1. Dataset Source

Dataset Name: Energy Efficiency Dataset

Source Link (Official & Reliable):
🔗 https://archive.ics.uci.edu/ml/datasets/Energy+efficiency

Used For This Experiment:
Predicting **Heating Load of buildings** using Multiple Linear Regression, Ridge Regression, and Lasso Regression.

✔ Each experiment uses a **different real-world dataset**
✔ Dataset is publicly available and widely accepted in ML research

---

2. Dataset Description

The Energy Efficiency Dataset contains measurements related to the energy efficiency of residential buildings.

Dataset Size

- **Total Instances:** 768
- **Total Features:** 8 input features
- **Target Variables:** 2 (Heating Load & Cooling Load)

In this experiment, **Heating Load** is used as the target.

Features Description

Feature	Description
Relative Compactness	Ratio indicating building compactness
Surface Area	Total surface area of the building
Wall Area	Area of walls
Roof Area	Area of roof

Overall Height	Height of the building
Orientation	Direction of the building
Glazing Area	Percentage of glass area
Glazing Area Distribution	Distribution of glazing

🎯 **Target Variable:**
Heating Load – amount of heat energy required to maintain indoor temperature.

Dataset Characteristics

- Numerical features only
- No missing values
- Strong linear relationships
- Suitable for regression and regularization techniques

3. Mathematical Formulation of the Algorithms

Multiple Linear Regression

y=β0+β1x1+β2x2+⋯+βnxny = \beta_0 + \beta_1x_1 + \beta_2x_2 + \cdots + \beta_nx_ny=β0+β1x1+β2x2+⋯+βnxn

Minimizes:

∑(y−y^)2\sum (y - \hat{y})^2∑(y−y^)2

Ridge Regression (L2 Regularization)

Loss=∑(y−y^)2+λ∑β2\text{Loss} = \sum (y - \hat{y})^2 + \lambda \sum \beta^2Loss=∑(y−y^)2+λ∑β2

- Penalizes large coefficients
- Reduces overfitting

- Keeps all features

Lasso Regression (L1 Regularization)

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

- Forces some coefficients to zero
- Performs automatic feature selection

---

4. Algorithm Limitations

Linear Regression

- Sensitive to outliers
- Assumes linear relationship
- Performs poorly with multicollinearity

Ridge Regression

- Does not remove irrelevant features
- Requires tuning of regularization parameter

Lasso Regression

- Can remove important features if alpha is too high
- Unstable when features are highly correlated

---

5. Methodology / Workflow

Steps Followed

1. Dataset loading
2. Feature and target selection
3. Train-test split (80%-20%)
4. Feature scaling using StandardScaler
5. Model training (Linear, Ridge, Lasso)
6. Prediction on test data
7. Performance evaluation (MSE & R^2)
8. Feature selection analysis (Lasso)
9. Visualization and comparison

Dataset
↓
Data Preprocessing
↓
Train-Test Split
↓
Feature Scaling
↓
Model Training
↓
Prediction
↓
Performance Evaluation
↓
Result Analysis

```python
# ================================
# 1. Import Required Libraries
# ================================
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import zipfile # Added for unzipping the dataset

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, r2_score


# ================================
# 2. Load Dataset
# ================================
# Unzip the uploaded dataset
with zipfile.ZipFile('energy+efficiency.zip', 'r') as zip_ref:
    zip_ref.extractall('.')

df = pd.read_excel('ENB2012_data.xlsx')

df.columns = [
    'Relative_Compactness', 'Surface_Area', 'Wall_Area', 'Roof_Area',
    'Overall_Height', 'Orientation', 'Glazing_Area',
    'Glazing_Area_Distribution', 'Heating_Load', 'Cooling_Load'
]

print("Dataset Loaded Successfully")
print(df.head())


# ================================
# 3. Features & Target
```

```python
# ================================
X = df.drop(['Heating_Load', 'Cooling_Load'], axis=1)
y = df['Heating_Load']


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


# ================================
# 5. Feature Scaling
# ================================
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)


# ================================
# 6. Train Models
# ================================
linear = LinearRegression()
ridge = Ridge(alpha=1.0)
lasso = Lasso(alpha=0.1)

linear.fit(X_train_scaled, y_train)
ridge.fit(X_train_scaled, y_train)
lasso.fit(X_train_scaled, y_train)


# ================================
# 7. Predictions
# ================================
y_pred_linear = linear.predict(X_test_scaled)
y_pred_ridge = ridge.predict(X_test_scaled)
y_pred_lasso = lasso.predict(X_test_scaled)
```

