เรื่อง Prediction and Recommend Model

จัดทำโดย

นาย กิตติพงศ์ พวงสินธ์ รหัสนักศึกษา 66070016

นาย ศุภณัฐ จันทรสาขา รหัสนักศึกษา 66070196

นาย สิรภพ สรรค์ศิลา รหัสนักศึกษา 66070204

นาย ธีรศานต์ ชูเชิด รหัสนักศึกษา 66070274

นาย วรวุฒิ มหาทอง รหัสนักศึกษา 66070307

เสนอ

ดร. ปาณิตา ธูสรานนท์

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา

06026211 Applied Machine Learning

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2568

คำนำ

บริษัท ABC ดำเนินธุรกิจเกี่ยวกับการมอบสิทธิประโยชน์และกิจกรรมพิเศษต่างๆ แก่ลูกค้าสมาชิก ลูกค้าสามารถสะสมคะแนน (Point) จากยอดใช้จ่าย เพื่อนำไปแลกสิทธิประโยชน์ เช่น ส่วนลดร้านค้า, โรงแรม, โรงพยาบาล หรือใช้ในการเข้าร่วมกิจกรรมที่สอดคล้องกับไลฟ์สไตล์ของตนเอง กิจกรรมเหล่านี้มีความหลากหลาย เช่น คอนเสิร์ต, กิจกรรมท่องเที่ยว, เวิร์กช็อปทำอาหาร, การจัดดอกไม้ หรือกิจกรรมสำหรับครอบครัว

โดยปกติบริษัทจะใช้ข้อมูลความสนใจที่ลูกค้าระบุไว้ (lifestyle และ favorite) ในการพิจารณาเชิญลูกค้าเข้าร่วมกิจกรรม อย่างไรก็ตาม บริษัทประสบปัญหาในการแนะนำกิจกรรมให้กับลูกค้ากลุ่มหนึ่ง ที่ไม่ได้ระบุข้อมูลดังกล่าวไว้ โครงการนี้จึงจัดทำขึ้นเพื่อวิเคราะห์ข้อมูลสมาชิกและสร้างโมเดลคาดการณ์ (Prediction Model) เพื่อช่วยให้บริษัทสามารถแนะนำกิจกรรมที่เหมาะสมให้กับลูกค้ากลุ่มนี้ได้ แม้จะไม่มีข้อมูล lifestyle และ favorite

สารบัญ

ขั้นตอนที่ 1. วิเคราะห์ความต้องการ (Business Understanding)

1.1 ปัญหาทางธุรกิจ

ในการดำเนินกลยุทธ์การตลาดและการรักษาความสัมพันธ์กับลูกค้าบริษัท ABC จำเป็นต้องสามารถแนะนำกิจกรรมและสิทธิประโยชน์ที่ตรงกับความสนใจของสมาชิกแต่ละคน อย่างไรก็ตาม พบปัญหาสำคัญ คือ บริษัท ABC ไม่สามารถแนะนำกิจกรรมที่สอดคล้องกับความสนใจของกลุ่มลูกค้าที่ไม่ได้ระบุ favorite และ lifestyle

โดยสถานการณ์นี้ถือเป็นอุปสรรคสำคัญต่อการแนะนำกิจกรรมให้กับสมาชิกกลุ่มดังกล่าว ทำให้บริษัทสูญเสียโอกาสในการมีส่วนร่วมกับลูกค้ากลุ่มนี้

1.2 วัตถุประสงค์ของโครงงาน

เพื่อแก้ปัญหาดังกล่าว โครงงานนี้จึงมีวัตถุประสงค์เพื่อสร้างโมเดล Machine Learning สำหรับการจำแนกประเภท (Classification) โดยอาศัยข้อมูลประชากรของลูกค้า เช่น อายุ, เพศ, อาชีพ, การศึกษา และเขตที่อยู่อาศัย เพื่อทำนาย กิจกรรม (activity) ที่ลูกค้ากลุ่มนี้ (กลุ่มที่ไม่ได้ระบุความชอบ) น่าจะสนใจเข้าร่วมมากที่สุด

ขั้นตอนที่ 2. การจัดการข้อมูล (Data Preparation)

2.1 ภาพรวมชุดข้อมูล (Data Overview)

ชุดข้อมูลที่ใช้ในการวิเคราะห์คือ MemberActivity.csv ซึ่งมีข้อมูลทั้งหมด 4,124 แถว และ 10 คอลัมน์ โดยสามารถแบ่งประเภทข้อมูลได้ดังนี้

1. **ข้อมูลประชากร (Demographic)** ได้แก่ Age (อายุ), gender (เพศ), occupation (อาชีพ), education (ระดับการศึกษา), zone (เขต/พื้นที่อยู่อาศัย)
2. **ข้อมูลความชอบ (Preference)** ได้แก่ lifestyle (ไลฟ์สไตล์ที่ลูกค้าระบุ), favorite (กิจกรรมโปรดที่ลูกค้าระบุ)
3. **ข้อมูลพฤติกรรม (Behavioral)** ได้แก่ activity (กิจกรรมที่ลูกค้าเข้าร่วมจริง), month (เดือนที่เกิด), place (สถานที่จัดกิจกรรม)

ตัวแปรเป้าหมายของโครงการนี้คือ activity

2.2 การสำรวจและทำความสะอาดข้อมูล (Data Cleaning)

จากการสำรวจข้อมูลในไฟล์นำเสนอและโค้ด พบประเด็นที่ต้องจัดการดังนี้

1. พบข้อมูลอายุที่ผิดปกติ คือ ค่าอายุติดลบ (เช่น -64) และ อายุน้อยกว่า 18 ปีแต่จบการศึกษาปริญญาตรี จำเป็นต้องมีการลบแถวข้อมูลนี้ทิ้งหรือแทนที่ด้วยค่าที่เหมาะสม

```
member_df[member_df['Age'] < 0]
```

	Age	gender	occupation	education	lifestyle	favorite	zone	activity	month	place
1201	-64	female	employee	bachelor	family	family	Khlong Sam Wa	stageplay	9	Rachadalai Theatre

```
member_df[member_df['Age'] > 100]
```

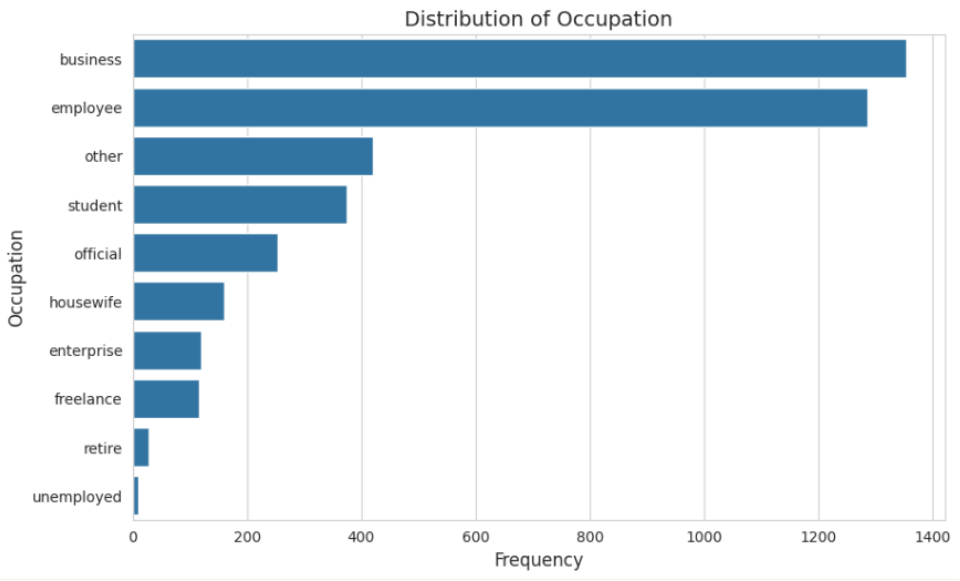
	Age	gender	occupation	education	lifestyle	favorite	zone	activity	month	place
1194	145	male	business	graduate	family	family	Pom Prap Sattru Phai	stageplay	10	Rachadalai Theatre

```
young = member_df[(member_df['Age'] < 18) & (member_df['education'].isin(['bachelor', 'graduate']))]
young
```

