**Program Assignment(week4)**

201635825 오성원

**Ex1\_StandardScaler**

|  |
| --- |
| Source code |
| import pandas as pd  import numpy as np  import seaborn as sns  import matplotlib.pyplot as plt  from sklearn import preprocessing  np.random.seed(1)  df =pd.DataFrame({  'x1': np.random.normal (0, 2,10000),  'x2': np.random.normal (5, 3,10000),  'x3': np.random.normal (5, 5, 10000)  })  scaler = preprocessing.StandardScaler()  scaled\_df = scaler.fit\_transform(df)  scaled\_df = pd.DataFrame(scaled\_df, columns=['x1','x2','x3'])  fig, (ax1, ax2) = plt.subplots(ncols =2, figsize=(6,5))  ax1.set\_title('Before Scaling')  sns.kdeplot(df['x1'],ax=ax1)  sns.kdeplot(df['x2'],ax=ax1)  sns.kdeplot(df['x3'],ax=ax1)  ax2.set\_title('After Standard Scaler')  sns.kdeplot(scaled\_df['x1'], ax=ax2)  sns.kdeplot(scaled\_df['x2'], ax=ax2)  sns.kdeplot(scaled\_df['x3'], ax=ax2)  plt.show() |

|  |
| --- |
| Result screen |
|  |

**Ex2\_MinMaxScaler**

|  |
| --- |
| Source code |
| import pandas as pd  import numpy as np  import seaborn as sns  import matplotlib.pyplot as plt  from sklearn import preprocessing  np.random.seed(1)  df =pd.DataFrame({  'x1': np.random.normal (0, 2,10000),  'x2': np.random.normal (5, 3,10000),  'x3': np.random.normal (5, 5, 10000)  })  scaler = preprocessing.MinMaxScaler()  scaled\_df = scaler.fit\_transform(df)  scaled\_df = pd.DataFrame(scaled\_df, columns=['x1','x2','x3'])  fig, (ax1, ax2) = plt.subplots(ncols =2, figsize=(6,5))  ax1.set\_title('Before Scaling')  sns.kdeplot(df['x1'],ax=ax1)  sns.kdeplot(df['x2'],ax=ax1)  sns.kdeplot(df['x3'],ax=ax1)  ax2.set\_title('After MinMax Scaler')  sns.kdeplot(scaled\_df['x1'], ax=ax2)  sns.kdeplot(scaled\_df['x2'], ax=ax2)  sns.kdeplot(scaled\_df['x3'], ax=ax2)  plt.show() |

|  |
| --- |
| Result screen |
|  |

**Ex3\_RobustScaler**

|  |
| --- |
| Source code |
| import pandas as pd  import numpy as np  import seaborn as sns  import matplotlib.pyplot as plt  from sklearn import preprocessing  np.random.seed(1)  df =pd.DataFrame({  'x1': np.random.normal (0, 2,10000),  'x2': np.random.normal (5, 3,10000),  'x3': np.random.normal (5, 5, 10000)  })  scaler = preprocessing.RobustScaler()  scaled\_df = scaler.fit\_transform(df)  scaled\_df = pd.DataFrame(scaled\_df, columns=['x1','x2','x3'])  fig, (ax1, ax2) = plt.subplots(ncols =2, figsize=(6,5))  ax1.set\_title('Before Scaling')  sns.kdeplot(df['x1'],ax=ax1)  sns.kdeplot(df['x2'],ax=ax1)  sns.kdeplot(df['x3'],ax=ax1)  ax2.set\_title('After Rodust Scaler')  sns.kdeplot(scaled\_df['x1'], ax=ax2)  sns.kdeplot(scaled\_df['x2'], ax=ax2)  sns.kdeplot(scaled\_df['x3'], ax=ax2)  plt.show() |

|  |
| --- |
| Result Screen |
|  |

**Ex4**

|  |
| --- |
| Source code |
| import numpy as np  score = [20, 15, 26, 32, 18, 28, 35, 14, 26, 22, 17]  mean = sum(score) / len(score)  print("The mean: ",round(mean,2))  sd = np.std(score)  print("The standard deviation: ",round(sd,2))  sc=[]  for i in score:  z= (i-mean)/sd  sc.append(z)  print("The standard scores: ")  for i in sc:  print(round(i,2), end=" ")  print("\nThe student scores",end=" ")  count=0  for i in range(0,len(sc)):  if sc[i]<-1 and count==0:  print(score[i], end=" ")  count+=1  elif sc[i]<-1 and count==1:  print("and",score[i], end=" ")  print("stand out to receive an F") |

|  |
| --- |
| Result screen |
|  |

**Ex5**

|  |
| --- |
| Source code |
| import numpy as np  score = [28, 35, 26, 32, 28, 28, 35, 34, 46, 42, 37]  mean = sum(score) / len(score)  print("The mean: ",round(mean,2))  sd = np.std(score)  print("The standard deviation: ",round(sd,2))  sc=[]  for i in score:  z= (i-mean)/sd  sc.append(z)  print("The standard scores: ")  for i in sc:  print(round(i,2), end=" ")  print("\nThe student scores",end=" ")  count=0  for i in range(0,len(sc)):  if sc[i]<-1 and count==0:  print(score[i], end=" ")  count+=1  elif sc[i]<-1 and count==1:  print("and",score[i], end=" ")  print("stand out to receive an F") |

|  |
| --- |
| Result screen |
|  |