



GROUP 5

Customer Segmentation Classification

ที่มา : [https://www.kaggle.com/datasets/kaushiksuresh147/
customer-segmentationfbclid=IwAR0ZMZjddjyjm0BuNxa
61cON_8ngUvlMkkKjhKC13A-60jAA5JTZ-EFIalo](https://www.kaggle.com/datasets/kaushiksuresh147/customer-segmentationfbclid=IwAR0ZMZjddjyjm0BuNxa61cON_8ngUvlMkkKjhKC13A-60jAA5JTZ-EFIalo)



Context

บริษัทรถยนต์มีแผนที่จะเข้าสู่ตลาดใหม่ด้วยผลิตภัณฑ์ที่มีอยู่ (P1, P2, P3, P4 และ P5) หลังจากการวิจัยตลาดอย่างเข้มข้น พวกเขางานรุปได้ว่าพฤติกรรมของตลาดใหม่นั้นคล้ายคลึงกับตลาดเดิมของพวกเข้า ในตลาดที่มีอยู่ ก็มขายได้จัดประเพณีลูกค้า กึ่งหนดอออกเป็น 4 กลุ่ม (A, B, C, D) จากนั้นจึงทำการเข้าถึง และสื่อสารแบบแบ่งส่วนสำหรับลูกค้ากลุ่มต่างๆ กลยุทธ์นี้ทำงานได้ดีเป็นพิเศษสำหรับพวกเข้า พวกเขาวางแผนที่จะใช้กลยุทธ์เดียวกันสำหรับตลาดใหม่และได้ระบุผู้มีโอกาสเป็นลูกค้าใหม่



Content >>



Variable	Definition
ID	Unique ID
Gender	Gender of the customer
Ever_Married	Marital status of the customer
Age	Age of the customer
Graduated	Is the customer a graduate?
Profession	Profession of the customer
Work_Experience	Work Experience in years
Spending_Score	Spending score of the customer
Family_Size	Number of family members for the customer (including the customer)
Var_1	Anonymised Category for the customer
Segmentation	(target) Customer Segment of the customer

Data

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Segmentation
0	462809	Male	No	22	No	Healthcare	1.0	Low	4.0	D
1	462643	Female	Yes	38	Yes	Engineer	NaN	Average	3.0	A
2	466315	Female	Yes	67	Yes	Engineer	1.0	Low	1.0	B
3	461735	Male	Yes	67	Yes	Lawyer	0.0	High	2.0	B
4	462669	Female	Yes	40	Yes	Entertainment	NaN	High	6.0	A
5	461319	Male	Yes	56	No	Artist	0.0	Average	2.0	C
6	460156	Male	No	32	Yes	Healthcare	1.0	Low	3.0	C
7	464347	Female	No	33	Yes	Healthcare	1.0	Low	3.0	D
8	465015	Female	Yes	61	Yes	Engineer	0.0	Low	3.0	D
9	465176	Female	Yes	55	Yes	Artist	1.0	Average	4.0	C

Import Data

นำข้อมูลที่สนใจจาก Kaggle



นำข้อมูลลงในไดร์ฟ Project

เชื่อมไดร์ฟใน colab



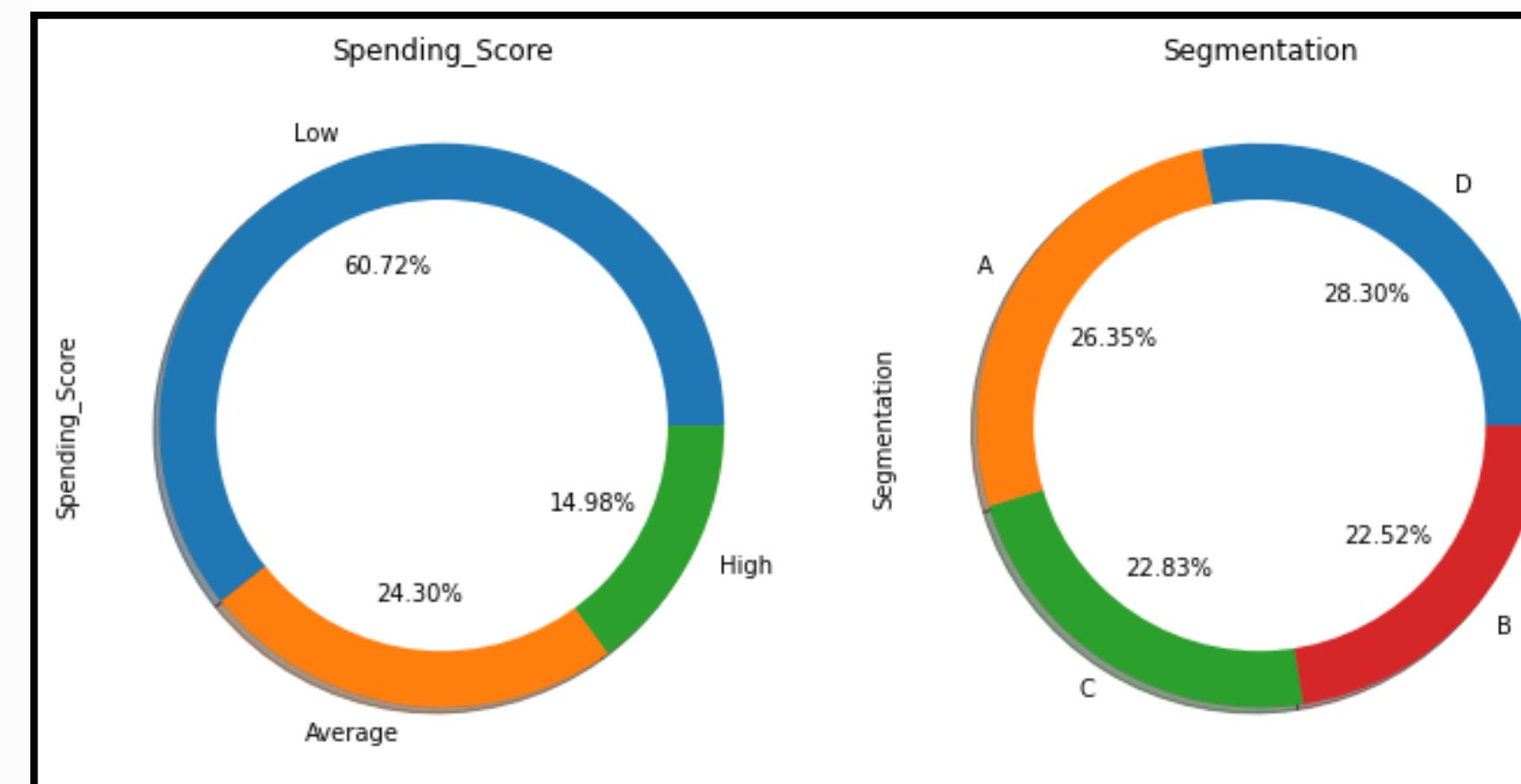
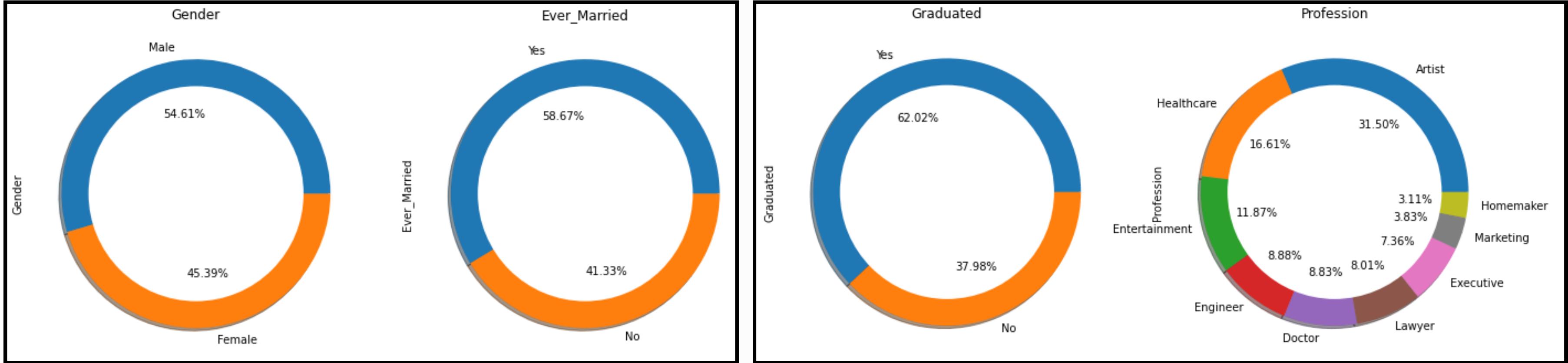
```
#เชื่อม google drive  
from google.colab import drive  
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

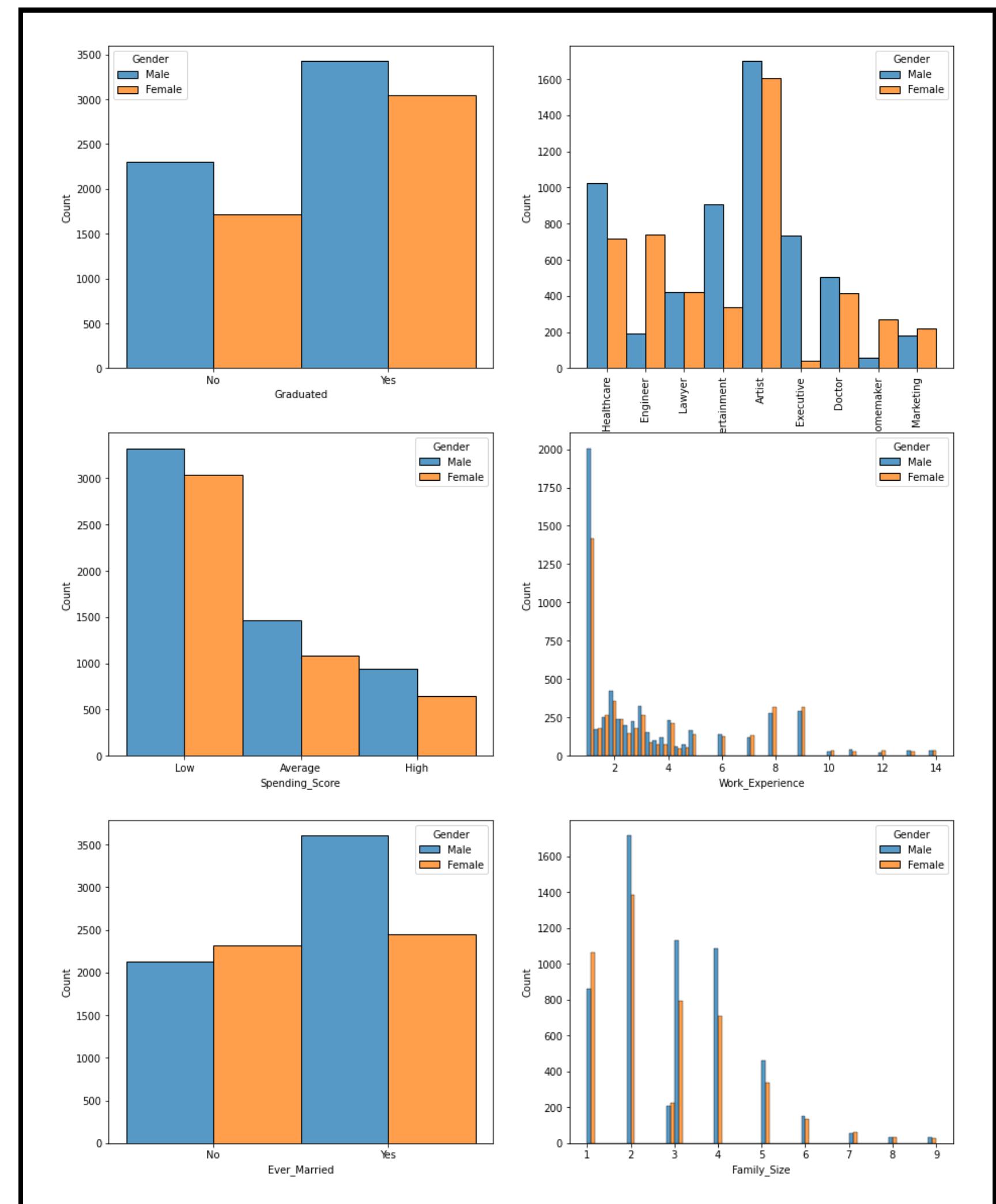
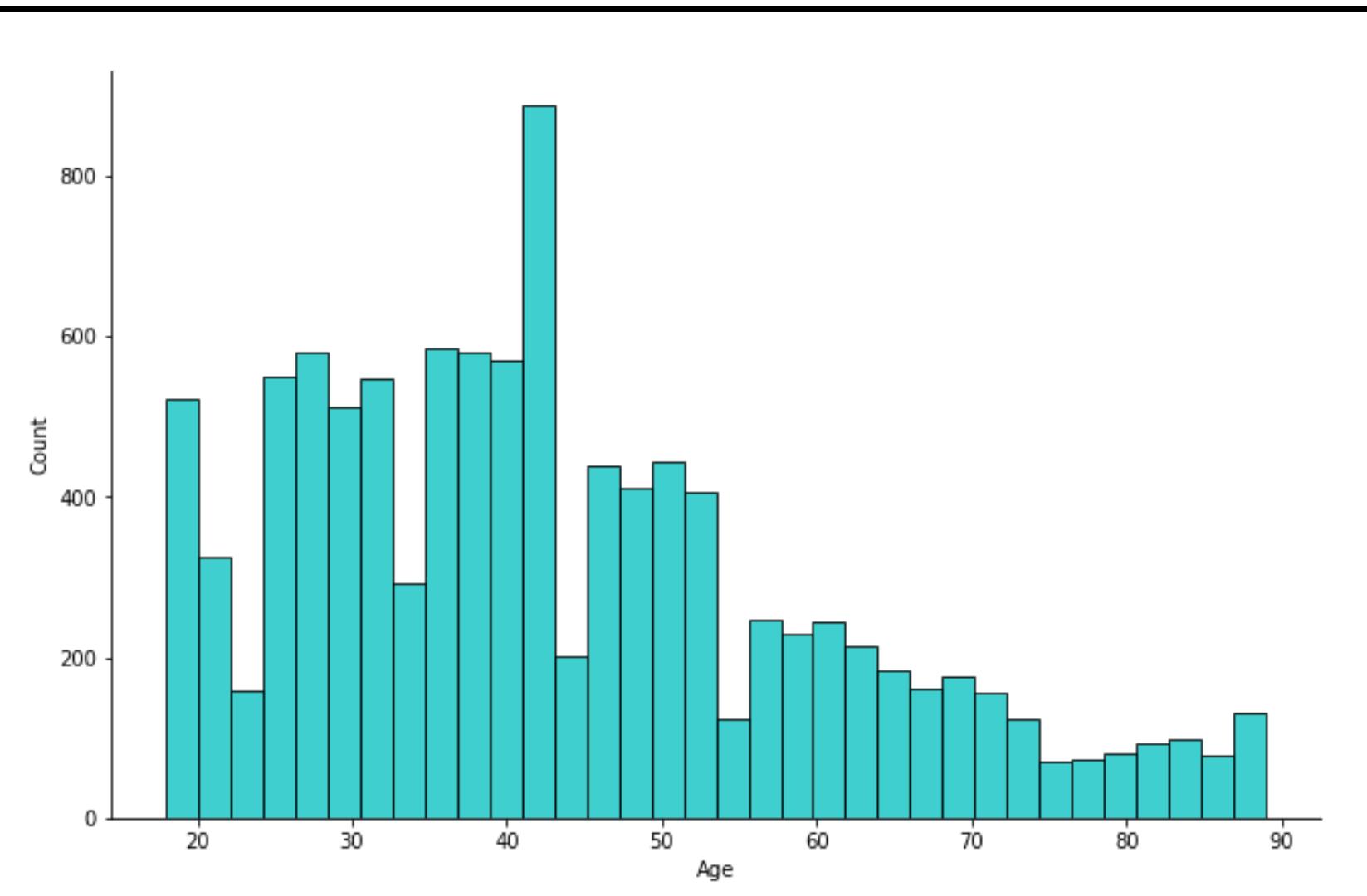
```
data_train = pd.read_csv('/content/drive/MyDrive/BPDM/Project/Train.csv', sep=',')  
data_test = pd.read_csv('/content/drive/MyDrive/BPDM/Project/Test.csv', sep=',')  
data_t = pd.concat([data_train, data_test])  
data_t = data_t.drop('Var_1', axis=1)  
data_t.head(5)
```

นำข้อมูลในไดร์ฟ project ลง colab

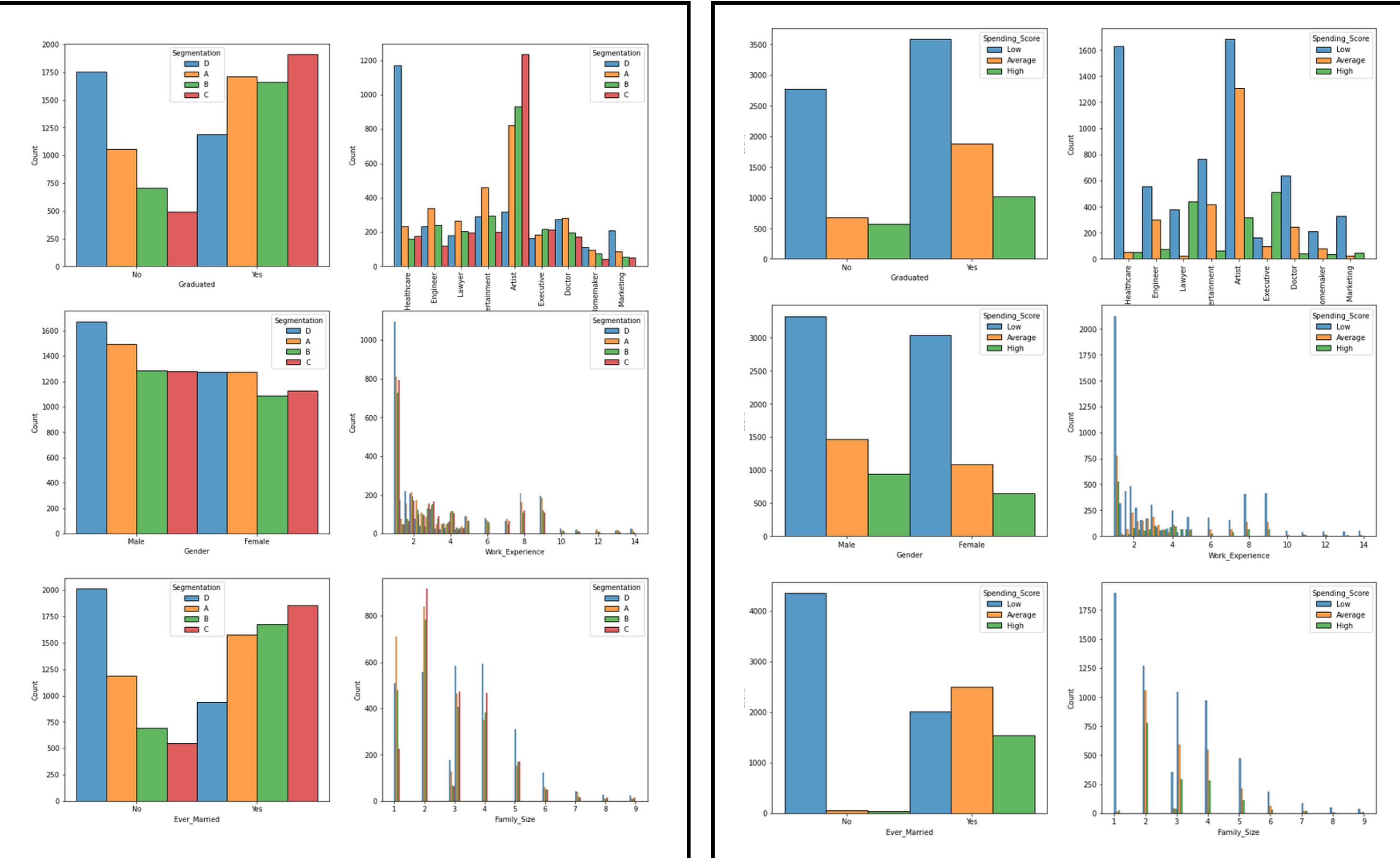