	Age	gender	occupation	education	lifestyle	favorite	zone	activity	month	place
94	4	male	other	bachelor	family	family	Min Buri	journey	10	The Mall Bang Kapi
111	14	female	employee	bachelor	family	family	Din Daeng	journey	11	The Mall Bang Kapi
135	11	male	business	bachelor	family	family	Saphan Sung	journey	11	The Mall Bang Kapi
191	13	female	employee	bachelor	family	family	Ratchathewi	journey	10	The Mall Bang Kapi
197	15	female	official	bachelor	family	travel	Phasi Charoen	journey	11	The Mall Bang Kapi
207	10	male	employee	bachelor	family	family	Ratchathewi	journey	10	The Mall Bang Kapi
304	12	female	business	bachelor	family	family	Bang Khen	camping	3	Khao Yai
309	11	female	business	bachelor	family	travel	Khlong Sam Wa	camping	3	Khao Yai
376	3	male	employee	bachelor	family	family	Watthana	movie	3	Office
380	6	male	employee	graduate	family	family	Bang Khen	movie	3	Office

ปรับคุณภาพข้อมูลตัวแปรอายุให้สมเหตุสม จากนั้นทดแทนค่าที่หายไปด้วยค่ากลางแบบ และจัดช่วงอายุเป็นกลุ่ม เพื่อให้พร้อมต่อการทำกฎความสัมพันธ์หรือระบบแนะนำ

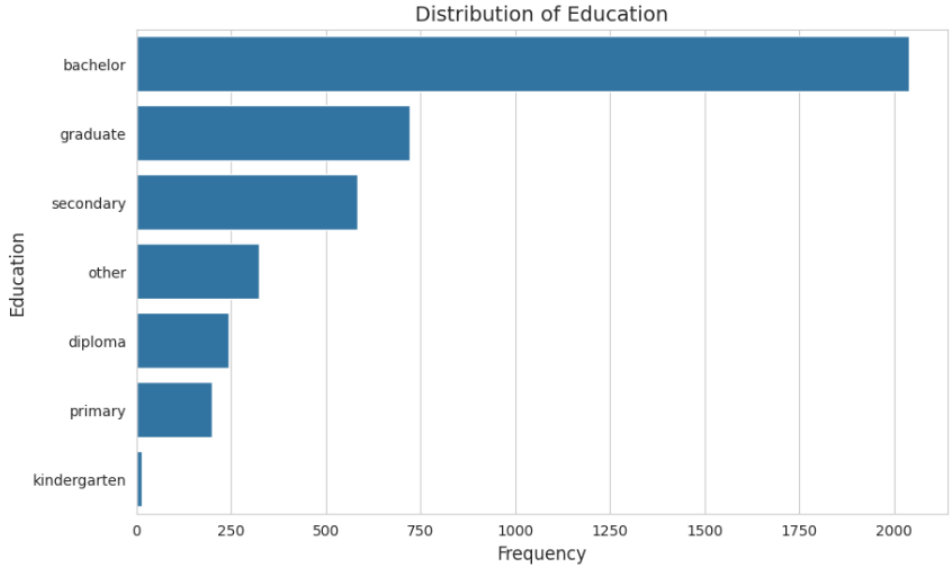
2. **Occupation** ข้อมูลอาชีพส่วนใหญ่ 3 อันดับแรก คือ business, employee และ other




```
member_df['occupation'].value_counts()
```

occupation	count
business	1354
employee	1287
other	420
student	374
official	254
housewife	161
enterprise	120
freelance	116
retire	29
unemployed	9

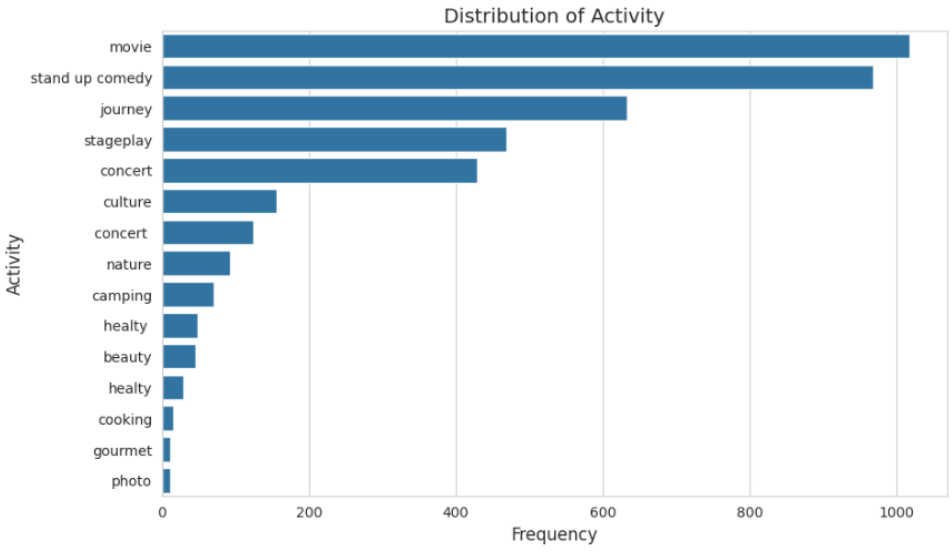
3. **Education** ข้อมูลการศึกษาส่วนใหญ่ 3 อันดับแรก คือ bachelor, graduate และ secondary




```
member_df['education'].value_counts()
```

education	count
bachelor	2039
graduate	720
secondary	584
other	323
diploma	244
primary	199
kindergarten	15

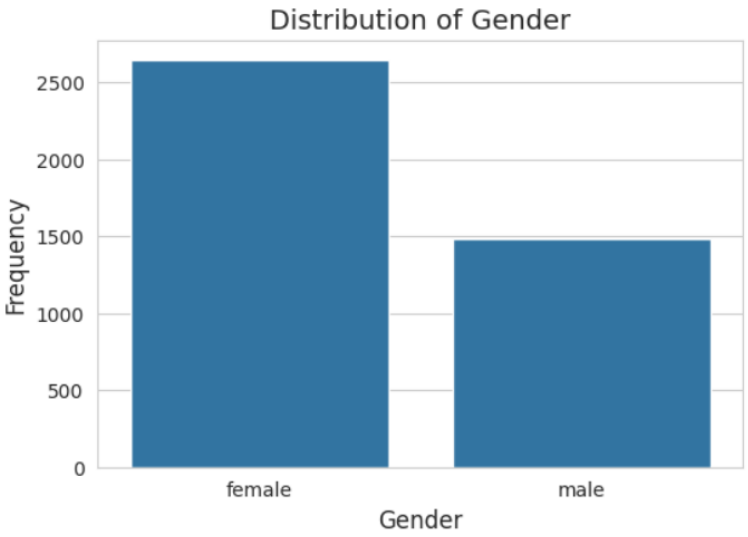
4. **Activity** ข้อมูลกิจกรรมที่เข้าร่วมส่วนใหญ่ 3 อันดับแรก คือ movie, concert และ travel




```
member_df['activity'].value_counts()
```

activity	count
movie	1018
stand up comedy	968
journey	634
stageplay	469
concert	429
culture	156
concert	125
nature	93
camping	70
healty	48
beauty	46
healty	29
cooking	16
gourmet	12
photo	11