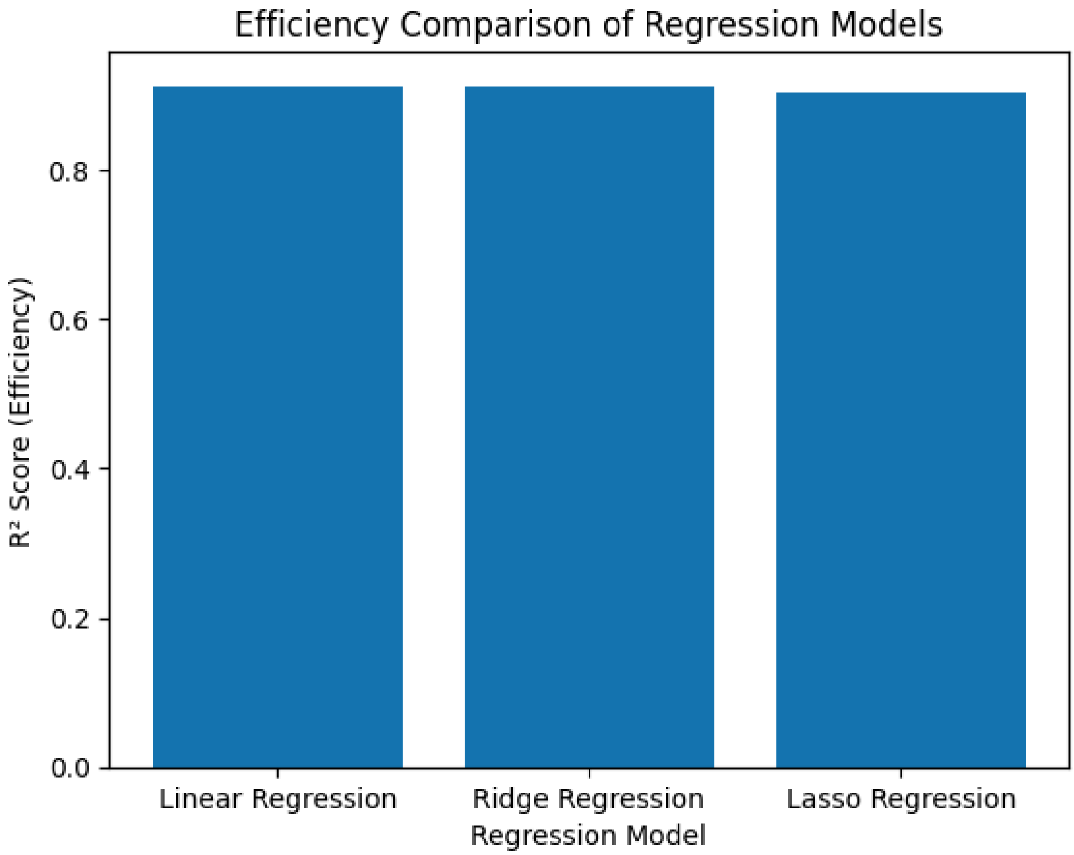
```python
# ================================
# 8. Performance Evaluation
# ================================
results = pd.DataFrame({
    "Model": ["Linear Regression", "Ridge Regression", "Lasso
Regression"],
    "MSE": [
        mean_squared_error(y_test, y_pred_linear),
        mean_squared_error(y_test, y_pred_ridge),
        mean_squared_error(y_test, y_pred_lasso)
    ],
    "R2 Score": [
        r2_score(y_test, y_pred_linear),
        r2_score(y_test, y_pred_ridge),
        r2_score(y_test, y_pred_lasso)
    ]
})

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


# ================================
# 9. Lasso Feature Selection
# ================================
lasso_coeff = pd.Series(lasso.coef_, index=X.columns)

kept_features = lasso_coeff[lasso_coeff != 0].index.tolist()
removed_features = lasso_coeff[lasso_coeff == 0].index.tolist()

print("\n✅ Features KEPT by Lasso:")
print(kept_features)

print("\n❌ Features REMOVED by Lasso:")
print(removed_features)


# ================================
# 10. Efficiency Graph (R² Score)
```

```
# ================================
plt.figure()
plt.bar(results["Model"], results["R2 Score"])
plt.xlabel("Regression Model")
plt.ylabel("R² Score (Efficiency)")
plt.title("Efficiency Comparison of Regression Models")
plt.show()


# ================================
# 11. Coefficient Comparison
# ================================
coef_df = pd.DataFrame({
    "Linear": linear.coef_,
    "Ridge": ridge.coef_,
    "Lasso": lasso.coef_
}, index=X.columns)

coef_df.plot(kind='bar', figsize=(10,5))
plt.ylabel("Coefficient Value")
plt.title("Feature Importance Comparison")
plt.show()
```

