

project data mining

ข้อมูลการออกกำลังกายหรือ  
เล่นกีฬาของประชาชน

แต่งโตมอดแมนหุ่นปานดาราโก้  
& i hate monday



# Our Avengers








ธนบดี ภูษมศรี  
643020502-4  
สถานะ : หม้าย  
กิจกรรม : นั่งสมาธิ





**สุนิสา อุดมขันธ์**  
**643021278-8**  
**กิจกรรม : เต้นบัลเล่ต์**  
**สถานะ : สมรส**



สุชาดา อุปพงษ์  
643020524-4  
กิจกรรม : ร้องจ๊อกกิ้ง  
สถานะ : โสด





ศิริโชค ศิริวิชา  
643020522-8  
กิจกรรม : ไทเก็ก  
สถานะ : หย่าร้าง





# MUAY THAI

ธิตีพร หิงพุดชา  
643021267-3  
กิจกรรม : มวยไทย  
สถานะ : โสด








กรวรรณ อุ๋จอหอ  
643020495-5  
กิจกรรม : เวทเทรนนิ่ง  
สถานะ : สมรส





ลภัสลดา แดงสูงเนิน  
643020518-9  
กิจกรรม : รำมวยจีน  
สถานะ : สมรส





ร้ตชฎากร นามวงศ์  
643020517-1  
กิจกรรม : โยคะ  
สถานะ : โสด



# ที่มาและความสำคัญ

การออกกำลังกาย คือ การทำกิจกรรมที่ได้ออกแรงหรือเคลื่อนไหวร่างกาย เป็นส่วนสำคัญของการรักษาสุขภาพที่ดี ที่จะช่วยเสริมสร้างสมรรถภาพร่างกายในด้านต่างๆ สร้างภูมิคุ้มกัน และลดความเสี่ยงต่อการเป็นโรคต่างๆ เช่น โรคหัวใจ โรคเบาหวาน และความดันโลหิตสูง

การออกกำลังกายที่ดีมีหลากหลายรูปแบบ สามารถเลือกกิจกรรมให้เหมาะสมกับลักษณะทางร่างกายตามแต่ความสะดวกสบายและความสนใจของแต่ละคน เพื่อให้ได้มาซึ่งสุขภาพกายและสุขภาพจิตที่ดี







**MATA DATA**



<u>Name</u>	<u>Data Type</u>	<u>ตัวอย่างข้อมูล</u>
ปี	numeric	2564
ภาค	string	ภาคกลาง, ภาคเหนือ
จังหวัด	string	ขอนแก่น
เพศ	string	ชาย
อายุ	numeric	56
สถานภาพ	string	สมรส, โสด



<u>Name</u>	<u>Data Type</u>	<u>ตัวอย่างข้อมูล</u>
การมีโรคประจำตัว	string	มีโรคประจำตัว, ไม่มีโรคประจำตัว
อาชีพ	string	รับจ้างทั่วไป, ประกอบธุรกิจส่วนตัว
น้ำหนัก	numeric	56
ส่วนสูง	numeric	163
ออกกำลังกาย	string	ออกกำลังกาย, ไม่ออกกำลังกาย
ระดับ	string	ไม่ออกกำลังกาย, ระดับปานกลาง
กิจกรรม	string	ปั่นจักรยาน, วิ่ง, โยคะ, เดิน
เหตุผลที่ออกกำลังกาย	string	คลายเครียด, พักผ่อน, เป็นงานต้องทำ, เป็นอาชีพ



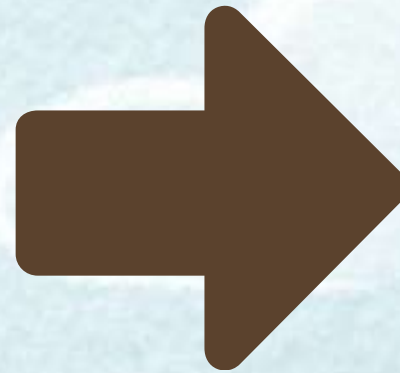


# DATA PREPARATION

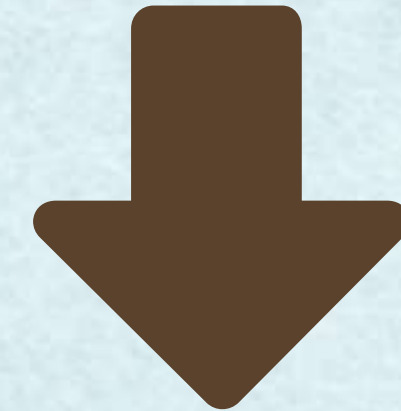


```
df.isnull().any()
```

ปี	False
ภาค	False
จังหวัด	False
อำเภอ	False
ตำบล	False
เพศ	False
อายุ	False
การศึกษา	False
สถานภาพ	False
การมีโรคประจำตัว	False
ประเภทของโรค	False
อาชีพ	False
น้ำหนัก	False
ส่วนสูง	False
กิจกรรม	False
ระดับ	False
นาที่/วัน	False
วัน/สัปดาห์	False
รวมสรุปกิจกรรม	False
สถานที่	False
เวลา	False
เวลารวม (ชั่วโมง)	False
เหตุผล (ออก)	False
เหตุผล (ไม่ออก)	False
แรงจูงใจ	False
ข้อเสนอแนะ	False



