

# Linear Regression Model on Machine Learning

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```
In [2]: data = [0.0 ,1.0 ,2.0 ,3.0 ,4.0 ,5.0 ,6.0 ,7.0 ,8.0 ,9.0]
label = [ 1.0,  4.0,  7.0, 10.0, 13.0, 16.0, 19.0, 22.0, 25.0, 28.0]
```

```
In [3]: x = np.array(data).reshape(len(data),1)
y = np.array(label)
```

```
In [4]: x, y
```

```
Out[4]: (array([[0.],
                [1.],
                [2.],
                [3.],
                [4.],
                [5.],
                [6.],
                [7.],
                [8.],
                [9.])),
        array([ 1.,  4.,  7., 10., 13., 16., 19., 22., 25., 28.]))
```

```
In [5]: import pandas as pd
df = pd.DataFrame(x,y)
df
```

```
Out[5]:
```

	0
1.0	0.0
4.0	1.0
7.0	2.0
10.0	3.0
13.0	4.0
16.0	5.0
19.0	6.0
22.0	7.0
25.0	8.0
28.0	9.0

```
In [6]: model = LinearRegression()
```

```
In [7]: model.fit(x,y)
```

```
Out[7]: LinearRegression()
```

```
In [8]: print(f"coefficients: {model.coef_}")
        print(f"intercept: {model.intercept_}")
```

```
coefficients: [3.]
intercept: 0.9999999999999982
```

## $y = mx + b$

$m$  = Coefficient  $b$  = intercept  $y = 3x + 0.99$

Nilai  $b_0$  adalah sekitar 0.9999999999999982. Ini menggambarkan bahwa model prediksi anda merespons 0.9999999999999982 ketika  $x$  adalah nol. Nilai = 3.0 berarti respons yang diprediksi naik menjadi 3,0 ketika  $x$  bertambah satu

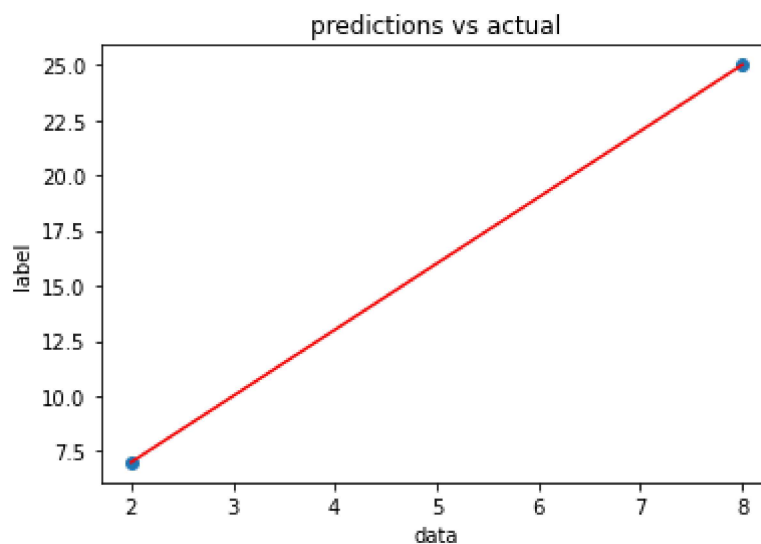
```
In [9]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
        r_squared = model.score(X_test, y_test)
        print(r_squared)
        y_pred = model.predict(X_test)
```

```
1.0
```

Nilai akurasi 100%

```
In [10]: plt.scatter(X_test, y_pred)
         plt.plot(X_test, y_pred, color='r')
         plt.xlabel('data')
         plt.ylabel('label')
         plt.title('predictions vs actual')
```

```
Out[10]: Text(0.5, 1.0, 'predictions vs actual')
```



```
In [11]: prediksi = model.predict([[20]]).reshape(-1,1)
         prediksi
```

```
Out[11]: array([[61.]])
```