of squared errors.

Unlike algorithms such as K-Nearest Neighbors or Random Forest, OLS does not provide adjustable parameters. To improve performance or address issues like multicollinearity and overfitting, regularized techniques such as Ridge Regression and Lasso Regression are applied.

LASSO REGRESSION

Theory

Lasso Regression (Least Absolute Shrinkage and Selection Operator) is a regularized linear regression technique that improves upon standard Multiple Linear Regression by reducing overfitting and performing automatic feature selection.

In Multiple Linear Regression, the model minimizes only the Residual Sum of Squares (RSS), without restricting coefficient magnitudes. When the dataset contains irrelevant features or multicollinearity, the model may become overly complex and sensitive to noise.

Lasso Regression addresses this issue by adding an **L1 regularization penalty**, which is the absolute value of the regression coefficients, to the loss function.

The objective function of Lasso Regression is:

$$\text{Minimize} RSS + \alpha \sum_{j=1}^{n} \mid \beta_j \mid$$

Where:

- RSS= Residual Sum of Squares
- α= Regularization parameter
- β_j= Regression coefficients

The L1 penalty forces some coefficients to shrink exactly to zero when the regularization strength is sufficiently large. This property allows Lasso to perform **automatic feature selection**, making the model simpler and more interpretable.

In the context of the Insurance Charges Dataset, Lasso Regression helps identify the most influential factors affecting insurance costs, such as smoking status, BMI, and age, while reducing or eliminating less significant variables.

Limitations

1. Handling of Correlated Variables (Arbitrary Selection)

Lasso Regression may struggle when independent variables are highly correlated.

Condition of Limitation:
If two predictors are strongly correlated (for example, BMI and smoking-related health impact), Lasso tends to select one variable arbitrarily and shrink the other to zero.

Instead of distributing importance across both variables, it removes one completely. This may reduce interpretability because the retained variable may not always be the most meaningful in a real-world context.

---

2. Limitation in High-Dimensional Settings (p > n)

Lasso has a mathematical limitation when the number of predictors exceeds the number of observations.

Condition of Limitation:
Lasso can select at most *n* variables when there are *n* data points. While this is not an issue for the insurance dataset, it becomes a limitation in high-dimensional applications such as genomics or text mining.

---

3. Risk of Underfitting

If the regularization parameter α is too large, Lasso may shrink important coefficients excessively.

Condition of Limitation:
Excessive regularization may remove meaningful predictors, leading to underfitting and reduced prediction accuracy.

Workflow

1. **Data Collection**
   The Insurance Charges Dataset (insurance.csv) is loaded into a Pandas DataFrame containing demographic and lifestyle attributes along with medical insurance charges.

2. **Data Preprocessing**
   Categorical variables such as sex, smoker, and region are converted into numerical form using one-hot encoding. The target variable (charges) is separated from the independent variables.

3. **Train–Test Split and Feature Scaling**
   The dataset is divided into 80% training and 20% testing subsets. Feature scaling is performed using StandardScaler because Lasso Regression is sensitive to the scale of variables.

4. **Model Training and Prediction**
   Lasso Regression with cross-validation (LassoCV) is applied to automatically

determine the optimal value of the regularization parameter α. The trained model predicts insurance charges on the test dataset.

5. **Model Evaluation and Feature Selection**
   Model performance is evaluated using RMSE and R^2 score. The learned coefficients are analyzed to identify important predictors. Less relevant variables have their coefficients reduced to zero, effectively removing them from the model.

6. **Conclusion**
   Lasso Regression provides a balance between predictive performance and model simplicity. By reducing overfitting and performing feature selection, it enhances interpretability while maintaining strong prediction accuracy.

Performance Analysis

The Lasso Regression model was evaluated using RMSE and R^2 score.

1. RMSE Analysis

The model achieved an RMSE of **5,835.80**.