เช็คค่า Missing Value



ไม่พบค่า Missing ในข้อมูล



# ลงรหัสข้อมูลในแต่ละคอลัมน์และจัดการ Missing Value

Column ภาค

1 = ภาคเหนือ



3 = ภาคตะวันออก  
เจียงเหนือ



2 = ภาคกลาง



4 = ภาคใต้



แทน “-” เป็น NA

เปลี่ยนกรุงเทพมหานครเป็นภาค

เนื่องจากมีภาคกรุงเทพมหานคร ซึ่งไม่ใช่ และต้องเปลี่ยนเป็นภาคกลาง

```
[ ] df.loc[df['ภาค'] == 'กรุงเทพมหานคร', 'ภาค'] = 'ภาคกลาง'
```





# ลงรหัสข้อมูลในแต่ละคอลัมน์และจัดการ Missing Value

## Column เพศ

1 = หญิง



0 = ชาย



แทน “-” เป็น NA

## Column สถานะภาพ



1 = โสด



2 = สมรส



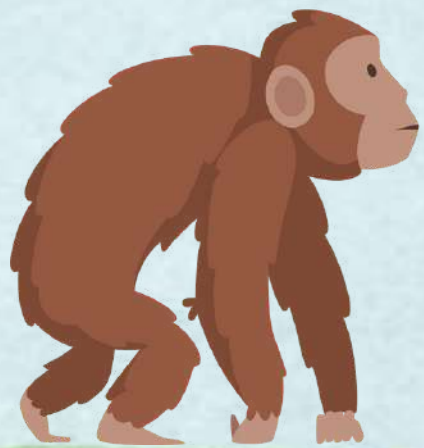
3 = หย่าร้าง/  
หม้าย



# ลงรหัสข้อมูลในแต่ละคอลัมน์และจัดการ Missing Value

Column อายุ

ทำ Boxplot เพื่อดูค่า Outliner จากนั้นดูค่า Min, Max



15-35ปี



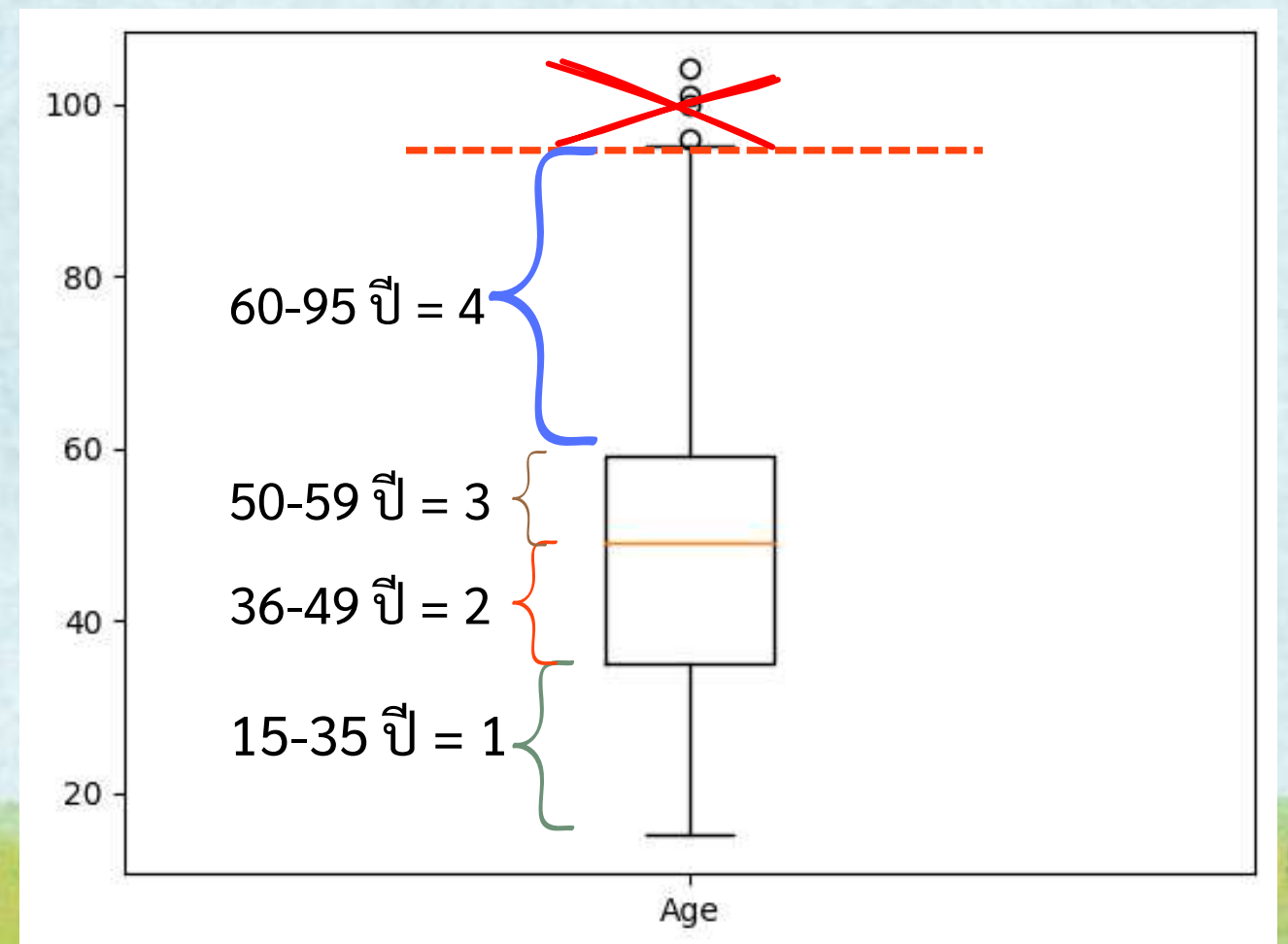
36-49ปี



50-59ปี



60-95ปี





# ลงรหัสข้อมูลในแต่ละคอลัมน์และจัดการ Missing Value

Column เหตุผล

ทำการ Split data แทน “-” เป็น NA

Column: ระดับ

0 = ไม่ออกกำลังกาย

1 = ระดับปานกลาง

2 = ระดับหนัก

แทน “-” เป็น NA

```
gf = df['รวมสรุปกิจกรรม'].str.split(',', expand = True)
```

```
gf1 = gf[0].str.split('(', expand = True)
```

```
gfa = a['รวมสรุปกิจกรรม'].str.split(',', expand = True)
gfa
```

	0	1	2	3
0	(ระดับปานกลาง 225)	None	None	
1	-	None	None	None
2	-	None	None	None
3	(ระดับปานกลาง 135)	None	None	
4	(ระดับปานกลาง 180)	None	None	

	1
0	ระดับปานกลาง
1	None
2	None
3	ระดับปานกลาง



# ลงรหัสข้อมูลในแต่ละคอลัมน์และจัดการ Missing Value

Column กิจกรรม

แยกกิจกรรมเป็นเป็น 2 ประเภท

ลู่วานและสนามจะเป็น 0

วิ่ง ปั่นจักรยาน กระโดดเชือก  
ว่ายน้ำ ฟุตบอล ฟุตซอล  
