Kth Nearest Neighbor

ชุดข้อมูล: Psychology.xls

Source Code: 6610450951_KNN.ipynb

Import library ที่สำคัญและอ่านข้อมูลจากชุดข้อมูล

Import libs

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Read data

```
df = pd.read_excel("Psycology.xls")
df.head()
```

```
df = pd.read_excel("Psycology.xls")
df.head()
```

Python

	เพศ	อายุ	เคยมีแฟนมาแล้ว (คน)	จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์	ระยะเวลาการนอน	นิสัยการกินอาหาร	เคยคิดฆ่าตัวตาย	จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์	ระดับความเครียดด้านการเงิน	เคยกินยานอนหลับ	ภาวะซึมเศร้า
0	Male	28	5	3	5-6 ชั่วโมง	อาหารสุขภาพ	Yes	8	3	Yes	Yes
1	Male	23	5	2	มากกว่า 8 ชั่วโมง	อาหารทั่วไป	No	10	4	No	Yes
2	Female	23	1	3	น้อยกว่า 5 ชั่วโมง	อาหารสุขภาพ	Yes	0	3	No	No
3	Female	20	5	5	มากกว่า 8 ชั่วโมง	Junkfood	Yes	2	5	No	Yes
4	Male	29	4	3	มากกว่า 8 ชั่วโมง	Junkfood	Yes	1	3	No	Yes

ตรวจสอบคอลัมน์ว่ามีคอลัมน์ใดบ้าง และแต่ละคอลัมน์ประกอบไปด้วยค่าอะไรบ้าง

ด้วยคำสั่ง .unique สำหรับทุกคอลัมน์ที่ปรากฏ

```
genders = df['เพศ'].unique()
ages = df['อายุ'].unique()
girlfriend_amount = df['เคยมีแฟนมาแล้ว  (คน)'].unique()
exercise = df['จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์'].unique()
sleep_time = df['ระยะเวลาการนอน'].unique()
food_habit = df['นิสัยการกินอาหาร'].unique()
sucide = df['เคยคิดฆ่าตัวตาย'].unique()
library_time = df['จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์'].unique()
money_status = df['ระดับความเครียดด้านการเงิน'].unique()
sleeping_pill = df['เคยกินยานอนหลับ'].unique()
is_sadness = df['ภาวะซึมเศร้า'].unique()

print('เพศ ', genders)
print('อายุ ', ages)
print('เคยมีแฟนมาแล้ว  (คน) ', girlfriend_amount)
print('จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์ ', exercise)
print('ระยะเวลาการนอน', sleep_time)
print('นิสัยการกินอาหาร ', food_habit)
print('เคยคิดฆ่าตัวตาย ', sucide)
print('จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์ ', library_time)
print('ระดับความเครียดด้านการเงิน ', money_status)
print('เคยกินยานอนหลับ ', sleeping_pill)
print('ภาวะซึมเศร้า', is_sadness)
```

```
เพศ  ['Male' 'Female']
อายุ  [28 23 20 29 31 24 33 25 19 34 21 30 32 26 22 27 18]
เคยมีแฟนมาแล้ว  (คน)  [5 1 4 2 3 '?']
จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์  [3 2 5 4 '?' 1]
ระยะเวลาการนอน ['5-6 ชั่วโมง' 'มากกว่า 8 ชั่วโมง' 'น้อยกว่า 5 ชั่วโมง' '7-8 ชั่วโมง']
นิสัยการกินอาหาร  ['อาหารสุขภาพ' 'อาหารทั่วไป' 'Junkfood']
เคยคิดฆ่าตัวตาย  ['Yes' 'No']
จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์  [8 10 0 2 1 3 11 12 9 7 4 6 '?' 5]
ระดับความเครียดด้านการเงิน  [3 4 5 '?' 1 2]
เคยกินยานอนหลับ  ['Yes' 'No']
ภาวะซึมเศร้า ['Yes' 'No']
```

