```
import pandas as pd
```

```
df = pd.read_csv('/content/boston.csv')
```

```
df.columns
```

```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'MEDV'],
      dtype='object')
```

```
x = df.drop('MEDV', axis=1)
y = df['MEDV']
```

```
import numpy as np
```

```
x = np.array(x)
y = np.array(y).reshape(-1,1)

# Apply log transformation to the target variable
y = np.log1p(y)
```

```
df.isnull().sum()
```

	0
CRIM	0
ZN	0
INDUS	0
CHAS	0
NOX	0
RM	0
AGE	0
DIS	0
RAD	0
TAX	0
PTRATIO	0
B	0
LSTAT	0
MEDV	0

dtype: int64

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(
    x,y, test_size=0.3, random_state=42
)
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

#Fit only on training data
x_train = scaler.fit_transform(x_train)

#Transform test data using parameters
x_test = scaler.transform(x_test)
```

```
from sklearn.linear_model import Ridge
```

```
from sklearn.linear_model import RidgeCV

alphas = np.logspace(-3,3,100)
ridge_cv = RidgeCV(alphas=alphas,cv=10)
ridge_cv.fit(x_train,y_train)
```

```
print("Best alpha:", ridge_cv.alpha_)
```

```
Best alpha: 17.47528400007683
```

```
best_ridge = Ridge(alpha=ridge_cv.alpha_)
best_ridge.fit(x_train,y_train)

y_pred = best_ridge.predict(x_test)
```

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

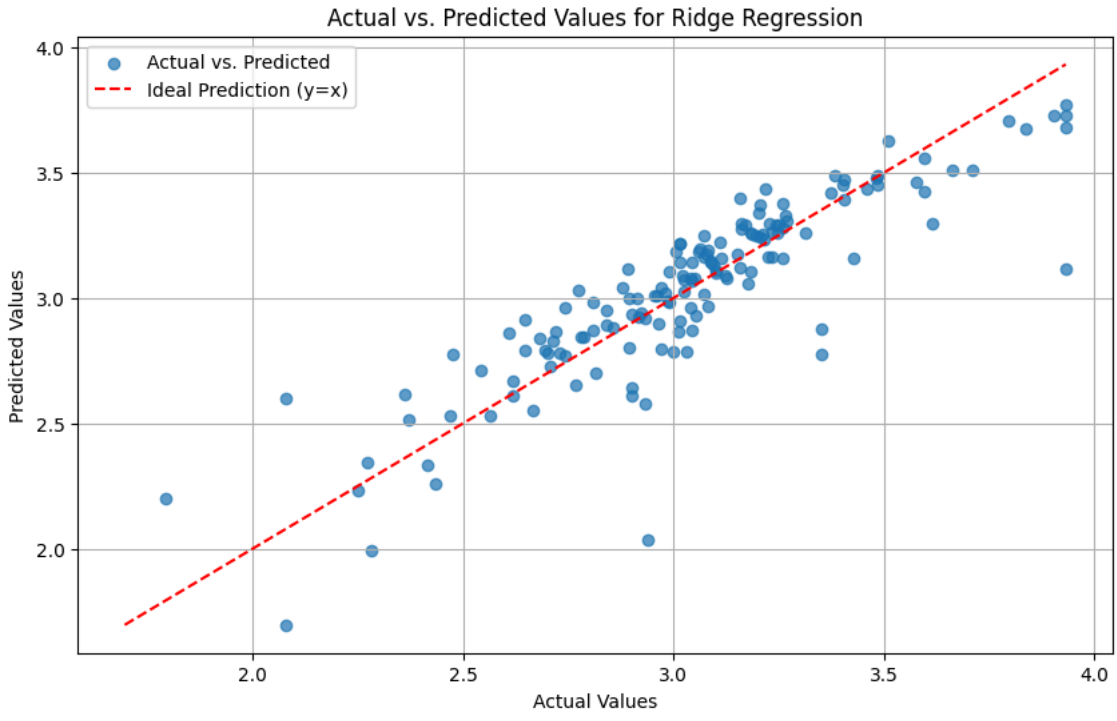
```
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared:", r2)
```

```
Mean Squared Error: 0.03378315962753483
Root Mean Squared Error: 0.18380195762704712
R-squared: 0.7543477598580829
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.7, label='Actual vs. Predicted')
plt.plot([min(y_test.min(), y_pred.min()), max(y_test.max(), y_pred.max())],
         [min(y_test.min(), y_pred.min()), max(y_test.max(), y_pred.max())],
         color='red', linestyle='--', label='Ideal Prediction (y=x)')
plt.title('Actual vs. Predicted Values for Ridge Regression')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.legend()
plt.grid(True)
plt.show()
```




```
Start coding or generate with AI.
```