

# Would customer purchase the SUV?

	User ID	Gender	Age	EstimatedSalary	Purchased
1					
2	15624510	Male	19	19000	0
3	15810944	Male	35	20000	0
4	15668575	Female	26	43000	0
5	15603246	Female	27	57000	0
6	15804002	Male	19	76000	0
7	15728773	Male	27	58000	0
8	15598044	Female	27	84000	0
9	15694829	Female	32	150000	1
10	15600575	Male	25	33000	0
11	15727311	Female	35	65000	0
12	15570769	Female	26	80000	0

0: not purchase 1: purchase

In [1]:

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

## Importing the dataset

In [2]:

```
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 [3]:

```
#พิมพ์ค่า X
X
```

Out[3]:

```
array([[ 19, 19000],
       [ 35, 20000],
       [ 26, 43000],
       [ 27, 57000],
       [ 19, 76000],
       [ 27, 58000],
       [ 27, 84000],
       [ 32, 150000],
       [ 25, 33000],
       [ 35, 65000],
       [ 26, 80000],
       [ 26, 52000],
       [ 20, 86000],
       [ 32, 18000],
       [ 18, 82000],
       [ 29, 80000],
       [ 47, 25000],
```

In [4]:



```
#พิมพ์ค่า y
y
```

Out[4]:

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1], dtype=int64)
```

## Splitting the dataset into the Training set and Test set

In [5]:

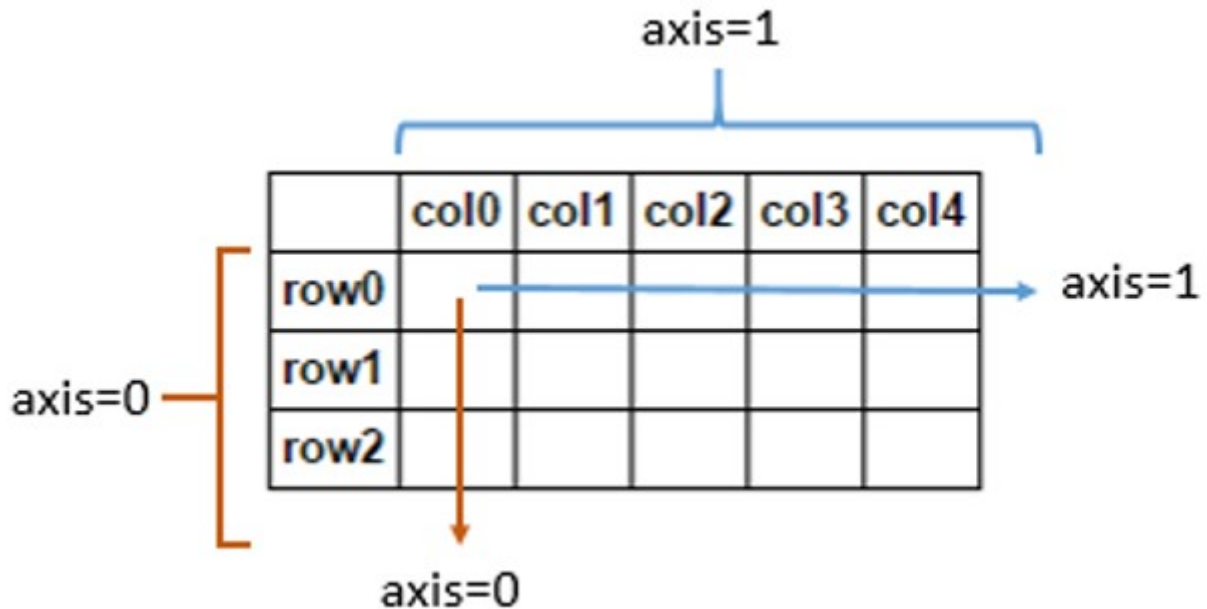


```
#แบ่งข้อมูลมาทำการ test 25% ของข้อมูลทั้งหมด
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state =
```

## Feature Scaling

เนื่องจากข้อมูลของเราอยู่กันคนละ scale จึงต้องมีการปรับข้อมูลให้อยู่ใน scale เดียวกัน เพื่อปรับให้ความแปรปรวนข้อมูลเหล่านี้มีค่าเท่ากันและปรับให้ค่าเฉลี่ยมีค่าเท่ากันก่อนที่จะนำข้อมูลไปใช้

การทำ Feature Scaling อยู่ในขั้นตอนของการทำ preprocessing



In [6]:

▶

```
#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
X_trainT = X_train

#กำหนดค่า X_testT ให้มีค่าเท่ากับ X_test
X_testT = X_test

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

#Scale features of X according to feature_range
#ทำการ scale ค่าของทั้ง X_trainT และ X_testT
X_trainT=scaler.transform(X_trainT)
X_testT = scaler.transform(X_testT)

#สร้างตัวแปร X_testT_min สำหรับเก็บค่า scaler.data_min_ เพื่อนำไปใช้สำหรับทดสอบข้อมูลใหม่ที่ไม่ได้อยู่ใน Tes
#สร้างตัวแปร X_testT_max สำหรับเก็บค่า scaler.data_max_ เพื่อนำไปใช้สำหรับทดสอบข้อมูลใหม่ที่ไม่ได้อยู่ใน Tes
X_testT_min = scaler.data_min_
X_testT_max = scaler.data_max_
```

In [7]:

```
#พิมพ์ค่า X_testT_min และ X_testT_max
print('X_testT_min = ',X_testT_min)
print('X_testT_max = ',X_testT_max)
```

```
X_testT_min = [ 18. 15000.]
X_testT_max = [6.0e+01 1.5e+05]
```

In [8]:

```
#แสดงค่าของ X_trainT ที่ได้หลังจากการทำ scaling
X_trainT
```

Out[8]:

```
array([[0.61904762, 0.17777778],
       [0.33333333, 0.77777778],
       [0.47619048, 0.25925926],
       [0.33333333, 0.88888889],
       [0.80952381, 0.04444444],
       [0.83333333, 0.65925926],
       [0.5       , 0.2       ],
       [0.47619048, 0.34074074],
       [0.42857143, 0.25925926],
       [0.42857143, 0.35555556],
       [0.4047619 , 0.07407407],
       [0.4047619 , 0.25925926],
       [0.57142857, 0.42962963],
       [0.69047619, 0.25185185],
       [0.97619048, 0.1037037 ],
       [0.73809524, 0.37037037],
       [0.64285714, 0.85925926],
```

## Choose Model: DecisionTreeClassifier

In [9]:

```
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 [10]:

```
#คำนวณหาค่า Accuracy ของการทำนายข้อมูล
from sklearn.metrics import accuracy_score #Accuracy classification score
accuracy_score(y_test, y_pred) #comparing real target (y_test) with predicted target (y_pre
```

Out[10]:

0.91

# Confusion Matrix

In [11]:

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

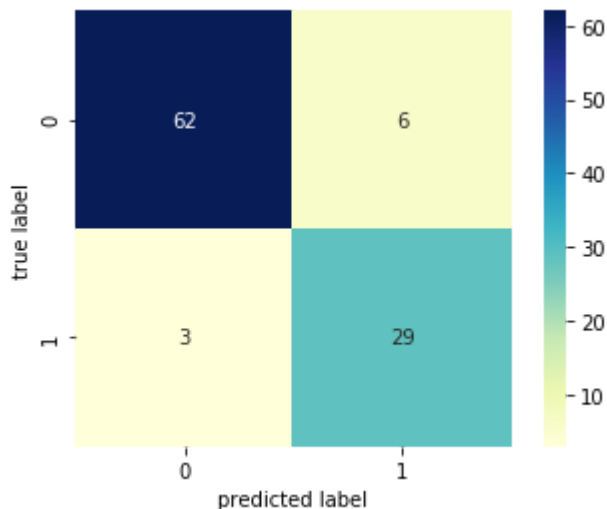
In [12]:

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

label = [0, 1]

import seaborn as sns
sns.heatmap(cm, square=True, annot=True, fmt='d', xticklabels=label, yticklabels=label, cmap="YlGnBu")

plt.xlabel('predicted label')
plt.ylabel('true label');
```



## newcustomer

age = 19, EstimatedSalary = 25000 ลูกค้าคนนี้จะซื้อ SUV หรือไม่

In [13]:

```
#สร้าง array ข้อมูลลูกค้า โดยใช้ numpy และตั้งชื่ออาร์เรย์ว่า customer

import numpy as np
#customer = np.array([32, 150000])
customer = np.array([19, 25000])

customer
```

Out[13]:

```
array([ 19, 25000])
```

In [14]:



```
# reshape อาร์เรย์ customer ให้อยู่ในรูปแบบของ feature format
customer = customer.reshape(1,-1)

customer.shape
```

Out[14]:

(1, 2)

In [15]:



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

customerN = (customer - X_testT_min) / (X_testT_max - X_testT_min)

customerN
```

Out[15]:

array([[0.02380952, 0.07407407]])

In [16]:



```
# predict การซื้อของลูกค้าจาก model ที่ได้สร้างไว้
cus_pred = model.predict(customerN)

#พิมพ์ผลการทำนาย ว่าได้ผลลัพธ์ เป็น 0 หรือ 1
print(cus_pred)
```

[0]

In [17]:



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

if cus_pred == 1:
    print('Purchase')
else:
    print('Not purchase')
```

Not purchase