

AIM

To implement and compare Multiple Linear Regression, Ridge Regression, and Lasso Regression on a real-world healthcare dataset to predict Body Mass Index (BMI) and evaluate model performance using standard regression metrics.

OBJECTIVES

1. To understand the concept of Multiple Linear Regression and its application in predicting continuous outcomes.
 2. To implement Ridge Regression and analyze the impact of L2 regularization on model performance.
 3. To implement Lasso Regression and study the effect of L1 regularization and feature selection.
 4. To compare the performance of all three models using evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score.
 5. To analyze the role of regularization in reducing overfitting and improving model generalization.
 6. To understand the importance of feature scaling and preprocessing in regression models.
 7. To interpret model coefficients and understand their significance in prediction.
-

THEORY

Regression is a supervised machine learning technique used to model the relationship between dependent and independent variables. It is widely used for prediction and forecasting tasks. In this experiment, three types of regression techniques are used: Multiple Linear Regression, Ridge Regression, and Lasso Regression.

Multiple Linear Regression

Multiple Linear Regression is an extension of simple linear regression where the dependent variable is predicted using multiple independent variables. It assumes a linear relationship between the input variables and the output.

Mathematical representation:

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

Where:

y is the dependent variable (BMI)

x_1, x_2, \dots, x_n are independent variables

β_0 is the intercept

$\beta_1, \beta_2, \dots, \beta_n$ are coefficients

The model estimates coefficients by minimizing the Residual Sum of Squares (RSS), which is the difference between actual and predicted values. Multiple Linear Regression works well when the relationship between variables is linear and there is low multicollinearity.

However, it has limitations such as sensitivity to outliers, overfitting when too many features are present, and poor performance when features are highly correlated.

Ridge Regression (L2 Regularization)

Ridge Regression is a regularization technique used to overcome the limitations of Multiple Linear Regression, especially overfitting and multicollinearity. It introduces a penalty term to the loss function that is proportional to the square of the magnitude of coefficients.

Mathematical representation:

$$\text{Loss} = \text{RSS} + \lambda \sum (\beta^2)$$

Where:

RSS is the Residual Sum of Squares

λ (lambda) is the regularization parameter

β represents model coefficients

The penalty term discourages large coefficients, thereby reducing model complexity and variance. Ridge Regression does not eliminate any feature completely; instead, it shrinks coefficients towards zero.

Advantages:

- Reduces overfitting
- Handles multicollinearity effectively
- Improves model generalization

Limitations:

- Does not perform feature selection
 - All features remain in the model
-

Lasso Regression (L1 Regularization)

Lasso Regression is another regularization technique that adds a penalty equal to the absolute value of coefficients.

Mathematical representation:

$$\text{Loss} = \text{RSS} + \lambda \sum |\beta|$$

This penalty forces some coefficients to become exactly zero, effectively performing feature selection. As a result, Lasso produces simpler and more interpretable models.

Advantages:

- Performs feature selection
- Produces sparse models
- Reduces model complexity

Limitations:

- Can eliminate important features if λ is too high
 - Not stable when features are highly correlated
-

Dataset Description

The dataset used in this experiment is a healthcare stroke dataset containing patient information such as age, gender, hypertension, heart disease, average glucose level, BMI, smoking status, and other attributes.

- The dataset is structured and consists of both numerical and categorical variables.
 - The target variable for regression is BMI, which is continuous.
 - Missing values in BMI are handled using mean imputation.
 - Categorical variables are converted into numerical form using encoding techniques such as Label Encoding.
-

Data Preprocessing

Data preprocessing is an important step in machine learning. The following steps are performed:

1. Removal of irrelevant features such as ID.
2. Handling missing values using mean imputation.
3. Encoding categorical variables into numerical form.

4. Splitting data into training and testing sets.
5. Applying feature scaling using StandardScaler to normalize the data.

Feature scaling is particularly important for Ridge and Lasso regression because these models are sensitive to the magnitude of input features.

