

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

```
x=np.array([95,85,80,70,60])
y=np.array([85,95,70,65,70])
```

```
model = np.polyfit(x,y,1)
```

```
model
```

```
↗ array([ 0.64383562, 26.78082192])
```

```
predict = np.poly1d(model)
predict(65)
```

```
↗ 68.63013698630135
```

```
y_pred = predict(x)
y_pred
```

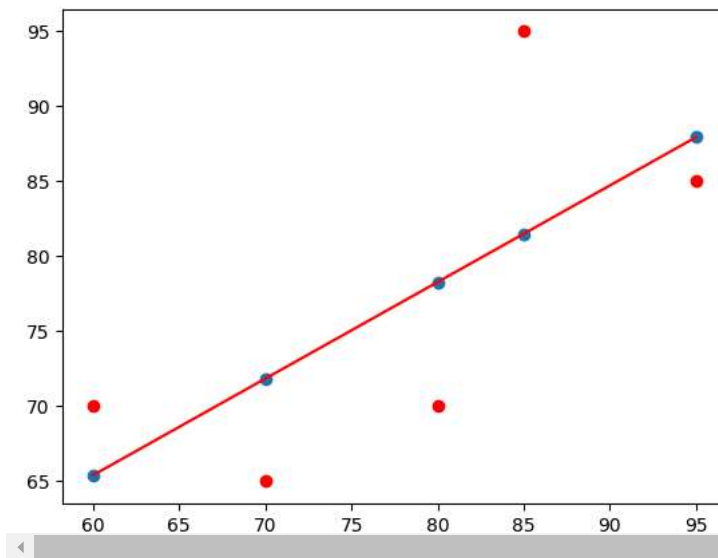
```
↗ array([87.94520548, 81.50684932, 78.28767123, 71.84931507, 65.4109589 ])
```

```
from sklearn.metrics import r2_score
r2_score(y, y_pred)
```


```
↗ 0.4803218090889323
```

```
y_line = model[1] + model[0]* x
plt.plot(x, y_line, c = 'r')
plt.scatter(x, y_pred)
plt.scatter(x,y,c='r')
```



```
↗ <matplotlib.collections.PathCollection at 0x78216234a750>
```



```
df = pd.read_csv("/housing.csv")
df
```



	RM	LSTAT	PTRATIO	MEDV
0	6.575	4.98	15.3	504000.0
1	6.421	9.14	17.8	453600.0
2	7.185	4.03	17.8	728700.0
3	6.998	2.94	18.7	701400.0
4	7.147	5.33	18.7	760200.0
...
484	6.593	9.67	21.0	470400.0
485	6.120	9.08	21.0	432600.0
486	6.976	5.64	21.0	501900.0
487	6.794	6.48	21.0	462000.0
488	6.030	7.88	21.0	249900.0

489 rows × 4 columns


Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

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

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
lm = LinearRegression()
lm.fit(X_train, y_train)
```




LinearRegression ⓘ ?
 LinearRegression()

```
y_train_pred = lm.predict(X_train)
y_test_pred = lm.predict(X_test)
```

```
df=pd.DataFrame(y_train_pred,y_train)
df=pd.DataFrame(y_test_pred,y_test)
```

```
mse_train = mean_squared_error(y_train, y_train_pred)
print(mse_train)
mse_test = mean_squared_error(y_test, y_test_pred)
print(mse_test)
```



```
7949715203.592581
6789025559.265892
```

```
plt.scatter(y_train ,y_train_pred,c='blue',marker='o',label='Training data')
plt.scatter(y_test,y_test_pred ,c='lightgreen',marker='s',label='Test data')
plt.xlabel('True Values')
plt.ylabel('Predicted')
plt.title('True Value vs Predicted Value')
plt.plot()
plt.show()
```