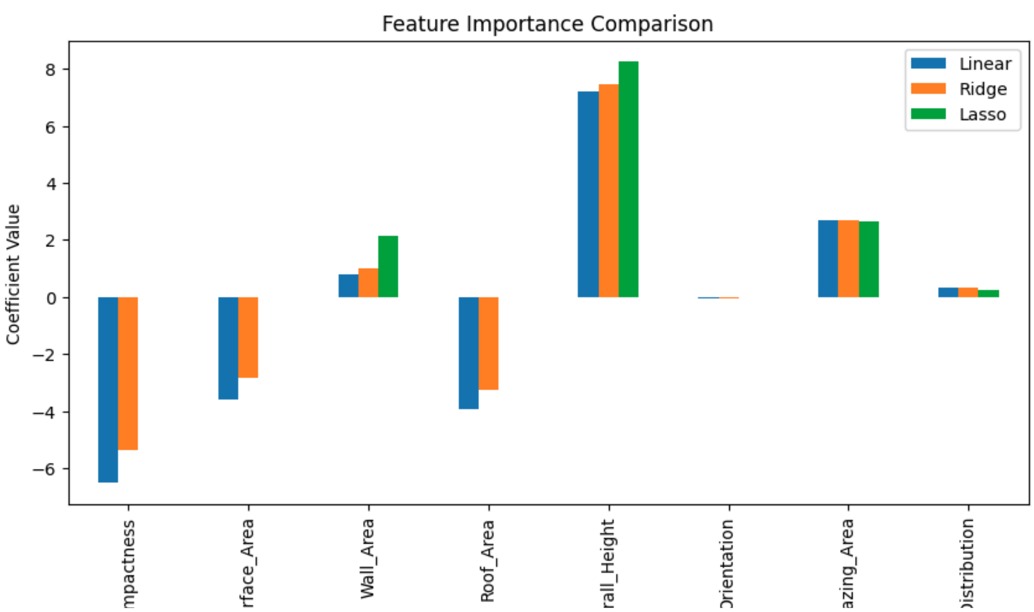
```
Dataset Loaded Successfully
   Relative_Compactness  Surface_Area  Wall_Area  Roof_Area  Overall_Height  \
0                  0.98         514.5      294.0     110.25             7.0
1                  0.98         514.5      294.0     110.25             7.0
2                  0.98         514.5      294.0     110.25             7.0
3                  0.98         514.5      294.0     110.25             7.0
4                  0.90         563.5      318.5     122.50             7.0

   Orientation  Glazing_Area  Glazing_Area_Distribution  Heating_Load  \
0            2           0.0                          0         15.55
1            3           0.0                          0         15.55
2            4           0.0                          0         15.55
3            5           0.0                          0         15.55
4            2           0.0                          0         20.84

   Cooling_Load
0         21.33
1         21.33
2         21.33
3         21.33
4         28.28
```

```
Model Efficiency Comparison:
               Model       MSE  R2 Score
0  Linear Regression  9.153208  0.912185
1   Ridge Regression  9.213843  0.911603
2   Lasso Regression  9.938754  0.904648

✅ Features KEPT by Lasso:
['Relative_Compactness', 'Wall_Area', 'Overall_Height', 'Glazing_Area', 'Glazing_Area_Distribution']

❌ Features REMOVED by Lasso:
['Surface_Area', 'Roof_Area', 'Orientation']
```