Data Illusion: Column



Data



Data



Prepare
Data



Step1: Data Cleaning

1. ลบแถวที่ซ้ำกัน (ทั้งหมด 38 แถว)
2. สร้างคอลัมน์จากการคำนวณ (Work_Experience_to_Age_Ratio) ที่อาจเป็นคุณสมบัติที่เกี่ยวข้องกับโมเดล (ทดสอบในภายหลัง)
3. จัดการ missing data: Ever_Married: แก้ missing ด้วย No, Graduated: แก้ missing ด้วย No, อัชีพ: ลบ (ไม่ใช่จำนวนช่องว่าง), Family_Size: ใส่ค่าเฉลี่ย, Var_1: ลบ (ไม่ใช่จำนวนช่องว่าง), Work_Experience: ประมาณการตามอายุ
4. ลบคอลัมน์ ID ออก
5. แปลงข้อมูล(เพศ, สถานภาพการสมรส, จบการศึกษาปริญญา, อัชีพ, คะแนนค่าใช้จ่าย, Var_1, การแบ่งกลุ่ม) ด้วยวิธี dummies
6. แปลงตัวแปรตัวเลขทั้งหมด (Age, Work_Experience, Family_Size, Work_Experience_to_Age_Ratio) เป็นช่วงคะแนน

Step2: Data Modelling

- กำหนดตัวแปร x และ y

