Lanvaya09 / AIML

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

AIML
/
230901503 EXP 2 COPPER USING LINEAR REGRESSION.ipynb

Go to file t

Lanvaya09 Add files via upload 34216ec · 1 minute ago

277 lines (277 loc) · 27.6 KB

Preview Code Blame Raw

In [12]:

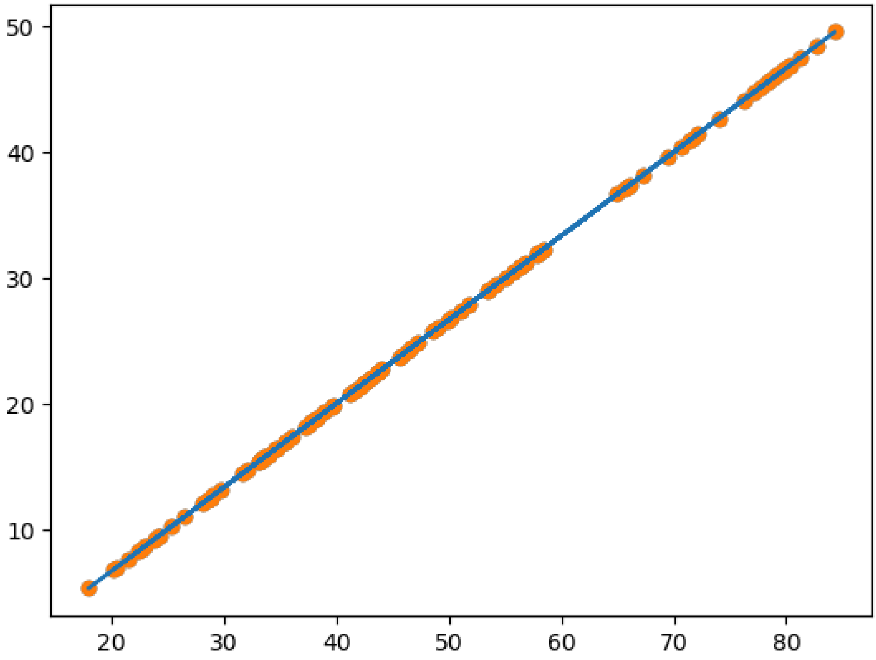
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
a=pd.read_excel("EXP 2 AIML.xlsx")
a.head(4)
```

Out[12]:

	power_rating	weight_of_copper
0	16.46	34.69
1	12.52	28.78
2	7.66	21.48
3	21.09	41.64

In [13]:

```
x = a["weight_of_copper"]
y = a["power_rating"]

x = np.array(x)

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.33, random_state=42
)

slope, intercept, r, p, std_err = stats.linregress(x, y)

plt.scatter(x, y)

def myfunc(x):
    return slope * x + intercept

mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```



In [14]:

```
x = x.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=0.33, random_state=42
)
```

```
)

from sklearn.linear_model import LinearRegression
clf = LinearRegression()
clf.fit(x_train, y_train.values.ravel())

y_pred = clf.predict(x_test)
y_pred
```

Out[14]:

```
array([21.59910719, 46.53423137, 10.9916787 ,  6.76470845, 30.479745  ,
       16.97210821, 23.83260094, 37.29356771, 12.52512217, 16.45873801,
       15.79202346, 41.40719648, 15.95203495, 26.81948212, 15.36532615,
       24.42597689,  9.47823667, 47.52763605, 19.27894056, 13.18516957,
       22.4791704 , 40.89382628, 16.95877392, 41.10050779, 25.71273597,
       15.71201772,  8.39149195, 39.63373578, 48.45436928, 46.48756135,
       24.23929681, 10.22495697, 26.02609181, 19.71230502])
```

In [15]:

```
print("Intercept:", intercept)
print("Slope:", slope)
```

```
Intercept: -6.667788099017734
Slope: 0.6666851985873647
```

In [16]:

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

print("r2_score:", r2_score(y_test, y_pred))
```

```
r2_score: 0.9999998999354417
```

In [17]:

```
new_value = [[40]]
x = clf.predict(new_value)
x
```

Out[17]:

```
array([19.99899227])
```

In [18]:

```
new_value = [[50]]
x = clf.predict(new_value)
print(x)
```

```
[26.66613778]
```

In [ ]: