Homework Assignment Hw 5

보고서 및 논문 윤리 서약

1. 나는 보고서 및 논문의 내용을 조작하지 않겠습니다.
2. 나는 다른 사람의 보고서 및 논문의 내용을 내 것처럼 무단으로 복사하지 않겠습니다.
3. 나는 다른 사람의 보고서 및 논문의 내용을 참고하거나 인용할 시 참고 및 인용 형식을 갖추고 출처를 반드시 밝히겠습니다.
4. 나는 보고서 및 논문을 대신하여 작성하도록 청탁하지도 청탁받지도 않겠습니다.

나는 보고서 및 논문 작성 시 위법 행위를 하지 않고, 명지인으로서 또한 공학인으로서 나의 양심과 명예를 지킬 것을 약속합니다.

학 과 : 융합소프트웨어학부 데이터테크놀로지전공

과 목 : 인공지능

담당교수 : 전종훈

강좌 번호: 6019

학 번 : 60182196

이 름 : 이동혁 (서명)

1.

```
# 1
import numpy as np

arr = [200, 300, 400, 600, 1000]
minMax = []
for num in arr:
    minMax.append((num-min(arr))/(max(arr)-min(arr)))
print('(a)₩n')
print('Min-max normalization :',minMax,'₩n')

z_score = []
for num in arr:
    z_score.append((num-np.mean(arr))/np.std(arr))
print('(b)₩n')
print('Z-score normalization :',z_score)
```

```
(a)

Min-max normalization : [0.0, 0.125, 0.25, 0.5, 1.0]

(b)

Z-score normalization : [-1.0606601717798212, -0.7071067811865475, -0.35355339059327373, 0.35355339059327373, 1.7677669529663687]
```

2.

```
# 2
import pandas as pd
x = pd.DataFrame({'age' : [23, 23, 27, 27, 39, 41, 47, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61],
                  '%fat' : [9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2, 34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7]})

z_age = []
for num in x['age']:
    z_age.append((num-np.mean(x['age']))/np.std(x['age']))

z_fat = []
for num in x['%fat']:
    z_fat.append((num-np.mean(x['%fat']))/np.std(x['%fat']))

y = pd.DataFrame({'age' : z_age, '%fat' : z_fat})
print('(a)₩n')
print(y,'₩n')

print('(b)₩n')
corr = x.corr(method = 'pearson')
print(corr) # 상관계수 확인
print(x.cov()) # 공분산 (두 변수간 편차 곱의 평균) 확인
```

(a)

	age	%fat
0	-1.825011	-2.144104
1	-1.825011	-0.253883
2	-1.513635	-2.333126
3	-1.513635	-1.221231
4	-0.579506	0.290946
5	-0.423818	-0.320596
6	0.043247	-0.153812
7	0.198935	-0.176050
8	0.276779	0.268708
9	0.432467	0.646752
10	0.588155	1.525149
11	0.588155	0.001853
12	0.743843	0.513325
13	0.821687	0.157518
14	0.899531	0.591157
15	0.899531	0.457730
16	1.055220	1.380603
17	1.133064	0.769061

(b)

	age	%fat
age	1.000000	0.817619
%fat	0.817619	1.000000

	age	%fat
age	174.732026	100.019608
%fat	100.019608	85.643824

두 변수의 correlation coefficient는 0.817619이다. 두 변수는 positively correlated 관계이다. 두 변수의 공분산은 100.019608이다.

3.

(a)

```
# 3

age = [13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70]

bin1 = [13, 15, 16]
bin2 = [16, 19, 20]
bin3 = [20, 21, 22]
bin4 = [22, 25, 25]
bin5 = [25, 25, 30]
bin6 = [33, 33, 35]
bin7 = [35, 35, 35]
bin8 = [36, 40, 45]
bin9 = [46, 52, 70]

#print(np.mean(bin1)) -> 14.6
#print(np.mean(bin2)) -> 18.3
#print(np.mean(bin3)) -> 21
#print(np.mean(bin4)) -> 24
#print(np.mean(bin5)) -> 26.6
#print(np.mean(bin6)) -> 33.6
#print(np.mean(bin7)) -> 35
#print(np.mean(bin8)) -> 40.3
#print(np.mean(bin9)) -> 56

m_bin1 = [14.6, 14.6, 14.6]
m_bin2 = [18.3, 18.3, 18.3]
m_bin3 = [21, 21, 21]
m_bin4 = [24, 24, 24]
m_bin5 = [26.6, 26.6, 26.6]
m_bin6 = [33.6, 33.6, 33.6]
m_bin7 = [35, 35, 35]
m_bin8 = [40.3, 40.3, 40.3]
m_bin9 = [56, 56, 56]
```

(b)

```
outliers = []
q1 = np.quantile(age,0.25)
q3 = np.quantile(age,0.75)
IQR = q3-q1

for num in age:
    if(num<q1-1.5*IQR or num>q3+1.5*IQR):
        outliers.append(num)
print('(b)₩n')
print(outliers) # IQR 방식을 사용하여 q1-1.5*IQR 미만이거나 q3+ 1.5*IQR 초과인 것은 outliers이다.
```

(b)

[70]

IQR 방식을 사용하여 Q1-1.5*IQR 미만이거나 Q3+1.5IQR 초과인 것은 outliers이다.

따라서 70은 outlier이다.

4.

```
# 4

price = [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]
m = 3

def equifreq(arr1, m):
    a = len(arr1)
    n = int(a / m)
    for i in range(0, m):
        arr = []
        for j in range(i * n, (i + 1) * n):
            if j >= a:
                break
            arr = arr + [arr1[j]]
        print(arr)

def equiwidth(arr1, m):
    a = len(arr1)
    w = int((max(arr1) - min(arr1)) / m)
    min1 = min(arr1)
    arr = []
    for i in range(0, m + 1):
        arr = arr + [min1 + w * i]
    arri=[]

    for i in range(0, m):
        temp = []
        for j in arr1:
            if j >= arr[i] and j <= arr[i+1]:
                temp += [j]
        arri += [temp]
    print(arri)
```

```
print('(a)₩n')
print("equal frequency binning")
equifreq(price, m)

print('₩n(b)₩n')
print("equal width binning")
equiwidth(price, m)

# https://www.geeksforgeeks.org/binning-in-data-mining/ (참고)
```

```
(a)

equal frequency binning
[5, 10, 11, 13]
[15, 35, 50, 55]
[72, 92, 204, 215]

(b)

equal width binning
[[5, 10, 11, 13, 15, 35, 50, 55, 72], [92], [204, 215]]
```

5.

```
# 5

from sklearn.preprocessing import minmax_scale

df = pd.DataFrame({'A' : [100, 0, 40, 80, 20],
                   'B' : ['big', 'small', 'medium', 'big', 'small']})

df['A'] = minmax_scale(df.A.astype(float))

one_hot = pd.get_dummies(df['B'])
df = df.drop('B', axis = 1)
df = df.join(one_hot)

print(df)
```

```
     A  big  medium  small
0  1.0    1       0      0
1  0.0    0       0      1
2  0.4    0       1      0
3  0.8    1       0      0
4  0.2    0       0      1
```