<u>ข้อมูลที่ได้จากชุดข้อมูลนี้</u>

- เพศ
    - ชาย (Male)
    - หญิง (Female)
- อายุ
    - ตัวเลขจำนวนเต็มที่อยู่ในช่วง 18 – 34 ปี
- เคยมีแฟนมาแล้ว (คน)
    - ตัวเลขจำนวนเต็มที่อยู่ในช่วง 1 – 5
    - พบค่าที่หายไป “?”
- จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์
    - ตัวเลขจำนวนเต็มที่อยู่ในช่วง 1 – 5
    - พบค่าที่หายไป “?”
- ระยะเวลาการนอน
    - น้อยกว่า 5 ชั่วโมง
    - 5-6 ชั่วโมง
    - 7-8 ชั่วโมง
    - มากกว่า 8 ชั่วโมง
- นิสัยการกินอาหาร
    - อาหารสุขภาพ
    - อาหารทั่วไป
    - Junkfood
- เคยคิดฆ่าตัวตาย
    - Yes
    - No
- จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์
    - จำนวนเต็มที่อยู่ในช่วง 0 – 12
    - พบค่าที่หายไป “?”
- ระดับความเครียดด้านการเงิน
    - จำนวนเต็มที่อยู่ในช่วง 1 - 5
    - พบค่าที่หายไป “?”
- เคยกินยานอนหลับ
    - Yes
    - No

<u>และผลลัพธ์ของการเป็นโรคซึมเศร้า</u>

- Yes และ No

ทดลองตัดชุดข้อมูลตัวอย่างทุกแถวที่มีสิ่งที่ไม่ทราบค่าปรากฏ

Ignore all rows that has '?' in it

```
#Before
print(f"Data size before cut is {df.shape[0]}")

#cut
df = df[~df.isin(['?']).any(axis=1)]

#After
print(f"Data size after cut is {df.shape[0]}")
```

```
Data size before cut is 502
Data size after cut is 467
```

พบว่ามีชุดข้อมูลที่ถูกตัดออกไปทั้งหมด 35 ตัวอย่างจาก 502 ตัวอย่าง

Map ชุดข้อมูลทั้งตารางให้อยู่ในรูปตัวเลข เพื่อสะดวกต่อการนำไปคิดต่อ

```
df['เพศ'] = df['เพศ'].replace({'Male': 1, 'Female': 0})

#'อายุ' is no need for mapping
#'เคยมีแฟนมาแล้ว  (คน)' is no need for mapping
#'จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์' is no need for mapping

df['ระยะเวลาการนอน'] = df['ระยะเวลาการนอน'].replace({
    '5-6 ชั่วโมง': 5.5,
    'มากกว่า 8 ชั่วโมง': 8.5,
    'น้อยกว่า 5 ชั่วโมง': 4.5,
    '7-8 ชั่วโมง': 7.5
})

df['นิสัยการกินอาหาร'] = df['นิสัยการกินอาหาร'].replace({
    'อาหารสุขภาพ': 0,
    'อาหารทั่วไป': 1,
    'Junkfood': 2
})

df['เคยคิดฆ่าตัวตาย'] = df['เคยคิดฆ่าตัวตาย'].replace({'Yes': 1, 'No': 0})

#'จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์' is no need for mapping
#'ระดับความเครียดด้านการเงิน' is no need for mapping

df['เคยกินยานอนหลับ'] = df['เคยกินยานอนหลับ'].replace({'Yes': 1, 'No': 0})
df['ความซึมเศร้า'] = df['ความซึมเศร้า'].replace({'Yes': 1, 'No': 0})
```

กำหนดให้การ Map ข้อมูลเป็นดังนี้

- เพศ
    - ชาย -> 1
    - หญิง -> 0
- ระยะเวลาการนอน จะใช้การประมาณค่า
    - น้อยกว่า 5 ชั่วโมง -> 4.5
    - 5-6 ชั่วโมง -> 5.5
    - 7-8 ชั่วโมง -> 7.5
    - มากกว่า 8 ชั่วโมง -> 8.5
- นิสัยการกินอาหาร
    - อาหารสุขภาพ -> 0
    - อาหารทั่วไป -> 1
    - Junkfood -> 2

- เคยคิดฆ่าตัวตาย
    - Yes -> 1
    - No -> 0
- เคยกินยานอนหลับ
    - Yes -> 1
    - No - > 0

<u>และ Map ข้อมูลผลลัพธ์เป็น</u>

ภาวะซึมเศร้า

- Yes -> 1
- No -> 0

เตรียม Split ข้อมูล Features และ Target (Training Data)

