```
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

```
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']
```

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

```
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()
#frst only on training data
x_train=scaler.fit_transform(x_train)
#transform test data using same parameters
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: 0.001
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_coordinate_descent.py:1664: DataConversionWarning: A column-ve
  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,r2_score
```

```
#MSE
mse=mean_squared_error(y_test,y_pred)
```

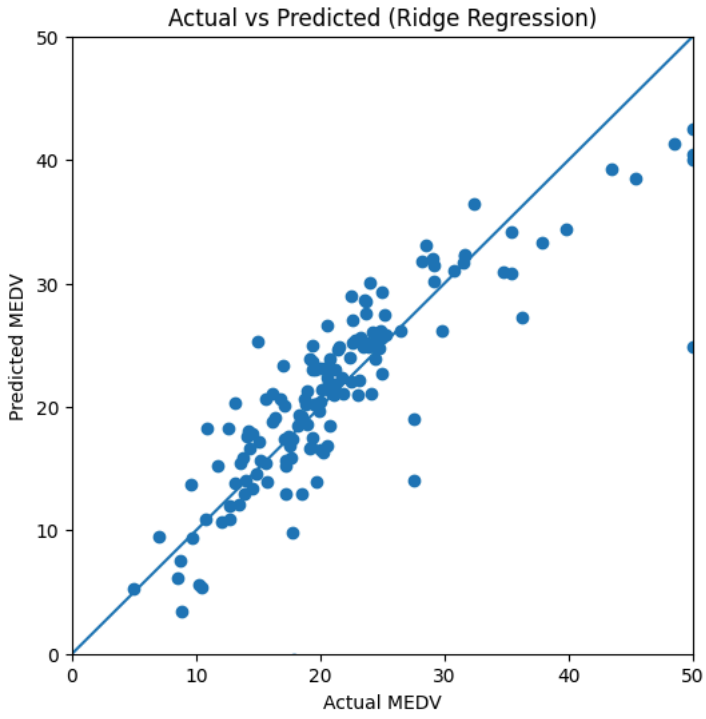
```
#RMSE
rmse=np.sqrt(mse)
```

```
#R2 score
r2=r2_score(y_test,y_pred)
```

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

```
MSE: 21.52151071849254
RMSE: 4.639128228287351
R2 score: 0.7111714316191494
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(6,6))
plt.scatter(y_test,y_pred)
#plot prediction line
max_val=max(y_test.max(),y_pred.max())
plt.plot([0,max_val],[0,max_val], )
#start axes from 0
plt.xlim(0,max_val)
plt.ylim(0,max_val)
plt.xlabel('Actual MEDV')
plt.ylabel('Predicted MEDV')
plt.title('Actual vs Predicted (Ridge Regression)')
plt.show()
```




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

```
y_train_pred=best_ridge.predict(x_train)
```

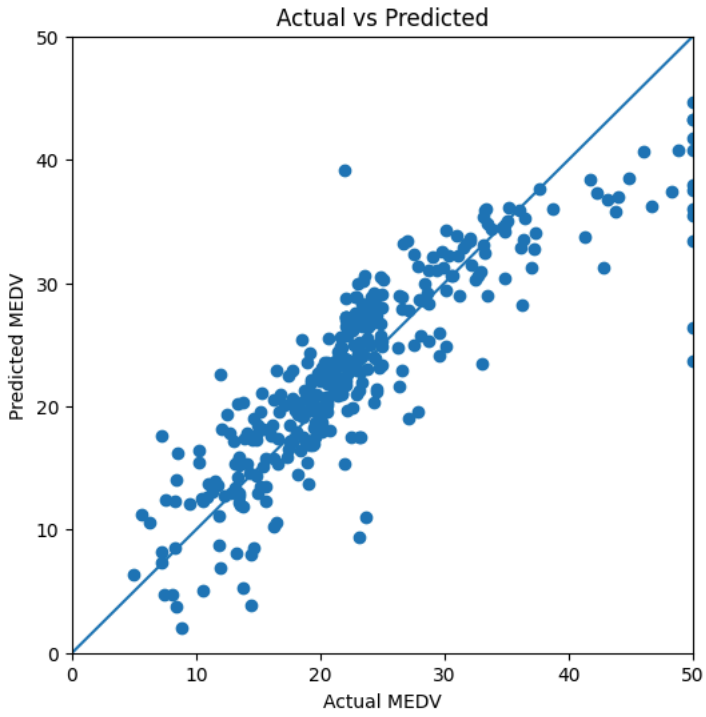
```
from sklearn.metrics import mean_squared_error,r2_score
```

```
mse_train=mean_squared_error(y_train,y_train_pred)
rms_traine=np.sqrt(mse_train)
r2_train=r2_score(y_train,y_train_pred)
```

```
print("MSE:",mse_train)
print("RMSE:",rms_traine)
print("R2 score:",r2_train)
```

```
MSE: 22.54555429526183
RMSE: 4.74821590655499
R2 score: 0.7434989248646879
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(6,6))
plt.scatter(y_train,y_pred)
#plot prediction line
max_val=max(y_test.max(),y_train_pred.max())
plt.plot([0,max_val],[0,max_val], )
#start axes from 0
plt.xlim(0,max_val)
plt.ylim(0,max_val)
plt.xlabel('Actual MEDV')
plt.ylabel('Predicted MEDV')
plt.title('Actual vs Predicted')
plt.show()
```




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