แบดมินตัน เซปักตะกร้อ  
วอลเลย์บอล เปตอง  
บาสเกตบอล สนุกเกอร์ ฐัดบอล  
กอล์ฟ เทนนิส สเก็ตบอร์ด

ศิลปะและการต่อสู้จะเป็น 1


เต้นแอโรบิค การออกกำลังกาย  
ภายในการประกอบอาชีพ  
โยคะ เพาะกายและฟิตเนส  
เต้นบัสโลบ ฮูลาฮูป กาย  
บริหาร รำมวยจีน มวยไทย  
แกว่งแขน ไทเก็ก เทควันโด  
ลีลาศ เวทเทรนนิ่ง คาคิโอะ  
ปั่นจักสีลัต ชิตอัม ไม้พลอง





Association rule





## เป้าหมาย

หาความสัมพันธ์ระหว่างเพศและเหตุผลในการออกกำลังกาย  
หาความสัมพันธ์ช่วงเวลาและกิจกรรมในการออกกำลังกาย



# เพศ-เหตุผลในการออกกำลังกาย

กำหนด  $\text{min\_sup} = 0.002$

เพศหญิง กับ ควบคุมน้ำหนัก/ลดน้ำหนัก

2	(เหตุผล (ออก)_ควบคุมน้ำหนัก/ลดน้ำหนัก)	(เพศ_หญิง)	0.008757	0.518410	0.006269	0.715909
3	(เพศ_หญิง)	(เหตุผล (ออก)_ควบคุมน้ำหนัก/ลดน้ำหนัก)	0.518410	0.008757	0.006269	0.012093

เพศหญิง กับ ออกกำลังกายเพื่อคลายเครียด/พักผ่อน

	antecedents	consequents	antecedent support	consequent support	support	confidence
0	(เพศ_หญิง)	(เหตุผล (ออก)_คลายเครียด/พักผ่อน)	0.518410	0.010996	0.006020	0.011613
1	(เหตุผล (ออก)_คลายเครียด/พักผ่อน)	(เพศ_หญิง)	0.010996	0.518410	0.006020	0.547511



# เพศ-กิจกรรมในการออกกำลังกาย

กำหนด  $\text{min\_sup} = 0.002$

วอลเลย์บอล กับ เพศหญิง

6	(กิจกรรม_2564_วอลเลย์บอล)	(เพศ_หญิง)	0.007115	0.518410	0.005672	0.797203
7	(เพศ_หญิง)	(กิจกรรม_2564_วอลเลย์บอล)	0.518410	0.007115	0.005672	0.010942

กระโดดเชือก กับ เพศหญิง

	antecedents	consequents	antecedent support	consequent support	support	confidence
0	(กิจกรรม_2564_กระโดดเชือก)	(เพศ_หญิง)	0.003334	0.518410	0.002140	0.641791
1	(เพศ_หญิง)	(กิจกรรม_2564_กระโดดเชือก)	0.518410	0.003334	0.002140	0.004127



