

▼ Task

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

▼ Upload file

Subtask:

Create a cell for uploading the csv files.

Reasoning: Create a code cell to handle the file upload process using google.colab.files.upload and print a message to the user.

```
# =====
# Multiple Linear, Ridge & Lasso Regression
# On California Housing Dataset
# =====

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.datasets import fetch_california_housing
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

# -----
# 1. Load Dataset
# -----
data = fetch_california_housing(as_frame=True)
df = data.frame

X = df.drop("MedHouseVal", axis=1)
y = df["MedHouseVal"]

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

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

# -----
# 3. Feature Scaling (IMPORTANT for Ridge & Lasso)
# -----
scaler = StandardScaler()

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

# -----
# 4. Model Training
# -----
# Multiple Linear Regression
linear_model = LinearRegression()
linear_model.fit(X_train_scaled, y_train)

# Ridge Regression
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train_scaled, y_train)

# Lasso Regression
lasso_model = Lasso(alpha=0.01)
lasso_model.fit(X_train_scaled, y_train)

# -----
# 5. Predictions
# -----
y_pred_linear = linear_model.predict(X_test_scaled)
y_pred_ridge = ridge_model.predict(X_test_scaled)
```

```
y_pred_lasso = lasso_model.predict(X_test_scaled)

# -----
# 6. Evaluation Function
# -----
def evaluate_model(name, y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"\n{name}:")
    print(f"  MSE : {mse:.4f}")
    print(f"  R² : {r2:.4f}")
    print("-" * 30)

# -----
# 7. Model Evaluation
# -----
evaluate_model("Multiple Linear Regression", y_test, y_pred_linear)
evaluate_model("Ridge Regression", y_test, y_pred_ridge)
evaluate_model("Lasso Regression", y_test, y_pred_lasso)

# -----
# 8. Coefficient Comparison
# -----
coefficients = pd.DataFrame({
    "Feature": X.columns,
    "Linear": linear_model.coef_,
    "Ridge": ridge_model.coef_,
    "Lasso": lasso_model.coef_
})

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

# -----
# 9. Visualization (Optional)
# -----
coefficients.set_index("Feature").plot(kind="bar", figsize=(12,6))
plt.title("Coefficient Comparison: Linear vs Ridge vs Lasso")
plt.ylabel("Coefficient Value")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Dataset Shape: (20640, 9)
   MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude \
0    8.3252      41.0   6.984127   1.023810     322.0  2.555556    37.88
1    8.3014      21.0   6.238137   0.971880    2401.0  2.109842    37.86
2    7.2574      52.0   8.288136   1.073446     496.0  2.802260    37.85
3    5.6431      52.0   5.817352   1.073059     558.0  2.547945    37.85
4    3.8462      52.0   6.281853   1.081081     565.0  2.181467    37.85
```

```
   Longitude  MedHouseVal
0     -122.23      4.526
1     -122.22      3.585
2     -122.24      3.521
3     -122.25      3.413
4     -122.25      3.422
```

Multiple Linear Regression
MSE : 0.5559
R² : 0.5758

Ridge Regression
MSE : 0.5559
R² : 0.5758

Lasso Regression
MSE : 0.5483
R² : 0.5816

Coefficient Comparison:

	Feature	Linear	Ridge	Lasso
0	MedInc	0.854383	0.854327	0.800957
1	HouseAge	0.122546	0.122624	0.127087
2	AveRooms	-0.294410	-0.294210	-0.162759
3	AveBedrms	0.339259	0.339008	0.206207
4	Population	-0.002308	-0.002282	-0.000000
5	AveOccup	-0.040829	-0.040833	-0.030602
6	Latitude	-0.896929	-0.896168	-0.790113
7	Longitude	-0.869842	-0.869071	-0.755674

Coefficient Comparison: Linear vs Ridge vs Lasso

