



ยอดขายสินค้า/บริการของบริษัท A

DS512/513 Data Analytics และ DS514/515 Data Science

68199160261 : นางสาวชนิศา อินทร์โชคิ

68199160267 : นางสาวทัศนันทน์ สัตย์มุชnum

68199160307 : นางสาวอัจฉราภรณ์ พรพัฒนทรัพย์

วันที่นำเสนอด : 14/12/2025

Title: ยอดขายสินค้า/บริการของบริษัท A

1. Problem Statement/Background

- บริษัทมีหลายสินค้า/บริการที่ขายผ่านช่องทางต่าง ๆ เช่น Direct Sales, Tele Sales, Organic (ลูกค้าซื้อเอง) ข้อมูลการขายที่มีข้อมูลเก็บอยู่คือ Package Group, Package Type, Solution Type, และ Price After Discount เป็นต้น
- ต้องการทำนายช่องทางการขายที่เหมาะสม เพื่อช่วยให้ทีมขายเลือกช่องทางได้ตรงกับประเภทสินค้าและราคาที่เสนอ
- ปัจจุบันยังไม่มีข้อมูลที่ช่วยในการตัดสินใจเลือกช่องทางการขาย ส่งผลให้โอกาสลดลง และเสียทรัพยากรของทีมขาย

2. SMART Objectives/ Value Propositions

- Specific : พัฒนาโมเดลเพื่อทำนายช่องทางการขาย
- Measurable : ให้ได้ Accuracy $\geq 70\%$
- Achievable : ในข้อมูลการขายที่มีอยู่ในการสร้างและปรับแต่งโมเดลให้เหมาะสม
- Relevant: ช่วยทีมขายนำผลทำนายไปใช้วางกลยุทธ์ช่องทางให้มีประสิทธิภาพมากขึ้น
- Time-bound: ภายใน 6 สัปดาห์
- พัฒนาโมเดลเพื่อทำนายช่องทางการขายให้ได้ Accuracy $\geq 70\%$ ภายใน 6 สัปดาห์ เพื่อช่วยให้ทีมขายใช้ผลทำนายในการวางแผนกลยุทธ์ช่องทางให้มีประสิทธิภาพมากขึ้น

3. Questions/Hypothesis

- Analytical Questions
 - performance ปีนี้เป็นอย่างไร?
 - ลูกค้าส่วนใหญ่ซื้อสินค้า/บริการผ่านช่องทางใดมากที่สุด?
 - สัดส่วนการขายแต่ละช่องทางเป็นอย่างไร?
 - ลูกค้ามีลักษณะเป็นอย่างไร?
- Predictive Hypothesis
 - สามารถทำนายช่องทางการขายที่เหมาะสม เพื่อช่วยให้ทีมขายเลือกช่องทางได้ตรงกับประเภทสินค้าและราคาที่เสนอได้

4. Data Sources/Attributes

- Data sources & collection
 - ข้อมูลบริษัท A จากไฟล์ Excel
- Data cleaning & preprocessing
 - แยกคอลัมน์ที่ข้อมูลถูกรวมกันไว้ เช่น อายุ ตำบล จังหวัด รหัสไปรษณีย์
 - ลบแ眷ที่มีค่าว่าง
 - ลบข้อมูลที่อาจจะผิดพลาด เช่น คนที่อายุต่ำกว่า 14 ปี
- Target variables & feature
 - Target : Sales_Channel
 - Feature :
 - Package_Group, Subscription_Type, Package_Type, Price_After_Discount, Price, Age
- Encoding & scaling strategies **
 - Label Encoding สำหรับ categorical features (เนื่องจากมีกลุ่มข้อมูลเยอะ)
 - scaling ข้อมูล

5. Analysis/Model Development

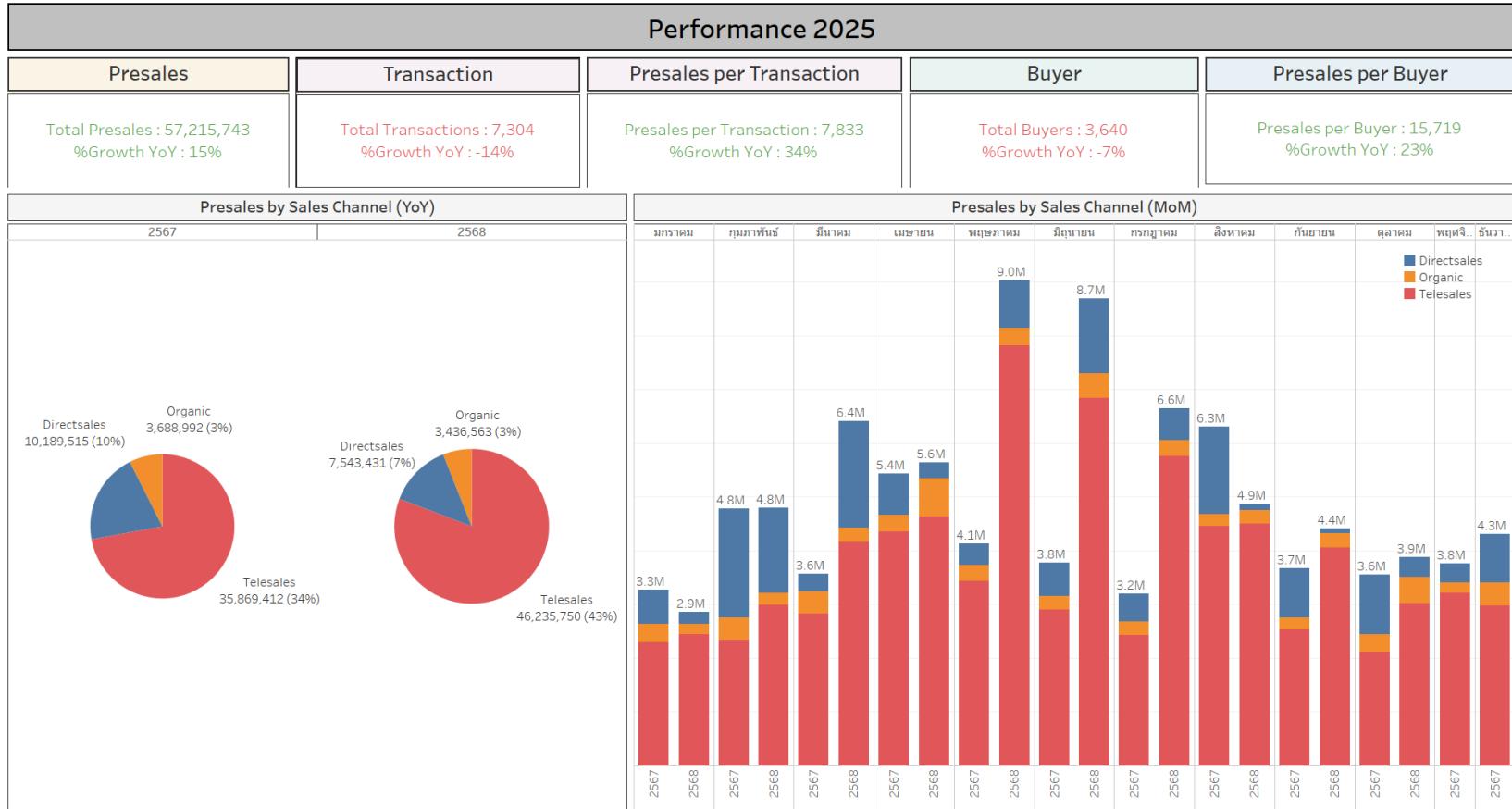
- Analytics Methodology & Exploratory Data Analysis (EDA)
 - เปรียบเทียบจำนวนและยอดขายแต่ละช่องทาง
 - หาความสัมพันธ์ของข้อมูล ปีเกิด อายุ ราคา ราคาหลังหักส่วนลด
 - เปรียบเทียบยอดขายรายปี รายเดือน
 - เปรียบเทียบยอดขายตามกลุ่มลูกค้า/บริการ
 - ดูข้อมูล demographics เพศ อายุ จังหวัด
- Modeling Methodology
 - K-Nearest Neighbors, Logistic Regression (Classification)
 - Training & hyperparameter tuning: Grid Search
 - Evaluation metrics : Accuracy, Precision, Recall, F1-Score, Confusion Matrix, Classification Report

6. Findings and Insights

7. Recommendation/Action and Impact



Findings and Insights



Data

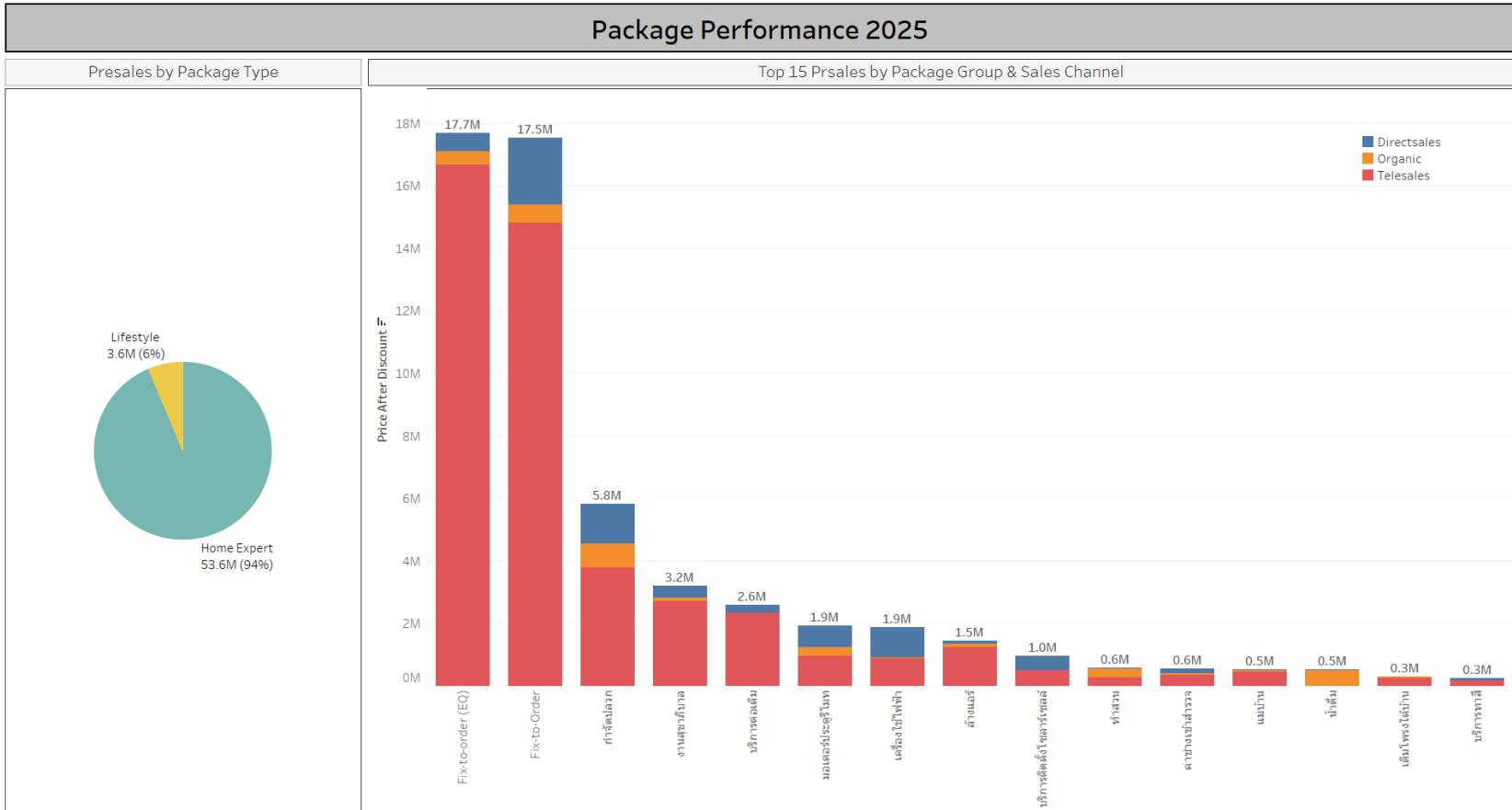
ยอดขายแบ่งตามช่องทางการขายแยกรายปี และรายเดือน

Major Findings and Insights...

- ยอดขายเพิ่มขึ้น แต่จำนวนขายและผู้ซื้อลดลงจากปีก่อนหน้า นั่นหมายความว่าขายสินค้า/บริการที่มีราคาสูงได้เพิ่มขึ้น
- ช่องทางที่ทำยอดขายได้มากที่สุดยังคงเป็นช่องทาง TeleSales และ DirectSales ซึ่งมีการเติบโตผ่านช่องทาง TeleSales มากขึ้น



Findings and Insights



Data

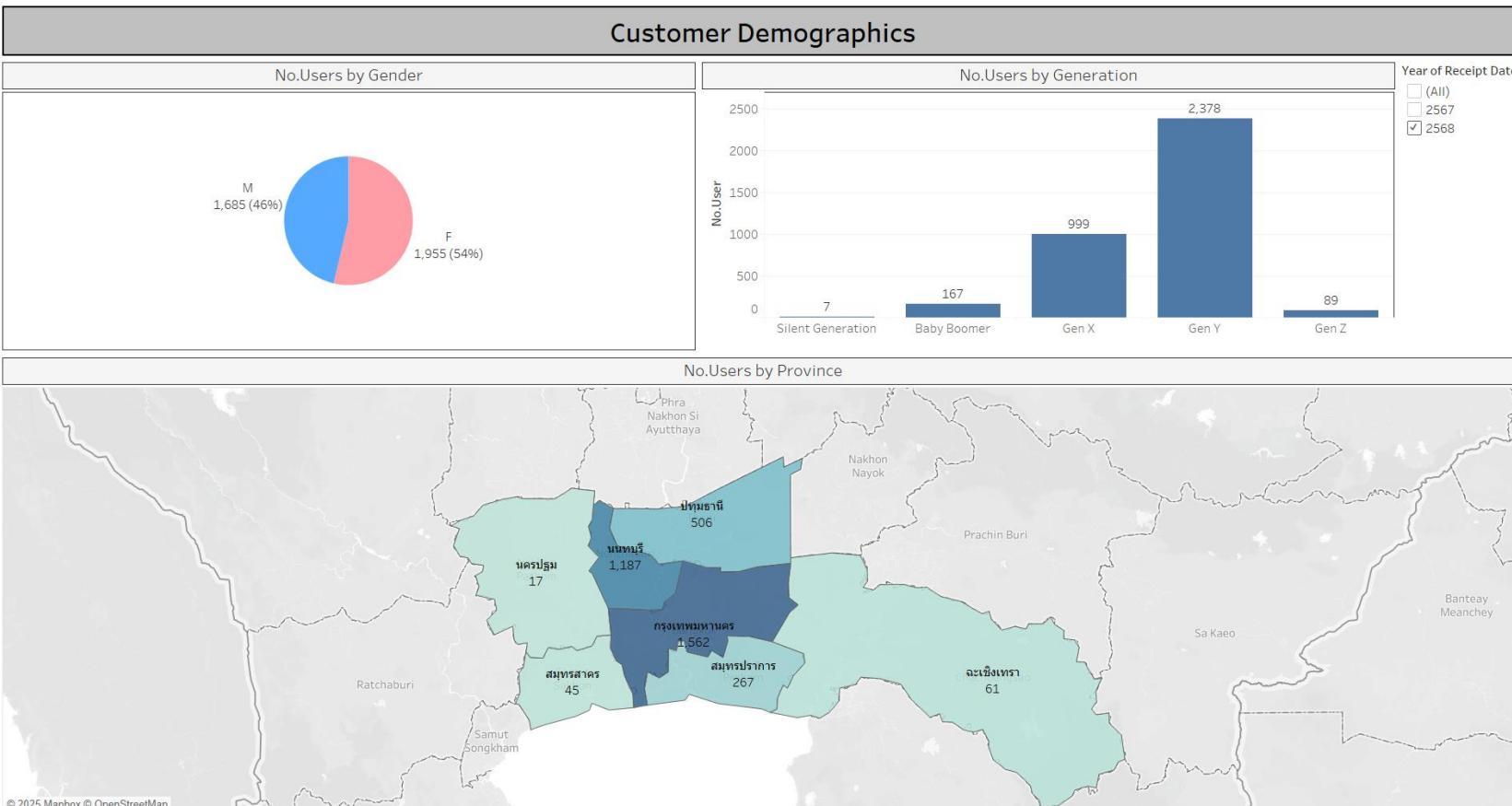
ยอดขายแบ่งตามประเภทและกลุ่มสินค้า/บริการ

Major Findings and Insights...