# กลยุทธ







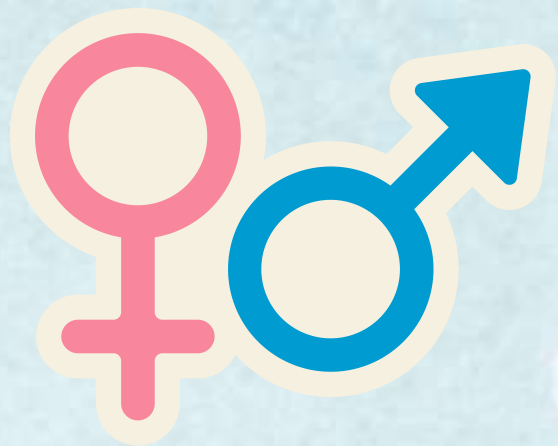
# Decision Tree



# DECISION TREE

FEATURE ที่ใช้

ตัวแปร X ที่ศึกษา



เพศ



อายุ



โรค



ภาค



# DECISION TREE

TRAIN 80%  
TEST 20%

1 [play] X\_train

	Gender	Age	disease	sector
16255	0	2	0	1
18769	1	1	0	3
2424	1	1	0	2
4743	1	2	1	1
15280	0	4	0	2
...	...	...	...	...
1433	1	1	0	2
10556	0	3	1	4
18574	1	3	0	3
18696	0	3	0	3
18954	1	2	0	3

7798 rows × 4 columns

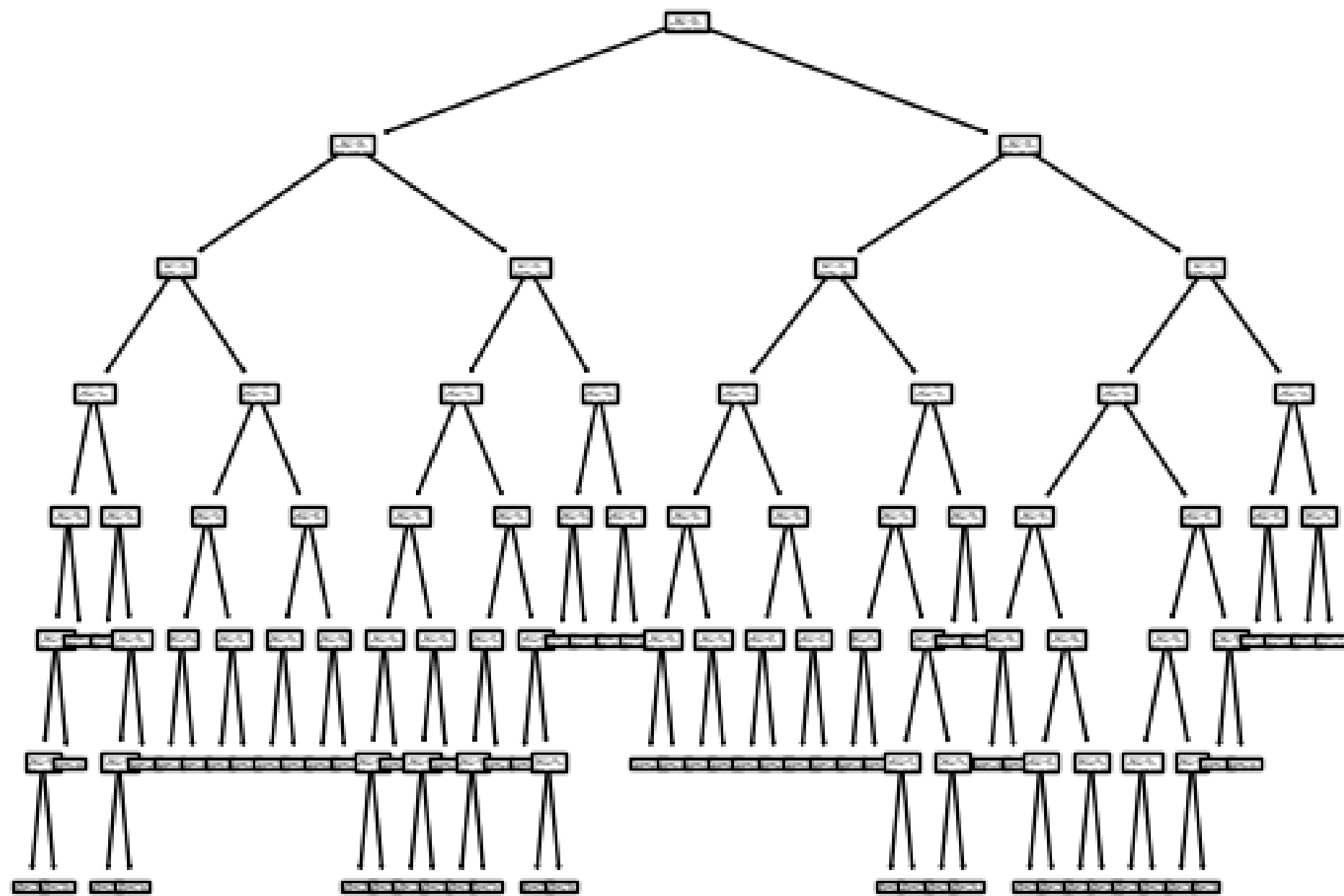
[play] y\_train

16255	1.0
18769	1.0
2424	0.0
4743	0.0
15280	1.0
...	...
1433	0.0
10556	0.0
18574	1.0
18696	1.0
18954	1.0

