41071102H 徐敏皓 HW4

Dataset: https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset

Code: https://colab.research.google.com/drive/1wO4C_wjygeAm_TS4W-axBfJlFzTsjCmK#scrollTo=SyAwBnPGQn4H

資料前處理-Missing value (處理前)

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	NaN
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea

印出的前五筆資料中有發現Sleep Disorder出現NaN，看似是missing value，但此attribute是在調查受試者是否有睡眠障礙(Sleep Disorder)，Nan代表受試者沒有此問題，為後續前處理方便，將NaN值替換成NO

處理前missing value的數量

```
資料處理前的missing values數量:
Person ID                    0
Gender                       0
Age                          0
Occupation                   0
Sleep Duration               0
Quality of Sleep             0
Physical Activity Level      0
Stress Level                 0
BMI Category                 0
Blood Pressure               0
Heart Rate                   0
Daily Steps                  0
Sleep Disorder             219
dtype: int64
```

資料前處理-Missing value (處理後)

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	NO
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NO
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NO
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea

處理後NaN值替換成NO

處理後missing value的數量

```
資料處理後的missing values數量
Person ID                  0
Gender                     0
Age                        0
Occupation                 0
Sleep Duration             0
Quality of Sleep           0
Physical Activity Level    0
Stress Level               0
BMI Category               0
Blood Pressure             0
Heart Rate                 0
Daily Steps                0
Sleep Disorder             0
dtype: int64
```

實作程式碼片段

```
# 將 NaN 值替換成 "NO"
print("將NaN替換成NO")
df.fillna('NO', inplace=True)
```

資料前處理-Sampling

資料處理前數目共有 374 筆

隨機抽樣後的前5筆資料:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
102	103	Female	36	Teacher	7.2	8	60	4	Normal	115/75	68	7000	NO
55	56	Male	32	Doctor	6.0	6	30	8	Normal	125/80	72	5000	NO
306	307	Female	52	Accountant	6.5	7	45	7	Overweight	130/85	72	6000	Insomnia
188	189	Female	43	Teacher	6.7	7	45	4	Overweight	135/90	65	6000	Insomnia
246	247	Male	44	Salesperson	6.3	6	45	7	Overweight	130/85	72	6000	Insomnia

---

資料處理後數目共有 200 筆

利用sample_without_replacement將資料隨機取樣，原本有374筆資料，隨機取樣後剩200筆資料

實作程式碼片段

```
#  設定母體大小（總行數）和要抽取的樣本數
population  =  len(df)
samples  =  200    #  假設我們想抽取  200  個樣本

#  使用  sample_without_replacement  進行抽樣
indices  =  sample_without_replacement(n_population=population,  n_samples=samples)
```

資料前處理-OneHotEncoder (處理前)

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	103	Female	36	Teacher	7.2	8	60	4	Normal	115/75	68	7000	NO
1	56	Male	32	Doctor	6.0	6	30	8	Normal	125/80	72	5000	NO
2	307	Female	52	Accountant	6.5	7	45	7	Overweight	130/85	72	6000	Insomnia
3	189	Female	43	Teacher	6.7	7	45	4	Overweight	135/90	65	6000	Insomnia
4	247	Male	44	Salesperson	6.3	6	45	7	Overweight	130/85	72	6000	Insomnia

印出的前五筆資料作為參考，對“Gender”這個attribute做OneHotEncoder

資料前處理-OneHotEncoder (處理後)

OneHotEncoder後的數據:

	Person ID	Gender_Female	Gender_Male	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	103	1.0	0.0	36	Teacher	7.2	8	60	4	Normal	115/75	68	7000	NO
1	56	0.0	1.0	32	Doctor	6.0	6	30	8	Normal	125/80	72	5000	NO
2	307	1.0	0.0	52	Accountant	6.5	7	45	7	Overweight	130/85	72	6000	Insomnia
3	189	1.0	0.0	43	Teacher	6.7	7	45	4	Overweight	135/90	65	6000	Insomnia
4	247	0.0	1.0	44	Salesperson	6.3	6	45	7	Overweight	130/85	72	6000	Insomnia

資料處理後，會把原attribute中所有的值當成新的attributes。

此範例中原本”Gender”欄位中的值只有”Female”和“Male”，因此這兩個值就變成新的attributes，所有的資料在這兩個新欄位中只會出現唯一一個1，不會同時兩個欄位皆為0或皆為1

