2000080110_ML skill2

September 2, 2021

[1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
data=pd.read_csv(r'E:\M&L excel\fisherman_mercury_levels.csv')
data
```

[1]:

```
    NumFishMeals  TotalMercury
0              2          2.60
1             15          7.06
2             17          8.63
3             23         14.11
4             21          8.05
..           …             …
95            32         20.23
96            23         11.27
97            17          8.47
98            13          5.78
99            32         17.80

[100 rows x 2 columns]
```

[2]:

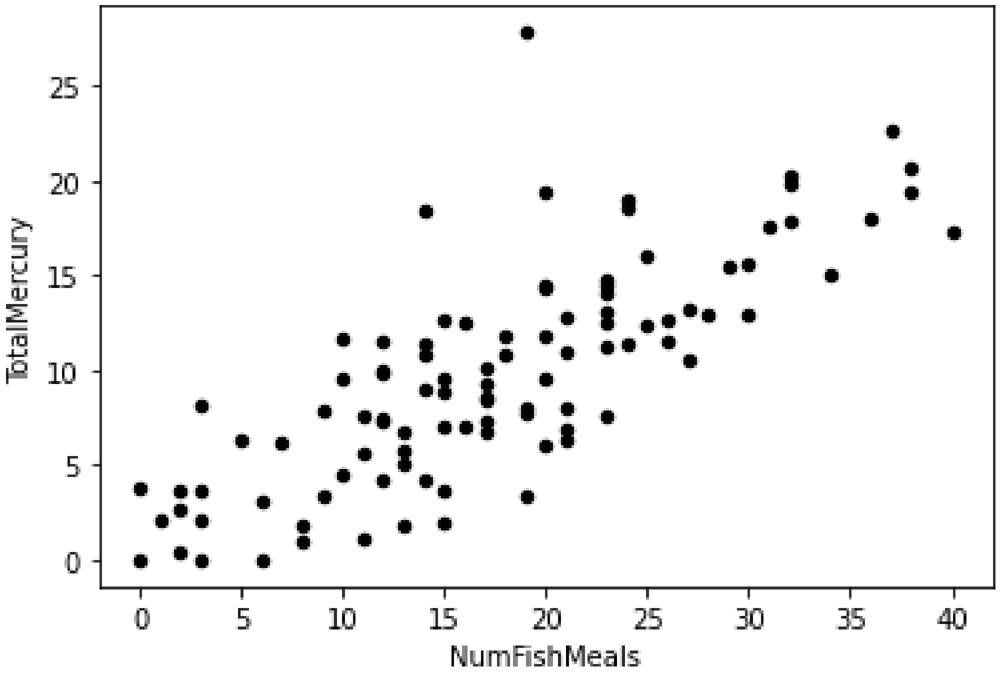
```
#preprocess
print("No.of unassigned values are:::",data.isna().sum().sum())
print("\nInformation and description of data\n")
print(data.info())
print('\n',data.describe())
```

```
No.of unassigned values are::: 0

Information and description of data

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   NumFishMeals  100 non-null    int64
 1   TotalMercury  100 non-null    float64
dtypes: float64(1), int64(1)
memory usage: 1.7 KB
None

        NumFishMeals  TotalMercury
count     100.000000    100.000000
mean       17.510000      9.811300
std         9.298088      5.771316
min         0.000000      0.000000
25%        12.000000      6.012500
50%        17.000000      9.525000
75%        23.000000     12.942500
max        40.000000     27.800000
```

[3]:

```
print("                          SCATTER PLOT")
g=data.plot.scatter(x='NumFishMeals',y='TotalMercury',c='Black')
```

```
                          SCATTER PLOT
```



[4]:

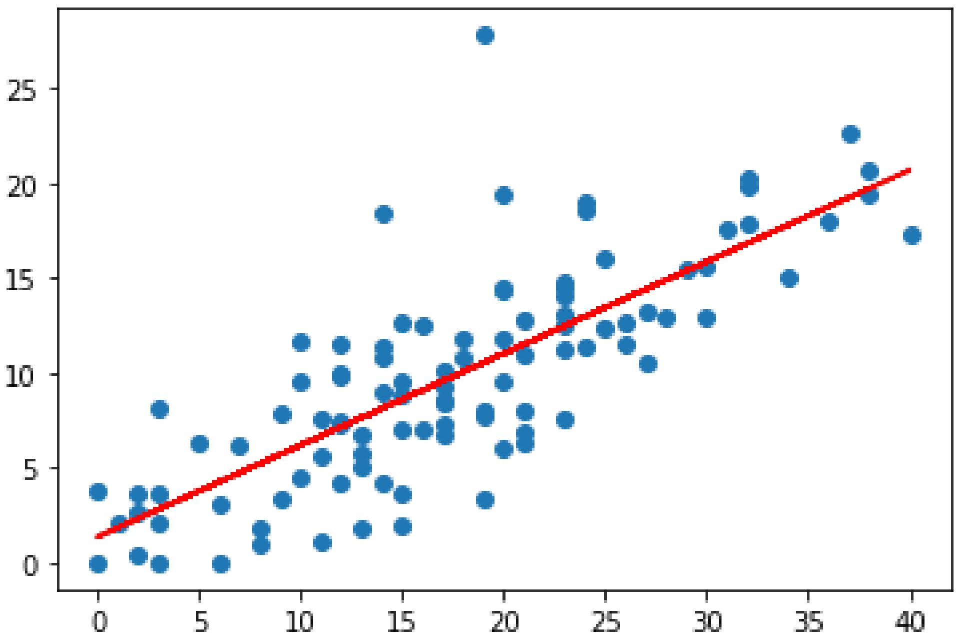
```
#correlation coefficient
pearsoncorr=data.corr(method='pearson')
print(pearsoncorr)
from scipy.stats import pearsonr
c,_=pearsonr(data.iloc[:,0],data.iloc[:,1])
```

```
print("Correlation coefficient is",c)
if -1<c and c<-0.75:
    print("The relation is STRONG NEGATIVE")
if -0.75<c and c<-0.5:
    print("The relation is MODERATE NEGATIVE")
if -0.5<c and c<0.5:
    print("There is NO RELATION")
if 0.5<c and c<0.75:
    print("The relation is MODERATE NEGATIVE")
if 0.75<c and c<1:
    print("The relation is STRONG POSITIVE")
```

```
              NumFishMeals  TotalMercury
NumFishMeals      1.000000      0.777984
TotalMercury      0.777984      1.000000
Correlation coefficient is 0.7779836869474293
The relation is STRONG POSITIVE
```

[5]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
X=data['NumFishMeals'].values.reshape(-1,1)
y=data['TotalMercury'].values.reshape(-1,1)
regressor=LinearRegression()
regressor.fit(X,y)
y_pred=regressor.predict(X)
plt.scatter(X, y)
plt.plot(X, y_pred, color='red')
plt.show()
print("SLOPE---",regressor.coef_[0][0],"INTERCEPT---",regressor.intercept_[0])
print("PERCENT OF ACCURACY(R-squared)--",r2_score(y,y_pred))
print(regressor.score(X,y))
```



```
SLOPE--- 0.48289385780331556 INTERCEPT--- 1.3558285498639435
PERCENT OF ACCURACY(R-squared)-- 0.6052586171563162
0.6052586171563162
```