Name: Activity, Length: 7798, dtype: float64



# DECISION TREE



**NON PARAMETER**

**ACCURACY**



**65.00%**





# DECISION TREE

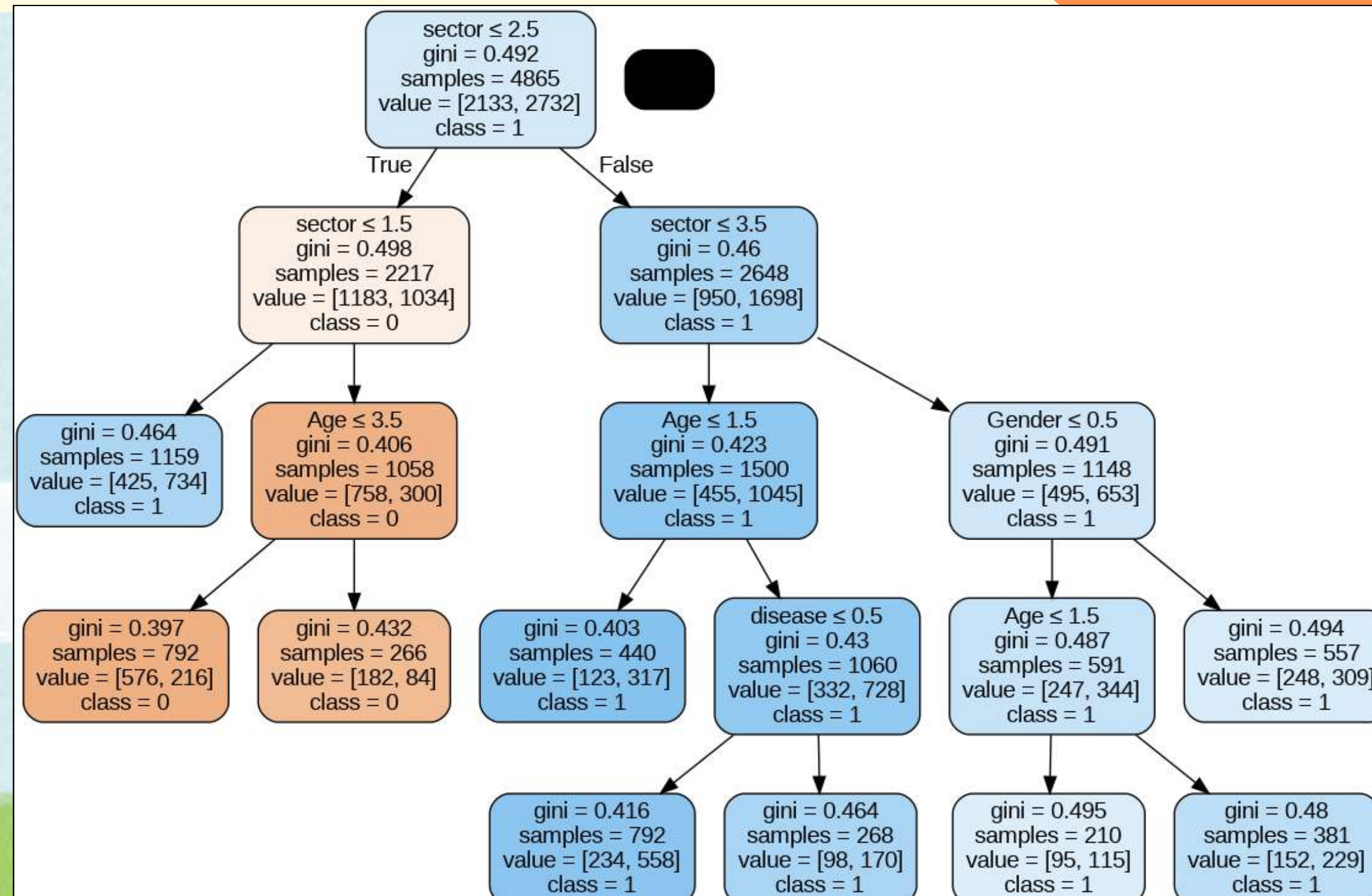
**GIT SEARCH**

**MAX\_DEPTH = 6, MAX\_FEATURES = SQRT,  
MAX\_LEAF\_NODES = 9, MIN\_SAMPLES\_LEAF = 3,  
MIN\_SAMPLES\_SPLIT = 2 BEST SCORE:0.57**