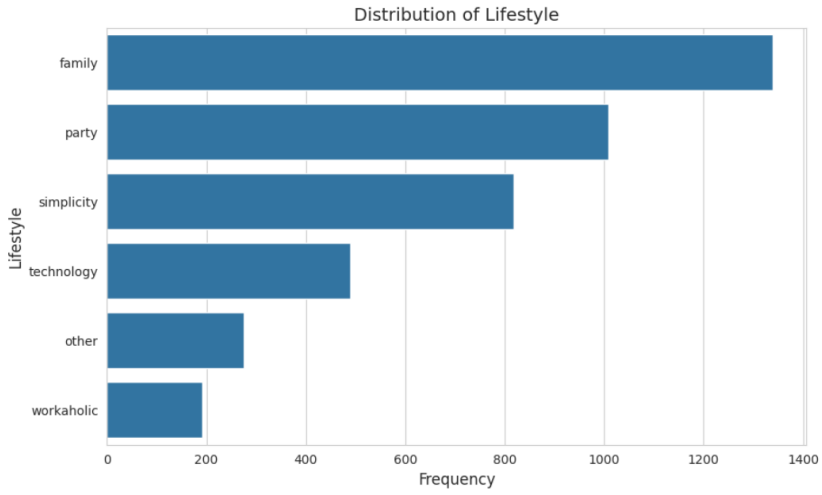
5. Gender ข้อมูลส่วนใหญ่ คือ female




```
member_df['gender'].value_counts()
```

gender	count
female	2643
male	1481

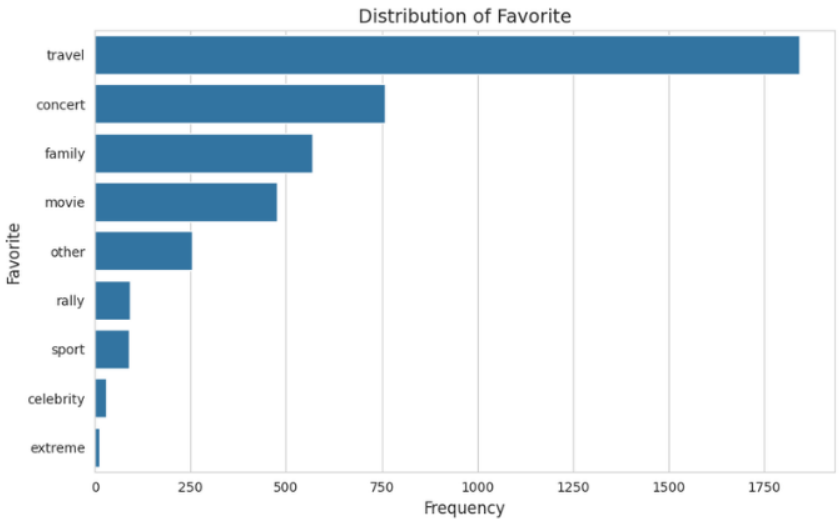
6. Lifestyle ข้อมูลไลฟ์สไตล์ส่วนใหญ่ 3 อันดับแรก คือ other, family และ simplicity




```
member_df['lifestyle'].value_counts()
```

lifestyle	count
family	1340
party	1009
simplicity	818
technology	490
other	276
workaholic	191

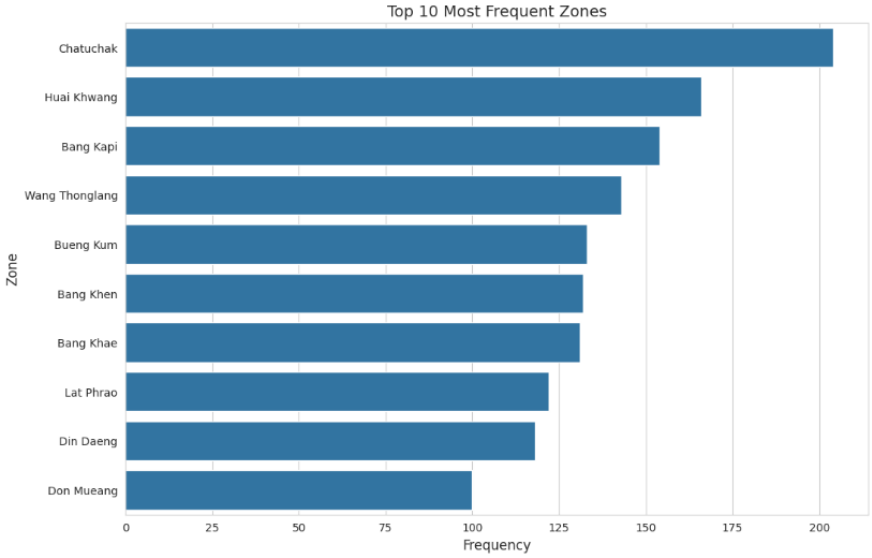
7. **Favorite** ข้อมูลกิจกรรมโปรดส่วนใหญ่ 3 อันดับแรก คือ other, travel และ movie




```
member_df['favorite'].value_counts()
```

favorite	count
travel	1842
concert	758
family	570
movie	476
other	254
rally	92
sport	89
celebrity	30
extreme	13

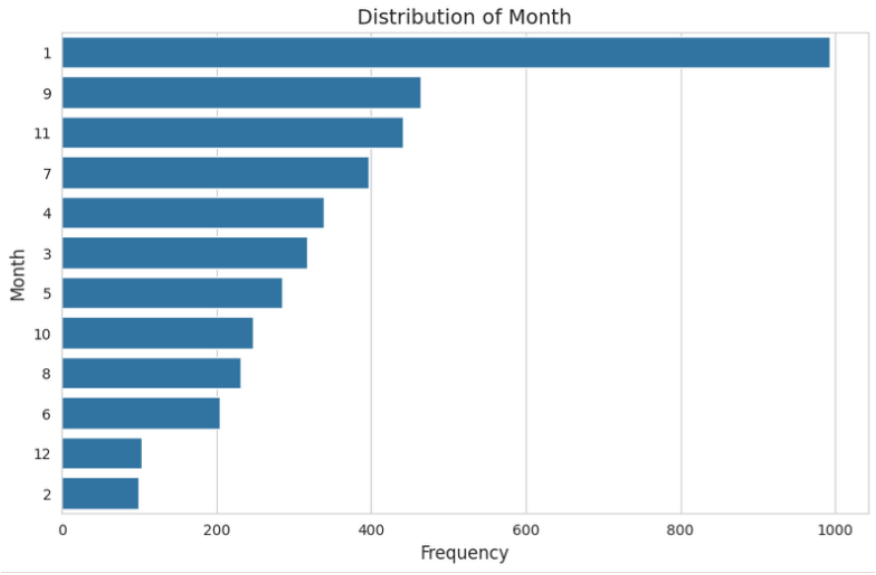
8. **Zone** ข้อมูลเขตที่อยู่อาศัยส่วนใหญ่ 3 อันดับแรก คือ Chatuchak, Bang Kapi และ Lat Phrao




```
member_df['zone'].value_counts()
```

zone	count
Chatuchak	204
Huai Khwang	166
Bang Kapi	154
Wang Thonglang	143
Bueng Kum	133
Bang Khen	132
Bang Khae	131
Lat Phrao	122
Din Daeng	118
Don Mueang	100
Sathon	91

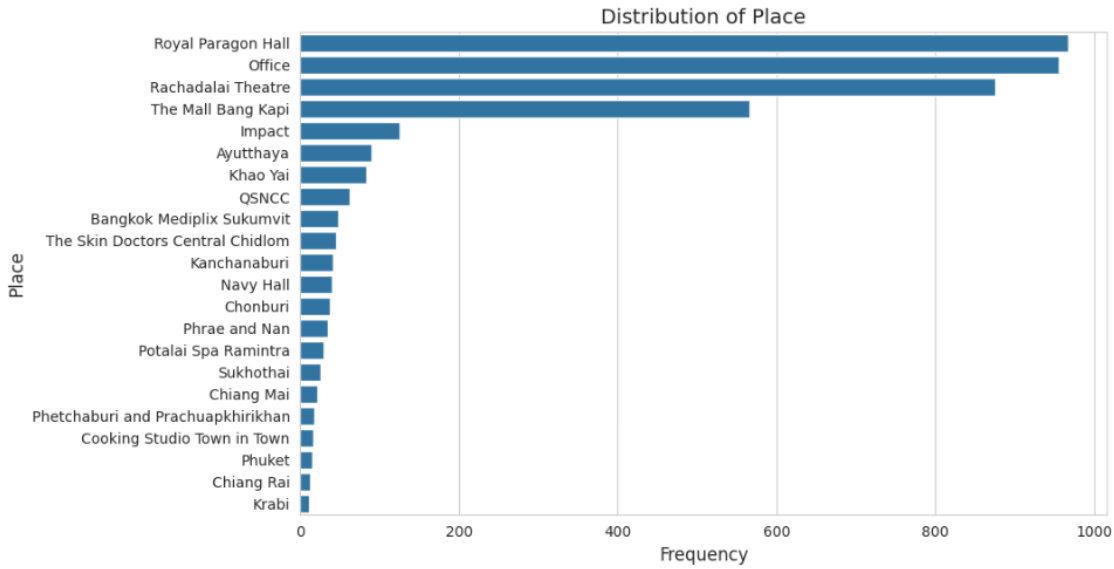
9. **Month** ข้อมูลเดือนส่วนใหญ่ 3 อันดับแรก คือ **9 (เดือนกันยายน)**, **1 (เดือนมกราคม)** และ **10 (เดือนตุลาคม)**




```
member_df['month'].value_counts()
```

month	count
1	993
9	464
11	441
7	397
4	339
3	318
5	285
10	248
8	232
6	204
12	103
2	100

10. **Place** ข้อมูลสถานที่จัดกิจกรรมส่วนใหญ่ 3 อันดับแรก คือ **Paragon Cineplex**, **Major Cineplex Ratchayothin** และ **Impact Arena**



2.3การเตรียมข้อมูล (Data Preparation)

1. รวมข้อมูล ของ Activity ที่ซ้ำกัน

2. ปรับอายุที่ < 0 และ เกิน > 100 ให้อยู่ใน median

```
member_df['Age'].describe()
```

	Age
count	4123.0
mean	41.17366
std	14.243683
min	-64.0
25%	33.0
50%	43.0
75%	51.0
max	145.0

```
df['Age'].describe()
```

	Age
count	4124.000000
mean	42.655432
std	12.379513
min	12.000000
25%	35.000000
50%	44.000000
75%	51.000000
max	81.000000

ขั้นตอนที่ 3 การออกแบบและสร้างโมเดล (Modeling)

ขั้นตอนนี้สร้างและประเมินแบบจำลองเพื่อทำนายค่าของ Favorite และ Lifestyle จากข้อมูลสมาชิก โดยใช้ชุดข้อมูลที่เตรียมไว้เพื่อฝึกและทดสอบโมเดล เพื่อเลือกแบบจำลองที่ให้ความแม่นยำสูงสุด

3.1 การแบ่งข้อมูล (Train-Test Split)

การทำ Test-Train Split สำหรับการทำนายตัวแปร **Favorite**

เลือกใช้ตัวแปรสำคัญ 5 ตัว ได้แก่ Age, gender, occupation, education และ activity

จากนั้นทำการแบ่งข้อมูลออกเป็น 2 ส่วน ได้แก่

1. Training set (favorite ≠ other) ใช้สำหรับฝึกโมเดล จำนวน 3,096 แถว
2. Testing set (favorite = other) ใช้สำหรับทำนายภายหลัง จำนวน 774 แถว

จากนั้นทำการแปลงตัวแปรสำคัญ ด้วย **One-Hot Encoding** เพื่อให้โมเดลสามารถประมวลผลได้อย่างถูกต้อง โดยปรับให้คอลัมน์ของชุดฝึกและชุดทดสอบมีจำนวนตรงกัน

ผลลัพธ์หลังเข้ารหัสคือ

```
รูปร่างหลัง One-Hot Encoding:
  - X_train: (3096, 29)
  - X_test: (774, 29)
```

1. Training set: 3,096 แถว, 29 ตัวแปร
2. Testing set: 774 แถว, 29 ตัวแปร
3. จะได้อัตราส่วนที่ใช้แบ่งคือ 80:20

การทำ Test-Train Split สำหรับการทำนายตัวแปร **Lifestyle**

เลือกใช้ตัวแปรสำคัญ 5 ตัว ได้แก่ Age, gender, occupation, education, และ activity

ข้อมูลถูกแบ่งออกเป็น 2 ส่วน คือ

1. Training set (lifestyle ≠ other) สำหรับฝึกโมเดล
2. Testing set (lifestyle = other) สำหรับนำมาทำนายภายหลัง

จากนั้นทำการแปลงตัวแปรสำคัญ ด้วย **One-Hot Encoding** เพื่อให้โมเดลสามารถประมวลผลได้อย่างถูกต้อง โดยปรับให้คอลัมน์ของชุดฝึกและชุดทดสอบมีจำนวนตรงกัน

ผลลัพธ์หลังการเตรียมข้อมูลพบว่า

```
รูปร่างหลัง One-Hot Encoding:
  - X_train: (3078, 29)
  - X_test: (770, 29)
```

1. Training set: 3,096 แถว, 29 ตัวแปร
2. Testing set: 774 แถว, 29 ตัวแปร
3. จะได้อัตราส่วนที่ใช้แบ่งคือ 80:20

3.2 การทดสอบโมเดล (Model Selection)

โครงงานนี้ทำการทดลองสร้างโมเดลจำแนกประเภทหลายรูปแบบเพื่อเปรียบเทียบประสิทธิภาพและเลือกโมเดลที่ดีที่สุด โดยโมเดลที่มีการเรียกใช้งานจากไลบรารี sklearn

การทดสอบโมเดลของ Favorite

1. Logistic Regression โมเดลทางสถิติที่ใช้สำหรับการจำแนกประเภท

```
# ฝึกโมเดล Logistic Regression
log_model_fav = LogisticRegression(max_iter=1000, random_state=42, n_jobs=-1)
log_model_fav.fit(X_train_fav_encoded, y_train_fav)
y_pred_log_fav = log_model_fav.predict(X_test_fav_encoded)

acc_log_fav = accuracy_score(y_test_fav, y_pred_log_fav)
print(f"Accuracy: {acc_log_fav:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_fav, y_pred_log_fav,
                            target_names=le_fav.classes_, zero_division=0))
```

```
Accuracy: 0.4599

Classification Report:
              precision    recall  f1-score   support

   celebrity       0.00      0.00      0.00         6
     concert       0.22      0.10      0.14       152
     extreme       0.00      0.00      0.00         3
      family       0.60      0.03      0.05       114
       movie       0.00      0.00      0.00        95
       rally       0.00      0.00      0.00        18
       sport       0.00      0.00      0.00        18
      travel       0.49      0.92      0.64       368

    accuracy                           0.46       774
   macro avg       0.16      0.13      0.10       774
weighted avg       0.36      0.46      0.34       774
```

2. Decision Tree โมเดลต้นไม้ตัดสินใจ

```
# ฝึกโมเดล Decision Tree
dt_model_fav = DecisionTreeClassifier(random_state=42, max_depth=10)
dt_model_fav.fit(X_train_fav_encoded, y_train_fav)
y_pred_dt_fav = dt_model_fav.predict(X_test_fav_encoded)

acc_dt_fav = accuracy_score(y_test_fav, y_pred_dt_fav)
print(f"Accuracy: {acc_dt_fav:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_fav, y_pred_dt_fav,
                            target_names=le_fav.classes_, zero_division=0))
```

```
Accuracy: 0.4264

Classification Report:
              precision    recall  f1-score   support

   celebrity       0.00      0.00      0.00         6
     concert       0.31      0.22      0.25       152
     extreme       0.00      0.00      0.00         3
      family       0.29      0.12      0.17       114
       movie       0.20      0.11      0.14        95
       rally       0.00      0.00      0.00        18
       sport       0.00      0.00      0.00        18
      travel       0.49      0.74      0.59       368

    accuracy                           0.43       774
   macro avg       0.16      0.15      0.14       774
weighted avg       0.36      0.43      0.37       774
```

3. Random Forest การรวมกันของ Decision Trees หลายๆ ต้นเพื่อเพิ่มความแม่นยำ

```
# ฝึกโมเดล Random Forest
rf_model_fav = RandomForestClassifier(n_estimators=100, random_state=42,
                                      max_depth=15, n_jobs=-1)
rf_model_fav.fit(X_train_fav_encoded, y_train_fav)
y_pred_rf_fav = rf_model_fav.predict(X_test_fav_encoded)

acc_rf_fav = accuracy_score(y_test_fav, y_pred_rf_fav)
print(f"Accuracy: {acc_rf_fav:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_fav, y_pred_rf_fav,
                            target_names=le_fav.classes_, zero_division=0))
```

```
Accuracy: 0.4470

Classification Report:
              precision    recall  f1-score   support

   celebrity       0.00      0.00      0.00         6
     concert       0.37      0.19      0.25       152
     extreme       0.00      0.00      0.00         3
      family       0.26      0.11      0.15       114
       movie       0.20      0.11      0.14        95
       rally       0.00      0.00      0.00        18
       sport       0.50      0.06      0.10        18
      travel       0.50      0.80      0.61       368

    accuracy                           0.45       774
   macro avg       0.23      0.16      0.16       774
weighted avg       0.38      0.45      0.38       774
```

4. Gradient Boosting โมเดลที่สร้างต้นไม้ทีละต้นเพื่อแก้ไขข้อผิดพลาดของต้นก่อนหน้า

```
# ฝึกโมเดล Gradient Boosting
GR_model_fav = GradientBoostingClassifier(n_estimators=100,
                                          learning_rate=0.1,
                                          max_depth=3,
                                          random_state=42)
GR_model_fav.fit(X_train_fav_encoded, y_train_fav)
y_pred_GR_fav = rf_model_fav.predict(X_test_fav_encoded)

acc_GR_fav = accuracy_score(y_test_fav, y_pred_GR_fav)
print(f"Accuracy: {acc_GR_fav:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_fav, y_pred_GR_fav,
                            target_names=le_fav.classes_, zero_division=0))
```

```
Accuracy: 0.4470

Classification Report:
              precision    recall  f1-score   support

   celebrity       0.00      0.00      0.00         6
     concert       0.37      0.19      0.25       152
     extreme       0.00      0.00      0.00         3
      family       0.26      0.11      0.15       114
       movie       0.20      0.11      0.14        95
       rally       0.00      0.00      0.00        18
       sport       0.50      0.06      0.10        18
      travel       0.50      0.80      0.61       368

    accuracy                           0.45       774
   macro avg       0.23      0.16      0.16       774
weighted avg       0.38      0.45      0.38       774
```

เปรียบเทียบโมเดล

```
============================================================
📊 สรุปผลการเปรียบเทียบโมเดล (FAVORITE)
============================================================
Logistic Regression      : 0.4599
Random Forest            : 0.4470
Gradient Boosting        : 0.4470
Decision Tree            : 0.4264

🏆 Best Model: Logistic Regression (0.4599)
```

การทดสอบโมเดลของ Lifestyle

1. Logistic Regression โมเดลทางสถิติที่ใช้สำหรับการจำแนกประเภท

```
# ฝึกโมเดล Logistic Regression
log_model_life = LogisticRegression(max_iter=1000, random_state=42, n_jobs=-1)
log_model_life.fit(X_train_life_encoded, y_train_life)
y_pred_log_life = log_model_life.predict(X_test_life_encoded)

acc_log_life = accuracy_score(y_test_life, y_pred_log_life)
print(f"Accuracy: {acc_log_life:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_life, y_pred_log_life,
                           target_names=le_life.classes_, zero_division=0))
```

```
Accuracy: 0.3896

Classification Report:
              precision    recall  f1-score   support

      family       0.41      0.69      0.51       268
       party       0.36      0.43      0.39       202
  simplicity       0.37      0.10      0.16       164
  technology       0.37      0.11      0.17        98
  workaholic       0.00      0.00      0.00        38

    accuracy                           0.39       770
   macro avg       0.30      0.27      0.25       770
weighted avg       0.36      0.39      0.34       770
```

2. Decision Tree โมเดลต้นไม้ตัดสินใจ

```
# ฝึกโมเดล Decision Tree
dt_model_life = DecisionTreeClassifier(random_state=42, max_depth=10)
dt_model_life.fit(X_train_life_encoded, y_train_life)
y_pred_dt_life = dt_model_life.predict(X_test_life_encoded)

acc_dt_life = accuracy_score(y_test_life, y_pred_dt_life)
print(f"Accuracy: {acc_dt_life:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_life, y_pred_dt_life,
                           target_names=le_life.classes_,
                           zero_division=0))
```

```
Accuracy: 0.4000

Classification Report:
              precision    recall  f1-score   support

      family       0.42      0.74      0.54       268
       party       0.40      0.32      0.35       202
  simplicity       0.38      0.18      0.24       164
  technology       0.29      0.14      0.19        98
  workaholic       0.17      0.05      0.08        38

    accuracy                           0.40       770
   macro avg       0.33      0.29      0.28       770
weighted avg       0.38      0.40      0.36       770
```