實作程式碼片段

```
# 初始化 OneHotEncoder
encoder = OneHotEncoder(sparse_output=False)

# 選擇需要編碼的分類變數
X = df['Gender']

# 套用 OneHotEncoder
X_encoded = encoder.fit_transform(X)
```

資料前處理-Discretization

年齡最小為: 27
年齡最大為: 59

	Person ID	Gender_Female	Gender_Male	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder	Discretized Age
0	103	1.0	0.0	36	Teacher	7.2	8	60	4	Normal	115/75	68	7000	NO	1.0
1	56	0.0	1.0	32	Doctor	6.0	6	30	8	Normal	125/80	72	5000	NO	0.0
2	307	1.0	0.0	52	Accountant	6.5	7	45	7	Overweight	130/85	72	6000	Insomnia	3.0
3	189	1.0	0.0	43	Teacher	6.7	7	45	4	Overweight	135/90	65	6000	Insomnia	2.0
4	247	0.0	1.0	44	Salesperson	6.3	6	45	7	Overweight	130/85	72	6000	Insomnia	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
195	167	0.0	1.0	41	Engineer	7.3	8	70	6	Normal Weight	121/79	72	6200	NO	1.0
196	69	1.0	0.0	33	Scientist	6.2	6	50	6	Overweight	128/85	76	5500	NO	0.0
197	313	1.0	0.0	52	Engineer	8.4	9	30	3	Normal	125/80	65	5000	NO	3.0
198	298	1.0	0.0	50	Nurse	6.1	6	90	8	Overweight	140/95	75	10000	Sleep Apnea	2.0
199	368	1.0	0.0	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	Sleep Apnea	3.0

區間化前先確認年齡最小與最大的歲數後，確認兩者相差33歲

在此打算分成4個區間(約每8歲一區間)，區間化後同一區間的年齡會以同一個數字表示

0: [27, 34]，1: [35, 42]，2: [43, 50]，3: [51, 59]

實作程式碼片段

```
# 假設我們對 Age 做離散化
features = df[['Age']]

# 設置 K-bins Discretizer
kbin_discretizer = KBinsDiscretizer(n_bins=4, encode='ordinal', strategy='uniform')

# 執行離散化
discretized_features = kbin_discretizer.fit_transform(features)

# 將結果存回 DataFrame
df['Discretized Age'] = discretized_features
```

P.S.區間化後的資料不直接替換掉Age是為了方便做對照

資料前處理-Normalize (處理前)

	Person ID	Gender_Female	Gender_Male	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder	Discretized Age
0	103	1.0	0.0	36	Teacher	7.2	8	60	4	Normal	115/75	68	7000	NO	1.0
1	56	0.0	1.0	32	Doctor	6.0	6	30	8	Normal	125/80	72	5000	NO	0.0
2	307	1.0	0.0	52	Accountant	6.5	7	45	7	Overweight	130/85	72	6000	Insomnia	3.0
3	189	1.0	0.0	43	Teacher	6.7	7	45	4	Overweight	135/90	65	6000	Insomnia	2.0
4	247	0.0	1.0	44	Salesperson	6.3	6	45	7	Overweight	130/85	72	6000	Insomnia	2.0

印出的前五筆資料作為參考，對這三個attributes做normalize

資料前處理-Normalize (處理後)

將 Quality of Sleep, Physical Activity Level, Stress Level 正規化後:

	Person ID	Gender_Female	Gender_Male	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder	Discretized Age
0	103	1.0	0.0	36	Teacher	7.2	0.8	0.50	0.2	Normal	115/75	68	7000	NO	1.0
1	56	0.0	1.0	32	Doctor	6.0	0.4	0.00	1.0	Normal	125/80	72	5000	NO	0.0
2	307	1.0	0.0	52	Accountant	6.5	0.6	0.25	0.8	Overweight	130/85	72	6000	Insomnia	3.0
3	189	1.0	0.0	43	Teacher	6.7	0.6	0.25	0.2	Overweight	135/90	65	6000	Insomnia	2.0
4	247	0.0	1.0	44	Salesperson	6.3	0.4	0.25	0.8	Overweight	130/85	72	6000	Insomnia	2.0