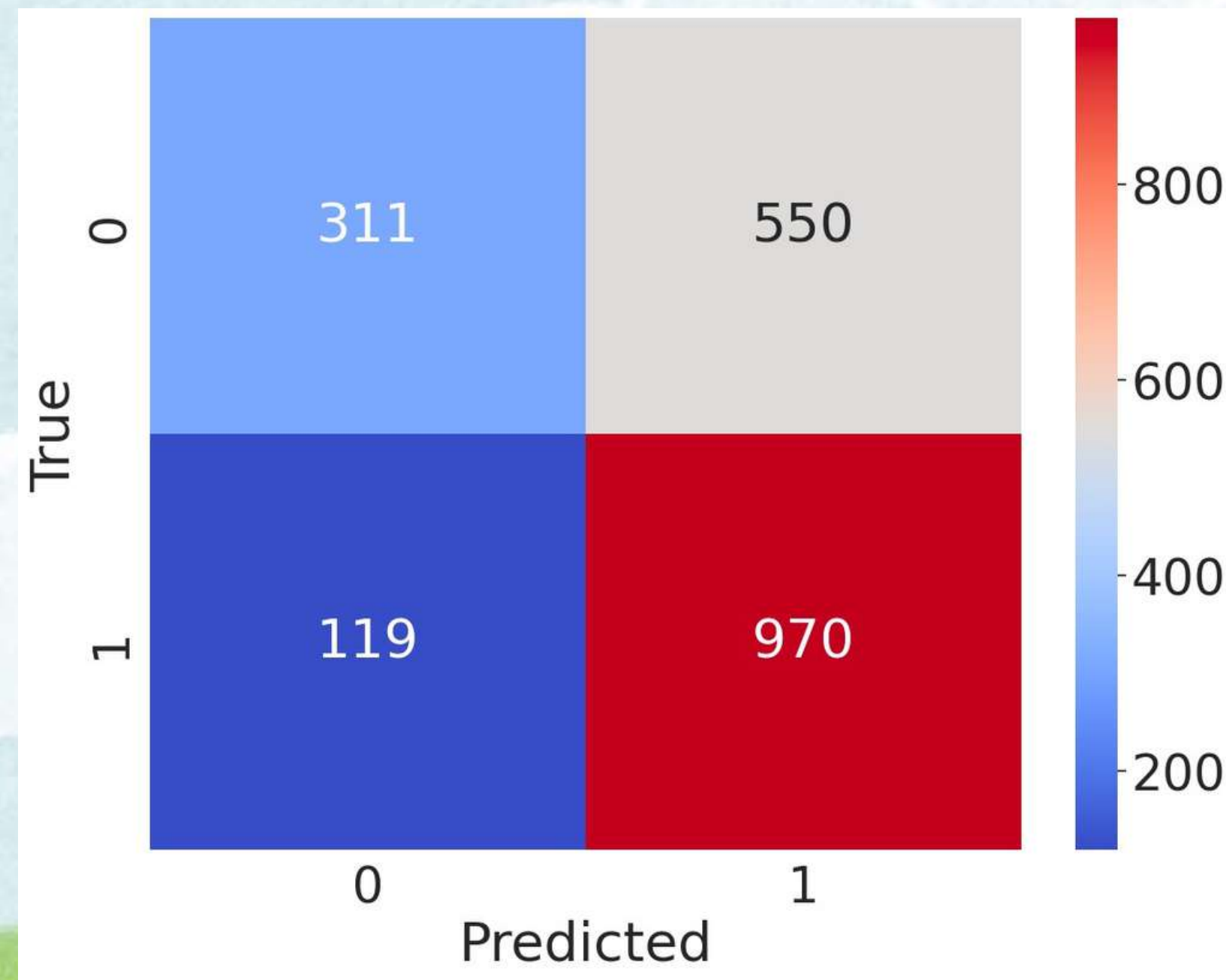
# DECISION TREE

OPTIMAL PARAMETERS





# CONFUSION MATRIX



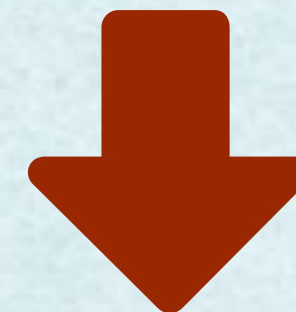


# MODEL EVALUATION

```
accuracy = 0.6569230769230769
```

	precision	recall	f1-score	support
0.0	0.72	0.36	0.48	861
1.0	0.64	0.89	0.74	1089

ACCURACY



65.63%





# K-NEAREST NEIGHBOR

ในที่นี้จะลองกำหนดให้ K มีค่าตั้งแต่ 3,5,7,9,11,13  
เพื่อเลือกจำนวน K ที่มีค่า ACCURACY ที่ดีที่สุด

```
[100] # Instantiate KNN model
      knn1 = KNeighborsClassifier(n_neighbors=3, weights='uniform', algorithm='ball_tree')

      # Fit the model to the training data
      knn1.fit(X_train, y_train)

      # Predict the test data
      y_pred = knn1.predict(X_test)

      #Evaluation
      from sklearn.metrics import accuracy_score, classification_report
      print(f'accuracy = {accuracy_score(y_test, y_pred)}')

      accuracy = 0.5835897435897436
```

K = 3 มีค่า ACCURACY = 58.35%





# K-NEAREST NEIGHBOR

```
# Instantiate KNN model
knn1 = KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='ball_tree')

# Fit the model to the training data
knn1.fit(X_train, y_train)

# Predict the test data
y_pred = knn1.predict(X_test)

#Evaluation
from sklearn.metrics import accuracy_score, classification_report
print(f'accuracy = {accuracy_score(y_test, y_pred)}')

accuracy = 0.5897435897435898
```

K = 5 มีค่า ACCURACY = 58.97%



# K-NEAREST NEIGHBOR

```
[94] # Instantiate KNN model
knn1 = KNeighborsClassifier(n_neighbors=7, weights='uniform', algorithm='ball_tree')

# Fit the model to the training data
knn1.fit(X_train, y_train)

# Predict the test data
y_pred = knn1.predict(X_test)

#Evaluation
from sklearn.metrics import accuracy_score, classification_report
print(f'accuracy = {accuracy_score(y_test, y_pred)}')

accuracy = 0.6082051282051282
```

K = 7 มีค่า ACCURACY = 60.82%



# K-NEAREST NEIGHBOR

```
# Instantiate KNN model
knn1 = KNeighborsClassifier(n_neighbors=9, weights='uniform', algorithm='ball_tree')

# Fit the model to the training data
knn1.fit(X_train, y_train)

# Predict the test data
y_pred = knn1.predict(X_test)

#Evaluation
from sklearn.metrics import accuracy_score, classification_report
print(f'accuracy = {accuracy_score(y_test, y_pred)}')

accuracy = 0.6128205128205129
```