```
X = data_t[["Age", "Family_Size", "Profession_Healthcare","Profession_Entertainment",
"Profession_Artist", "Ever_Married_Yes","Graduated_No","Spending_Score_Level"]]
```



```
y = data_t["Segmentation_Level"]
```
- การปรับหรือแปลง feature ทั้งหมดในชุดข้อมูลให้มีมาตราส่วนหรือค่าเดียวกัน ทำเพื่อหลีกเลี่ยงอคติที่มีต่อ feature เช่นในกรณีที่ขนาดของ feature ไม่เท่ากัน
- สุ่มแบ่งข้อมูลออกเป็นสองส่วนย่อย: ชุดหนึ่งสำหรับฝึกโมเดล (trainning) และอีกชุดสำหรับทดสอบประสิทธิภาพ (testing)

Classification

Decision tree, Naive Bayes and KNN



Decision tree, Naive Bayes and KNN

1. ทำการ Import Library ที่จำเป็น
2. ให้เครื่องเรียนรู้จากข้อมูลที่เตรียมไว้ โดยใช้ โมเดลในการเรียนรู้ที่เลือก และกำหนดไว้ คือเรียนรู้ข้อมูล (X_{train}) กับ (y_{train}) เพื่อให้ได้ โมเดลที่พร้อมใช้งาน
3. นำ Model ที่ได้มาทำการ Predict กับข้อมูล (X_{test}), (y_{test})
4. นำ Model ที่ได้มาทำการ Predict และ Evaluate

Deciding on a model

ผลลัพธ์จากการทดสอบ Model

- Descition tree

ROC AUC (train): 0.923 | ROC AUC (test): 0.621

- Naive bayes

ROC AUC (train): 0.707 | ROC AUC (test): 0.697

- KNN

ROC AUC (train): 0.820 | ROC AUC (test): 0.667

Deciding on a model

จากผลการทดสอบของ Decision Tree, Naive Bayes, และ KNN โดยใช้การประเมินผลด้วย ROC AUC บนชุดข้อมูล train และ test พบว่า Decision Tree มีค่า ROC AUC สูงที่สุดบนชุดข้อมูล train คือ 0.923 แต่มีค่า ROC AUC ต่ำกว่าชุดข้อมูล test ที่ 0.621 ซึ่งเป็นสัญญาณว่าโมเดลอาจเกิดปัญหา overfitting จึงไม่เหมาะสมสำหรับการใช้งานจริง

ส่วน Naive Bayes และ KNN ค่า ROC AUC บนชุดข้อมูล train และ test ไม่ต่างกันมากนัก แต่ ROC AUC ของ Naive Bayes บนชุดข้อมูล test สูงกว่า KNN นั่นหมายความว่า Naive Bayes อาจเหมาะสมกว่าในการใช้งานจริง ดังนั้น จึงควรเลือกใช้โมเดล Naive Bayes สำหรับการทำนาย Segmentation ของลูกค้า



Association rule

เราจะใช้ association rule เพื่อหาความสัมพันธ์ระหว่างข้อมูล data train และ data test เพื่อกำหนดข้อมูลกลุ่มลูกค้าใหม่ที่เหมาะสม

Step1: Data Cleaning

existing_customers

1. existing = data_train ที่ทำการ drop ค่า 'Var_1'
2. ลบແກວທີ່ໜ້າກົນ (ທັງໝາດ 38 ແກວ)
3. สร้างຄອລິ້ນຈາກການคำนວณ (Work_Experience_to_Age_Ratio) ທີ່ຈະເປັນຄຸນສມບັດທີ່
ເກື່ອງບ້ອງກັບໂມເດລ (ທດສອບໃນກາຍຫຼັງ)
4. ຈັດການ missing data: Ever_Married: ແກ້ missing ດ້ວຍ No,Graduated: ແກ້
missing ດ້ວຍ No,ອາຊື່ພ: ລບ (ໄມ່ໃຊ່ຈຳນວນໜ່ອງວ່າງ),Family_Size: ໃສ່ຄ່າ
ເຂົ້າ,Work_Experience: ປະມານກາຣຕາມອາຍຸ
5. ລບຄອລິ້ນ ID ອອກ
6. ແປລັງບ້ອນມູລ(ເພີ່ມ, ສຖານກາພກາຮສມຣລ, ຈບກາຣສຶກຫາປຣິລູນາ, ອາຊື່ພ, ຄະແນນຄ່າໃຊ້ຈ່າຍ, ກາຣ
ແປ່ງກລຸ່ມ) ດ້ວຍວິຣີ dummies
7. ແປລັງຕົວແປຣຕົວເລບທັງໝາດ (Age, Work_Experience, Family_Size,
Work_Experience_to_Age_Ratio)ເປັນໜ່ວງຄະແນນ

Step 1: Data Cleaning

existing_customers

8. เลือกจาก existing เนพะคอลัมน์ที่ต้องการ ใช้ชื่อเป็น existing1

	Gender	Ever_Married	Age_Range	Graduated	Profession	Spending_Score_Level	Segmentation	Work_Experience_Range	Family_Size_Range
0	Male	No	18-30	No	Healthcare	1	D	0-5	3-6
1	Female	Yes	30-40	Yes	Engineer	2	A	0-5	3-6
2	Female	Yes	60+	Yes	Engineer	1	B	0-5	0-3
3	Male	Yes	60+	Yes	Lawyer	3	B	0-5	0-3
4	Female	Yes	40-50	Yes	Entertainment	3	A	0-5	6+
...
8062	Male	Yes	40-50	Yes	Artist	3	B	0-5	3-6
8064	Male	No	30-40	No	Executive	1	D	0-5	3-6
8065	Female	No	30-40	Yes	Healthcare	1	D	0-5	0-3
8066	Female	No	18-30	Yes	Healthcare	1	B	0-5	3-6
8067	Male	Yes	30-40	Yes	Executive	2	B	0-5	3-6

Step1: Data Cleaning

new_customers

ทำบันทุณ 1-7 แบบเดียวกับ existing_customers

8. เลือกจาก new เฉพาะคอลัมน์ที่ต้องการ ใช้ชื่อเป็น new1

	Gender	Ever_Married	Age_Range	Graduated	Profession	Spending_Score_Level	Segmentation	Work_Experience_Range	Family_Size_Range
0	Female	Yes	30-40	Yes	Engineer	1	B	0-5	0-3
1	Male	Yes	30-40	Yes	Healthcare	2	A	5-10	3-6
3	Male	Yes	50-60	No	Executive	3	B	10+	0-3
4	Female	No	18-30	No	Marketing	1	A	0-5	3-6
5	Male	Yes	40-50	Yes	Doctor	3	C	0-5	3-6
...
2622	Male	No	18-30	No	Healthcare	1	B	5-10	3-6
2623	Female	No	30-40	Yes	Doctor	1	A	0-5	0-3
2624	Female	No	50-60	Yes	Entertainment	1	C	0-5	0-3
2625	Male	Yes	40-50	Yes	Executive	3	C	0-5	3-6
2626	Female	No	40-50	Yes	Healthcare	1	A	5-10	3-6

Step2: แปลงข้อมูลให้เป็นใบหน้า existing_customers

9. นำ existing1 ไปแปลงเป็น ใบหน้า ค่า 0 กับ 1 โดยใช้ dummies เปลี่ยนชื่อเป็น existing1_encode
10. เปลี่ยนชื่อคอลัมน์ของ existing1_encode เพื่อให้ง่ายต่อการนำไปใช้

	Female	Male	Singer	Married	Age_Range_18-30	Age_Range_30-40	Age_Range_40-50	Age_Range_50-60	Age_Range_60+	Graduated_No	...	A	B	C	D	
0	0	1	1	0	1	0	0	0	0	0	1	...	0	0	0	1
1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0
2	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0

3 rows × 33 columns

Work_Experience_Range_0-5	Work_Experience_Range_10+	Work_Experience_Range_5-10	Family_Size_Range_0-3	Family_Size_Range_3-6	Family_Size_Range_6+
1	0	0	0	1	0
1	0	0	0	1	0
1	0	0	1	0	0

Step2: แปลงข้อมูลให้เป็นไบนารี

new_customers

9. นำ new1 ไปแปลงเป็น ไบนารี ค่า 0 กับ 1 โดยใช้ dummies เปลี่ยนซึ่อเป็น new1_encode

10. เปลี่ยนชื่อคอลัมน์ของ new1_encode เพื่อให้ง่ายต่อการนำไปใช้

	Female	Male	Singer	Married	Age_Range_18-30	Age_Range_30-40	Age_Range_40-50	Age_Range_50-60	Age_Range_60+	Graduated_No	...	A	B	C	D	
0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0
1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0
3	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0

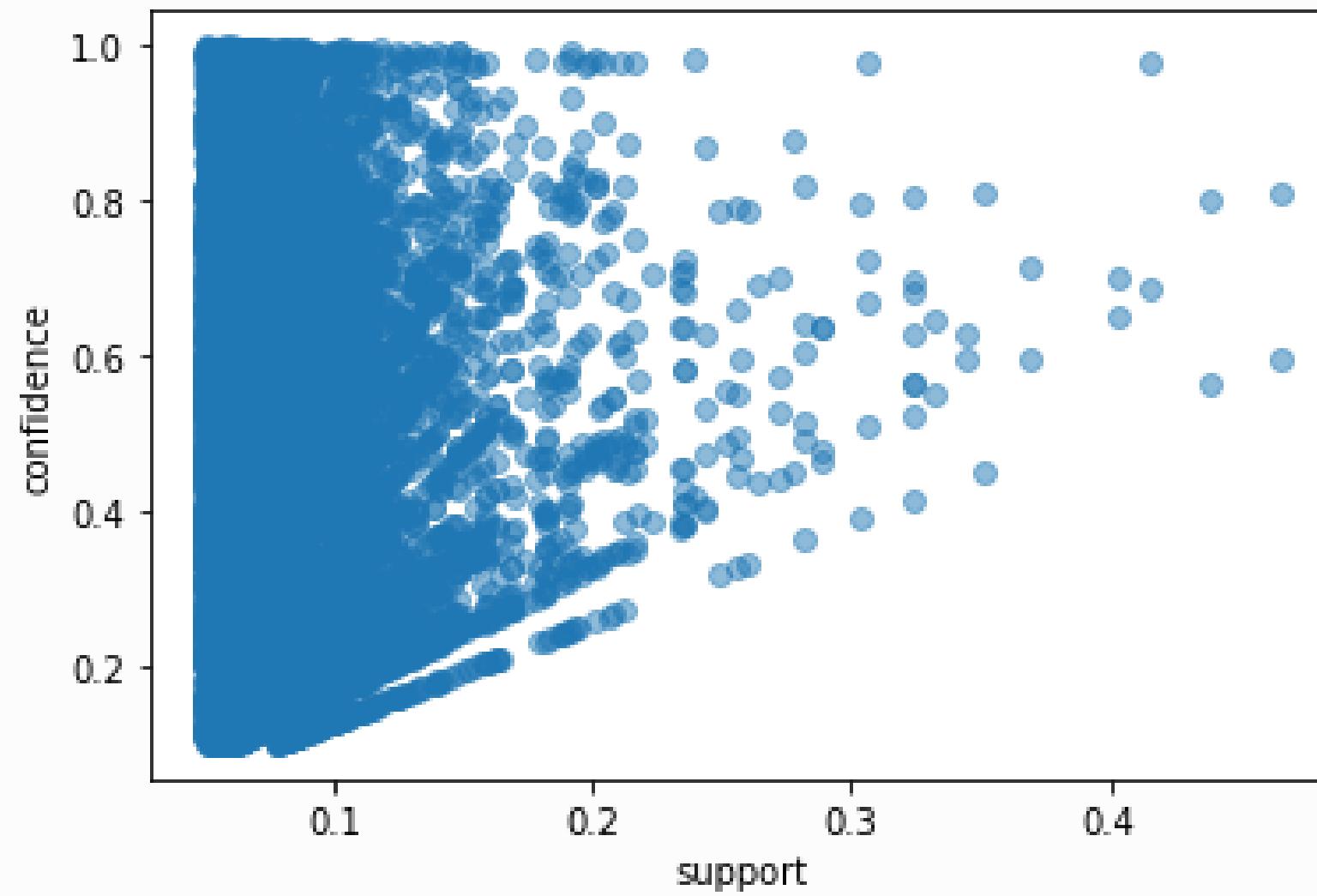
3 rows × 33 columns

Work_Experience_Range_0-5	Work_Experience_Range_10+	Work_Experience_Range_5-10	Family_Size_Range_0-3	Family_Size_Range_3-6	Family_Size_Range_6+
1	0	0	1	0	0
0	0	1	0	1	0
0	1	0	1	0	0

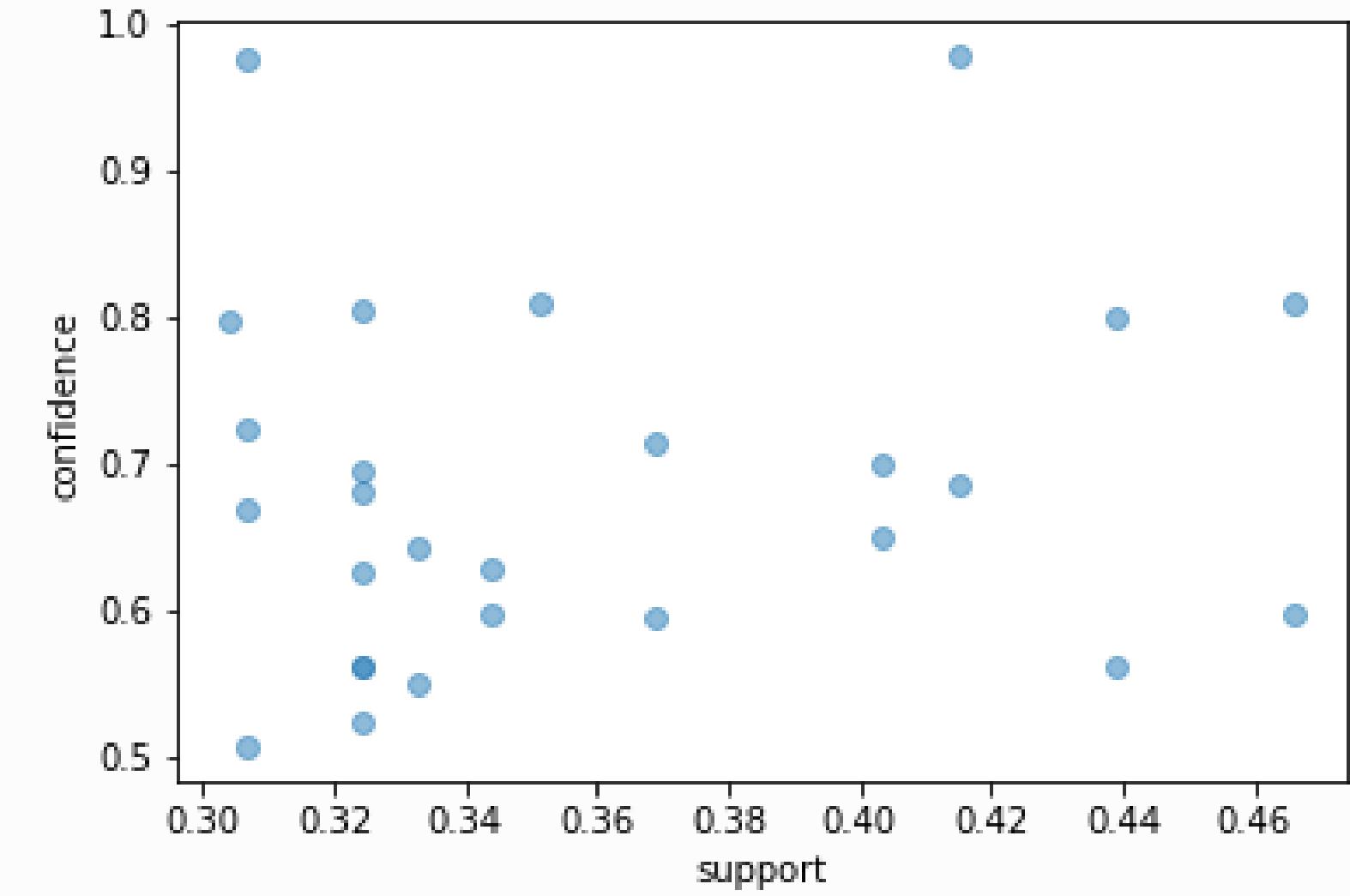
Step3: กรองชุดของกฎการเชื่อมโยง existing_customers และ new_customers

11. Install mlxtend และ import mlxtend
from mlxtend.frequent_patterns import association_rules, apriori
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import matplotlib.pyplot as plt
12. กรองชุดของกฎการเชื่อมโยง: ตามเกณฑ์การสนับสนุนและความเชื่อมั่น
ขึ้นต่อ

existing_customers

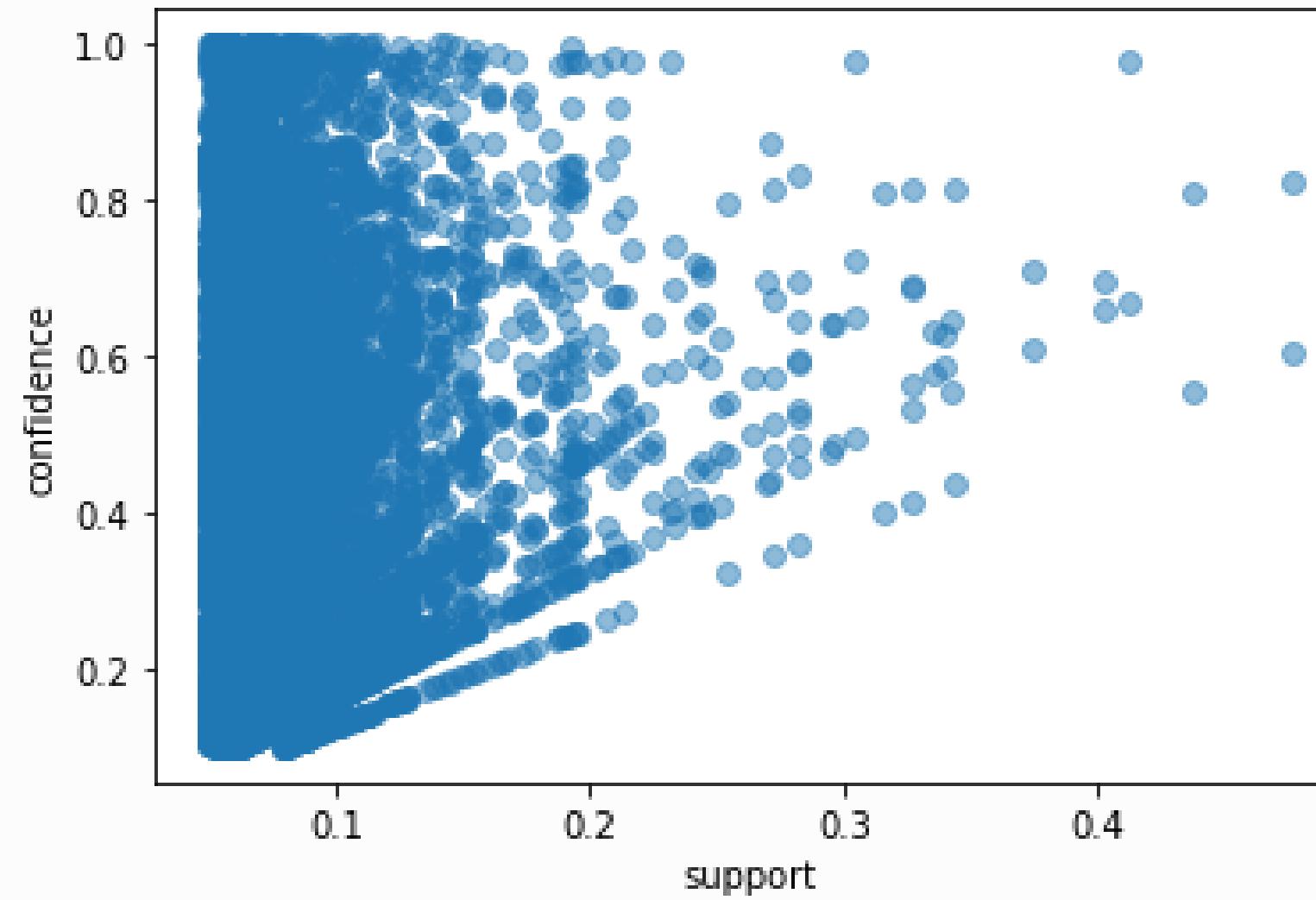


กรองกฎตามเกณฑ์การสนับสนุนและความเชื่อมั่น
rules0 = rules0[(rules0['support'] >= 0.05) & (rules0['confidence'] >= 0.1)]

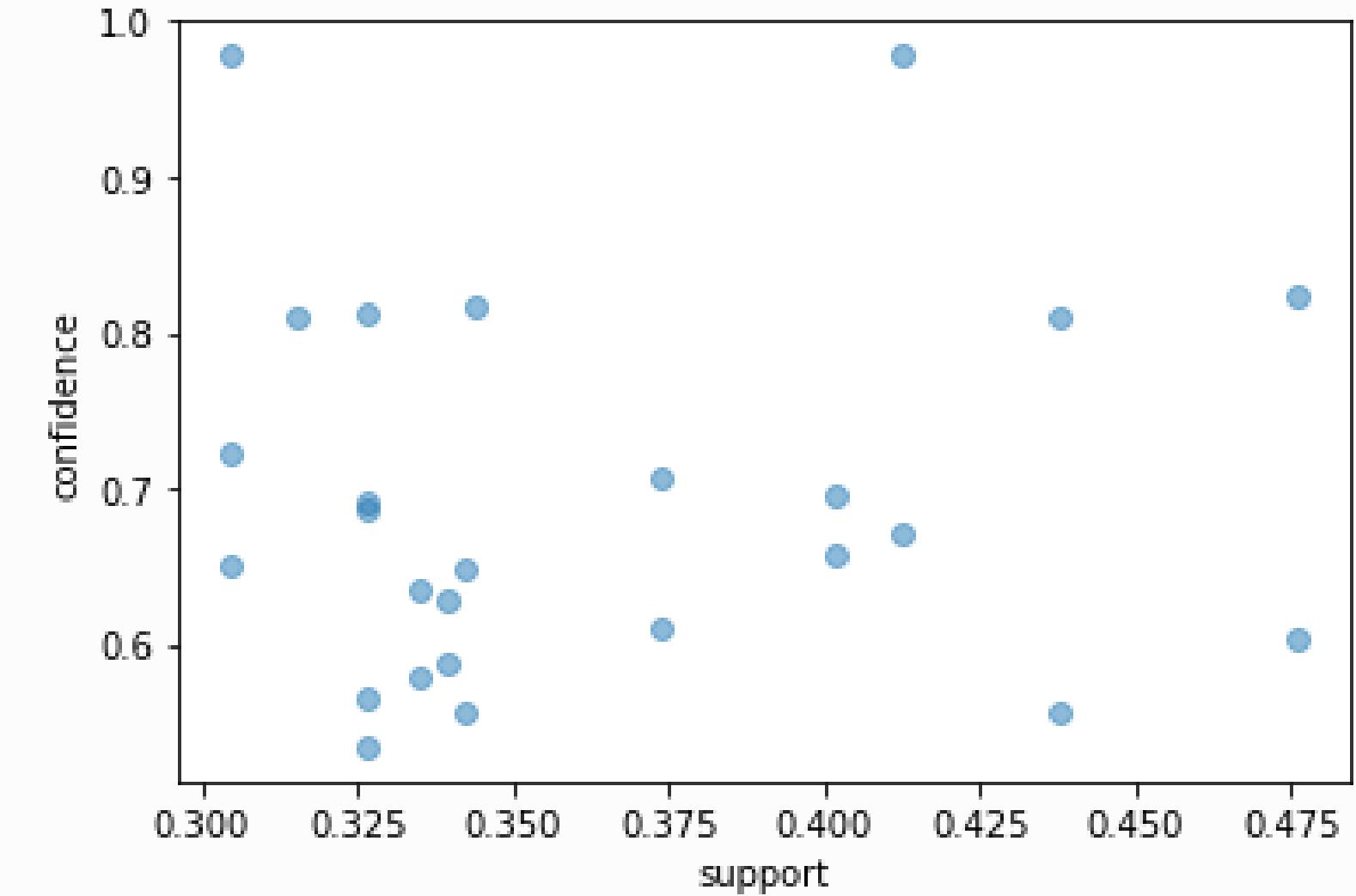


กรองกฎตามเกณฑ์การสนับสนุนและความเชื่อมั่น
rules2 = rules0[(rules0['support'] >= 0.3) & (rules0['confidence'] >= 0.5)]

new_customers



กรองกฎตามเกณฑ์การสนับสนุนและความเชื่อมั่น
`rules5 = rules3[(rules3['support'] >= 0.05)
& (rules3['confidence'] >= 0.1)]`



กรองกฎตามเกณฑ์การสนับสนุนและความเชื่อมั่น
`rules6 = rules3[(rules3['support'] >= 0.3)
& (rules3['confidence'] >= 0.5)]`

จากราฟ existing_customers และ new_customers มีลักษณะของข้อมูลที่คล้ายคลึงกัน
ข้อมูลทั้งสองชุดมีความสัมพันธ์กัน

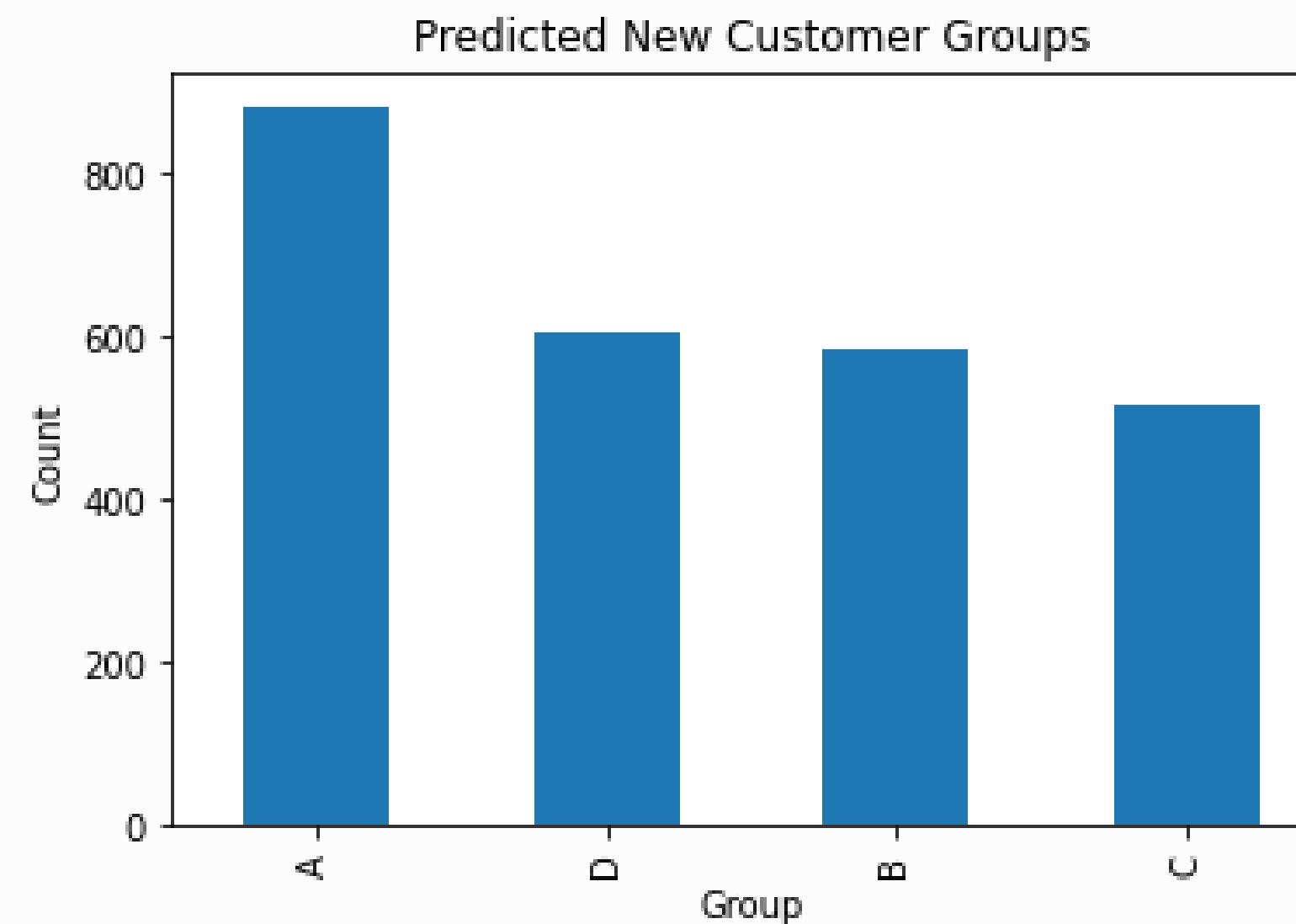
Step4: นำ้งยกลุ่มลูกค้าใหม่ที่เหมาะสม

ใช้ `existing_customers` เพื่อหากลุ่มของ `new_customers` ที่เหมาะสม

3.แบ่งลูกค้าออกเป็นกลุ่ม: แบ่งลูกค้าที่มีอยู่ออกเป็น 4 กลุ่ม (A, B, C, D) ตามลักษณะของข้อมูล ใช้อัลกอริทึมการจัดกลุ่มแบบ k-mean

4. เชื่อมโยงกลุ่มลูกค้าเข้ากับคุณลักษณะ: วิเคราะห์ข้อมูลการจัดกลุ่ม k-mean บันชุดข้อมูล ลูกค้าที่มีอยู่ และ ผู้ที่คาดว่าจะเป็นลูกค้าใหม่ เพื่อแบ่งกลุ่มลูกค้าตามความเหมือนและความต่างใน feature ต่างๆ ของบุคคลทั้งสอง สร้างคลัสเตอร์ 4 กลุ่มตาม ลูกค้าที่มีอยู่ (`existing1_encode`) จากนั้นจะสร้าง dataframes 4 ชุด (`group_a`, `group_b`, `group_c` และ`group_d`) สำหรับลูกค้าแต่ละกลุ่ม

5. นำกลุ่มลูกค้าใหม่ : ใช้ข้อมูล ผู้ที่คาดว่าจะเป็นลูกค้าใหม่ ที่ทำแบ่งลูกค้าออก เป็นกลุ่ม เชื่อมโยงกลุ่มลูกค้าเข้ากับคุณลักษณะแบบเดียวกันกับลูกค้าที่มีอยู่ เพื่อคาดการณ์ว่าลูกค้าใหม่แต่ละรายอยู่ในกลุ่มใด



จากการนิยาม new_customers_predicted กลุ่ม A เป็นกลุ่มลูกค้าใหม่ที่เหมาะสม

Step5: ດູກງານ Association rule ອຳນັ້ນສະໄວ

- การสนับสนุน(support): การสนับสนุนชุดไอเท็มคือสัดส่วนของธุรกรรมในชุดข้อมูลที่มีไอเท็มทั้งหมดในชุดไอเท็มนั้น กล่าวอีกนัยหนึ่ง วัดความถี่หรือความนิยมของชุดรายการในชุดข้อมูล ค่าการสนับสนุนสูงปั่งชี้ว่าชุดรายการเกิดขึ้นบ่อยครั้งในชุดข้อมูล และถ้าว่ามีความสำคัญสำหรับการวิเคราะห์เพิ่มเติม
 - ความเชื่อมั่น(confidence): ความเชื่อมั่นของกฎ $A \rightarrow B$ คือสัดส่วนของการทำธุรกรรมที่มีทั้ง A และ B (เช่น การสนับสนุนของชุดรายการ $A \cup B$) หารด้วย การสนับสนุนของ A กล่าวอีกนัยหนึ่ง การวัดความน่าจะเป็นที่ รายการใน A เป็นของ B เช่นกัน ค่าความเชื่อมั่นสูงปั่งชี้ว่ากฎมีความน่าเชื่อถือและสามารถใช้สำหรับการคาดคะเนได้

Step5: គូករួម Association rule នៃផ្លូវការ

ໃច្ច new_customers_predicted ដើម្បីគូករួម Association rule នៃផ្លូវការ

