# ProblemSet2

# Jiang Yufeng 21009200038

## P-1

### Code

|  |
| --- |
| import numpy  import pandas  from algorithm import statistical\_model\_analysis  #生成案例数据  data = pandas.DataFrame({  "A": numpy.random.random(size=20),  "B": numpy.random.random(size=20)  })  result = statistical\_model\_analysis.cluster\_analysis(data, cluster\_num=3)  print(result) |

### Results

|  |  |  |  |
| --- | --- | --- | --- |
| Type | x | y | Data # |
| 2 | 1.9 | 0.97 | 1 |
| 2 | 1.76 | 0.84 | 2 |
| 2 | 2.32 | 1.63 | 3 |
| 2 | 2.31 | 2.09 | 4 |
| 2 | 1.14 | 2.11 | 5 |
| 1 | 5.02 | 3.02 | 6 |
| 1 | 5.74 | 3.84 | 7 |
| 3 | 2.25 | 3.98 | 8 |
| 3 | 4.71 | 3.6 | 9 |
| 3 | 3.17 | 4.96 | 10 |

## P-2

### Code

|  |
| --- |
| import numpy  import pandas  from algorithm import statistical\_model\_analysis  #Generate case data  data\_x1 = pandas.DataFrame({  "A": numpy.random.random(size=100),  "B": numpy.random.random(size=100)  })  data\_x2 = pandas.DataFrame({"C": numpy.random.choice(["1", "2", "3"], size=100)})  data\_y = pandas.Series(data=numpy.random.choice([1, 2], size=100), name="Y")  result = statistical\_model\_analysis.linear\_regression(data\_y=data\_y, data\_x1=data\_x1, data\_x2=data\_x2)  print(result) |

|  |
| --- |
| x=[4,8,12,16]  y=[10,45,69,87]  poly=polyfit(x,y,2) |

|  |
| --- |
| x=[4,8,12,16]  y=[10,45,69,87]  poly =[-0.2656,11.6875,-32.2500]  mse=0  for i in range(4):  mse+=(y[i]-sum(poly[j]\*(x[i]\*\*(2-j)) for j in range(3)))\*\*2  print(mse) |

### Result

The MSE for the first method is 73.5, and for the second one is only 1.25. Obviously, the better method is polynomial regression.

### P-3

### Q1.

The recall score of both classifiers are 0.772 and 0.4.

### Q2

The precision of both classifiers are 0.85 and 0.8.

### Q3

Classifier 2 is better.