# practical-04

February 5, 2025

## 1 Using Synthesis Dataset

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
x=np.array([95,85,80,70,60])
y=np.array([85,95,70,65,70])
```

[3]:
```
model= np.polyfit(x, y, 1)
```

[4]:
```
model
```

[4]:
```
array([ 0.64383562, 26.78082192])
```

[5]:
```
predict = np.poly1d(model)
predict(65)
```

[5]:
```
68.63013698630135
```

[6]:
```
y_pred= predict(x)
y_pred
```

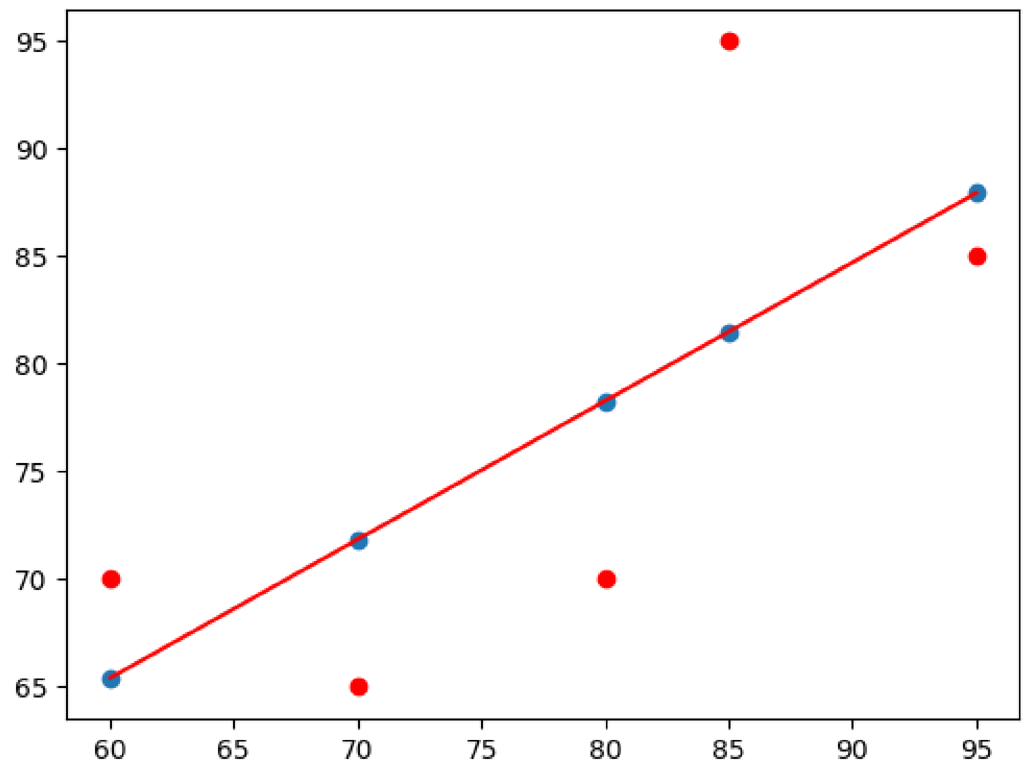
[6]:
```
array([87.94520548, 81.50684932, 78.28767123, 71.84931507, 65.4109589 ])
```

[9]:
```
from sklearn.metrics import r2_score
r2_score(y, y_pred)
```

[9]:
```
0.4803218090889323
```

[13]:
```
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')
```

[13]:
```
<matplotlib.collections.PathCollection at 0x1117fb4cf90>
```



## 2 Using California Housing Dataset

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
from sklearn.datasets import fetch_california_housing
boston = fetch_california_housing()
```

[3]:
```
data = pd.DataFrame(boston.data, columns=boston.feature_names)
data['PRICE'] = boston.target
data.isnull().sum()
```

[3]:
```
MedInc        0
HouseAge      0
AveRooms      0
AveBedrms     0
Population    0
AveOccup      0
Latitude      0
Longitude     0
```

```
PRICE         0
dtype: int64
```

