

**Import Scikit-Learn: DecisionTreeClassifier, LabelEncoder****Import: Graphviz**

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_text, export_graphviz
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
import graphviz
```

**Pre-Process**

**Load Dataset** แล้วลอง **Display** เพื่อตรวจสอบ

```
df = pd.read_csv('fertility2.csv')
print(df.head())
```

	Age	kidney diseases	Fasting Blood sugar	Uri infection	exercise habit	Frequency of alcohol consumption
0	30	no	80	yes	more than 3 months ago	once a week
1	35	yes	140	yes	less than 3 hours a week	once a week
2	27	yes	99	no	no	hardly ever or never
3	44	no	96	yes	more than 3 months ago	hardly ever or never
4	30	yes	134	no	less than 3 hours a week	once a week

  

Smoking habit	profession	#hours spent sitting per day	Diagnosis
occasional	engineer	16	Normal
daily	engineer	6	weak
never	engineer	9	Normal
never	pilot	7	Normal
never	engineer	9	weak

(139, 10)

**Age** = อายุ

**kidney diseases** = ภาวะโรคไต

**Fasting Blood sugar** = ระดับน้ำตาลในเลือด

**Uri infection** = การติดเชื้อทางเดินปัสสาวะ

**exercise habit** = รวมถึงพฤติกรรมการใช้ชีวิต

**Frequency of alcohol consumption** = การดื่มสุรา

**Smoking habitprofession** = การสูบบุหรี่

**#hours spent sitting per day** = จำนวนชั่วโมงที่นั่งอยู่กับที่

**Profession** = อาชีพ

**Diagnosis** = ว่าปกติหรืออ่อนแอ

## Process

แปลง **Raw Data** จาก **String to Int** และจัดการ **feature** บางตัวให้เป็นค่าที่นับได้เพื่อใช้ในการเทรนและลอง **Display** เพื่อตรวจสอบ

```
label_encoder = LabelEncoder()
for col in ['Age', 'kidney diseases', 'Fasting Blood sugar', 'Uri infection', 'Frequency of
alcohol consumption', 'profession', '#hours spent sitting per day']:
    df[col] = label_encoder.fit_transform(df[col])

df['exercise habit'] = df['exercise habit'].map({
    'more than 3 months ago': 0,
    'less than 3 hours a week': 1,
    'no': 2
})

df = pd.get_dummies(df, columns=['Frequency of alcohol consumption'],
prefix='alcohol', drop_first=True)

df = pd.get_dummies(df, columns=['Smoking habit'], prefix='smoking',
drop_first=True)
print(df.head())
```

ให้ **X = feature**, **Y = class**

```
X = df.drop('Diagnosis', axis=1)
y = df['Diagnosis']
```

แบ่ง **Data** ออกเป็นส่วนสำหรับการ **train** และ **test** โดยที่ให้สัดส่วนของ **test = 0.2**

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

เทรนโมเดลและ **Prediction**

```
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

หาความแม่นยำและลอง **Display** ออกมา

```
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

	Age	kidney diseases	Fasting Blood sugar	Uri infection
0	5	0		2
1	10	1		25
2	2	1		17
3	17	0		15
4	5	1		24

Accuracy: 67.86%

```

|--- exercise habit <= 1.50
|   |--- Age <= 22.00
|       |--- exercise habit <= 0.50
|           |--- class: weak
|           |--- exercise habit > 0.50

```

Display tree ออกมา จะได้เป็นไฟล์ .png ดังรูป

```

dot_data = export_graphviz(clf, out_file=None, feature_names=list(X.columns),
class_names=['normal', 'weak'], filled=True, rounded=True)
graph = graphviz.Source(dot_data)
graph.render("fertility_tree2", format="png", cleanup=True)
graph.view("fertility_tree2")

```