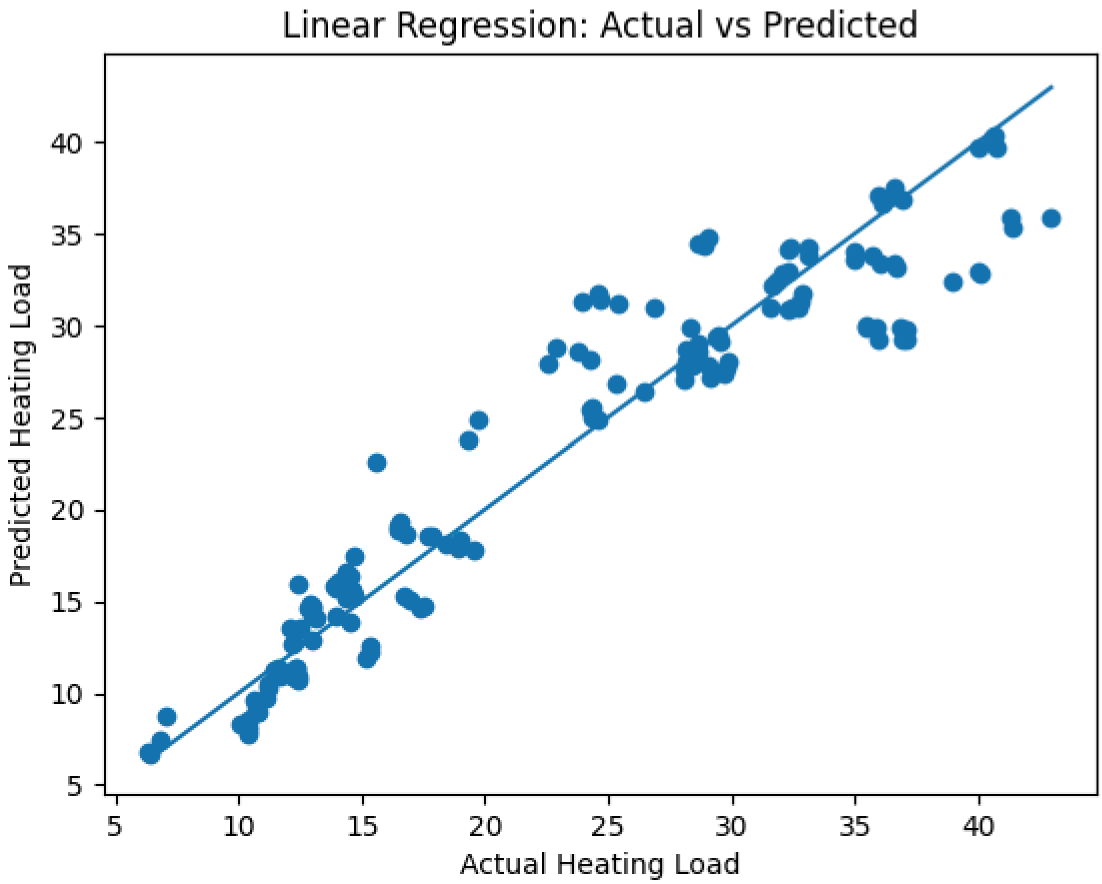
Efficiency Comparison of Regression Models



Efficiency Comparison of Regression Models
R² Score (Efficiency)
0.0
0.2
0.4
0.6
0.8
Linear Regression
Ridge Regression
Lasso Regression
Regression Model



Feature Importance Comparison
Coefficient Value
Linear
Ridge
Lasso
8
6
4
2
0
−2
−4
−6
mpactness
rface_Area
Wall_Area
Roof_Area
rall_Height
Drientation
azing_Area
Distribution


Linear Regression: Actual vs Predicted
Predicted Heating Load
Actual Heating Load
5
10
15
20
25
30
35
40