```
# Generating association
#เลือก metric="support" តាមតម្លៃស្ថិតិថ្មី 0.05
rules rules_predicted = association_rules(frequent_itemsets,
metric="support", min_threshold= 0.05)
#លើកត្រួត support >= 0.3 និង confidence >= 0.5
rules_predicted = rules_predicted[(rules_predicted['support'] >=
0.3) & (rules_predicted['confidence'] >= 0.5)]
#រៀនការណ៍តាម support
rules_predicted = rules_predicted.sort_values(['support'],
ascending=False)
```

เรียกดู rules_predicted

rules_predicted.head(15)

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
182	(Married)	(Work_Experience_Range_0-5)	0.578216	0.787563	0.475859	0.822979	1.044970	0.020478	1.200070
183	(Work_Experience_Range_0-5)	(Married)	0.787563	0.578216	0.475859	0.604218	1.044970	0.020478	1.065698
354	(Work_Experience_Range_0-5)	(Graduated_Yes)	0.787563	0.611433	0.472383	0.599804	0.980980	-0.009159	0.970941
355	(Graduated_Yes)	(Work_Experience_Range_0-5)	0.611433	0.787563	0.472383	0.772584	0.980980	-0.009159	0.934134
435	(Spending_Score_Level1)	(Work_Experience_Range_0-5)	0.615295	0.787563	0.468907	0.762084	0.967649	-0.015677	0.892909
434	(Work_Experience_Range_0-5)	(Spending_Score_Level1)	0.787563	0.615295	0.468907	0.595390	0.967649	-0.015677	0.950803
94	(Male)	(Work_Experience_Range_0-5)	0.539591	0.787563	0.437621	0.811024	1.029789	0.012659	1.124147
95	(Work_Experience_Range_0-5)	(Male)	0.787563	0.539591	0.437621	0.555665	1.029789	0.012659	1.036175
124	(Singer)	(Spending_Score_Level1)	0.421784	0.615295	0.412514	0.978022	1.589516	0.152992	17.504056
125	(Spending_Score_Level1)	(Singer)	0.615295	0.421784	0.412514	0.670433	1.589516	0.152992	1.754471
525	(Family_Size_Range_0-3)	(Work_Experience_Range_0-5)	0.528003	0.787563	0.403631	0.764448	0.970650	-0.012205	0.901869
524	(Work_Experience_Range_0-5)	(Family_Size_Range_0-3)	0.787563	0.528003	0.403631	0.512506	0.970650	-0.012205	0.968211
156	(Married)	(Graduated_Yes)	0.578216	0.611433	0.402086	0.695391	1.137313	0.048546	1.275625
157	(Graduated_Yes)	(Married)	0.611433	0.578216	0.402086	0.657612	1.137313	0.048546	1.231891
358	(Family_Size_Range_0-3)	(Graduated_Yes)	0.528003	0.611433	0.373890	0.708120	1.158132	0.051051	1.331256

กฎ Association rule ที่นำสนใจ โดยดูจากเกณฑ์ support ≥ 0.40

- คนที่แต่งงานแล้ว ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี (ที่เลือก Married \rightarrow Work_Experience_Range_0-5 เพราะมีค่า confidence สูงกว่า Work_Experience_Range_0-5 \rightarrow Married), (ค่า confidence = 0.822979)
- คนที่จบการศึกษาแล้ว ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี (ค่า confidence = 0.772584)
- คนที่มีคะแแนค่าใช้จ่ายต่ำ ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี (ค่า confidence = 0.762084)
- เพศชาย ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี (ค่า confidence = 0.811024)
- สถานะภาพโสด ส่วนใหญ่มีคะแแนค่าใช้จ่ายต่ำ (ค่า confidence = 0.978022)
- คนที่มีขนาดครอบครัว 1-3 คน ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี(ค่า confidence = 0.764448)
- คนที่แต่งงานแล้ว ส่วนใหญ่จบการศึกษาแล้ว (ค่า confidence = 0.695391)

ข้อมูล support ≥ 0.40 ถ้าเรียงตามค่า confidence จะได้ว่า

- คนที่มีสถานะภาพโสด ส่วนใหญ่มีคะแแนค่าใช้จ่ายต่ำ
- คนที่แต่งงานแล้ว ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี
- เพศชาย ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี
- คนที่จบการศึกษาแล้ว ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี
- คนที่มีขนาดครอบครัว 1-3 คน ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี
- คนที่มีคะแแนค่าใช้จ่ายต่ำ ส่วนใหญ่เป็นผู้มีประสบการณ์การทำงาน 0-5 ปี
- คนที่แต่งงานแล้ว ส่วนใหญ่จบการศึกษาแล้ว



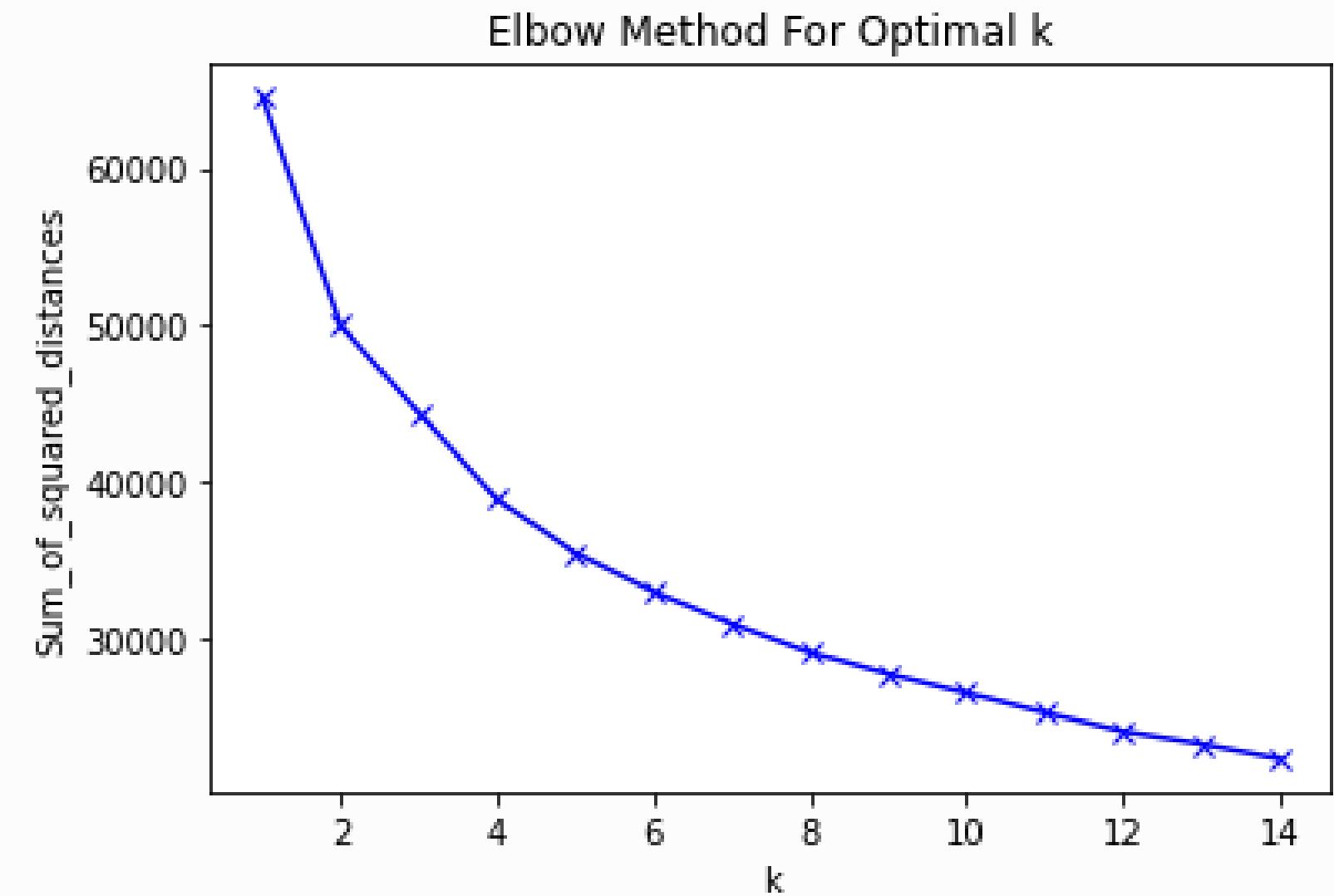
Clustering

K-Means

1. ปรับ encoder ให้พอดีกับข้อมูลคอลัมน์และแปลงข้อมูลเป็นรูปแบบตัวเลข
2. ปรับขนาดข้อมูลโดยใช้ StandardScaler
3. ใช้ Elbow method เพื่อหาจำนวน clusters ที่ดีที่สุด และ Plot the Elbow method
4. กำหนด cluster labels
5. คำนวณจำนวนการกำหนดที่ถูกต้อง และคำนวณความแม่นยำเป็นเปอร์เซ็นต์

K-Means

Elbow Method แสดงจุดที่เหมาะสม
สมของจำนวน cluster
โดยจุดที่ดีที่สุดที่เลือกคือ $k = 4$



K-Means

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1	Segmentation	cnt_IDS	new_Segmentation
0	462809	Male	No	22	No	Healthcare	1.0	Low	4.0	Cat_4	D	2268	D
1	462643	Female	Yes	38	Yes	Engineer	0.0	Average	3.0	Cat_4	A	1972	C
2	466315	Female	Yes	67	Yes	Engineer	1.0	Low	1.0	Cat_6	B	1858	A
3	461735	Male	Yes	67	Yes	Lawyer	0.0	High	2.0	Cat_6	B	1858	A
4	462669	Female	Yes	40	Yes	Entertainment	0.0	High	6.0	Cat_6	A	1972	C
...
8063	464018	Male	No	22	No	Unknown	0.0	Low	7.0	Cat_1	D	2268	D
8064	464685	Male	No	35	No	Executive	3.0	Low	4.0	Cat_4	D	2268	D
8065	465406	Female	No	33	Yes	Healthcare	1.0	Low	1.0	Cat_6	D	2268	A
8066	467299	Female	No	27	Yes	Healthcare	1.0	Low	4.0	Cat_6	B	1858	D
8067	461879	Male	Yes	37	Yes	Executive	0.0	Average	3.0	Cat_4	B	1858	C

K-Means