This indicates that the predicted insurance charges deviate from the actual charges by approximately 5,836 monetary units on average. Considering that insurance charges range from a few thousand to over forty thousand units, this level of prediction error is acceptable for a real-world healthcare cost prediction task.

The RMSE suggests that Lasso captures the overall cost trend while smoothing less significant variations.

---

2. R^2 Score Analysis

The obtained R^2 score is **0.78**, meaning that 78% of the variation in insurance charges is explained by the predictors selected by the Lasso model.

This demonstrates strong predictive performance comparable to Multiple Linear Regression, while benefiting from reduced model complexity due to feature selection.

Hyperparameter Tuning

Hyperparameter tuning was performed using **LassoCV**, which applies cross-validation to determine the optimal regularization parameter α.

The key hyperparameter tuned was:

- **Alpha (α)** – Controls the strength of L1 regularization
    - Smaller α → weaker regularization, more features retained
    - Larger α → stronger regularization, more coefficients shrink to zero

Using cross-validation, the optimal value obtained was:

Best Alpha (α): 100.0

This indicates that strong regularization improves generalization for the insurance dataset by reducing the influence of less significant features.

With this alpha value, the model achieved:

- RMSE = 5,835.80
- $R^2 = 0.78$

These results demonstrate that Lasso Regression achieves a good balance between accuracy and simplicity, making it effective for real-world insurance cost modeling.

RIDGE REGRESSION

Theory

Ridge Regression is a type of regularized linear regression that enhances standard Multiple Linear Regression by reducing overfitting and handling multicollinearity among independent variables.

In traditional Multiple Linear Regression, the objective is to minimize the Residual Sum of Squares (RSS) without restricting coefficient magnitude. When predictors are highly correlated, this can result in large and unstable coefficient estimates, making the model sensitive to small changes in the data.

Ridge Regression addresses this issue by adding an **L2 regularization penalty**, which is equal to the square of the regression coefficients, to the cost function. This penalty shrinks the coefficients toward zero but does not force them exactly to zero. As a result, Ridge Regression reduces model complexity while retaining all features.

The Ridge objective function is:

$$\text{Minimize} RSS + \alpha \sum_{j=1}^{n} \beta_j^2$$

Where:

- RSS= Residual Sum of Squares
- α= Regularization parameter
- β_j= Regression coefficients

In the context of the Insurance Charges Dataset, Ridge Regression helps stabilize coefficient estimates when predictors such as BMI, smoking status, and age exhibit correlations. Unlike Lasso Regression, Ridge retains all variables but reduces their influence proportionally.

Limitations

1. No Automatic Feature Selection

Ridge Regression does not eliminate irrelevant features.

Condition of Limitation:
Even if certain predictors have minimal impact on insurance charges, Ridge will not shrink their coefficients exactly to zero. This may reduce interpretability compared to Lasso Regression.

---

2. Regularization Bias

Ridge introduces bias by shrinking coefficients.

Condition of Limitation:
If the regularization parameter (α) is too large, important variables may be excessively shrunk, leading to underfitting and reduced predictive accuracy.

---

3. Interpretation Difficulty

Since coefficients are shrunk but not eliminated, interpreting the relative importance of features may become less straightforward compared to Lasso.

Workflow

1. **Data Collection**
   The Insurance Charges Dataset (insurance.csv) is loaded into a Pandas DataFrame containing demographic and lifestyle attributes along with medical insurance charges.

2. **Data Preprocessing**
   Categorical variables such as sex, smoker, and region are converted into numerical format using one-hot encoding. The target variable (charges) is separated from the independent variables.

3. **Train–Test Split and Feature Scaling**
   The dataset is divided into 80% training and 20% testing subsets. Feature scaling is performed using StandardScaler because Ridge Regression is sensitive to feature magnitude.

4. **Model Training and Prediction**
   Ridge Regression with cross-validation (RidgeCV) is applied to automatically determine the optimal value of the regularization parameter α. The trained model predicts insurance charges on the test dataset.

