linear-inc-age

March 21, 2024

[1]:
```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
df = pd.read_csv("income-age.csv")
df
```

[2]:

	age	experience	income
0	25	1	30450
1	30	3	35670
2	47	2	31580
3	32	5	40130
4	43	10	47830
5	51	7	41630
6	28	5	41340
7	33	4	37650
8	37	5	40250
9	39	8	45150
10	29	1	27840
11	47	9	46110
12	54	5	36720
13	51	4	34800
14	44	12	51300
15	41	6	38900
16	58	17	63600
17	23	1	30870
18	44	9	44190
19	37	10	48700

[3]:
```
df.shape
```

[3]:
```
(20, 3)
```

[4]:
```
df.isnull().sum()
```

[4]:
```
age           0
experience    0
income        0
```

```
dtype: int64
```

[5]:
```
x = df.drop("income",axis=1)
x
```

[5]:

	age	experience
0	25	1
1	30	3
2	47	2
3	32	5
4	43	10
5	51	7
6	28	5
7	33	4
8	37	5
9	39	8
10	29	1
11	47	9
12	54	5
13	51	4
14	44	12
15	41	6
16	58	17
17	23	1
18	44	9
19	37	10

[6]:
```
y = df["income"]
y
```

[6]:
```
0     30450
1     35670
2     31580
3     40130
4     47830
5     41630
6     41340
7     37650
8     40250
9     45150
10    27840
11    46110
12    36720
13    34800
14    51300
15    38900
16    63600
```

```
17    30870
18    44190
19    48700
Name: income, dtype: int64
```

[7]:
```
from sklearn.model_selection import train_test_split
```

[8]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
↪random_state=42)
```

[9]:
```
from sklearn.linear_model import LinearRegression
```

[10]:
```
model = LinearRegression().fit(x_train, y_train)
model
```

[10]:
```
LinearRegression()
```

[11]:
```
y_pred = model.predict(x_test)
y_pred
```

[11]:
```
array([31093.38107376, 31295.49954076, 40250.46080162, 34897.6958918 ])
```

[12]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

[13]:
```
r2_sc = r2_score(y_test, y_pred)
```

[14]:
```
print(f"R2 Score = ", r2_sc)
```

```
R2 Score =  0.9387098237077887
```

OPTIMIZATION

[15]:
```
from sklearn.model_selection import GridSearchCV
```

[16]:
```
model = LinearRegression()
model
```

[16]:
```
LinearRegression()
```

[17]:
```
param_grid = {
     'fit_intercept':[True,False],
    'copy_X':[True, False],
    'n_jobs':[-1, None],
    'positive':[False, True]
}
```

[18]:
```
grid_search = GridSearchCV(model, param_grid, cv=5, n_jobs=-1)
grid_search.fit(x_train, y_train)
```

[18]:
```
GridSearchCV(cv=5, estimator=LinearRegression(), n_jobs=-1,
             param_grid={'copy_X': [True, False],
                         'fit_intercept': [True, False], 'n_jobs': [-1, None],
                         'positive': [False, True]})
```

[19]:
```
best_params = grid_search.best_params_
print("Best Parameters :", best_params)
```

```
Best Parameters : {'copy_X': True, 'fit_intercept': True, 'n_jobs': -1,
'positive': False}
```

[20]:
```
best_model = LinearRegression(**best_params)
best_model.fit(x_train, y_train)
best_model
```

[20]:
```
LinearRegression(n_jobs=-1)
```

[21]:
```
y_pred = best_model.predict(x_test)
y_pred
```

[21]:
```
array([31093.38107376, 31295.49954076, 40250.46080162, 34897.6958918 ])
```

[22]:
```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2_sc = r2_score(y_test, y_pred)
```

[23]:
```
print(f"R2 Score = ", r2_sc)
print("Best Parameters :", best_params)
```

```
R2 Score =  0.9387098237077887
Best Parameters : {'copy_X': True, 'fit_intercept': True, 'n_jobs': -1,
'positive': False}
```

[ ]:

[ ]:

[ ]:

[ ]: