

Would customer purchase the SUV?

	User ID	Gender	Age	EstimatedSalary	Purchased
1	15624510	Male	19	19000	0
2	15810944	Male	35	20000	0
3	15668575	Female	26	43000	0
4	15603246	Female	27	57000	0
5	15804002	Male	19	76000	0
6	15728773	Male	27	58000	0
7	15598044	Female	27	84000	0
8	15694829	Female	32	150000	1
9	15600575	Male	25	33000	0
10	15727311	Female	35	65000	0
11	15570769	Female	26	80000	0

0: not purchase 1: purchase

In []:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

In []:

```
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values #select data in position 2 and 3 (Age, EstimatedSalary)
y = dataset.iloc[:, 4].values #select data in position 4 (Purchased)
```

In []:

```
#พิมพ์ค่า X
```

In []:

```
#พิมพ์ค่า y
```

Splitting the dataset into the Training set and Test set

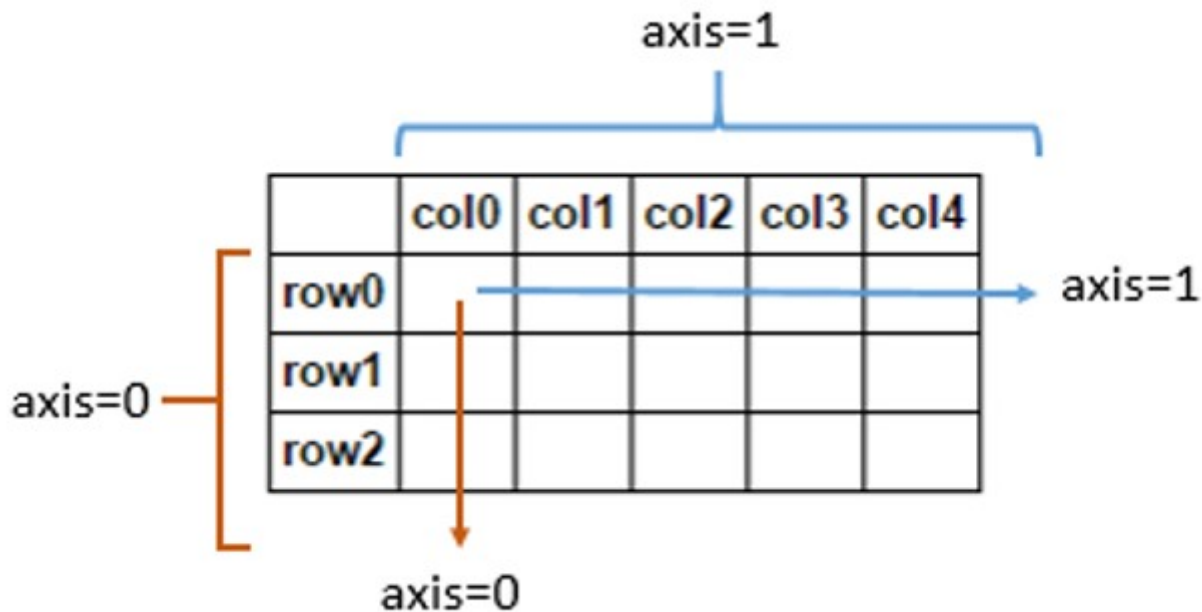
In []:

```
#แบ่งข้อมูลมาทำการ test 25% ของข้อมูลทั้งหมด
```

Feature Scaling

เนื่องจากข้อมูลของเราอยู่กันคนละ scale จึงต้องมีการปรับข้อมูลให้อยู่ใน scale เดียวกัน เพื่อปรับให้ความแปรปรวนข้อมูลเหล่านี้มีค่าเท่ากันและปรับให้ค่าเฉลี่ยมีค่าเท่าๆกันก่อนที่จะนำข้อมูลไปใช้

การทำ Feature Scaling อยู่ในขั้นตอนของการทำ preprocessing



In []:

```
#Standardize features by removing the mean and scaling to unit variance
#import MinMaxScaler() => data matrix lie between [0, 1] range
#(X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
#https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

#กำหนดค่า X_trainT ให้มีค่าเท่ากับ X_train

#กำหนดค่า X_testT ให้มีค่าเท่ากับ X_test

#Compute the minimum and maximum to be used for later scaling.
#ทำการคำนวณของทั้ง X_trainT และ X_testT

#Scale features of X according to feature_range
#ทำการ scale ค่าของทั้ง X_trainT และ X_testT

#สร้างตัวแปร X_testT_min สำหรับเก็บค่า scaler.data_min_ เพื่อนำไปใช้สำหรับทดสอบข้อมูลใหม่ที่ไม่ได้อยู่ใน Test
#สร้างตัวแปร X_testT_max สำหรับเก็บค่า scaler.data_max_ เพื่อนำไปใช้สำหรับทดสอบข้อมูลใหม่ที่ไม่ได้อยู่ใน Test
```

In []:

```
#พิมพ์ค่า X_testT_min และ X_testT_max
```

In []:

```
#แสดงค่าของ X_trainT ที่ได้หลังจากการทำ scaling
```

Choose Model: DecisionTreeClassifier

In []:

```
from sklearn.tree import DecisionTreeClassifier #เลือก model
model = DecisionTreeClassifier(criterion='entropy', random_state=0) #กำหนดค่าให้ model
model.fit(X_trainT, y_train) #สร้าง model
y_pred = model.predict(X_testT) #ทำนาย
```

Accuracy

In []:

```
#คำนวณหาค่า Accuracy ของการทำนายข้อมูล
```

Confusion Matrix

In []:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In []:

```
#ทำการ visualise confusion matrix โดยให้
#แกน x เป็น predicted label
#แกน y เป็น true label
```

newcustomer

age = 19, EstimatedSalary = 25000 ลูกค้าคนนี้จะซื้อ SUV หรือไม่

In []:

```
#สร้าง array ข้อมูลลูกค้า โดยใช้ numpy และตั้งชื่ออาร์เรย์ว่า customer
```

In []:



```
# reshape อาร์เรย์ customer ให้อยู่ในรูปแบบของ feature format
```

In []:



```
# scaling ข้อมูลลูกค้า โดยใช้ค่า X_testT_min และ X_testT_max ของข้อมูลทดสอบที่ได้มาตอนแรก  
#(X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
```

In []:



```
# predict การซื้อของลูกค้าจาก model ที่ได้สร้างไว้
```

```
#พิมพ์ผลการทำนาย ว่าได้ผลลัพธ์ เป็น 0 หรือ 1
```

In []:



```
#ถ้าผลการทำนายเป็น 1 ให้แสดงข้อความว่า Purchase  
#ถ้าผลการทำนายเป็น 0 ให้แสดงข้อความว่า Not purchase
```