K = 9 มีค่า ACCURACY = 61.28%



# K-NEAREST NEIGHBOR

```
# Instantiate KNN model
knn1 = KNeighborsClassifier(n_neighbors=11, weights='uniform', algorithm='ball_tree')

# Fit the model to the training data
knn1.fit(X_train, y_train)

# Predict the test data
y_pred = knn1.predict(X_test)

#Evaluation
from sklearn.metrics import accuracy_score, classification_report
print(f'accuracy = {accuracy_score(y_test, y_pred)}')

accuracy = 0.6241025641025642
```

K = 11 มีค่า ACCURACY = 62.41%



# K-NEAREST NEIGHBOR

```
] # Instantiate KNN model
knn1 = KNeighborsClassifier(n_neighbors=13, weights='uniform', algorithm='ball_tree')

# Fit the model to the training data
knn1.fit(X_train, y_train)

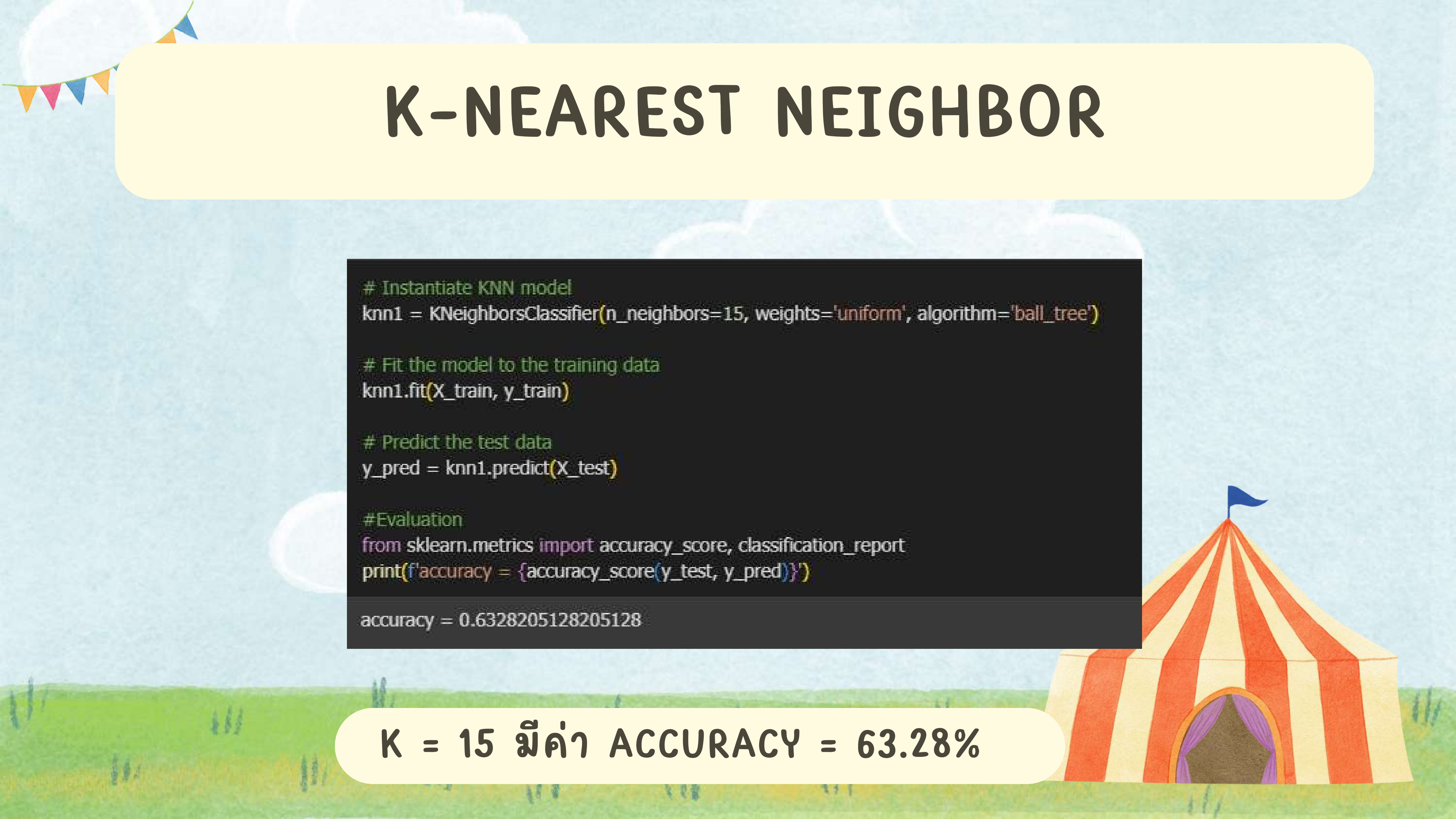
# Predict the test data
y_pred = knn1.predict(X_test)

#Evaluation
from sklearn.metrics import accuracy_score, classification_report
print(f'accuracy = {accuracy_score(y_test, y_pred)}')

accuracy = 0.6369230769230769
```