```
[]: # Calculate the number of correct predictions  
correct_predictions = sum(df_train['Segmentation'] == df_train['new_Segmentation'])  
  
# Calculate the total number of predictions  
total_predictions = df_train.shape[0]  
  
# Calculate the accuracy as a percentage  
accuracy = (correct_predictions / total_predictions) * 100  
print("Accuracy for K-Means: {:.2f}%".format(accuracy))  
  
Accuracy for K-Means: 44.94%
```

ค่าความแม่นยำของ K-Means Clustering Model ที่ใช้ในการทำนายกลุ่มลูกค้า (new_Segmentation) เท่ากับ 44.94 %

นั่นคือ K-Means Model สามารถจัดกลุ่มลูกค้าที่คล้ายคลึงกันเข้าด้วยกันใน cluster เดียวกัน ได้ไม่ดีนัก เนื่องจากบุคคลต่างๆ น้อยกว่าครึ่งหนึ่งถูกจัดกลุ่มเข้าด้วยกันอย่างถูกต้อง

Members

1. นางสาวกรณิศ เมืองเก่า 633020437-8
2. นางสาวสวิตา สมศรี 633020448-3
3. นางสาวอภิสรา ปราบนอก 633020450-6
4. นางสาวสุพรรษา ประกอบบุญ 633021026-4



THANK YOU FOR YOUR ATTENTION