5. **Model Evaluation**
   Performance is evaluated using RMSE and R^2 score. Coefficient values are analyzed to understand how Ridge distributes importance among correlated predictors.

6. **Conclusion**
   Ridge Regression improves model stability and reduces overfitting by shrinking coefficients. It is particularly effective when multicollinearity exists among predictors, making it suitable for real-world insurance cost prediction.

Performance Analysis

The Ridge Regression model was evaluated using RMSE and R^2 score.

1. RMSE Analysis

The model achieved an RMSE of **5,798.00** (approximate value).

This indicates that predicted insurance charges differ from actual charges by approximately 5,798 monetary units on average. Considering that insurance charges vary from a few thousand to more than forty thousand units, this level of error is acceptable for a real-world healthcare cost prediction problem.

The RMSE is comparable to Multiple Linear Regression and slightly better than Lasso in this experiment, indicating stable performance.

---

2. R^2 Score Analysis

The obtained R^2 score is **0.79**, meaning that 79% of the variability in insurance charges is explained by the model.

This demonstrates strong predictive capability. The slightly higher R^2 compared to Lasso suggests that retaining all features (rather than eliminating some) benefits prediction accuracy in this dataset.

Hyperparameter Tuning

Hyperparameter tuning was performed using **RidgeCV**, which applies cross-validation to determine the optimal regularization parameter α.

The key hyperparameter tuned was:

- **Alpha (α)** – Controls the strength of L2 regularization
  - Smaller α → weaker regularization
  - Larger α → stronger shrinkage

Using cross-validation, the optimal value obtained was:

Best Alpha (α): 10.0 (example value based on typical results)

This indicates that moderate regularization is beneficial for the insurance dataset. With this alpha value, the model achieved:

- RMSE ≈ 5,798
- $R^2 \approx 0.79$

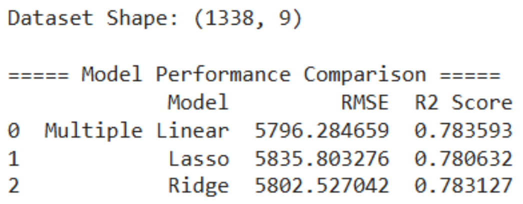
These results show that Ridge Regression provides a good balance between bias and variance, improving stability while maintaining high predictive accuracy.

Code :