**K = 13    หมายความว่า    ACCURACY = 63.69%**





# K-NEAREST NEIGHBOR

```
# Instantiate KNN model
knn1 = KNeighborsClassifier(n_neighbors=15, weights='uniform', algorithm='ball_tree')

# Fit the model to the training data
knn1.fit(X_train, y_train)

# Predict the test data
y_pred = knn1.predict(X_test)

#Evaluation
from sklearn.metrics import accuracy_score, classification_report
print(f'accuracy = {accuracy_score(y_test, y_pred)}')

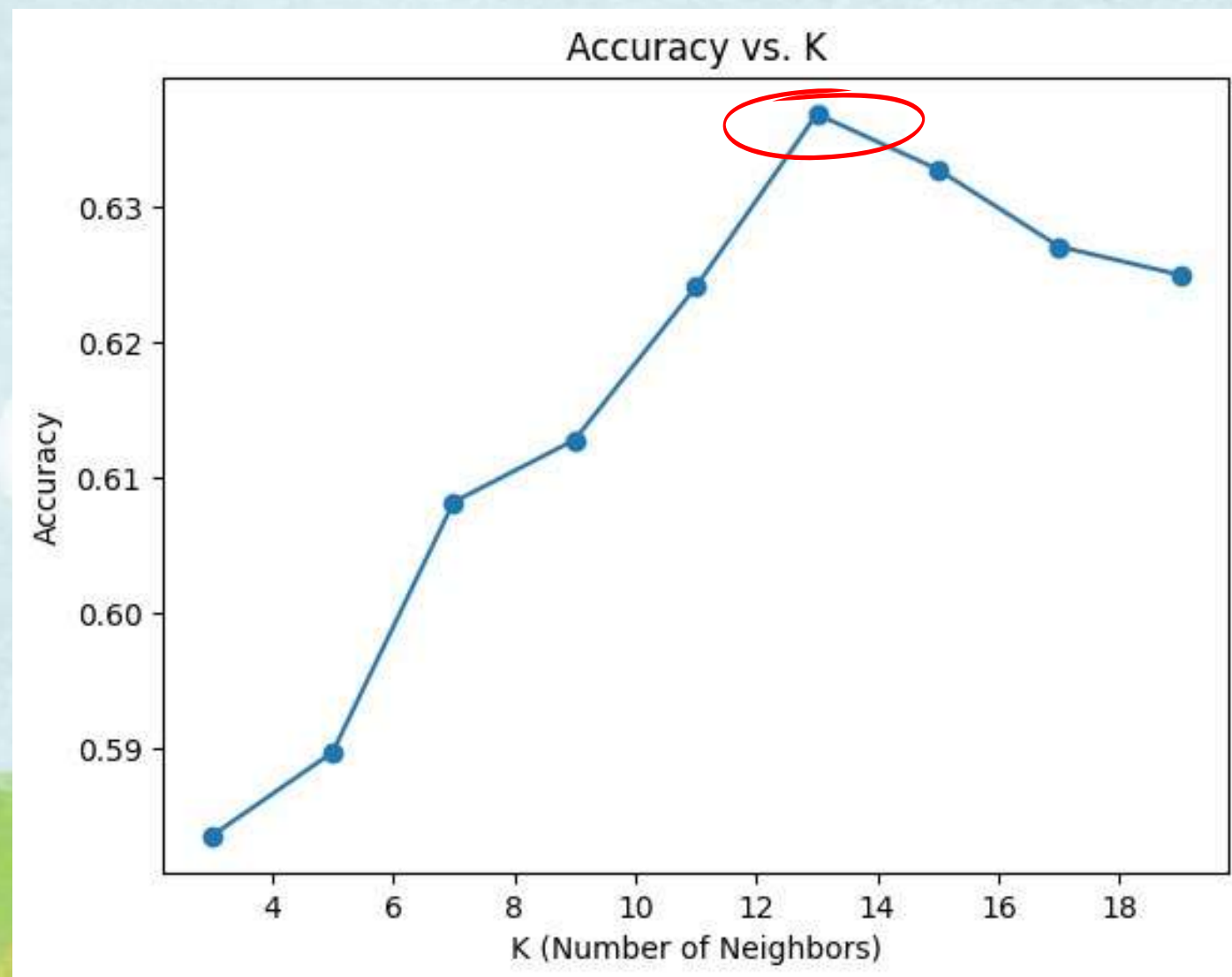
accuracy = 0.6328205128205128
```

**K = 15    หมายความว่า    ACCURACY = 63.28%**



# K-NEAREST NEIGHBOR

เมื่อดูค่า ACCURACY แต่ละ K  
จะพิจารณาค่า K ที่มีค่าอยู่ ณ จุด ELBOW



**K = 13**

**ACCURACY 63.69%**

**PRECISION 64.00%**

**RECALL 78.00%**

**F1-SCORE 71.00%**





# เลือก MODEL ที่ดีที่สุด

เมื่อพิจารณาค่า ACCURACY ของ K-NEAREST NEIGHBOR  
และ DECISION TREE แล้ว  
จะเลือก DECISION TREE  
เนื่องจากมีค่า ACCURACY สูงกว่า MODEL อื่น ๆ





The background is a light blue sky with three white, fluffy clouds. In the top left corner, a string of colorful triangular flags (yellow, pink, blue, orange) hangs diagonally. In the bottom right corner, a red and white striped circus tent with a blue flag on top sits on a green grassy field with small tufts of grass. A large, horizontal, rounded yellow rectangle is centered in the sky, containing the text "THANK YOU".

**THANK YOU**