因為設定的範圍是[0, 1]，因此所有值normalize後皆會落在此區間

實作程式碼片段

```
# 建立 MinMaxScaler 物件
scaler = MinMaxScaler(feature_range=(0, 1))


features = ['Quality of Sleep', 'Physical Activity Level', 'Stress Level']


# 套用 MinMaxScaler
df[features] = scaler.fit_transform(df[features])
```

資料前處理-Feature selection (處理前)

	Person ID	Gender_Female	Gender_Male	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder	Discretized Age
0	103	1.0	0.0	36	Teacher	7.2	0.8	0.50	0.2	Normal	115/75	68	7000	NO	1.0
1	56	0.0	1.0	32	Doctor	6.0	0.4	0.00	1.0	Normal	125/80	72	5000	NO	0.0
2	307	1.0	0.0	52	Accountant	6.5	0.6	0.25	0.8	Overweight	130/85	72	6000	Insomnia	3.0
3	189	1.0	0.0	43	Teacher	6.7	0.6	0.25	0.2	Overweight	135/90	65	6000	Insomnia	2.0
4	247	0.0	1.0	44	Salesperson	6.3	0.4	0.25	0.8	Overweight	130/85	72	6000	Insomnia	2.0

印出最原始的前五筆資料作為參考，會先將整份資料做些處理後才正式做Feature selection。

(因為做特徵選取時資料必須為數值才有辦法執行)

總共對以下三項欄位做處理:

1. 將Age欄位做整理，只保留discretize後的。
2. 將Blood Pressure中的收縮壓和舒張壓分離出來。
3. 將BMI Category的文字轉成數字後的資料

資料前處理-Feature selection (處理中)

將Age欄位做整理，只保留discretize後的。以及將Blood Pressure中的收縮壓和舒張壓分離出來。最後還有將BMI Category的文字轉成數字後的資料

	Person ID	Gender_Female	Gender_Male	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Systolic	Diastolic	Heart Rate	Daily Steps	Sleep Disorder
0	103	1.0	0.0	1.0	Teacher	7.2	0.8	0.50	0.2	0	115	75	68	7000	NO
1	56	0.0	1.0	0.0	Doctor	6.0	0.4	0.00	1.0	0	125	80	72	5000	NO
2	307	1.0	0.0	3.0	Accountant	6.5	0.6	0.25	0.8	1	130	85	72	6000	Insomnia
3	189	1.0	0.0	2.0	Teacher	6.7	0.6	0.25	0.2	1	135	90	65	6000	Insomnia
4	247	0.0	1.0	2.0	Salesperson	6.3	0.4	0.25	0.8	1	130	85	72	6000	Insomnia

印出最原始的前五筆資料作為參考，會先將整份資料做些處理後才正式做Feature selection。

(因為做特徵選取時資料必須為數值才有辦法執行)

總共對以下三項欄位做處理:

1. 將Age欄位做整理，只保留discretize後的。
2. 以及將Blood Pressure中的收縮壓和舒張壓分離出來。
3. 最後還有將BMI Category的文字轉成數字後的資料

資料前處理-Feature selection (處理後)

```
利用SelectKBest後選擇的特徵: Index(['Gender_Female', 'Gender_Male', 'BMI Category', 'Systolic',
       'Diastolic'],
      dtype='object')
```

處理後，由於k設定為5，因此會將5項與目標最相關的五個特徵挑選出來

實作程式碼片段

```
#  假設數據中有多個特徵
X  =  df.iloc[:,  1:-1]    #  取所有列，除了第一列ID不取(無意義)，最後一列作為特徵
X  =  X.drop(columns  =  ['Occupation'])  #我認為Occupation這個attribute分太細了
y  =  df.iloc[:,  -1]      #  取最後一列作為目標


#  切分資料集
X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,  test_size=0.2,  random_state=42)

#  初始化  SelectKBest，選擇  K=5  的最佳特徵
selector  =  SelectKBest(score_func=f_classif,  k=5)

#  擬合選擇器
X_new  =  selector.fit_transform(X_train,  y_train)

#  獲取選擇的特徵索引
selected_indices  =  selector.get_support(indices=True)

#  獲取選擇的特徵名稱
selected_features  =  X.columns[selected_indices]
```