3. Random Forest การรวมกันของ Decision Trees หลายต้น เพื่อเพิ่มความแม่นยำ

```python
# ฝึกโมเดล Random Forest
rf_model_life = RandomForestClassifier(n_estimators=100, random_state=42,
                                       max_depth=15, n_jobs=-1)
rf_model_life.fit(X_train_life_encoded, y_train_life)
y_pred_rf_life = rf_model_life.predict(X_test_life_encoded)

acc_rf_life = accuracy_score(y_test_life, y_pred_rf_life)
print(f"Accuracy: {acc_rf_life:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_life, y_pred_rf_life,
                            target_names=le_life.classes_, zero_division=0))
```

```
Accuracy: 0.3922

Classification Report:
              precision    recall  f1-score   support

      family       0.44      0.68      0.53       268
       party       0.36      0.35      0.35       202
  simplicity       0.33      0.19      0.24       164
  technology       0.28      0.16      0.21        98
  workaholic       0.27      0.08      0.12        38

    accuracy                           0.39       770
   macro avg       0.34      0.29      0.29       770
weighted avg       0.37      0.39      0.36       770
```

4. Gradient Boosting โมเดลที่สร้างต้นไม้ทีละต้นเพื่อแก้ไขข้อผิดพลาดของต้นก่อนหน้า

```python
# ฝึกโมเดล GradientBoosting
gr_model_life = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1,
                                           max_depth=3, random_state=42)
gr_model_life.fit(X_train_life_encoded, y_train_life)
y_pred_gr_life = gr_model_life.predict(X_test_life_encoded)

acc_gr_life = accuracy_score(y_test_life, y_pred_gr_life)
print(f"Accuracy: {acc_gr_life:.4f}")

print("\nClassification Report:")
print(classification_report(y_test_life, y_pred_gr_life,
                            target_names=le_life.classes_, zero_division=0))
```

```
Accuracy: 0.4026

Classification Report:
              precision    recall  f1-score   support

      family       0.41      0.79      0.54       268
       party       0.42      0.35      0.38       202
  simplicity       0.31      0.10      0.16       164
  technology       0.26      0.10      0.15        98
  workaholic       1.00      0.05      0.10        38

    accuracy                           0.40       770
   macro avg       0.48      0.28      0.27       770
weighted avg       0.40      0.40      0.35       770
```

เปรียบเทียบโมเดล

```
GradientBoosting         : 0.4026
Decision Tree            : 0.4000
Random Forest            : 0.3922
Logistic Regression      : 0.3896

🏆 Best Model: GradientBoosting (0.4026)
```

3.3 การเติมค่าข้อมูลที่ขาดหาย (Imputed Data)

หลังจากเลือกโมเดลที่มีประสิทธิภาพดีที่สุดแล้ว ขั้นตอนนี้ได้นำโมเดลดังกล่าวมาใช้ทำนายและเติมค่า Favorite ที่ขาดหาย (favorite = other) ในชุดข้อมูลจริง เพื่อให้ได้ข้อมูลสมบูรณ์สำหรับการนำไปวิเคราะห์ต่อไป

```
ส่วนที่ 5: Impute ค่า 'other' ใน FAVORITE
กำลังทำนาย 254 แถวที่มี favorite = 'other'...

Imputation สำเร็จ!

การกระจายของ favorite หลัง Impute:
favorite
travel       2067
concert       780
family        572
movie         481
rally          92
sport          89
celebrity      30
extreme        13
Name: count, dtype: int64

ตรวจสอบ: จำนวน 'other' ที่เหลือ = 0 (ควรเป็น 0)

💾 บันทึกไฟล์: dataset_imputed_favorite.csv
```

ต่อมาทำการเติมค่า Lifestyle ที่ขาดหาย (Lifestyle = other)

ในชุดข้อมูลจริงเพื่อให้ได้ข้อมูลสมบูรณ์สำหรับการนำไปวิเคราะห์ต่อไป

```
กำลังทำนาย 276 แถวที่มี lifestyle = 'other'...

 Imputation สำเร็จ!

การกระจายของ lifestyle หลัง Impute:
lifestyle
family        1521
party         1042
simplicity     853
technology     512
workaholic     196
Name: count, dtype: int64

ตรวจสอบ: จำนวน 'other' ที่เหลือ = 0 (ควรเป็น 0)

💾 บันทึกไฟล์: dataset_imputed_lifestyle.csv
```

3.4 การรวมชุดข้อมูลหลังการเติมค่าที่ขาดหาย

โค้ดนี้ใช้รวมข้อมูลที่ผ่านการเติมค่าจากสองโมเดล ได้แก่ favorite และ lifestyle ให้เป็นชุดข้อมูลเดียวที่สมบูรณ์ โดยจะได้ผลลัพธ์คือชุดข้อมูล df_combined ที่ประกอบด้วยค่าของ favorite_Imputed และ lifestyle_Imputed ซึ่งถูกทำนายและเติมครบทุกแถว พร้อมนำไปใช้ในขั้นตอนวิเคราะห์หรือระบบแนะนำกิจกรรมต่อไป

```
df_combined = df_fav_imputed.copy()
df_combined = df_combined.drop(columns=['lifestyle'])
df_combined['lifestyle'] = df_life_imputed['lifestyle']
df_combined = df_combined.rename(columns={
    'favorite': 'favorite_Imputed',
    'lifestyle': 'lifestyle_Imputed'})
df_combined
```

	Age	gender	occupation	education	favorite_Imputed	zone	activity	month	place	AgeGroup	lifestyle_Imputed
0	34.0	female	employee	graduate	family	Watthana	cooking	1	Cooking Studio Town in Town	25-34	family
1	63.0	female	business	bachelor	travel	Yan Nawa	cooking	1	Cooking Studio Town in Town	55-64	family
2	37.0	female	employee	bachelor	travel	Watthana	cooking	1	Cooking Studio Town in Town	35-44	family
3	51.0	female	business	bachelor	family	Lat Phrao	cooking	1	Cooking Studio Town in Town	45-54	family
4	61.0	female	business	bachelor	family	Suan Luang	cooking	1	Cooking Studio Town in Town	55-64	family
...	...	...	...	...	...	...	...	...	...	...	...
4119	46.0	female	other	other	travel	Thung Khru	stageplay	9	Rachadalai Theatre	45-54	family
4120	64.0	female	business	bachelor	travel	Bang Rak	culture	4	Sukhothai	55-64	family
4121	46.0	female	other	other	travel	Bang Khae	culture	5	Sukhothai	45-54	family
4122	40.0	male	enterprise	bachelor	travel	Bang Khen	culture	4	Sukhothai	35-44	family
4123	46.0	female	other	other	travel	Bang Khae	nature	3	Chonburi	45-54	family

ขั้นตอนที่ 4. การประเมินผลโมเดล (Model Evaluation)

หลังจากฝึกโมเดลด้วยข้อมูล Train Set แล้ว จะนำโมเดลมาทดสอบ กับข้อมูล Test Set จากนั้นประเมินผลโดยใช้ค่าชี้วัด เช่น

1. **Accuracy Score** ความแม่นยำโดยรวมของโมเดล
2. **Classification Report** รายงานที่แสดงค่า Precision, Recall และ F1-Score ของแต่ละกิจกรรม

ผลลัพธ์จากการประเมินนี้จะถูกใช้เพื่อตัดสินใจว่าโมเดลใดมีความสามารถในการทำนายกิจกรรม (activity) ได้ดีที่สุด เพื่อนำไปใช้งานจริง

