



SUPERSTORE MARKETING CAMPAIGN DATASET

ชุดข้อมูลแคมเปญการตลาดของ Superstore
ตัวอย่างข้อมูลลูกค้าสำหรับการวิเคราะห์ข้อเสนอสำหรับสมาชิกเป้าหมาย

Presented By

•••

Group 6

•••



CONTEXT

ชูปเปอร์สโตร์ได้วางแผนการขาย โดยการลดราคา สินค้าส่งท้ายปี ซึ่งเคมเปญจะเป็นการเปิดตัวข้อเสนอใหม่ โดยมีการวางแผนจัดทำเคมเปญผ่านໂຕຣສັພກเพื่อ ลดต้นทุนของเคมเปญ โดยให้ส่วนลด 20% สำหรับลูกค้าที่เป็นสมาชิกระดับโกลด์ เมื่อซื้อครบ 999\$ แต่จะลดไม่เกิน 499\$



OBJECTIVE

1. ต้องการนำความเป็นไปได้ที่ลูกค้าให้การตอบรับเชิงบวก เพื่อหาโมเดลที่ดีที่สุดที่จะใช้ในการนำยชุดข้อมูลนี้ ซึ่งจะดูแค่ลูกค้าระดับโกลด์เท่านั้น
2. เพื่อหากฎการซื้อขายที่เกิดขึ้นบ่อยที่สุดของชุดข้อมูล superstore_data
3. ทำการแบ่งกลุ่ม เพื่อหากลุ่มที่มีความเหมือนกันของชุดข้อมูล superstore_data





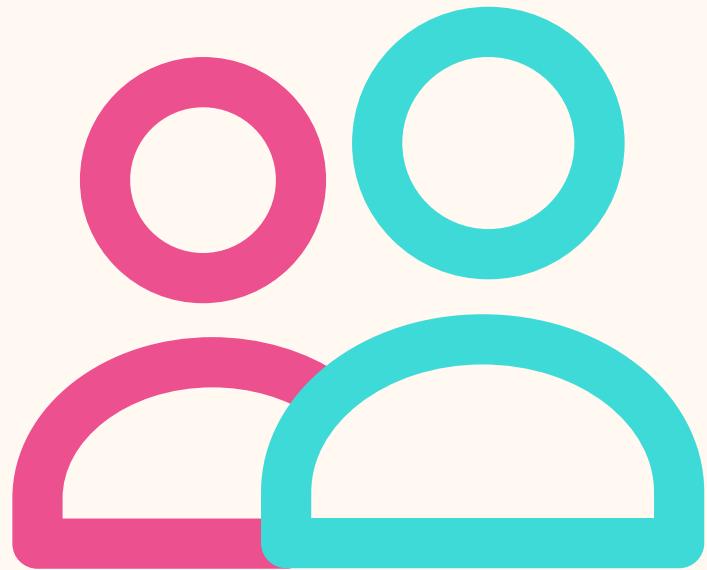
File: superstore_data.csv

BUY NOW

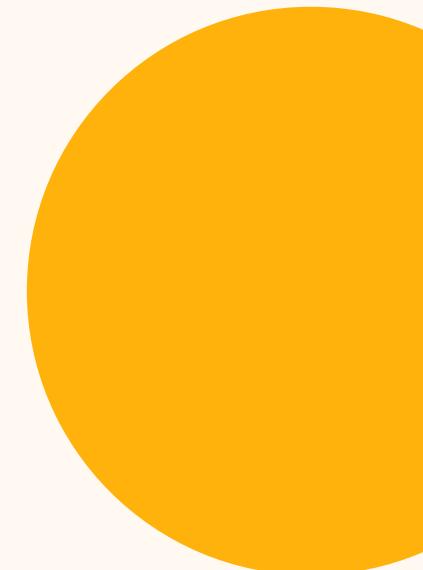
ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatPr	MntFishPr	I
1826	1970	Graduation	Divorced	84835	0	0	6/16/2014	0	189	104	379	111	
1	1961	Graduation	Single	57091	0	0	6/15/2014	0	464	5	64	7	
10476	1958	Graduation	Married	67267	0	1	5/13/2014	0	134	11	59	15	
1386	1967	Graduation	Together	32474	1	1	11/5/2014	0	10	0	1	0	
5371	1989	Graduation	Single	21474	1	0	8/4/2014	0	6	16	24	11	
7348	1958	PhD	Single	71691	0	0	3/17/2014	0	336	130	411	240	
4073	1954	2n Cycle	Married	63564	0	0	1/29/2014	0	769	80	252	15	
1991	1967	Graduation	Together	44931	0	1	1/18/2014	0	78	0	11	0	
4047	1954	PhD	Married	65324	0	1	11/1/2014	0	384	0	102	21	
9477	1954	PhD	Married	65324	0	1	11/1/2014	0	384	0	102	21	
2079	1947	2n Cycle	Married	81044	0	0	12/27/2013	0	490	26	535	73	
5642	1979	Master	Together	62499	1	0	9/12/2013	0	140	4	61	0	
10530	1959	PhD	Widow	67786	0	0	7/12/2013	0	431	82	441	80	
2964	1981	Graduation	Married	26872	0	0	10/16/2013	0	3	10	8	3	
10311	1969	Graduation	Married	4428	0	1	5/10/2013	0	16	4	12	2	
637	1977	Graduation	Married	54809	1	1	11/9/2013	0	63	6	57	13	
10521	1977	Graduation	Married	54809	1	1	11/9/2013	0	63	6	57	13	
10175	1958	PhD	Divorced	32173	0	1	1/8/2013	0	18	0	2	0	
1473	1960	2n Cycle	Single	47823	0	1	7/23/2013	0	53	1	5	2	
2795	1958	Master	Single	30523	2	1	1/7/2013	0	5	0	3	0	
2285	1954	Master	Together	36634	0	1	5/28/2013	0	213	9	76	4	
115	1966	Master	Single	43456	0	1	3/26/2013	0	275	11	68	25	
10470	1979	Master	Married	40662	1	0	3/15/2013	0	40	2	23	0	



DATA DESCRIPTION



Response (target) - 1 หากลูกค้ายอมรับข้อเสนอในแคมเปญล่าสุด 0 อย่างอื่น
ID - รหัส ID ที่ไม่ซ้ำกันของลูกค้าแต่ละคน
Year_Birth - อายุของลูกค้า
Complain - 1 หากลูกค้าร้องเรียนในช่วง 2 ปีที่ผ่านมา 0 อย่างอื่น
Dt_Customer - วันที่ลงทะเบียนของลูกค้ากับบริษัท
Education - ระดับการศึกษาของลูกค้า
Marital - สภาพการสมรสของลูกค้า
Kidhome - จำนวนเด็กแต่ละคนในครัวเรือนของลูกค้า
Teenhome - จำนวนวัยรุ่นแต่ละคนในครัวเรือนของลูกค้า
Income - รายได้รายปีของลูกค้า
MntFishProducts - จำนวนเงินที่ใช้สำหรับซื้อสินค้าปลาในช่วง 2 ปีที่ผ่านมา
MntMeatProducts - จำนวนเงินที่ใช้สำหรับซื้อเนื้อสัตว์ในช่วง 2 ปีที่ผ่านมา



DATA DESCRIPTION



MntFruits - จำนวนเงินที่ใช้สำหรับซื้อผลไม้ในช่วง 2 ปีที่ผ่านมา

MntSweetProducts - จำนวนเงินที่ใช้สำหรับซื้อบนมหวานในช่วง 2 ปีที่ผ่านมา

MntWines - จำนวนเงินที่ใช้สำหรับซื้อไวน์ในช่วง 2 ปีที่ผ่านมา

MntGoldProds - จำนวนเงินที่ใช้สำหรับซื้อผลิตภัณฑ์ทองคำในช่วง 2 ปีที่ผ่านมา

NumDealsPurchases - จำนวนการซื้อสินค้าที่มีส่วนลด

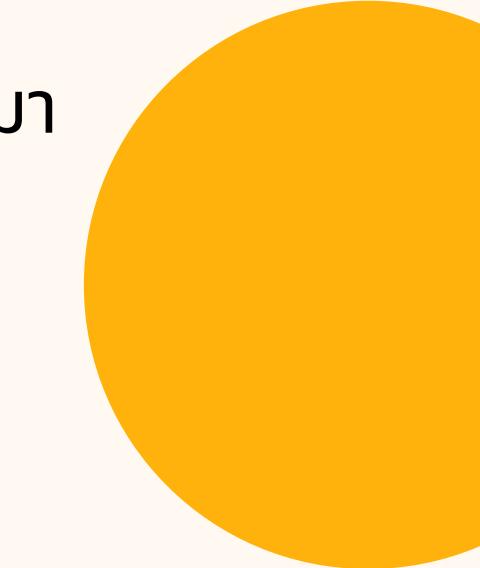
NumCatalogPurchases - จำนวนการซื้อสินค้าผ่านแคตตาล็อก
(การซื้อสินค้าที่จะส่งผ่านไปรษณีย์)

NumStorePurchases - จำนวนการซื้อสินค้าโดยตรงที่ร้านค้า

NumWebPurchases - จำนวนการซื้อสินค้าผ่านเว็บไซต์ของบริษัท

NumWebVisitsMonth - จำนวนการเข้าชมเว็บไซต์ของบริษัทในเดือนล่าสุด

Recency - จำนวนวันตั้งแต่การซื้อสินค้าครั้งล่าสุด





MODEL AND CLUSTERING

Decision tree

Naive Bayes

K-nearest
Neighbor

Association
Rules

K-Means
Clustering



DATA PREPARATION

IMPORT FILE

```
superstore_data = pd.read_csv('superstore_data.csv')
print(superstore_data.shape) # shown number of (row, column)
superstore_data.head()
```

(2240, 22)

	Id	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts
0	1826	1970	Graduation	Divorced	84835.0	0	0	6/16/2014	0	189	...	111	111
1	1	1961	Graduation	Single	57091.0	0	0	6/15/2014	0	464	...	7	7
2	10476	1958	Graduation	Married	67267.0	0	1	5/13/2014	0	134	...	15	15
3	1386	1967	Graduation	Together	32474.0	1	1	11/5/2014	0	10	...	0	0
4	5371	1989	Graduation	Single	21474.0	1	0	8/4/2014	0	6	...	11	11

5 rows × 22 columns



DATA PREPARATION

MISSING/NULL CHECKING

แทนค่าว่างด้วยค่าเฉลี่ยของคอลัมน์ Income

	Id	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Response
134	8996	1957	PhD	Married	NaN	2	1	False
262	1994	1983	Graduation	Married	NaN	1	0	False
394	3769	1972	PhD	Together	NaN	1	0	False
449	5255	1986	Graduation	Single	NaN	1	0	False
525	8268	1961	PhD	Married	NaN	0	1	False
590	10629	1973	2n Cycle	Married	NaN	1	0	False

Id	False
Year_Birth	False
Education	False
Marital_Status	False
Income	True
Kidhome	False
Teenhome	False
Dt_Customer	False
Recency	False
MinWines	False
MinFruits	False
MinMeatProducts	False
MinFishProducts	False
MinSweetProducts	False
MinGoldProducts	False
NumDealsPurchases	False
NumWebPurchases	False
NumCatalogPurchases	False
NumStorePurchases	False
NumWebVisitsMonth	False
Response	False
Complain	False
dtype:	bool



... CLASSIFICATION ...

1

DICISION TREE

2

K-NEAREST NEIGHBORS (KNN)

3

NAIVE BAYES



CLASSIFICATION

1. สร้างตัวแปร X และ y เพื่อเก็บ **features** และ **target variable**

X = Id, Year_Birth, Education, Marital_Status, Income, Kidhome,
Teenhome, Dt_Customer, Recency, MntWines, MntFruits,
MntMeatProducts, MntFishProducts, MntSweetProducts,
MntGoldProds, NumDealsPurchases, NumWebPurchases,
NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, Complain

y = Response



PREPARE INFORMATION FOR CLASSIFICATION

```
from sklearn.tree import DecisionTreeClassifier

feature_cols = ['Id', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
                'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
                'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
                'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
                'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth', 'Complain']

X = superstore_data_clean[feature_cols] # Features
y = superstore_data_clean['Response'] # Target variable

from sklearn.preprocessing import LabelEncoder

# encoding categorical variables
label_encoder = LabelEncoder()
X['Education'] = label_encoder.fit_transform(X['Education'])
X['Marital_Status'] = label_encoder.fit_transform(X['Marital_Status'])

# applying timestamp to the date column
X['Dt_Customer'] = pd.to_datetime(X['Dt_Customer']).apply(lambda x: pd.Timestamp(x).timestamp())
```

CLASSIFICATION

2. แบ่งข้อมูลเป็น training set และ testing set ด้วยสัดส่วน 80:20
3. สร้างตัวแบบของแต่ละโมเดล
4. ฝึกโมเดล (training) ด้วยชุดข้อมูลสำหรับการฝึก
5. ใช้โมเดลในการคำนวณ
6. ประเมินประสิทธิภาพของโมเดล โดยคำนวณค่า accuracy, precision, recall และ F1-score





DECISION TREE

10-fold cross validation and find the best parameters

Best hyperparameters: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 3}
Mean cross-validated accuracy score: 85.33%

สรุปผล

ผลลัพธ์ที่ได้จะเห็นว่าโมเดล Decision Tree ที่สร้างขึ้นมีความแม่นยำในการทำนายผลของชุดข้อมูลทดสอบอยู่ที่ 78.79% ซึ่งมีค่าต่ำกว่าค่าความแม่นยำที่ได้จากการทำ cross-validation และพารามิเตอร์ที่ดีที่สุดที่ค้นหาโดย GridSearchCV โดยค่าความแม่นยำที่ดีที่สุดที่ได้จาก cross-validation คือ 85.33% และเมื่อนำมาทดสอบที่สร้างขึ้นจากพารามิเตอร์ดังกล่าวมาใช้กับชุดข้อมูลสำหรับการทดสอบได้ค่าความแม่นยำอยู่ที่ 81.70% ซึ่งมีค่าต่ำกว่าค่าความแม่นยำที่ได้จากการ cross-validation



K-NEAREST NEIGHBORS (KNN)

10-fold cross validation and find the best parameters

Best hyperparameters: {'metric': 'manhattan', 'n_neighbors': 5, 'weights': 'distance'}

Mean cross-validated accuracy score: 87.22%

สรุปผล

จากการหาค่า k โดยการใช้ GridSearchCV พบว่าค่า k ที่เหมาะสมที่สุดสำหรับโมเดล KNN Classifier คือ 10 และความแม่นยำของโมเดล KNN Classifier บนชุดข้อมูลทดสอบที่สร้างขึ้นมีความแม่นยำในการทำนายผลของชุดข้อมูลทดสอบอยู่ที่ 83.71% ซึ่งมีค่าน้อยกว่าค่าความแม่นยำที่ได้จากการทำ cross-validation โดยค่าความแม่นยำที่ดีที่สุดที่ได้จาก cross-validation คือ 85.94% และเมื่อนำโมเดลที่สร้างขึ้นจากพารามิเตอร์ดังกล่าวมาใช้กับชุดข้อมูลสำหรับการทดสอบ ได้ค่าความแม่นยำอยู่ที่ 85.27% ซึ่งมีค่าใกล้เคียงกับค่าความแม่นยำที่ได้จากการ cross-validation



NAIVE BAYES

10-fold cross validation and find the best parameters

Best parameters: {}

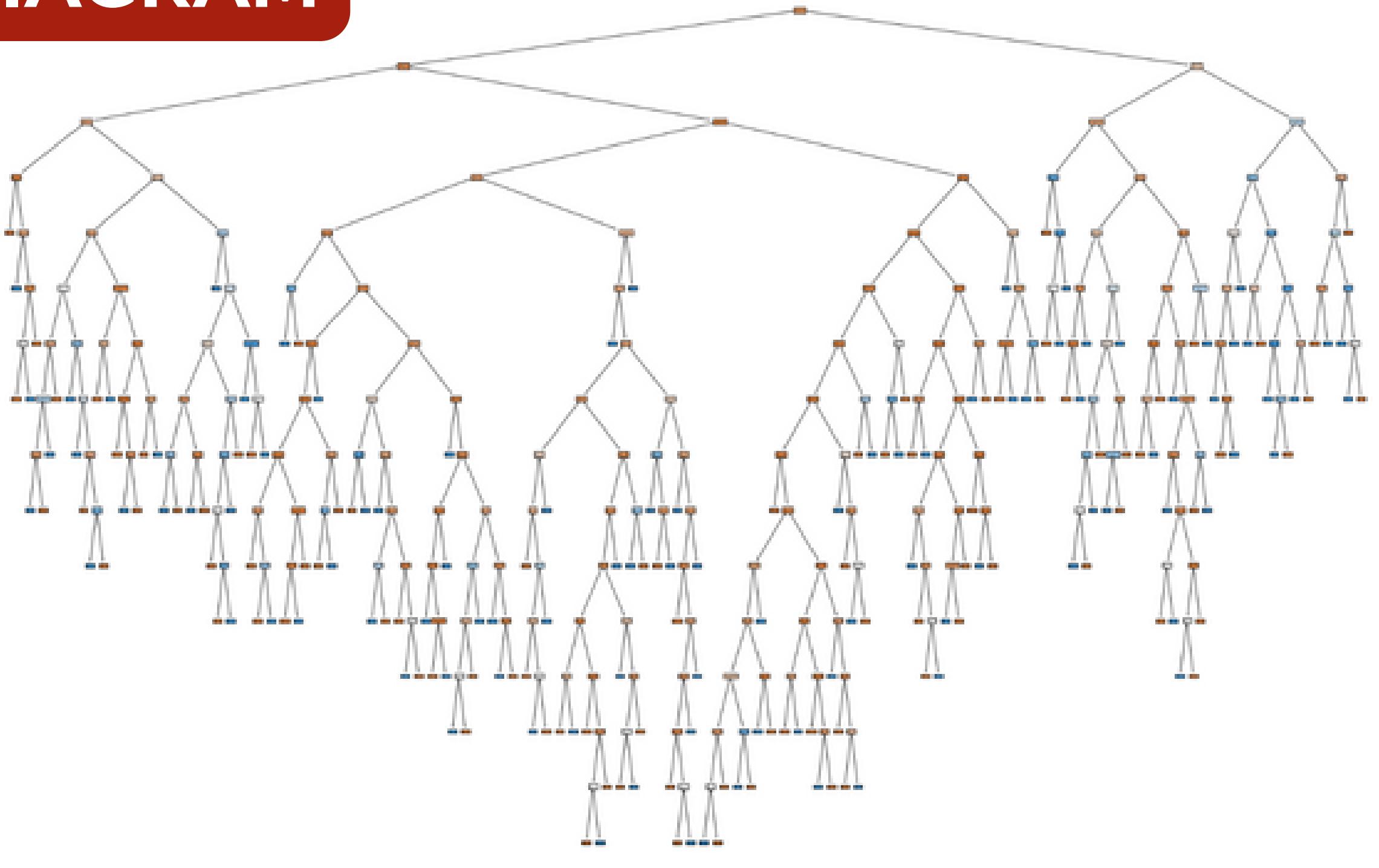
Mean cross-validated accuracy score: 85.13%

สรุปผล

โมเดล Naive Bayes ที่สร้างขึ้นมา มีความแม่นยำในการทำนายเทียบกับข้อมูลทดสอบอยู่ที่ 84.60% และหลังจากการประเมินโมเดลด้วยวิธี cross-validation 10-fold พบร่วมค่าความแม่นยำเฉลี่ยของโมเดลคือ 85.04% และเมื่อนำโมเดลที่สร้างขึ้นจากพารามิเตอร์ดังกล่าวมาใช้กับชุดข้อมูลสำหรับการทดสอบ ได้ค่าความแม่นยำอยู่ที่ 84.60%



TREE DIAGRAM





เปรียบเทียบ MODEL

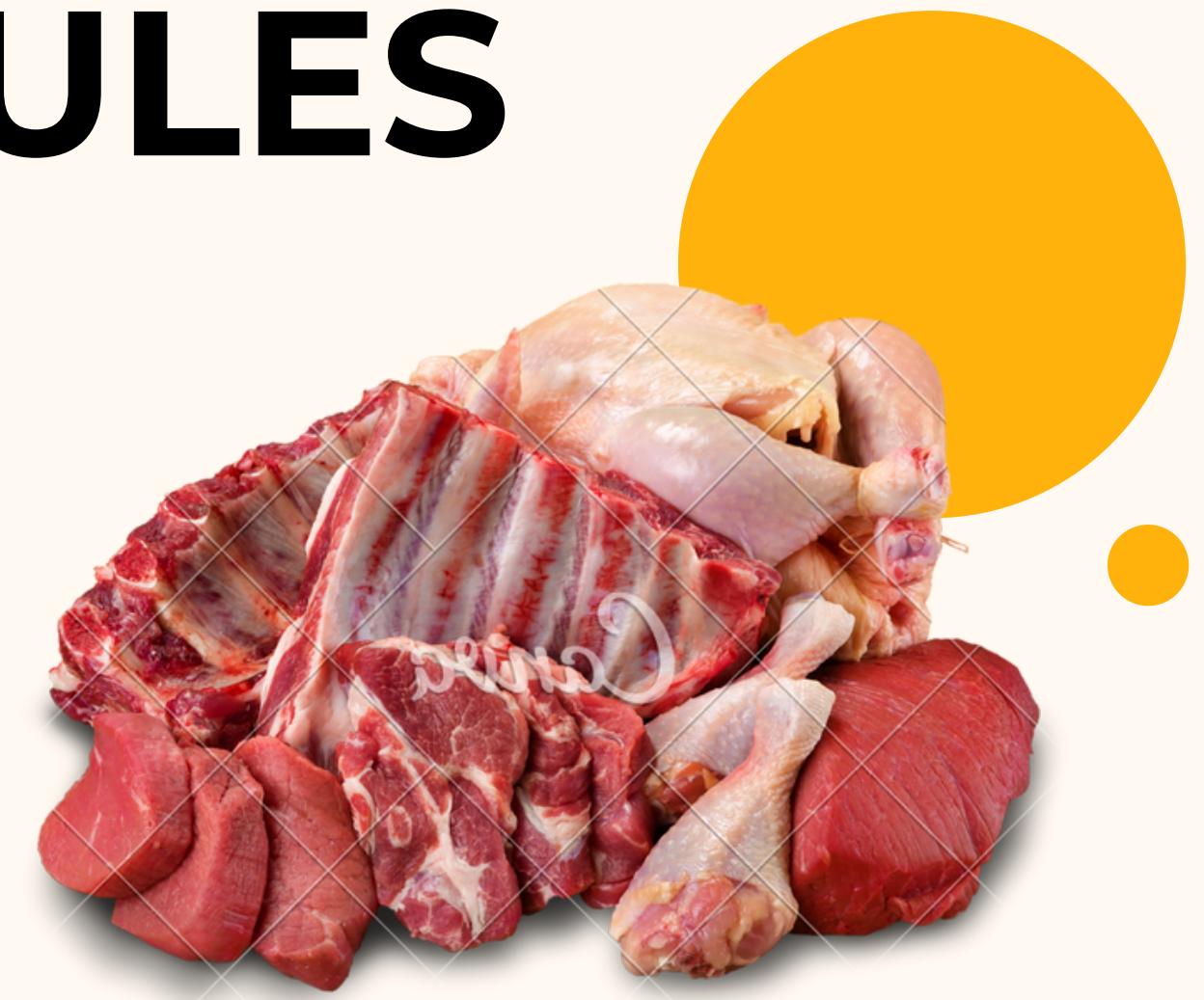
Model	Accuracy	Precision	Recall	F1-score
DICISION TREE	81.70%	class 0 = 87.91% class 1 = 33.33%	class 0 = 85.11% class 1 = 38.89%	class 0 = 86.49% class 1 = 35.90%
K-NEAREST NEIGHBORS (KNN)	85.27%	class 0 = 84.04% class 1 = 33.33%	class 0 = 99.47% class 1 = 1.39%	class 0 = 91.11% class 1 = 2.67%
NAIVE BAYES	84.60%	class 0 = 84.65% class 1 = 80.00%	class 0 = 99.73% class 1 = 5.56%	class 0 = 91.58% class 1 = 10.39%

แสดงผลลัพธ์จริงและผลลัพธ์ที่คำนายได้ ของโมเดล **NAIVE BAYES**



Response	predicted
324	1
96	0
2104	0
1259	0
1061	0
...	...
423	1
1340	1
753	0
2138	0
618	0
448 rows × 2 columns	

ASSOCIATION RULES





ASSOCIATION RULES

```
[1]: import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import numpy as np

subset = superstore_data_clean[['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProducts']]

# ตรวจสอบค่าที่ไม่ถูกต้องของตัวแปร subset ด้วย np.isnan(), np.inf, -np.inf
print(subset.isin([np.nan, np.inf, -np.inf]).sum())

# หันค่าที่ np.isfinite() ไม่สามารถใช้งานกับ subset ที่มีตัวเลข float หรือ decimal ได้ จึงต้องใช้ lambda แทน
print(subset.applymap(lambda x: isinstance(x, (int, float))).all())

# แปลงค่าใน subset ที่เป็น True หรือ False ตามที่ต้องการ ด้วย subset.replace(0, True) ดูตัวอย่างด้านล่าง 0 ให้เป็น True และ 1 ให้เป็น False
subset = subset.replace(0, True)
```

▪ ตรวจสอบค่าที่ไม่ถูกต้องของตัวแปร subset ด้วย np.isnan(), np.inf, -np.inf

หันค่าที่ np.isfinite() ไม่สามารถใช้งานกับ subset ที่มีตัวเลข float หรือ decimal ได้ จึงต้องใช้ lambda แทน

แปลงค่าใน subset ที่เป็น True หรือ False ตามที่ต้องการ ด้วย subset.replace(0, True) ดูตัวอย่างด้านล่าง 0 ให้เป็น True และ 1 ให้เป็น False

```
subset = subset.replace(0, True)
```

▪ กรณี NaN นั้น Inf/-Inf จะถูกแทนที่ด้วยความท้าทายที่น่าปวดหัว

```
# subset.fillna(subset.mean(), inplace=True)
# subset.replace([np.inf, -np.inf], np.nan, inplace=True)
# subset.fillna(subset.max(), inplace=True)
```

▪ กรณีที่ต้องการตัดสินใจว่าต้องการ min_support เท่าใด

ต้องคำนึงถึง min_support ที่ต้องการของ itemsets ด้วย rule of thumb ดังนี้

- กรณี itemsets ที่มีจำนวนตัวSKU 2-3 ตัว ให้ min_support ที่เป็น 0.05 แล้ว
- กรณี itemsets ที่มีจำนวน 4 ตัวขึ้นไป ให้ min_support ที่เป็น 0.01 แล้ว

```
frequent_itemsets = apriori(subset, min_support=0.05, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules
```



ASSOCIATION RULES

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(MntFruits)	(MntWines)	0.821429	0.994196	0.817857	0.995652	1.001464	0.001196	1.334821
1	(MntWines)	(MntFruits)	0.994196	0.821429	0.817857	0.822631	1.001464	0.001196	1.006781
2	(MntMeatProducts)	(MntWines)	0.999554	0.994196	0.994196	0.994640	1.000447	0.000444	1.082850
3	(MntWines)	(MntMeatProducts)	0.994196	0.999554	0.994196	1.000000	1.000447	0.000444	inf
4	(MntFruits)	(MntMeatProducts)	0.821429	0.999554	0.821429	1.000000	1.000447	0.000367	inf
...
543	(MntFishProducts)	(MntMeatProducts, MntGoldProds, MntSweetProdu...	0.828571	0.710268	0.646875	0.780711	1.099179	0.058367	1.321236
544	(MntGoldProds)	(MntMeatProducts, MntFishProducts, MntSweetPro...	0.972768	0.656696	0.646875	0.664984	1.012620	0.008062	1.024738
545	(MntSweetProducts)	(MntMeatProducts, MntFishProducts, MntGoldProd...	0.812946	0.719643	0.646875	0.795717	1.105710	0.061844	1.372393
546	(MntFruits)	(MntMeatProducts, MntFishProducts, MntGoldProd...	0.821429	0.714286	0.646875	0.787500	1.102500	0.060140	1.344538
547	(MntWines)	(MntMeatProducts, MntFishProducts, MntGoldProd...	0.994196	0.650000	0.646875	0.650651	1.001002	0.000647	1.001864

548 rows × 9 columns



ASSOCIATION RULES

LIFT

	antecedents	consequents	lift
421	(MntFishProducts, MntFruits)	(MntWines, MntSweetProducts, MntGoldProd)	1.109586
535	(MntFishProducts, MntFruits)	(MntMeetProducts, MntWines, MntSweetProducts, ...)	1.109586
498	(MntMeetProducts, MntWines, MntSweetProducts, ...)	(MntFishProducts, MntFruits)	1.109586
511	(MntWines, MntSweetProducts, MntGoldProd)	(MntFishProducts, MntFruits, MntMeetProducts)	1.109586
408	(MntWines, MntSweetProducts, MntGoldProd)	(MntFishProducts, MntFruits)	1.109586

จากผลลัพธ์ที่ได้ค่า lift ที่สูงที่สุด คือ 1.109586 ซึ่งมีค่าสูงกว่า 1
โดยถ้าค่า lift มา กกว่า 1 จะแสดงว่ามีความสัมพันธ์เชิงบวก (positive correlation)



ASSOCIATION RULES

CONFIDENCE

	antecedents	consequents	confidence
431	(MntFishProducts, MntWines, MntSweetProducts, ...)	(MntMeatProducts)	1.0
46	(MntWines, MntGoldProds)	(MntMeatProducts)	1.0
62	(MntFishProducts, MntFruits)	(MntMeatProducts)	1.0
130	(MntWines, MntFruits, MntSweetProducts)	(MntMeatProducts)	1.0
75	(MntFruits, MntGoldProds)	(MntMeatProducts)	1.0

ค่า confidence ที่สูงที่สุด คือ 1.0 แสดงว่า ความน่าจะเป็นที่จะมีการซื้อสินค้า antecedents และ consequents พร้อมกัน 100%



ASSOCIATION RULES

SUPPORT

	antecedents	consequents	support
2	(MintWines)	(MintMeatProducts)	0.994196
3	(MintMeatProducts)	(MintWines)	0.994196
47	(MintMeatProducts, MintGoldProd)	(MintWines)	0.960964
48	(MintWines)	(MintMeatProducts, MintGoldProd)	0.960964
49	(MintWines, MintGoldProd)	(MintMeatProducts)	0.960964

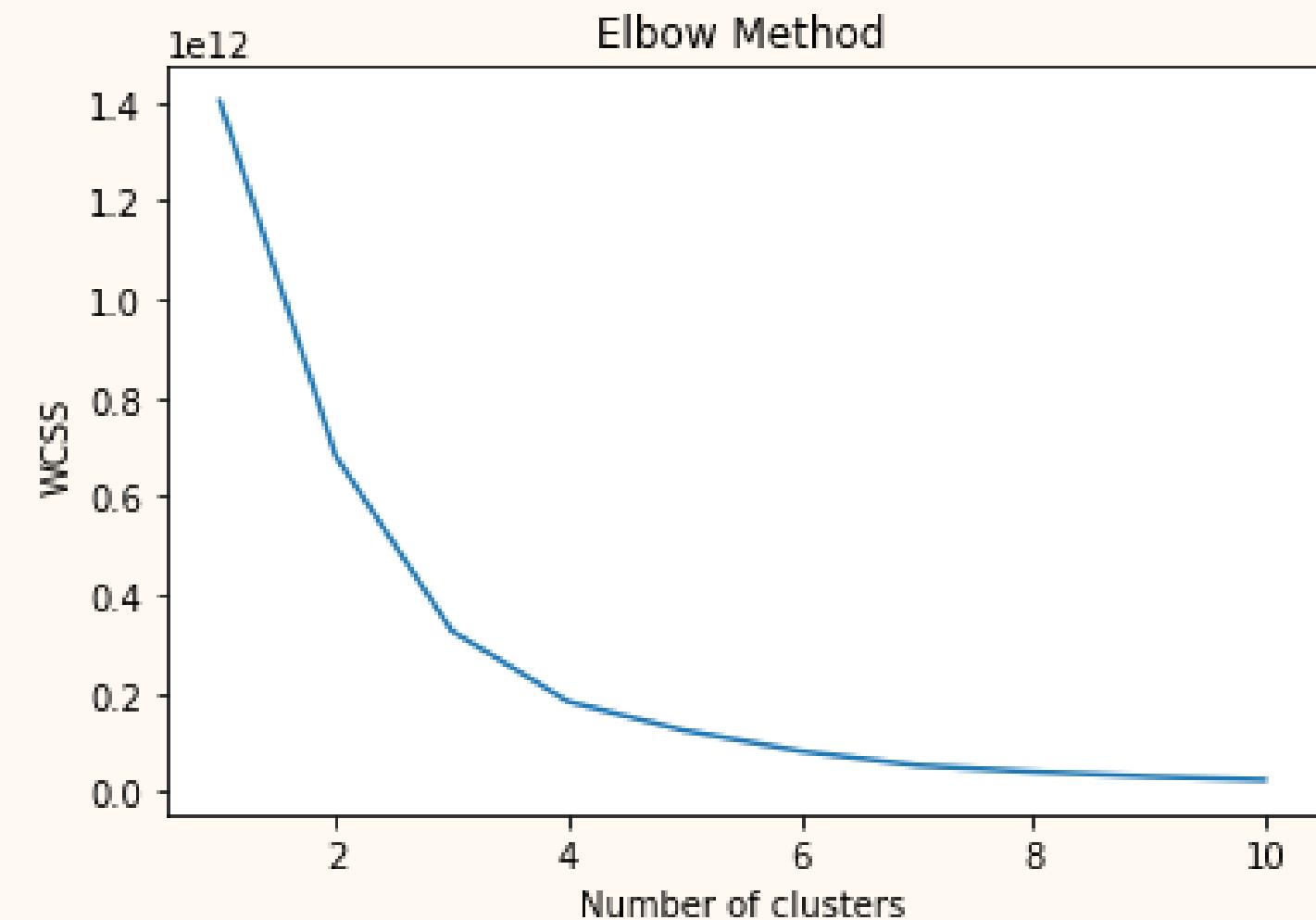
จากผลลัพธ์ที่ได้ค่า support ที่สูงที่สุด คือ มีความถี่ในการเกิดขึ้นสูง
แสดงว่า มีโอกาสเกิดกันบ่อย หรือมีความสัมพันธ์กันอย่างชัดเจน



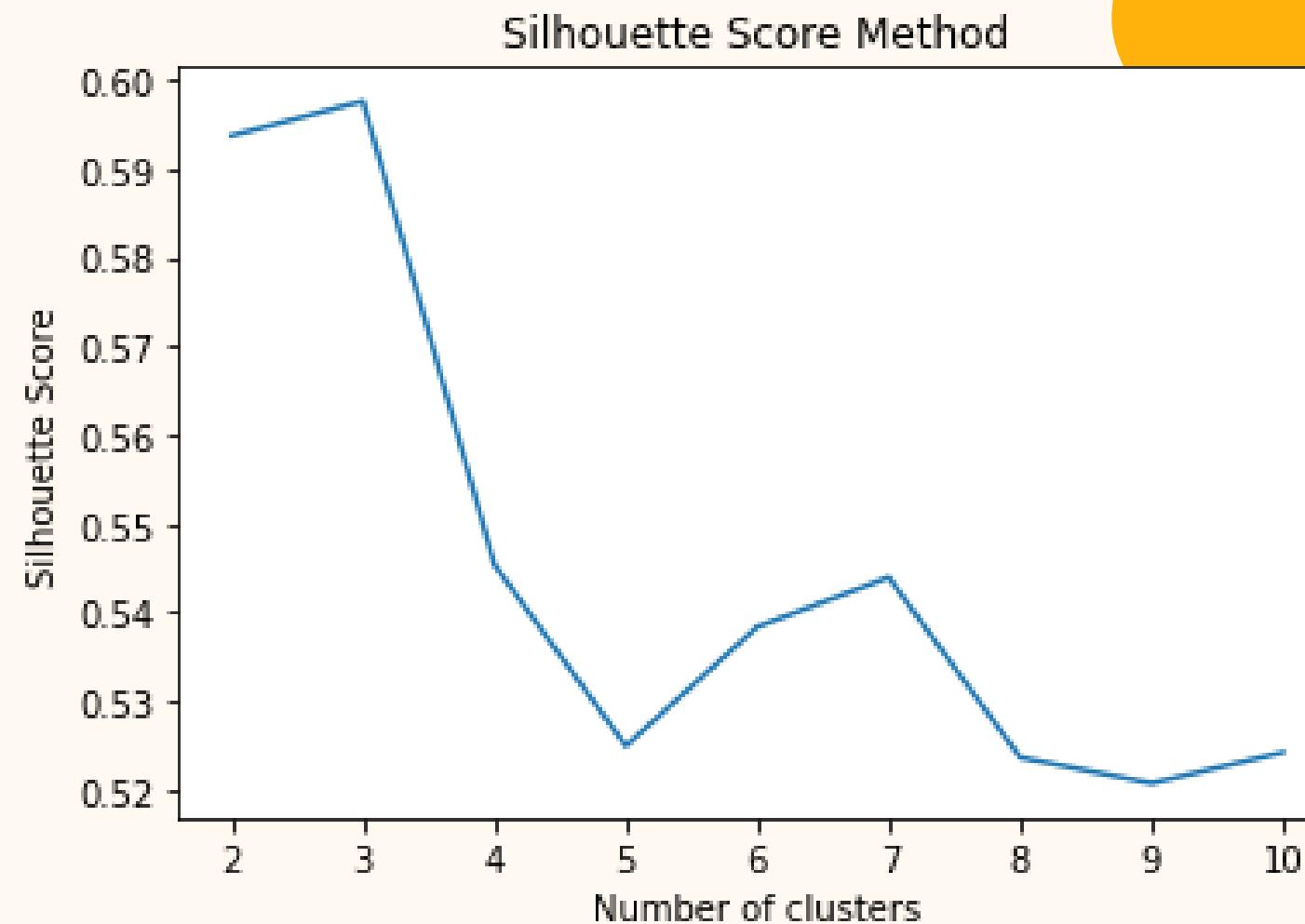
K-MEANS CLUSTERING



การหาจำนวน CLUSTER ที่เหมาะสมกับการแบ่งข้อมูล



Elbow Method เพื่อหาจำนวน cluster
ที่เหมาะสมสำหรับการแบ่งกลุ่ม (clustering)



Silhouette Score Method เพื่อหาจำนวน cluster ที่
เหมาะสมสำหรับการแบ่งกลุ่ม (clustering)

1

● นำ K-means Clustering จากคอลัม์ Income
แบ่งกลุ่มรายได้ของลูกค้า
เพื่อที่จะนำไปวางแผนยุทธ์การขายสินค้า



ขั้นตอนการทำ K-MEANS CLUSTERING

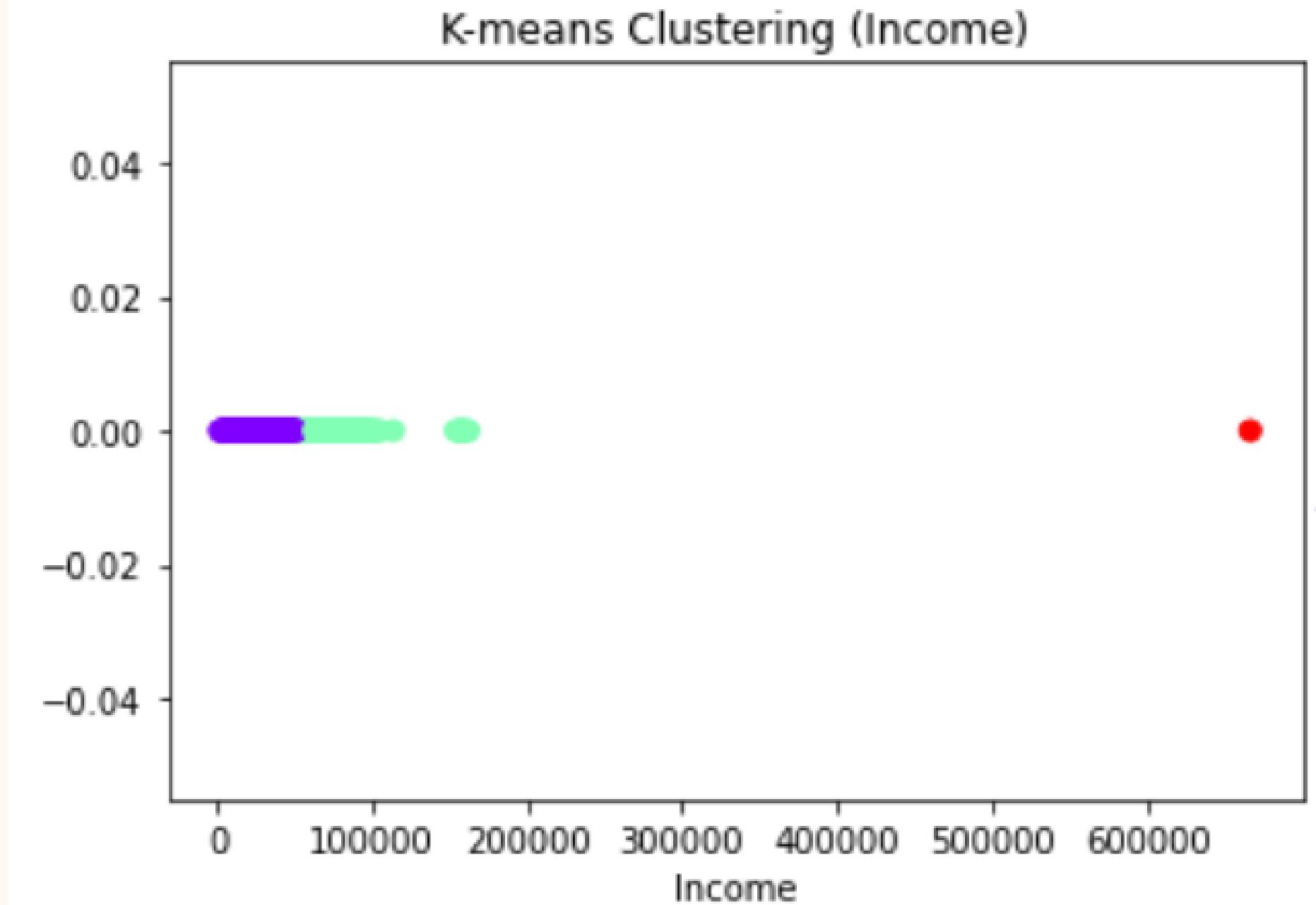
```
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
#init='k-means++' คือวิธีการที่เลือกจุดศูนย์กลางโดยจะเดือด
#แต่ละกูบุนมาคำนวณใหม่ในแต่ละรอบของการจัดกลุ่ม โดยใช้ค่าเฉลี่ยของจุดทั้งหมดในกูบุนนั้นๆ เพื่อมาเป็นจุดศูนย์กลางใหม่
kmeans.fit(income)
#.fit ใช้ในการสร้างโมเดล โดยใช้ชื่ออย่าง income เพื่อ หาจุดศูนย์กลางค่าที่มีรายได้ใกล้เคียงกันไว้
```

```
# คำนวนค่าเฉลี่ยรายได้
income_means = []
for i in range(3):
    income_means.append(np.mean(income[kmeans.labels_ == i]))
#การแปลงกูบุนของข้อมูลรายได้ (income) ออกเป็น 3 กลุ่ม โดยใช้ loop for เพื่อหาค่าเฉลี่ยของแต่ละกูบุน และเก็บค่าเข้าไปใน list ชื่อ income_means

# แสดงผลข้างรายได้สำหรับแต่ละคลัสเตอร์
for i in range(3):
    print("Cluster {} income range: {:.2f}- {:.2f}".format(i+1, income_means[i]-np.std(income[kmeans.labels_ == i]), income_means[i]+np.

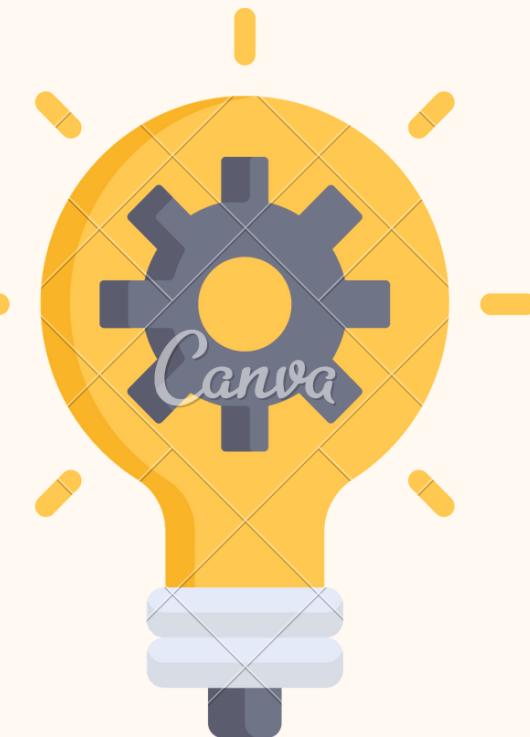
# Visualize clusters using scatter plot
plt.scatter(income, np.zeros_like(income), c=kmeans.labels_, cmap='rainbow')
plt.xlabel('Income')
plt.title('K-means Clustering (Income)')
plt.show()
```

Cluster 1 income range: 23727.56-46337.26
Cluster 2 income range: 57553.91-83275.10
Cluster 3 income range: 666666.00-666666.00



สรุปผล

- **กลุ่มลูกค้าที่ 1 สีน้ำเงิน เป็นกลุ่มที่มีรายได้รายเดือนต่อเดือนแต่ 23,727.56 ถึง 46,337.26 ดอลลาร์** ดังนั้นจึงต้องใช้กลยุทธ์การขายที่ตรงกับฐานข้อมูลลูกค้ากลุ่มนี้ คือ โปรโมชั่นส่วนลดมากกว่ากลุ่มลูกค้าอื่น ๆ เป็นต้น
- **กลุ่มลูกค้าที่ 2 สีเขียว เป็นกลุ่มที่มีรายได้รายเดือนปานกลาง** ตั้งแต่ 57,553.91 ถึง 83,275.10 ดอลลาร์ดังนั้นสามารถจัดทำแผนการตลาดที่เหมาะสมกับกลุ่มลูกค้านี้ได้ คือ จัดโปรโมชั่น
- **กลุ่มลูกค้าที่ 3 สีแดง เป็นกลุ่มที่มีรายได้รายเดือนสูงมาก** ตั้งแต่ 666,666.00 ดอลลาร์ขึ้นไป ดังนั้นจึงต้องให้ ความสำคัญกับประสิทธิภาพของการบริการและส่วนลดที่มีคุณภาพเพื่อสร้างความพึงพอใจ และความลงตัวของลูกค้ากลุ่มนี้ นอกจากนี้ยังสามารถสร้างโปรโมชั่นส่วนลดที่มีประสิทธิภาพสูง เช่น โปรโมชั่นซื้อเพียงหนึ่งครั้งก็สามารถรับสิทธิประโยชน์ได้เยอะกว่าลูกค้าอื่น ๆ เป็นต้น



2

ทำ K-means Clustering จากคอลัมน์ Income,MntFishProducts,
MntMeatProducts และ MntSweetProducts
แบ่งกลุ่มรายได้ของลูกค้ากับการซื้อสินค้า
เพื่อจะได้นำไปวางแผนยุทธ์การขายสินค้าให้เข้ากับแต่ละกลุ่มสินค้า



ขั้นตอนการทำ K-MEANS CLUSTERING

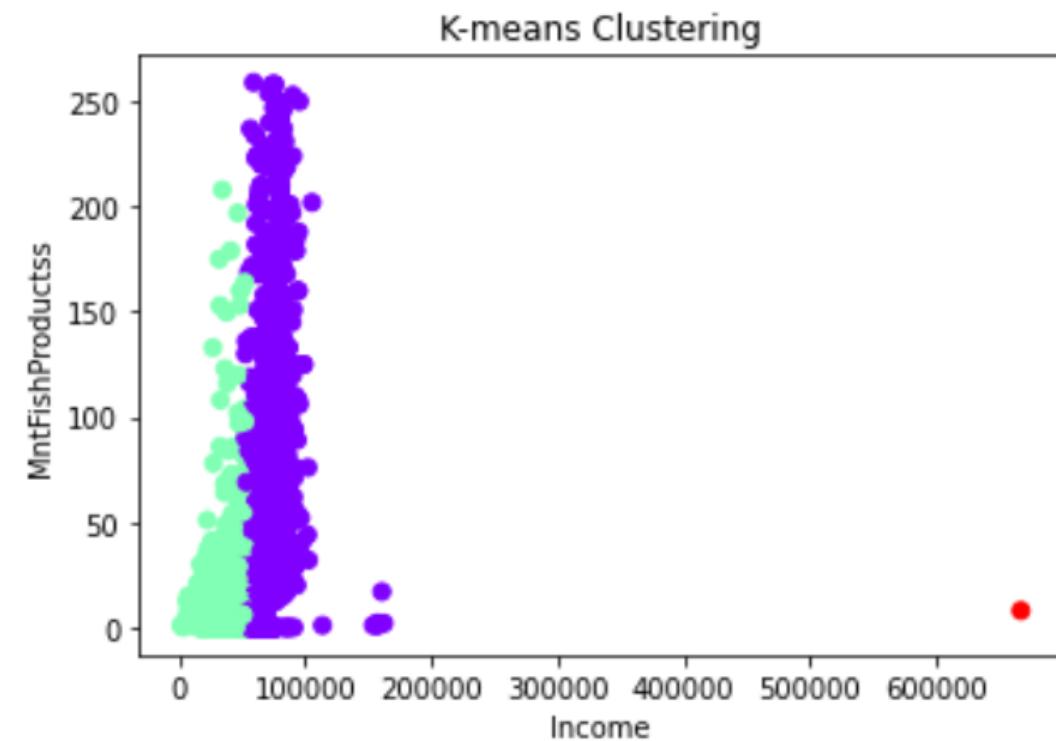
```
# แบ่งกลุ่มออกเป็น 3 กลุ่ม
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
kmeans.fit(df)

# วนลูปเก็บค่า cluster
df_means = pd.DataFrame(columns=['Income', 'MntFishProducts', 'MntMeatProducts', 'MntSweetProducts'])
for i in range(3):
    df_means.loc[i] = df[kmeans.labels_ == i].mean()#.loc[i] ใช้เพื่อเข้าถึงแนวข้อ DataFrame df_means

# วนลูปเพื่อหา ช่วงรายได้สำหรับแต่ละคอลัมน์
for i in range(3):
    print("Cluster {} income range: {:.2f}- {:.2f} ".format(i+1, df_means.loc[i, 'Income']-np.std(df[kmeans.labels_ == i]['Income']), df_means.loc[i, 'Income']+np.std(df[kmeans.labels_ == i]['Income'])))

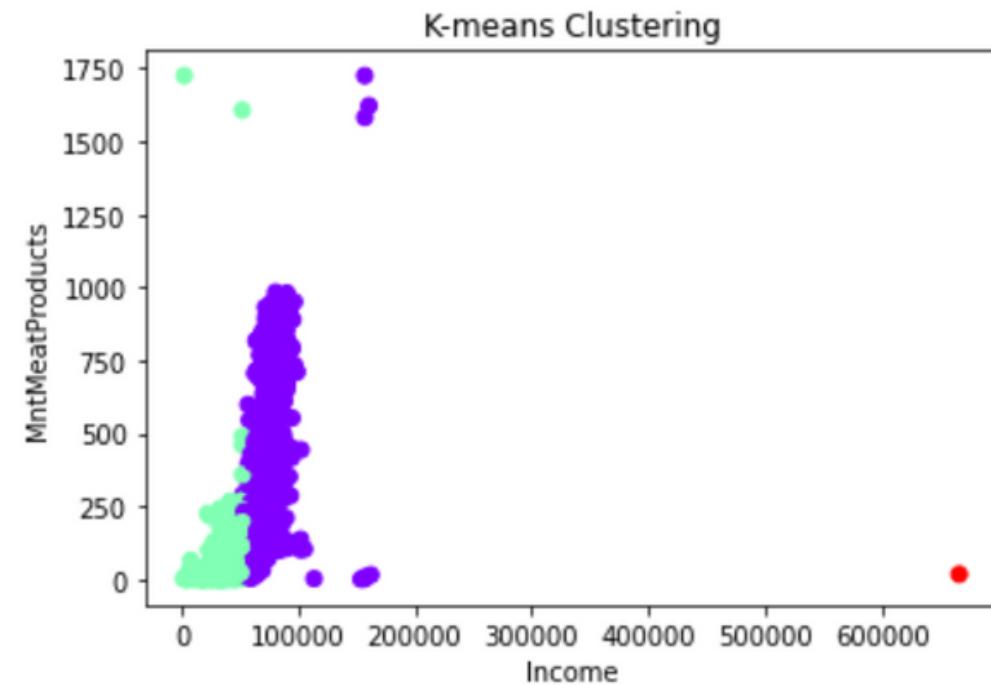
#แสดงกราฟการแบ่งกลุ่มที่ลูกค้ารายได้ของลูกค้า กับ จำนวนเงินที่ใช้ซื้อสินค้าในช่วง 2 ปีที่ผ่านมา
plt.scatter(df['Income'], df['MntFishProducts'], c=kmeans.labels_, cmap='rainbow')
plt.xlabel('Income')
plt.ylabel('MntFishProductss')
plt.title('K-means Clustering')
plt.show()
```

Cluster 1 income range: 57553.91-83275.10
 Cluster 2 income range: 23727.56-46337.26
 Cluster 3 income range: 666666.00-666666.00



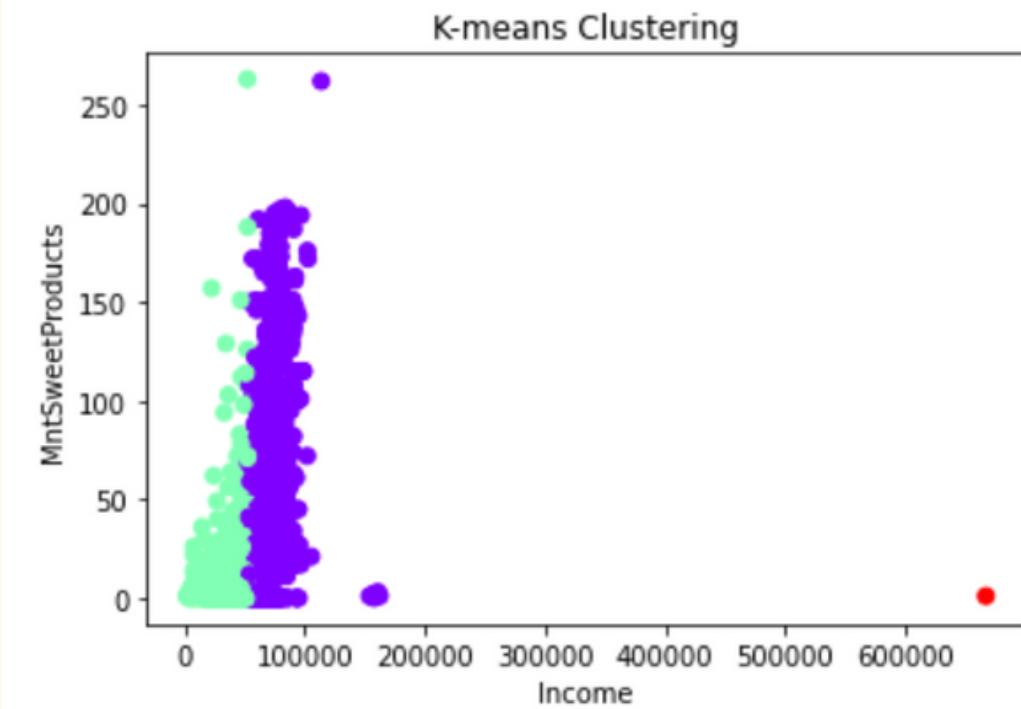
กราฟการแบ่งกลุ่มที่ลูกค้ารายได้ของลูกค้า กับ
จำนวนเงินที่ใช้สำหรับซื้อสินค้าปลาในช่วง 2 ปีที่ผ่านมา

Cluster 1 income range: 57553.91-83275.10
 Cluster 2 income range: 23727.56-46337.26
 Cluster 3 income range: 666666.00-666666.00



กราฟการแบ่งกลุ่มที่ลูกค้า รายได้ของลูกค้า
กับจำนวนเงินที่ใช้สำหรับซื้อสินค้าเนื้อในช่วง 2 ปีที่ผ่านมา

Cluster 1 income range: 57553.91-83275.10
 Cluster 2 income range: 23727.56-46337.26
 Cluster 3 income range: 666666.00-666666.00



กราฟการแบ่งกลุ่มที่ลูกค้า รายได้ของลูกค้า
กับจำนวนเงินที่ใช้สำหรับซื้อสินค้าหวาน
ในช่วง 2 ปีที่ผ่านมา

สรุปผล

- **กลุ่มที่ 1 มีรายได้เฉลี่ยต่อ กี่สุดอยู่ที่ประมาณ 23,727.56 ถึง 46,337.26 ดอลลาร์ และมียอดซื้อสินค้าประเภทเนื้อสัตว์สูงสุด รองลงเป็นขนมหวาน และน้ำออยที่สุดเป็น ปลา โดยจัดกลยุทธ์ในการตลาดดังนี้ ให้ส่วนลดพิเศษหรือโปรโมชั่นเมื่อซื้อครึ่งแรก เพื่อดึงดูดให้มาซื้อสินค้าสร้างสินค้าที่มีราคาถูกและคุณภาพดี เพื่อเป็นตัวเลือกสำหรับลูกค้าในกลุ่มนี้**
- **กลุ่มที่ 2 มีรายรายได้รายเฉลี่ยปานกลาง เฉลี่ยระหว่าง 57,553.91 ถึง 83,275.10 ดอลลาร์ และมียอดซื้อสินค้าประเภทเนื้อสัตว์สูงสุด รองลงมาเป็น ปลาและขนมหวาน โดยจัดกลยุทธ์ในการตลาดดังนี้ ให้ส่วนลดพิเศษหรือโปรโมชั่นเมื่อซื้อสินค้าจำนวนมาก โดยเน้นไปที่สินค้าที่เคยซื้อมา ก่อนเพื่อให้เกิดความพึงพอใจและติดต่อซื้อสินค้าอีก ส่งเสริมการสั่งซื้อออนไลน์โดยมีโปรโมชั่นพิเศษสำหรับการสั่งซื้อทางออนไลน์**
- **กลุ่มที่ 3 มีรายได้เฉลี่ยสูงที่สุด อยู่ที่ประมาณ 666666.00 ดอลลาร์ ขึ้นไป โดยจัดกลยุทธ์ในการตลาดดังนี้ สร้างแพ็คเกจสินค้าสำหรับลูกค้าที่มีราคาถูกและคุณภาพดี เพื่อตอบสนองความต้องการของกลุ่มนี้ ให้ส่วนลดพิเศษหรือโปรโมชั่นเมื่อซื้อสินค้าในปริมาณมาก**

3

ทำ K-means Clustering จากคอลัมน์ Income และ NumDealsPurchases เพื่อแบ่งกลุ่มรายได้ของลูกค้า
ที่มีจำนวนการซื้อสินค้าที่มีส่วนลด
เพื่อที่จะนำไปวางแผนกลยุทธ์การขายสินค้า



kmeans.cluster_centers_

```

x = superstore_data_clean[['Income', 'NumDealsPurchases']]
# ការបង្ការណ៍ cluster ដើម្បាននៅ តួ 3 គឺ n_clusters=3
# ការបង្ការនឹងការពិនិត្យស្ថិតិយវត្ថុ centroid តាម init='k-means++'
# ការបង្ការណ៍ការរាយចំណាំទៅការជួយ centroid តាម max_iter=300
# ការបង្ការណ៍ការរាយចំណាំទៅការជួយ centroid តាម n_init=10
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10, random_state=42)
y_kmeans = kmeans.fit_predict(x)

#y_kmeans គឺជាប្រភពនៃការបង្ការណ៍ក្នុងការរាយទៅការជួយ KMeans នៅលើការរាយទៅការជួយ fit_predict() ដើម្បាននឹងការរាយទៅក្នុងការជួយ KMeans
#គ្រប់គ្រង x ដើម្បាននៅ ការបង្ការណ៍ទៅការជួយ fit_predict() ដើម្បាននឹងការរាយទៅក្នុងការជួយ KMeans

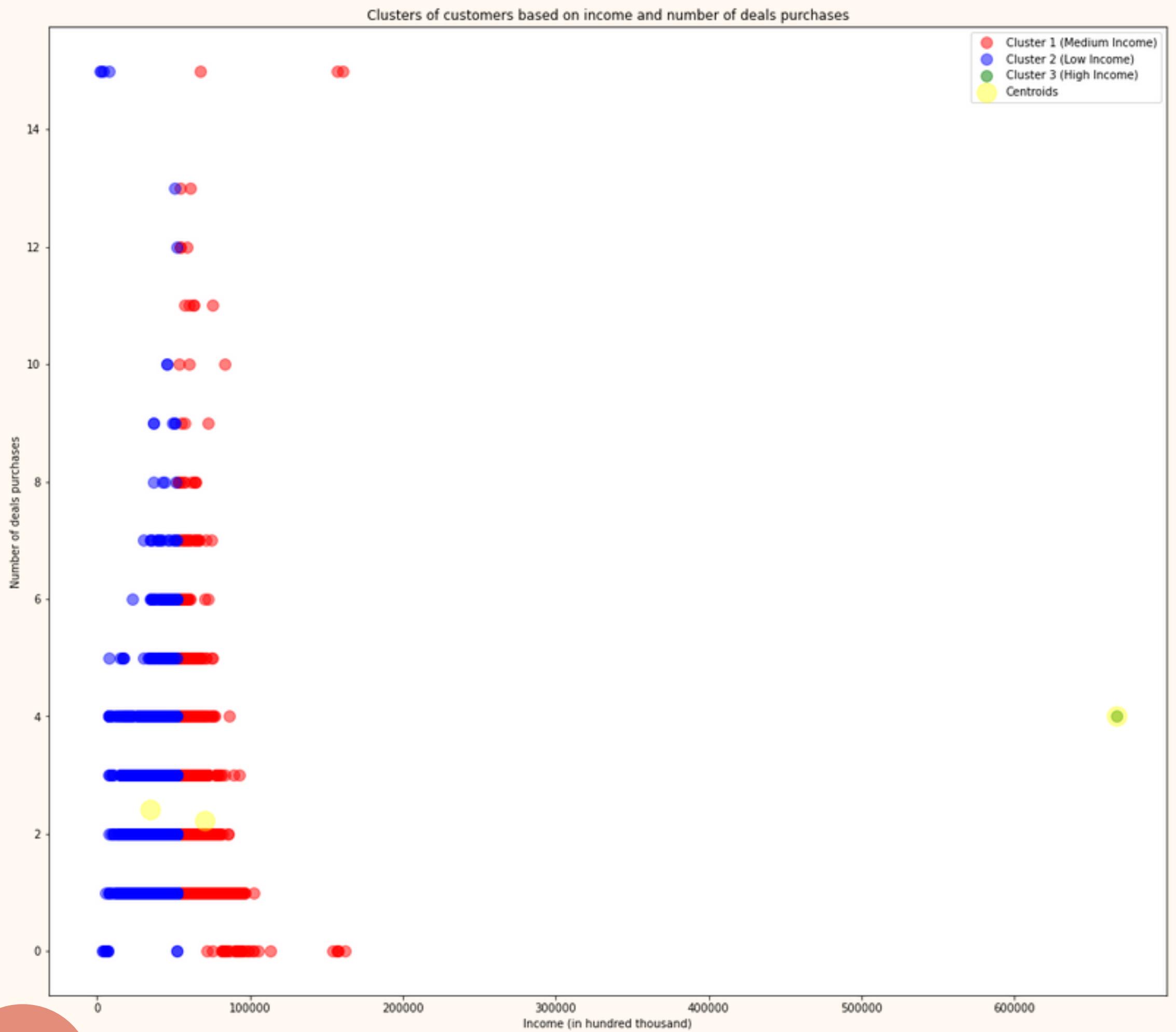
plt.figure(figsize=(18, 16))
# ការរាយទៅក្នុង scatter plot នៃក្នុងការជួយ Cluster 1 និងក្នុងការជួយ x នៅ 'Income' ជីវិសាស x
# និងក្នុងការជួយ x នៅ 'NumDealsPurchases' ជីវិសាស y និងក្នុង y_kmeans
# ពីនេះគឺជាប្រភពនៃការជួយ Cluster 1 ដែលនឹង និងក្នុងការជួយ x នៅ alpha និងក្នុងការជួយ y និងក្នុង y_kmeans
plt.scatter(x.iloc[y_kmeans == 0, 0], x.iloc[y_kmeans == 0, 1], s=100, c='red', alpha=0.5, label='Cluster 1 (Medium Income)')
plt.scatter(x.iloc[y_kmeans == 1, 0], x.iloc[y_kmeans == 1, 1], s=100, c='blue', alpha=0.5, label='Cluster 2 (Low Income)')
plt.scatter(x.iloc[y_kmeans == 2, 0], x.iloc[y_kmeans == 2, 1], s=100, c='green', alpha=0.5, label='Cluster 3 (High Income)')

kmeans.cluster_centers_[:, 0]: នៅ ក្នុងក្នុង x នៃ Centroids នៃក្នុង Cluster ដូចនាកំណើនរាយ attributes នៃក្នុងក្នុង clustering នៅ 'Income'
kmeans.cluster_centers_[:, 1]: នៅ ក្នុងក្នុង y នៃ Centroids នៃក្នុង Cluster ដូចនាកំណើនរាយ attributes នៃក្នុងក្នុង clustering នៅ 'NumDealsPurchases'
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='yellow', alpha=0.4, label='Centroids')
plt.title('Clusters of customers based on income and number of deals purchases')
plt.xlabel('Income (in hundred thousand)')
plt.ylabel('Number of deals purchases')
plt.legend()
plt.show()

kmeans.cluster_centers_

```

ແບລຜາ



ພລເລັກຮົ່ນ centroids ກັ້ງມາດ 3 ຈຸດ

- ຈຸດ centroids ຂອງ Cluster 1 ອີງກີ່ (74138.60, 2.19)
- ຈຸດ centroids ຂອງ Cluster 2 ອີງກີ່ (39667.73, 2.24)
- ຈຸດ centroids ຂອງ Cluster 3 ອີງກີ່ (134138.26, 2.16)

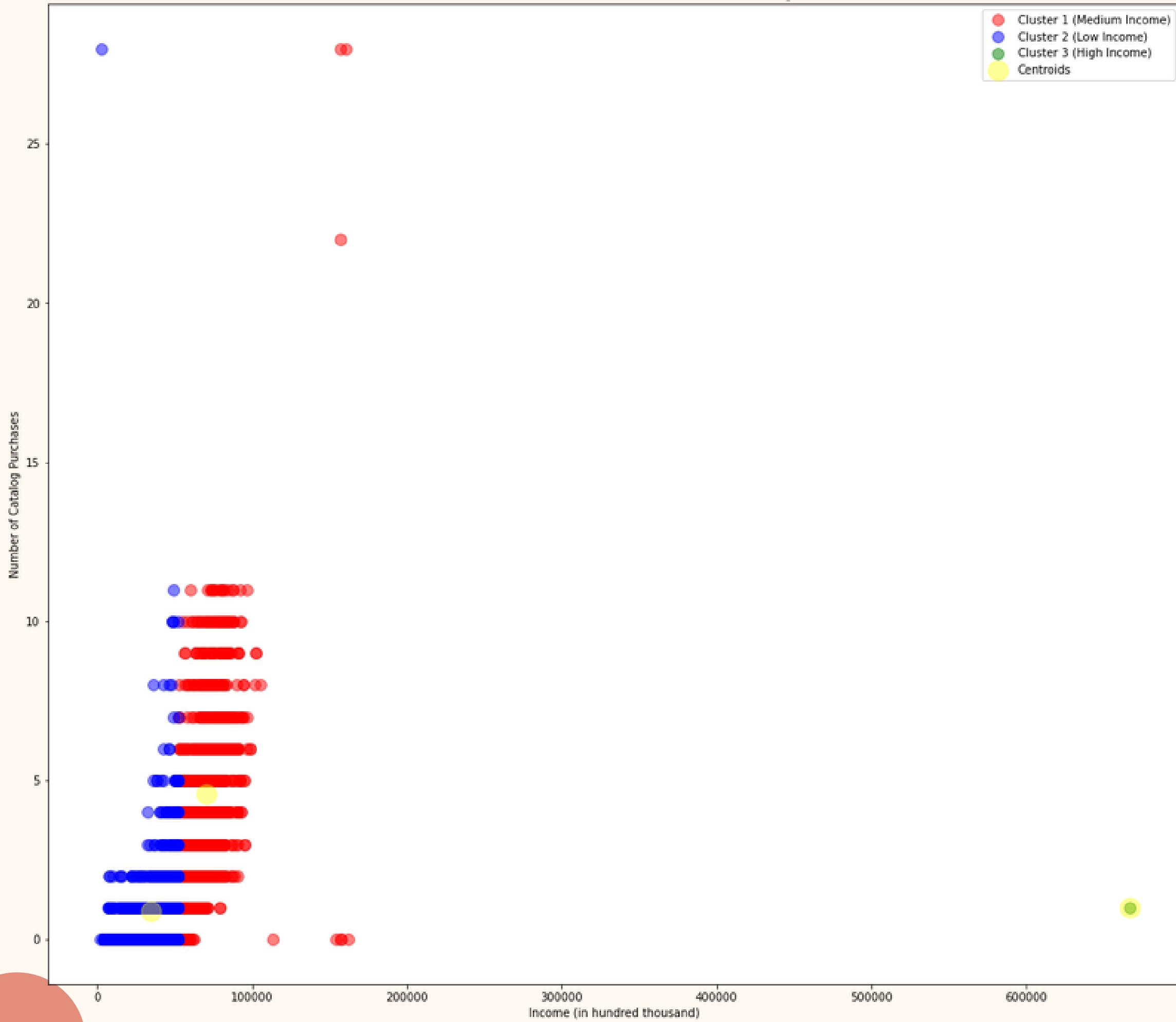


4

ทำ K-means Clustering จากคอลัม์ Income และ NumCatalogPurchases เพื่อแบ่งกลุ่มรายได้ของลูกค้าที่มีจำนวนการซื้อสินค้าผ่านแคตตาล็อก เพื่อที่จะนำไปวางแผนกลยุทธ์การขายสินค้า