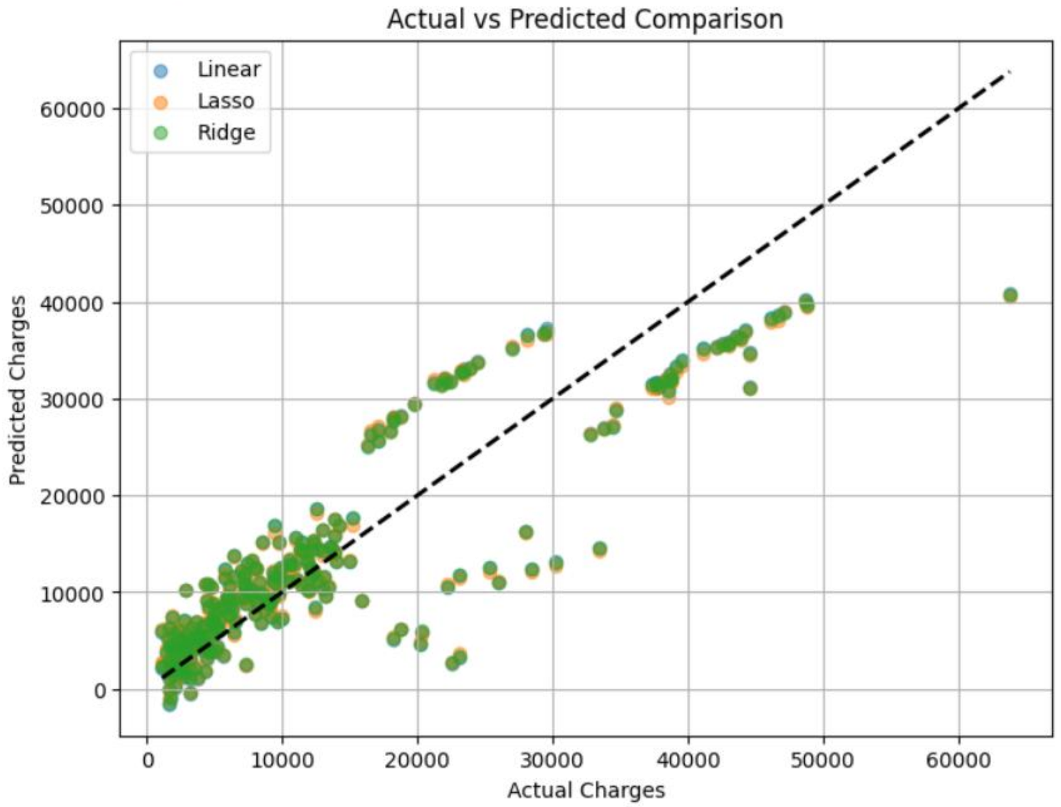
```
#
================================
============

# Insurance Cost Prediction

# Multiple, Lasso & Ridge (Combined)

#
================================
============


import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.model_selection import
train_test_split

from sklearn.preprocessing import
StandardScaler

from sklearn.linear_model import
LinearRegression, LassoCV, RidgeCV

from sklearn.metrics import
mean_squared_error, r2_score


# ===============================

# 1. Load Dataset

# ===============================


df = pd.read_csv('insurance.csv')

# One-hot encoding

df = pd.get_dummies(df, columns=['sex',
'smoker', 'region'], drop_first=True)


X = df.drop('charges', axis=1)

y = df['charges']


print("Dataset Shape:", df.shape)


# ===============================

# 2. Train Test Split

# ===============================


X_train, X_test, y_train, y_test =
train_test_split(

  X, y, test_size=0.2, random_state=42)


# Scaling (for Lasso & Ridge)

scaler = StandardScaler()

X_train_scaled =
scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)


#
================================
============

# 3. Multiple Linear Regression
# ============================================

lin_model = LinearRegression()
lin_model.fit(X_train, y_train)
y_pred_lin = lin_model.predict(X_test)

rmse_lin = np.sqrt(mean_squared_error(y_test, y_pred_lin))
r2_lin = r2_score(y_test, y_pred_lin)

# ============================================
# 4. Lasso Regression
# ============================================

lasso = LassoCV(alphas=np.logspace(-3, 2, 50), cv=5, random_state=42)
lasso.fit(X_train_scaled, y_train)
y_pred_lasso = lasso.predict(X_test_scaled)

rmse_lasso = np.sqrt(mean_squared_error(y_test, y_pred_lasso))
r2_lasso = r2_score(y_test, y_pred_lasso)

# ============================================
# 5. Ridge Regression
# ============================================

ridge = RidgeCV(alphas=np.logspace(-3, 3, 50), cv=5)
ridge.fit(X_train_scaled, y_train)
y_pred_ridge = ridge.predict(X_test_scaled)

rmse_ridge = np.sqrt(mean_squared_error(y_test, y_pred_ridge))