ผลการทดสอบโมเดลของ Lifestyle/Favorite

1. การทดสอบโมเดลของ Lifestyle เหมาะกับโมเดล **Gradient Boosting**
2. การทดสอบโมเดลของ Favorite เหมาะกับโมเดล **Logistic Regression**

```
FAVORITE:
- Best Model: Logistic Regression
- Accuracy: 0.4599
- Output File: dataset_imputed_favorite.csv
- จำนวนแถวที่ Impute: 254

LIFESTYLE:
- Best Model: GradientBoosting
- Accuracy: 0.4026
- Output File: dataset_imputed_lifestyle.csv
- จำนวนแถวที่ Impute: 276
```

ขั้นตอนที่ 5. การวิเคราะห์กฎความสัมพันธ์(Association Rules)

การวิเคราะห์กฎความสัมพันธ์เป็นขั้นตอนหนึ่งในงานวิเคราะห์ข้อมูลสมาชิกของบริษัท ABC เพื่อค้นหารูปแบบความสัมพันธ์ระหว่างพฤติกรรมหรือความสนใจของลูกค้า เช่น ไลฟ์สไตล์ กิจกรรมโปรด หรือประเภทกิจกรรมช่วยให้บริษัทสามารถนำผลลัพธ์ไปปรับใช้ในเชิงกลยุทธ์

เช่น การออกแบบกิจกรรมร่วมสนับสนุน การทำแคมเปญแบบ Cross-selling ตลอดจนการพัฒนาระบบแนะนำกิจกรรมที่เหมาะสมกับลูกค้าแต่ละกลุ่ม

5.1 การเตรียมข้อมูลสำหรับการทำ Association Rule

ในขั้นตอนนี้ได้เตรียมข้อมูลสำหรับการวิเคราะห์กฎความสัมพันธ์ โดยเลือกคอลัมน์ lifestyle, favorite, และ activity จากชุดข้อมูลที่ผ่านการเติมค่าแล้ว จากนั้นรวมข้อมูลแต่ละแถวเป็นรายการ Transaction เช่น [lifestyle_family, favorite_concert, activity_travel] แล้วใช้ TransactionEncoder แปลงข้อมูลให้อยู่ในรูปแบบตารางค่า 0 และ 1 เพื่อพร้อมนำไปใช้ในขั้นตอนการสร้างกฎความสัมพันธ์ต่อไป

```
assoc_df = df_combined[['lifestyle', 'favorite', 'activity']].copy()
transactions = assoc_df.apply(lambda row: [f"lifestyle_{row['lifestyle']}",
                                           f"favorite_{row['favorite']}",
                                           f"activity_{row['activity']}"],
                              axis=1).tolist()
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
trans_df = pd.DataFrame(te_ary, columns=te.columns_)

print("Sample Transactions:")
display(trans_df.head())
```

	activity_beauty	activity_camping	activity_concert	activity_cooking	activity_culture	activity_gourmet	activity_healthy
0	False	False	False	True	False	False	False
1	False	False	False	True	False	False	False
2	False	False	False	True	False	False	False
3	False	False	False	True	False	False	False
4	False	False	False	True	False	False	False