Clusters of customers based on income and number of Catalog Purchases



ผลลัพธ์นี้ centroids ทั้งหมด 3 จุด

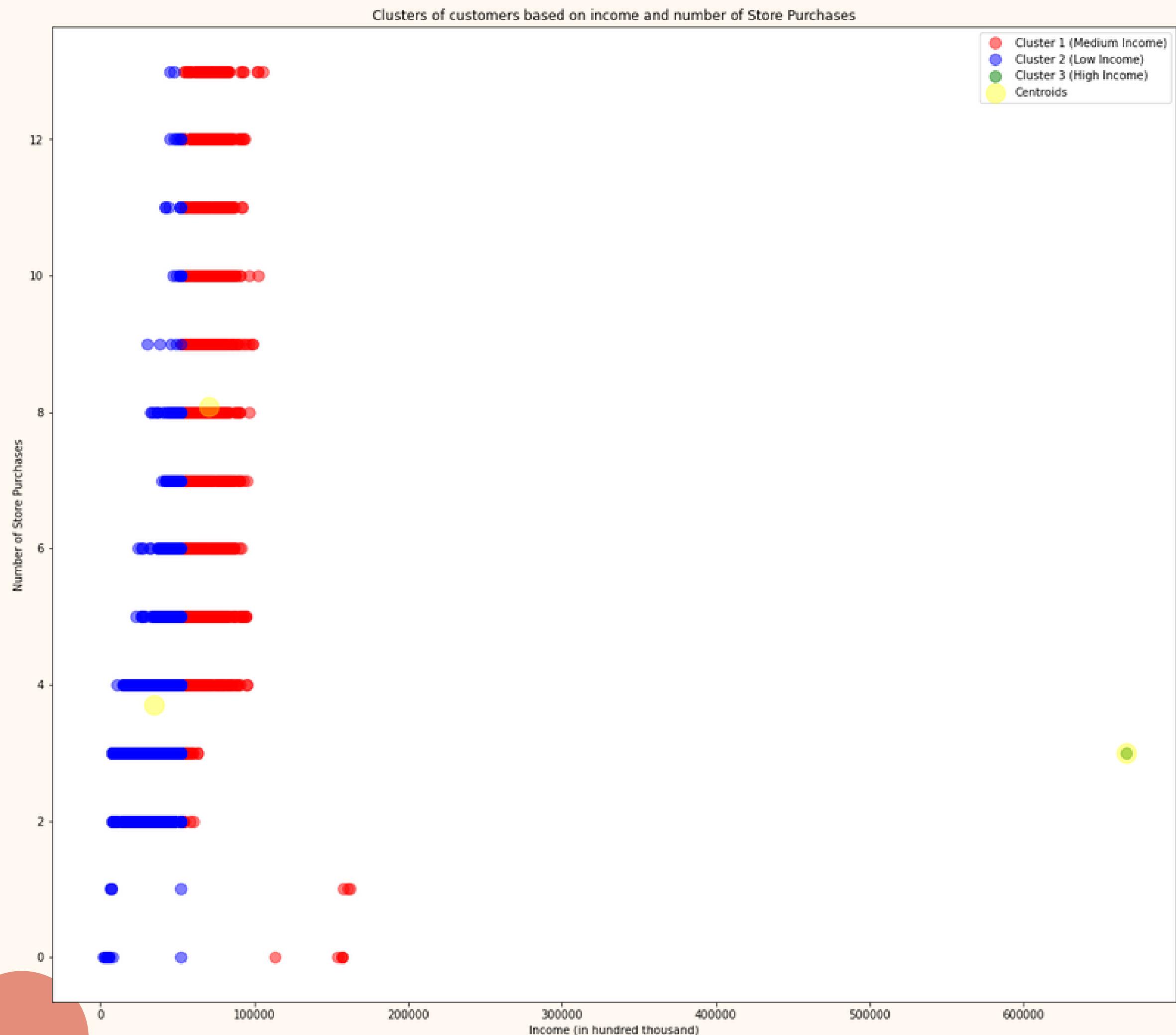
- จุด centroids ของ Cluster 1 อยู่ที่
(70431.0, 4.59)
- จุด centroids ของ Cluster 2 อยู่ที่
(35047.58, 0.89)
- จุด centroids ของ Cluster 3 อยู่ที่
(666666.0, 1.0)

5

ทำ K-means Clustering จากคอลัมน์ Income และ NumStorePurchases เพื่อแบ่งกลุ่มรายได้ของลูกค้า
ที่มีจำนวนการซื้อสินค้าโดยตรงที่ร้านค้า
เพื่อที่จะนำไปวางแผนกลยุทธ์การขายสินค้า



ແປລຜາ



ພລເລັພຣນີ centroids ກົ້ງໜູດ 3 ຈຸດ

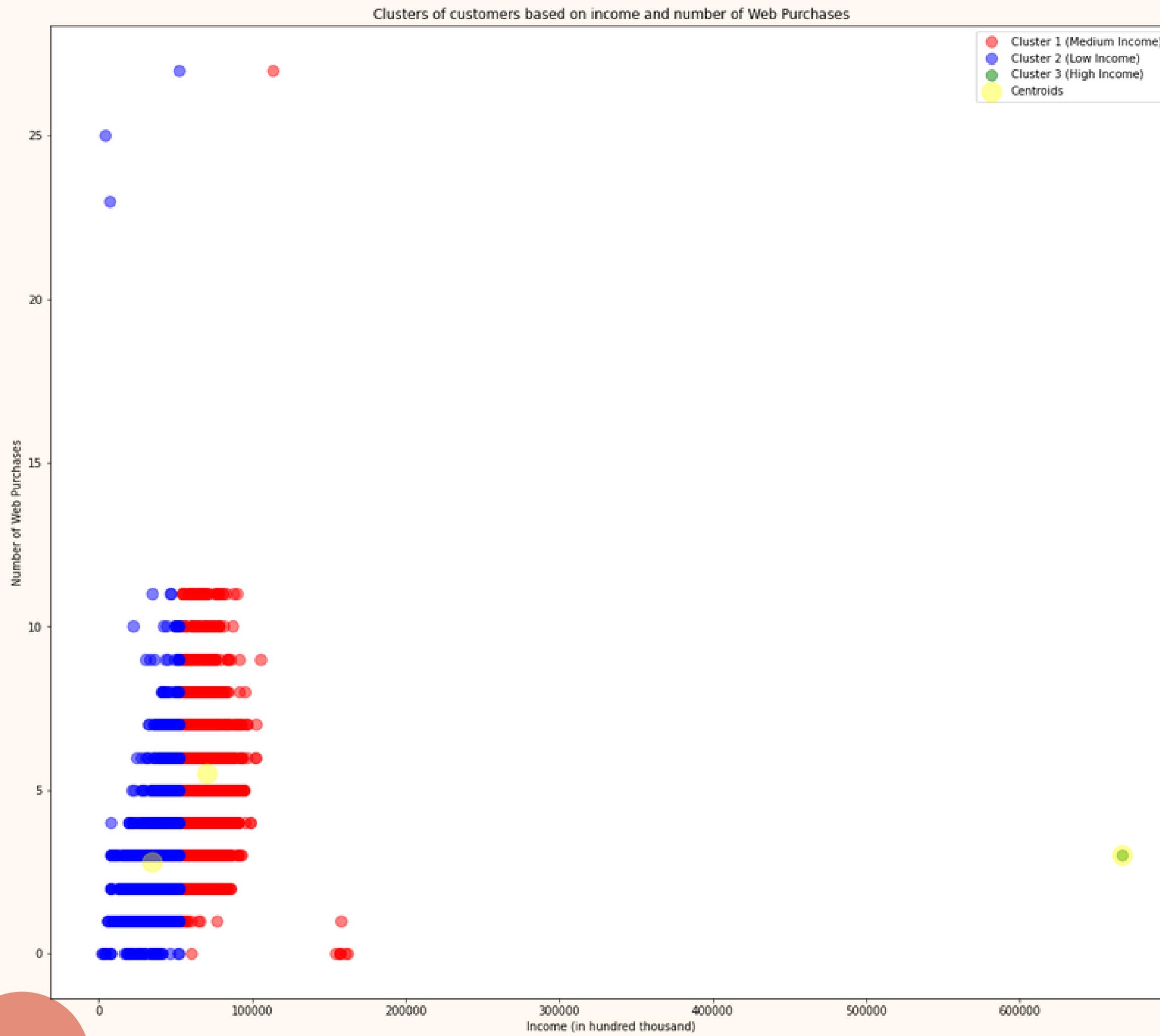
- ຈຸດ centroids ຂອງ Cluster 1 ອີງກີ່ (70431.00, 8.07)
- ຈຸດ centroids ຂອງ Cluster 2 ອີງກີ່ (35047.58, 3.70)
- ຈຸດ centroids ຂອງ Cluster 3 ອີງກີ່ (666666.00, 3.00)

6

ทำ K-means Clustering จากคอลัมน์ Income และ NumWebPurchases เพื่อแบ่งกลุ่มรายได้ของลูกค้า ที่มีจำนวนการซื้อสินค้าผ่านเว็บไซต์ของบริษัท เพื่อที่จะนำไปวางแผนยุทธ์การขายสินค้า



ແປລຜາ



ພລເລັກຮນ້ຳ centroids ກົ້ງໜົດ 3 ຈຸດ

- ຈຸດ centroids ຂອງ Cluster 1 ຄູ່ອ (70431.00, 5.49)
- ຈຸດ centroids ຂອງ Cluster 2 ຄູ່ອ (35047.58, 2.79)
- ຈຸດ centroids ຂອງ Cluster 3 ຄູ່ອ (666666.00, 3.00)

THANK YOU!

MEMBER



นางสาวกานุจนาสุดา พุยมูลตรี
633020438-6



นางสาววรรณนภา ส่งเสริม
633020550-2



นางสาวเกิดา เชื้อก้าว
633021011-7



นางสาวศิริขันยุ บุญศรี
633021024-8