[4]:

```python
x = data.drop(['PRICE'], axis=1)
y = data['PRICE']
```

[5]:

```python
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=0)
```

[6]:

```python
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(x_train, y_train)
```

[6]:

```
LinearRegression()
```

[7]:

```python
y_train_pred = lm.predict(x_train)
y_test_pred = lm.predict(x_test)
df_train = pd.DataFrame({'Actual': y_train, 'Predicted': y_train_pred})
df_test = pd.DataFrame({'Actual': y_test, 'Predicted': y_test_pred})
```

[8]:

```python
from sklearn.metrics import mean_squared_error, r2_score
mse_train = mean_squared_error(y_train, y_train_pred)
print("MSE (Train):", mse_train)
```

```
MSE (Train): 0.5234413607125448
```

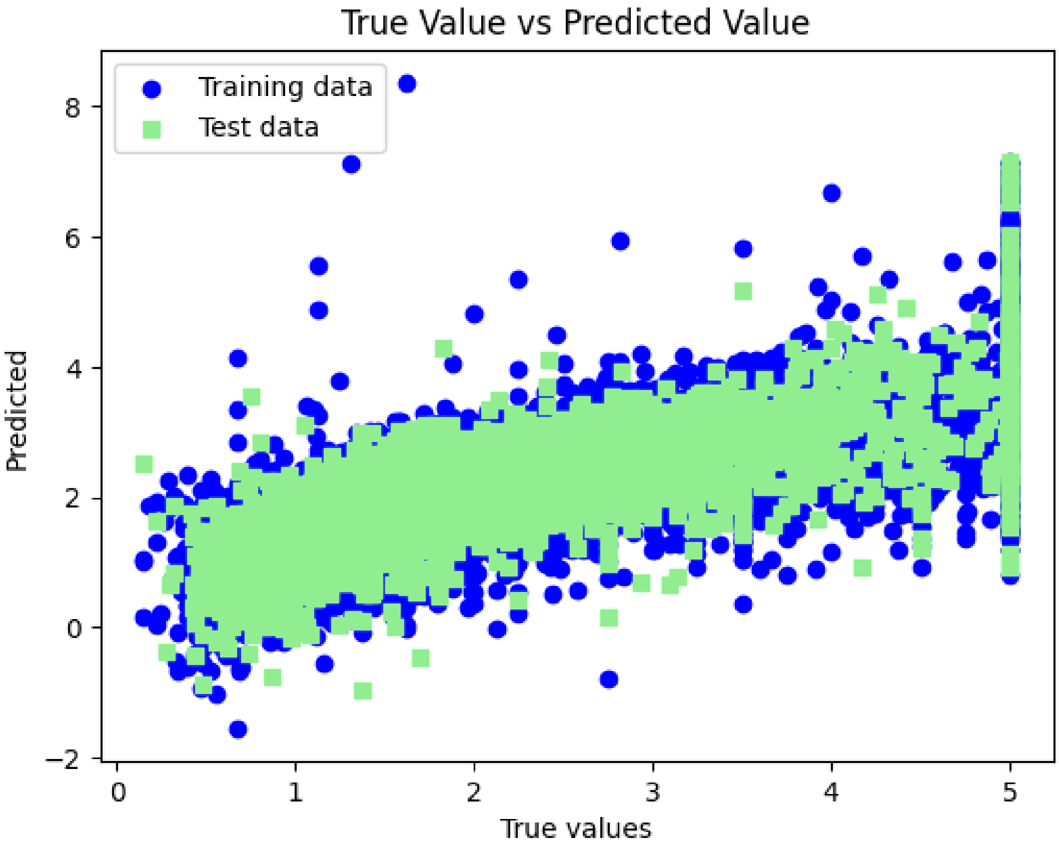
[9]:

```python
mse_test = mean_squared_error(y_test, y_test_pred)
print("MSE (Test):", mse_test)
```

```
MSE (Test): 0.5289841670367214
```

[10]:

```python
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.legend(loc='upper left')
plt.show()
```


True Value vs Predicted Value
● Training data
■ Test data
Predicted
True values
−2
0
2
4
6
8
0
1
2
3
4
5

```
[ ]:
```