5.2 ขั้นตอนการวิเคราะห์

ใช้เทคนิค FP-Growth โดยอาศัยตัวชี้วัดสำคัญ ได้แก่

1. Support สัดส่วนของรายการทั้งหมดที่พบ Itemset โดยมี minimum support เท่ากับ 0.02

2. Confidence ค่าความน่าเชื่อถือของกฎเมื่อเงื่อนไขเกิดขึ้น มากกว่า 0.4

3. Lift ค่าความสัมพันธ์เชิงสถิติระหว่าง Antecedent และ Consequent เทียบกับความเป็นไปได้แบบสุ่ม มากกว่า 1 เพื่อให้ได้เฉพาะความสัมพันธ์ที่มีความหมายเชิงบวก สำหรับการนำไปตีความทางธุรกิจ

```
# หา frequent itemsets ด้วย FP-Growth
fp = fpgrowth(trans_df, min_support=0.02, use_colnames=True)

# สร้าง association rules
rules_fp = association_rules(fp, metric='confidence', min_threshold=0.4)

# ดูเฉพาะกฎที่ lift > 1 (บ่งชี้ว่ามีความสัมพันธ์จริง)
rules_fp = rules_fp[rules_fp['lift'] > 1]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
1	(favorite_family)	(lifestyle_family)	0.138700	0.368817	0.085354	0.615385	1.668538
2	(favorite_family, activity_movie)	(lifestyle_family)	0.039767	0.368817	0.025703	0.646341	1.752474
8	(activity_stand up comedy)	(favorite_travel)	0.234724	0.501212	0.130698	0.556818	1.110943
9	(activity_stand up comedy, lifestyle_family)	(favorite_travel)	0.079534	0.501212	0.044132	0.554878	1.107072
10	(activity_stand up comedy, lifestyle_party)	(favorite_travel)	0.065228	0.501212	0.033220	0.509294	1.016123
12	(activity_culture)	(favorite_travel)	0.037827	0.501212	0.024976	0.660256	1.317319

5.3 Insight จาก Association Rule

จากการวิเคราะห์ข้อมูลด้วยเทคนิค Association Rule ทำให้เราค้นพบความเชื่อมโยงที่น่าสนใจระหว่างกิจกรรมและไลฟ์สไตล์ ของสมาชิก ซึ่งเผยให้เห็น Insight เชิงลึกที่โมเดลการจำแนกประเภทอาจมองไม่เห็นโดยมีข้อค้นพบหลักและโอกาสทางธุรกิจที่สามารถนำไปต่อยอด ดังนี้

1. กลุ่มแสวงหาประสบการณ์ (**Comedy + Travel**) เราพบความสัมพันธ์ที่ชัดเจนว่า ลูกค้าที่ชื่นชอบกิจกรรม Stand-up Comedy มีแนวโน้มสูงที่จะชื่นชอบ การท่องเที่ยว ด้วย **Insight** นี้ชี้ให้เห็นว่าลูกค้ากลุ่มนี้ไม่ได้มองหากิจกรรมใดกิจกรรมหนึ่ง แต่กำลังมองหาประสบการณ์ ที่ให้ความสุข ความผ่อนคลาย และการได้พบเจอสิ่งใหม่ๆ

Antecedents (สิ่งที่ชอบก่อน)	**Consequent (สิ่งที่มักตามมา)**	**Support**	**Confidence**	**Lift**
(activity_stand up comedy)	(favorite_travel)	0.13	**0.56**	**1.11**
(lifestyle_family, activity_stand up comedy)	(favorite_travel)	0.04	0.55	1.10
(lifestyle_party, activity_stand up comedy)	(favorite_travel)	0.03	0.50	1.01
(lifestyle_simplicity, activity_stand up comedy)	(favorite_travel)	0.03	**0.65**	**1.29**

2. กลุ่มสายสัมพันธ์ครอบครัว (**Movie + Family + Simplicity**) กิจกรรม ดูหนัง มีความเชื่อมโยงอย่างเหนียวแน่นกับไลฟ์สไตล์แบบ ครอบครัว และ ความเรียบง่าย ซึ่งสะท้อนว่าการดูหนังไม่ใช่แค่ความบันเทิงส่วนบุคคลแต่เป็นกิจกรรมหลักที่สมาชิกในครอบครัวใช้เวลา ร่วมกัน สร้างสายสัมพันธ์ในบรรยากาศที่ผ่อนคลาย ที่น่าสนใจคือ

Antecedents	**Consequents**	**Confidence**	**Lift**
(favorite_family, activity_movie)	(lifestyle_family)	**0.65**	**1.75**
(lifestyle_simplicity, activity_movie)	(favorite_travel)	**0.58**	**1.15**

3.กลุ่มนักสำรวจ (**Culture + Travel**) ลูกค้าที่เข้าร่วมกิจกรรมเชิง วัฒนธรรม มีแนวโน้มสูงที่จะรัก การท่องเที่ยว นี่คือกลุ่มลูกค้าที่ชัดเจนว่าต้องการ การท่องเที่ยวเชิงวัฒนธรรม พวกเขาไม่ได้เที่ยวเพื่อพักผ่อนเท่านั้น แต่เพื่อเปิดโลกทัศน์และเรียนรู้สิ่งใหม่ๆ จากประสบการณ์ตรง

Antecedents	Consequent	Confidence	Lift
(activity_culture)	(favorite_travel)	0.66	1.31

5.4 **โอกาสทางธุรกิจ จาก Association Rules**

ข้อมูลเชิงลึกเหล่านี้สามารถนำไปต่อยอดเป็นโอกาสทางธุรกิจที่จับต้องได้ โดยการ มัดรวม กิจกรรมที่ลูกค้ามีแนวโน้มจะสนใจเข้าด้วยกัน

1. Comedy on Tour จากความเชื่อมโยงของกลุ่ม **Stand-up Comedy** และการท่องเที่ยว บริษัทสามารถสร้างสรรค์กิจกรรมพิเศษที่ไม่เหมือนใคร เช่น การจัด **Comedy on Tour**อาจเป็นทริปท่องเที่ยวสุดสัปดาห์ที่มีการแสดงเดี่ยวไมโครโฟนเป็นไฮไลท์บนเรือสำราญ, บนรถทัวร์ หรือในโรงแรมที่พัก นี่คือการสร้างประสบการณ์ใหม่ที่ตอบโจทย์ความต้องการทั้งสองด้านของลูกค้ากลุ่มนี้พร้อมกัน

2. **Family Movie Night** เพื่อตอบสนองกลุ่มลูกค้าที่รักครอบครัวและความเรียบง่ายที่ชื่นชอบการดูหนัง บริษัทสามารถจัดกิจกรรม **Family Movie Night** เช่น การปิดโรงภาพยนตร์รอบพิเศษสำหรับสมาชิกและครอบครัว หรือการจัดฉายหนังกลางแปลงในบรรยากาศสบายๆ พร้อมสิทธิประโยชน์ด้านอาหารและเครื่องดื่มสำหรับครอบครัว

3. Cultural Tourสำหรับกลุ่มนักสำรวจที่สนใจทั้งวัฒนธรรมและการเดินทาง บริษัทไม่ควรหยุดแค่การมอบส่วนลดเข้าชมพิพิธภัณฑ์ แต่ควรก้าวไปสู่การจัด **ทัวร์เชิงศิลปะและวัฒนธรรมโดยเฉพาะ** เช่น ทริปทัวร์วัด, ทัวร์ชมสถาปัตยกรรมเก่าแก่ หรือเวิร์กช็อปงานฝีมือท้องถิ่น ซึ่งตอบโจทย์ความต้องการเรียนรู้และท่องเที่ยวไปพร้อมกัน

ขั้นตอนที่ 6. ระบบแนะนำกิจกรรม (Recommendation System)

การสร้างระบบแนะนำกิจกรรม โดยแปลงข้อมูลสมาชิกเป็นเวกเตอร์คุณลักษณะผ่าน MinMaxScaler สำหรับตัวแปร gender, occupation, lifestyle, favorite พร้อมเพิ่มน้ำหนักให้ lifestyle และ favorite 1.5 เท่า เนื่องจากมีอิทธิพลสูงสุด

จากนั้นสร้าง member_profiles ของสมาชิกแต่ละราย และคำนวณค่าเฉลี่ยตามกิจกรรมเพื่อสร้าง activity_profiles แล้วหาค่าความคล้ายคลึงระหว่างสมาชิกกับกิจกรรมด้วย Cosine Similarity

ได้ตาราง similarity_df ซึ่งใช้เลือกกิจกรรมที่ใกล้เคียงที่สุด ฟังก์ชัน recommend_activity() และ get_top_n_recommendations() คัดเลือกกิจกรรมที่เหมาะสม 3 อันดับต่อสมาชิก และเพิ่มผลลัพธ์ลงในคอลัมน์ Rec_Activity_1, Rec_Activity_2, Rec_Activity_3 เพื่อใช้เป็นผลแนะนำสุดท้ายของระบบ

	gender	Age	occupation	favorite	lifestyle	activity	Rec_Activity_1	Rec_Activity_2	Rec_Activity_3
0	female	34.0	employee	family	family	cooking	concert	camping	movie
1	female	63.0	business	travel	family	cooking	healthy	beauty	culture
2	female	37.0	employee	travel	family	cooking	beauty	healthy	photo
3	female	51.0	business	family	family	cooking	healthy	journey	camping
4	female	61.0	business	family	family	cooking	healthy	journey	camping
5	female	56.0	housewife	family	family	cooking	camping	journey	healthy
6	female	39.0	employee	concert	family	beauty	concert	stageplay	journey
7	female	56.0	official	family	family	beauty	cooking	camping	journey
8	female	36.0	employee	travel	family	beauty	healthy	photo	camping
9	female	61.0	business	movie	family	beauty	healthy	journey	stageplay

7.บทสรุปและข้อเสนอแนะ

7.1 บทสรุป

การวิเคราะห์ข้อมูลสมาชิกของบริษัท ABC และสร้างโมเดล Machine Learning ที่สามารถคาดการณ์กิจกรรมที่เหมาะสมสำหรับลูกค้าได้ โดยอาศัยข้อมูลประชากร โมเดลนี้ช่วยแก้ปัญหาหลักทางธุรกิจของบริษัท ABC ที่ก่อนหน้านี้ไม่สามารถแนะนำกิจกรรมใดๆ ให้กับกลุ่มลูกค้าที่ไม่ได้ระบุ lifestyle หรือ favorite ไว้ได้

7.2 ข้อเสนอแนะ

1.การนำไปใช้งาน บริษัท ABC ควรนำโมเดลที่ให้ประสิทธิภาพสูงสุด (เช่น ให้ค่า Accuracy หรือ F1-Score ที่ดีที่สุด) ไปติดตั้งใช้งานในระบบ CRM เพื่อให้สามารถส่งคำเชิญเข้าร่วมกิจกรรม หรือนำเสนอสิทธิประโยชน์ที่ตรงกับความสนใจ (ที่โมเดลทำนาย) ให้กับลูกค้ากลุ่มเป้าหมายได้ทันที

2.การปรับปรุงโมเดลในอนาคต ควรมีการเก็บข้อมูลพฤติกรรมการเข้าร่วมกิจกรรมของลูกค้าอย่างต่อเนื่อง และนำข้อมูลใหม่มาฝึกโมเดล (**Re-train)** เป็นระยะ เพื่อให้โมเดลมีความทันสมัยและแม่นยำอยู่เสมอ

3.การวิเคราะห์เพิ่มเติม ควรวิเคราะห์ผลลัพธ์จาก classification_report อย่างละเอียด เพื่อดูว่ามีกิจกรรมใดที่โมเดลทำนายผิดพลาดบ่อยเป็นพิเศษ และอาจต้องหาข้อมูล (**Features)** อื่นๆ เพิ่มเติมเพื่อช่วยให้โมเดลจำแนกกิจกรรมนั้นๆ ได้ดีขึ้น