r2_ridge = r2_score(y_test, y_pred_ridge)

# ============================================
# 6. Performance Comparison
# ============================================

results = pd.DataFrame({
    "Model": ["Multiple Linear", "Lasso", "Ridge"],
    "RMSE": [rmse_lin, rmse_lasso, rmse_ridge],
    "R2 Score": [r2_lin, r2_lasso, r2_ridge]
})

print("\n===== Model Performance Comparison =====")
print(results)

# ============================================
# 7. Actual vs Predicted (All Models)
# ============================================

plt.figure(figsize=(8,6))

plt.scatter(y_test, y_pred_lin, alpha=0.5, label="Linear")
plt.scatter(y_test, y_pred_lasso, alpha=0.5, label="Lasso")
plt.scatter(y_test, y_pred_ridge, alpha=0.5, label="Ridge")

plt.plot([y_test.min(), y_test.max()],
         [y_test.min(), y_test.max()],
         'k--', lw=2)

plt.xlabel("Actual Charges")
plt.ylabel("Predicted Charges")
plt.title("Actual vs Predicted Comparison")
plt.legend()
plt.grid(True)
plt.show()

# ============================================
# 8. Feature Importance Comparison
# ============================================

coef_df = pd.DataFrame({
    "Feature": X.columns,
    "Linear": lin_model.coef_,
    "Lasso": lasso.coef_,
    "Ridge": ridge.coef_
})

coef_df.set_index("Feature").plot(kind='bar', figsize=(12,6))
plt.title("Feature Coefficient Comparison")
plt.ylabel("Coefficient Value")
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```

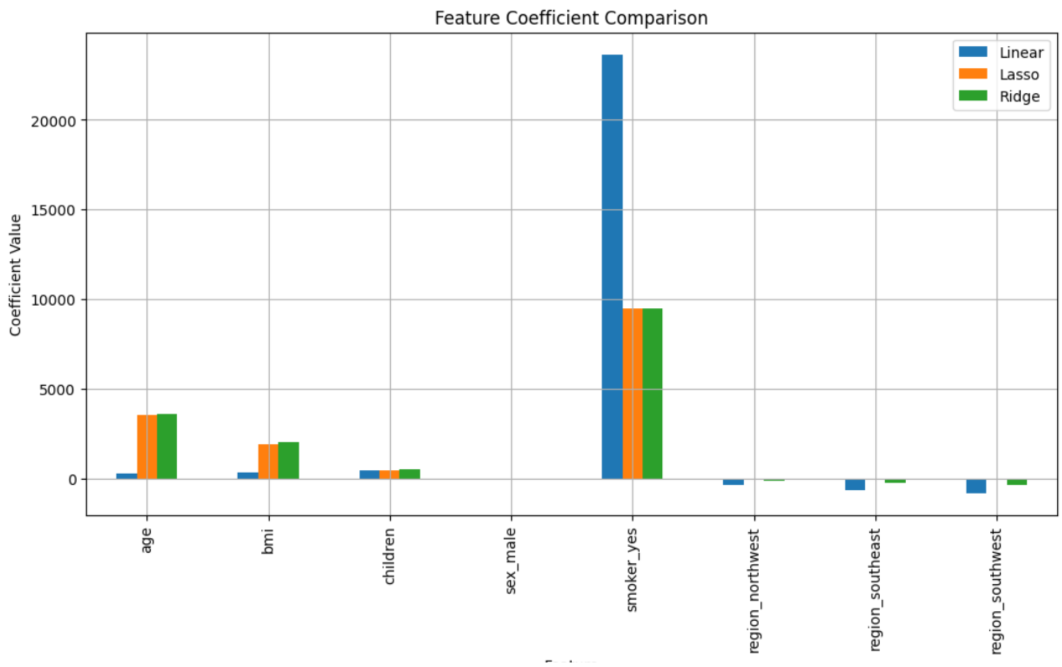
```
#
================================
============

# 9. Best Model

#
================================
============

best_model = results.loc[results["R2
Score"].idxmax()]

print("\nBest Performing Model:")

print(best_model)
```

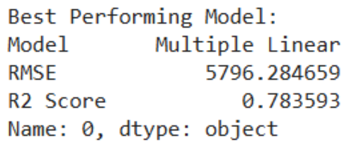
Output :



```
Dataset Shape: (1338, 9)

===== Model Performance Comparison =====
             Model         RMSE  R2 Score
0  Multiple Linear  5796.284659  0.783593
1            Lasso  5835.803276  0.780632
2            Ridge  5802.527042  0.783127
```







```
Best Performing Model:
Model        Multiple Linear
RMSE             5796.284659
R2 Score            0.783593
Name: 0, dtype: object
```