```
import pandas as pd
import numpy as np
```

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

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2

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

```
import numpy as np
x = np.array(df['MEDV'])
y = np.array(df['LSTAT'])
```

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

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x, y, test_size = 0.2, random_state=42)
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

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

```
from sklearn.linear_model import RidgeCV
```

```
alphas = np.logspace(-3, 3, 7)

rigde_cv = RidgeCV(alphas=alphas, cv=10)
rigde_cv.fit(x_train, y_train)

print('Best alpha:',rigde_cv.alpha_)
```

```
Best alpha: 1.0
```

```
best_ridge = Ridge(alpha=rigde_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)
mse
```

```
23.750529543766103
```

```
r2 = r2_score(y_test, y_pred)
r2
```

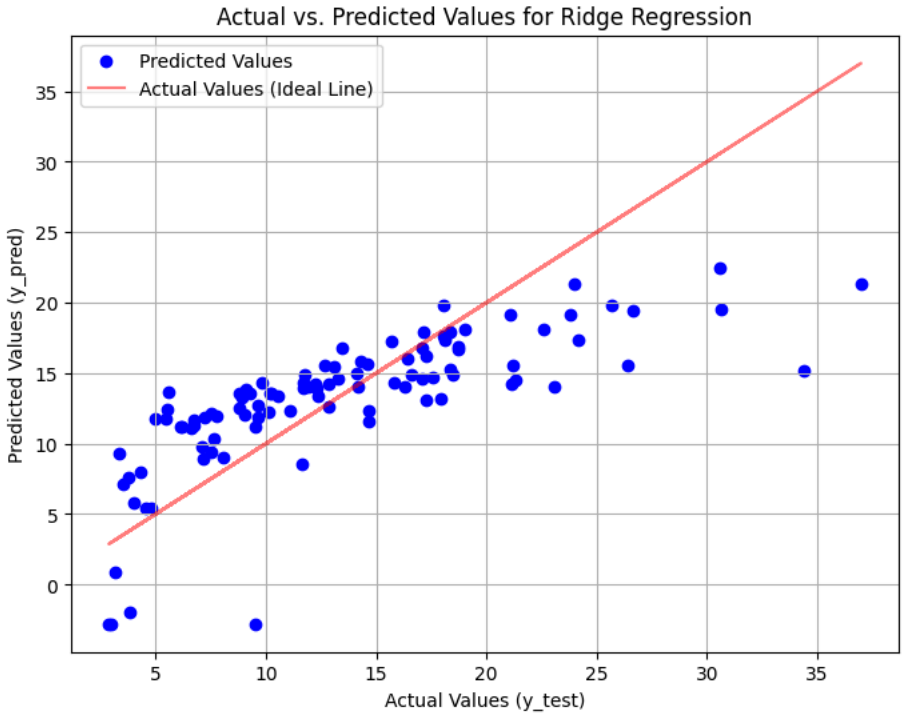
```
0.5429926562719491
```

```
rmse = np.sqrt(mse)
rmse
```

```
np.float64(4.873451502145692)
```

```
import matplotlib.pyplot as plt
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', label='Predicted Values')
plt.plot(y_test, y_test, color='red', alpha=0.5, label='Actual Values (Ideal Line)')
plt.xlabel('Actual Values (y_test)')
plt.ylabel('Predicted Values (y_pred)')
plt.title('Actual vs. Predicted Values for Ridge Regression')
plt.legend()
plt.grid(True)
plt.show()
```




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