โดยเตรียมฟีเจอร์ X สำหรับเรียนรู้ โดยตัดคอลัมน์สุดท้าย (เฉลย) ออก

และแปลงข้อมูล (Preprocess) ด้วย Standard Scaler รวมไปถึงกำหนดให้ y คือเฉลย

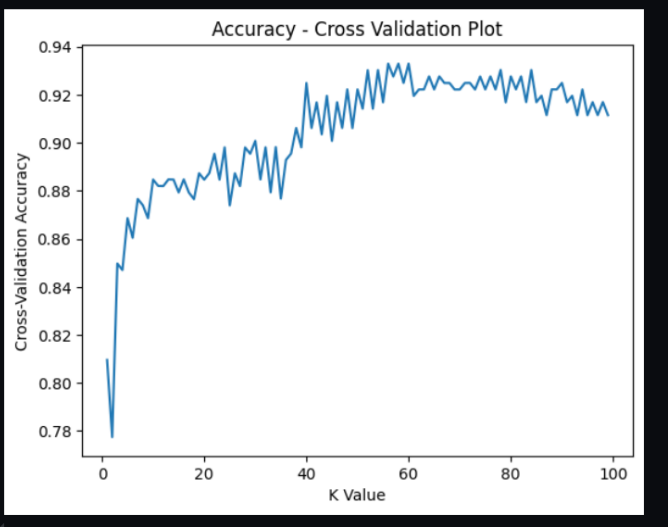
จากนั้น Split ข้อมูล Train: Test ด้วยสัดส่วน 80:20 และกำหนดให้ random_state (seed) คือ 12

<u>ทำ KNN 5 folds cross validation สำหรับ K ตั้งแต่ช่วง 1 ถึง 99</u>

พบว่า ค่า K ที่ดีที่สุด คือ 60

Visualization

```
plt.plot(k_values, cv_scores)
plt.xlabel('K Value')
plt.ylabel('Cross-Validation Accuracy')
plt.title('Accuracy - Cross Validation Plot')
plt.show()
```



จากกราฟพบว่าค่า accuracy ที่สูงที่สุด อยู่ที่จุด K มีค่าเท่ากับ 60

ดูผลลัพท์การ Predict ข้อมูล

โดยที่คอลัมน์แรก เป็นเฉลยจากชุดข้อมูล และคอลัมน์ที่สอง เป็นสิ่งที่เกิดจากการจำแนกด้วย KNN Classifier

Testing The Prediction

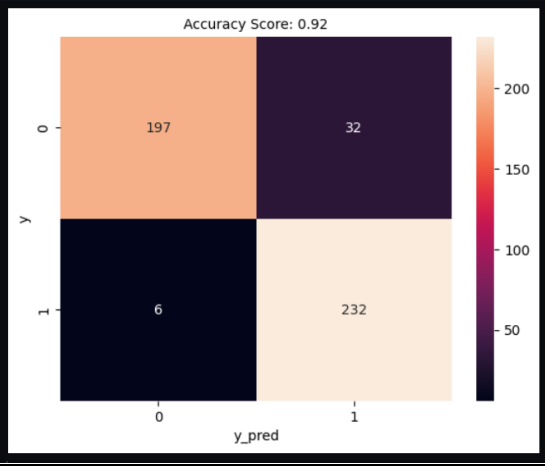
```
knn = KNeighborsClassifier(n_neighbors= optimal_k_values[0]).fit(X_train, y_train)

y_pred = knn.predict(X)
df['ภาวะซึมเศร้า_predict'] = pd.DataFrame(y_pred)
df[['ภาวะซึมเศร้า','ภาวะซึมเศร้า_predict']].head(15)
```

[18]

	ภาวะซึมเศร้า	ภาวะซึมเศร้า_predict
0	1	1.0
1	1	1.0
2	0	0.0
3	1	1.0
4	1	1.0
6	0	0.0
7	0	1.0
9	1	1.0
11	0	0.0
12	1	1.0
13	0	1.0
14	0	1.0
15	1	0.0
16	1	1.0
17	1	1.0

HeatMap

```
acc = round(accuracy_score(y, y_pred), 2)
cm = confusion_matrix(y, y_pred)
sns.heatmap(cm, annot=True, fmt=".0f")
plt.xlabel('y_pred')
plt.ylabel('y')
plt.title(f"Accuracy Score: {acc}", size=10)
plt.show()
```



Classification Report

Classification Report

```
[21] print(classification_report(y, y_pred))
```

```
...              precision    recall  f1-score   support

           0       0.97      0.86      0.91       229
           1       0.88      0.97      0.92       238

    accuracy                           0.92       467
   macro avg       0.92      0.92      0.92       467
weighted avg       0.92      0.92      0.92       467
```