

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

```
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

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

```
x = np.array(x)
y = np.array(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 Lasso
from sklearn.linear_model import LassoCV
```

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

lasso_cv = LassoCV(alphas=alphas, cv=10)
lasso_cv.fit(x_train,y_train)

print("Best alpha:", lasso_cv.alpha_)

Best alpha: 0.01
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_coordinate_descent.py:1664: DataConversionWarning: A co
y = column_or_1d(y, warn=True)
```

```
best_ridge = Lasso(alpha=lasso_cv.alpha_)
best_ridge.fit(x_train,y_train)

y_pred = best_ridge.predict(x_test)
```

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

```
mse = mean_squared_error(y_test,y_pred)

rmse = np.sqrt(mse)

r2 = r2_score(y_test,y_pred)

print("MSE:",mse)
print("RMSE:",rmse)
print("R2 score:",r2)
```

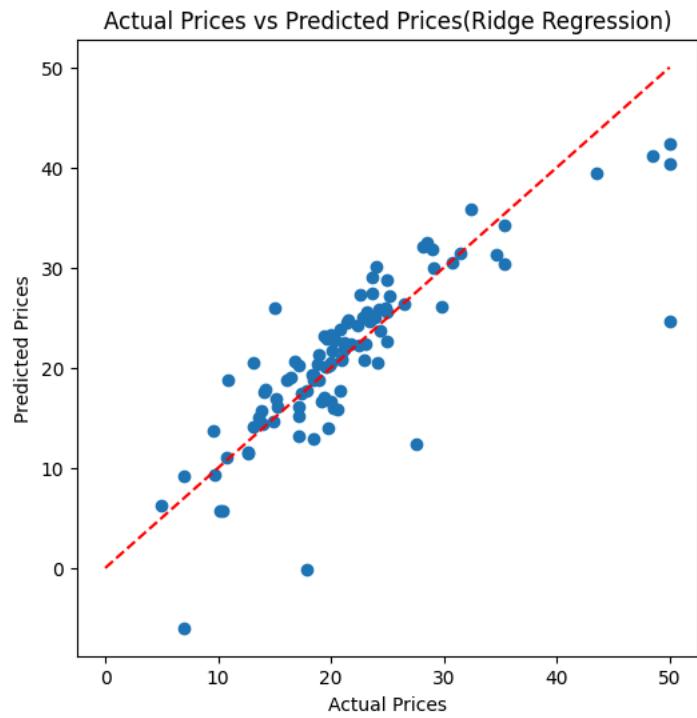
```
MSE: 24.333499160616334
RMSE: 4.932899670641634
```

R2 score: 0.6681815922762606

```
import matplotlib.pyplot as plt
plt.figure(figsize=(6,6))
plt.scatter(y_test,y_pred)

max_val = max(y_test.max(),y_pred.max())
plt.plot([0,max_val],[0,max_val],linestyle='--',color='red')

plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual Prices vs Predicted Prices(Ridge Regression)')
plt.show()
```



```
best_ridge = Lasso(alpha=lasso_cv.alpha_)
best_ridge.fit(x_train,y_train)
```

▼ Lasso ⓘ ?  
Lasso(alpha=np.float64(0.01))

```
mse_train = mean_squared_error(y_train,best_ridge.predict(x_train))
rmse_train = np.sqrt(mse_train)
```

```
mse_train = mean_squared_error(y_train,best_ridge.predict(x_train))
mse_train
```

21.648542729975865

Start coding or [generate](#) with AI.