Model Training

The dataset is divided into training and testing sets using an 80:20 ratio. Three models are trained:

1. Multiple Linear Regression model without regularization.
2. Ridge Regression model with L2 penalty.
3. Lasso Regression model with L1 penalty.

Each model learns the relationship between input features and the target variable (BMI) using training data.

Evaluation Metrics

The performance of the models is evaluated using the following metrics:

Mean Absolute Error (MAE):

Measures the average absolute difference between actual and predicted values.

Mean Squared Error (MSE):

Measures the average squared difference between actual and predicted values. It penalizes larger errors more heavily.

Root Mean Squared Error (RMSE):

Square root of MSE, representing error in the same unit as the target variable.

R² Score (Coefficient of Determination):

Measures how well the model explains the variance in the data. A value closer to 1 indicates better performance.

Comparison of Models

Multiple Linear Regression provides a baseline model but may suffer from overfitting when dealing with high-dimensional data.

Ridge Regression improves performance by reducing the impact of less important features through coefficient shrinkage.

Lasso Regression simplifies the model by removing less important features, making it more interpretable.

CODE:-

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

```
# STEP 1: Import Libraries
```

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

```
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.linear_model import LinearRegression, Ridge, Lasso
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

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

```
# STEP 2: Load Dataset
```

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

```
df = pd.read_csv("/content/healthcare-dataset-stroke-data.csv")
```

```
print(df.head())
```

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

```
# STEP 3: Data Cleaning
```

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

```
# Drop unnecessary column
```

```
df.drop("id", axis=1, inplace=True)
```

```
# Fill missing BMI
```

```
df["bmi"].fillna(df["bmi"].mean(), inplace=True)
```

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

```
# STEP 4: Encode Categorical Data
```

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

```
le = LabelEncoder()
```

```
cat_cols = ["gender", "ever_married", "work_type", "Residence_type", "smoking_status"]
```

```
for col in cat_cols:
```

```
    df[col] = le.fit_transform(df[col])
```

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

```
# STEP 5: Feature & Target Selection
```

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

```
X = df.drop("bmi", axis=1) # features
```

```
y = df["bmi"]           # target
```

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

```
# STEP 6: Train-Test Split
```

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

```
X_train, X_test, y_train, y_test = train_test_split(
```

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

```
)
```

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

```
# STEP 7: Feature Scaling (IMPORTANT)
```

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

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

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

```
# STEP 8: MULTIPLE LINEAR REGRESSION
```

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

```
lr = LinearRegression()
```

```
lr.fit(X_train, y_train)
```

```
y_pred_lr = lr.predict(X_test)
```

```
print("\n==== MULTIPLE LINEAR REGRESSION ====")
```

```
print("MAE:", mean_absolute_error(y_test, y_pred_lr))
```

```
print("MSE:", mean_squared_error(y_test, y_pred_lr))
```

```
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred_lr)))
```

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

```
# =====

# STEP 9: RIDGE REGRESSION

# =====

ridge = Ridge(alpha=1.0) # alpha = regularization strength
ridge.fit(X_train, y_train)

y_pred_ridge = ridge.predict(X_test)

print("\n===== RIDGE REGRESSION =====")
print("MAE:", mean_absolute_error(y_test, y_pred_ridge))
print("MSE:", mean_squared_error(y_test, y_pred_ridge))
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred_ridge)))
print("R2 Score:", r2_score(y_test, y_pred_ridge))

# =====

# STEP 10: LASSO REGRESSION

# =====

lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)

y_pred_lasso = lasso.predict(X_test)
```

```
print("\n===== LASSO REGRESSION =====")  
print("MAE:", mean_absolute_error(y_test, y_pred_lasso))  
print("MSE:", mean_squared_error(y_test, y_pred_lasso))  
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred_lasso)))  
print("R2 Score:", r2_score(y_test, y_pred_lasso))
```

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

```
# STEP 11: MODEL COMPARISON
```

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

```
models = ["Linear", "Ridge", "Lasso"]
```

```
r2_scores = [  
    r2_score(y_test, y_pred_lr),  
    r2_score(y_test, y_pred_ridge),  
    r2_score(y_test, y_pred_lasso)  
]
```

```
plt.bar(models, r2_scores)  
plt.title("Model Comparison (R2 Score)")  
plt.xlabel("Models")  
plt.ylabel("R2 Score")  
plt.show()
```

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

```
# STEP 12: COEFFICIENT COMPARISON
```

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

```
print("\nFeature Coefficients:")
```

```
coeff_df = pd.DataFrame({
```

```
    "Feature": X.columns,
```

```
    "Linear": lr.coef_,
```

```
    "Ridge": ridge.coef_,
```

```
    "Lasso": lasso.coef_
```

```
})
```

```
print(coeff_df)
```

OUTPUT

```

...      id  gender  age  hypertension  heart_disease  ever_married  \
0      9046   Male  67.0             0             1           Yes
1      51676  Female  61.0             0             0           Yes
2      31112   Male  80.0             0             1           Yes
3      60182  Female  49.0             0             0           Yes
4       1665  Female  79.0             1             0           Yes

      work_type  Residence_type  avg_glucose_level  bmi  smoking_status  \
0      Private      Urban      228.69      36.6  formerly smoked
1  Self-employed      Rural      202.21      NaN  never smoked
2      Private      Rural      105.92      32.5  never smoked
3      Private      Urban      171.23      34.4      smokes
4  Self-employed      Rural      174.12      24.0  never smoked

      stroke
0         1
1         1
2         1
3         1
4         1

```

```
===== MULTIPLE LINEAR REGRESSION =====
```

```
MAE: 5.018980827980883
```

```
MSE: 43.9725374897823
```

```
RMSE: 6.6311791930080055
```

```
R2 Score: 0.2071236662250704
```

```
===== RIDGE REGRESSION =====
```

```
MAE: 5.019002796451791
```

```
MSE: 43.97261801812743
```

```
RMSE: 6.631185264952822
```

```
R2 Score: 0.20712221420472543
```

```
===== LASSO REGRESSION =====
```

```
MAE: 5.017735963281367
```

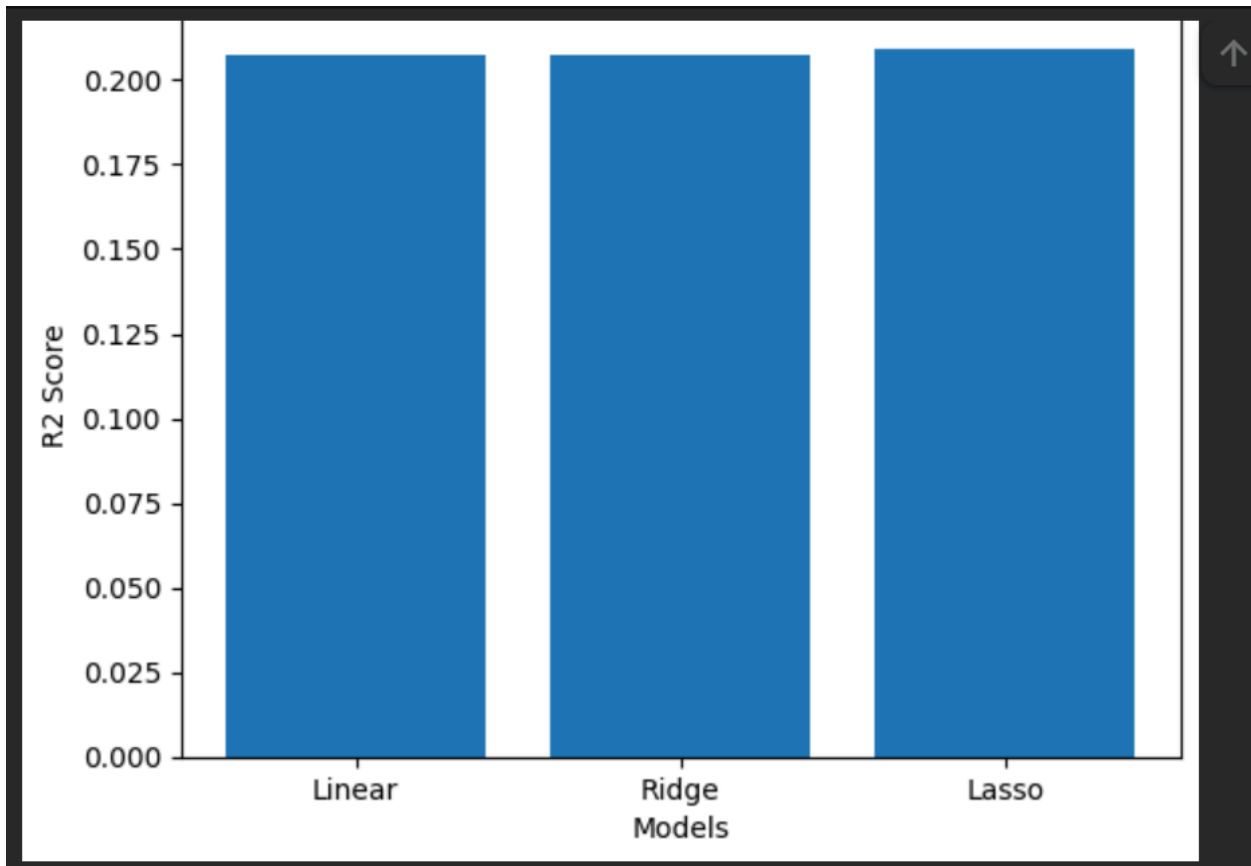
```
MSE: 43.85026601648348
```

```
RMSE: 6.621953338440515
```

```
R2 Score: 0.20932836404349742
```

```
/tmp/ipython-input-2487079396.py:32: FutureWarning: A value is trying to be set on a copy of a DataFrame.
The behavior will change in pandas 3.0. This inplace method will never work because the intermedi
```

```
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value},
```

**Feature Coefficients:**

	Feature	Linear	Ridge	Lasso
0	gender	-0.125071	-0.125060	-0.037853
1	age	0.861118	0.861137	0.783664
2	hypertension	0.655364	0.655251	0.575227
3	heart_disease	-0.291823	-0.291705	-0.165767
4	ever_married	1.125486	1.125313	1.109927
5	work_type	-1.290719	-1.290476	-1.245252
6	Residence_type	-0.163423	-0.163379	-0.062016
7	avg_glucose_level	0.769137	0.768978	0.671044
8	smoking_status	0.680314	0.680272	0.627318
9	stroke	-0.200867	-0.200781	-0.075224

CONCLUSION

In this experiment, Multiple Linear Regression, Ridge Regression, and Lasso Regression were successfully implemented on a real-world healthcare dataset to predict BMI.

Multiple Linear Regression served as a baseline model and demonstrated the relationship between independent variables and the target variable. However, it is prone to overfitting and is sensitive to multicollinearity.

Ridge Regression addressed these issues by introducing L2 regularization, which reduced the magnitude of coefficients and improved model stability. It performed better in handling correlated features and provided a more generalized model.

Lasso Regression further enhanced the model by applying L1 regularization, which performed automatic feature selection by shrinking some coefficients to zero. This resulted in a simpler and more interpretable model.

Overall, both Ridge and Lasso Regression outperformed Multiple Linear Regression in terms of generalization and robustness. Regularization techniques are essential when working with real-world datasets, as they help in reducing overfitting and improving predictive performance.

Thus, it can be concluded that Ridge and Lasso Regression are effective techniques for improving regression models and are widely used in practical applications such as healthcare analytics, financial forecasting, and predictive modeling.