- สินค้า/บริการส่วนใหญ่มากจากกลุ่ม Home Expert และที่ทำให้ยอดขายเพิ่มขึ้นมากที่สุดได้แก่ Fix-to-Order (การซ่อมบ้าน)
- ทำสวน และ น้ำดื่ม เป็นสินค้า/บริการที่มีการซื้อผ่านช่องทาง Organic เป็นส่วนใหญ่



Findings and Insights



Data

จำนวนลูกค้าแยกตามอายุ เพศและจังหวัด

Major Findings and Insights...

- ลูกค้าส่วนใหญ่เป็นเพศหญิงมากกว่า
- Gen Y เป็นกลุ่มลูกค้ารายใหญ่
- ลูกค้าอยู่ในกทม.มากที่สุด



Data Preprocessing

```
#ลบเเละเปลี่ยน null ของ column Gender Year_Birthday Subdistrict District Province Postal_Code Payment_Method
df2 = df.dropna(subset=['Gender', 'Year_Birthday', 'Subdistrict', 'District', 'Province', 'Postal_Code', 'Payment_Method'])
df2.info()
df2.isnull().sum()
```

```
... <class 'pandas.core.frame.DataFrame'>
Index: 16597 entries, 0 to 18250
Data columns (total 24 columns):
 #   Column      Non-Null Count Dtype  
--- 
 0   UserID      16597 non-null  object  
 1   Gender      16597 non-null  object  
 2   Year_Birthday  16597 non-null float64
 3   Subdistrict  16597 non-null  object  
 4   District     16597 non-null  object  
 5   Province     16597 non-null  object  
 6   Postal_Code  16597 non-null  float64
 7   Package_ID   16597 non-null  int64  
 8   Package_Group 16597 non-null  object  
 9   Subscription_Type 16597 non-null object  
 10  Package_Status 16597 non-null  object  
 11  Sales_Channel 16597 non-null  object  
 12  Customer_Type 16597 non-null  object  
 13  Package_Type  16597 non-null  object  
 14  Supplier_Type 16597 non-null  object  
 15  Receipt_Date  16597 non-null  datetime64[ns]
 16  Receipt_Week  16597 non-null  int64  
 17  Receipt_Month 16597 non-null  object  
 18  Receipt_Time  16597 non-null  object  
 19  Price        16597 non-null  float64
 20  Discount      732 non-null   object  
 21  Price_After_Discount 16597 non-null float64
 22  Payment_Method 16597 non-null  object  
 23  Supplier_Name 16597 non-null  object  
dtypes: datetime64[ns](1), float64(4), int64(2), object(17)
```

```
#เปลี่ยน data type Year_Birthday เป็น int
df2['Year_Birthday'] = df2['Year_Birthday'].astype(int)

#เปลี่ยน data type Postal_Code, Package_ID, Receipt_Week เป็น Obj
df2['Postal_Code'] = df2['Postal_Code'].astype(str)
df2['Package_ID'] = df2['Package_ID'].astype(str)
df2['Receipt_Week'] = df2['Receipt_Week'].astype(str)

df2.info()

<class 'pandas.core.frame.DataFrame'>
Index: 16597 entries, 0 to 18250
Data columns (total 24 columns):
 #   Column      Non-Null Count Dtype  
--- 
 0   UserID      16597 non-null  object  
 1   Gender      16597 non-null  object  
 2   Year_Birthday  16597 non-null int64  
 3   Subdistrict  16597 non-null  object  
 4   District     16597 non-null  object  
 5   Province     16597 non-null  object  
 6   Postal_Code  16597 non-null  object  
 7   Package_ID   16597 non-null  object  
 8   Package_Group 16597 non-null  object  
 9   Subscription_Type 16597 non-null object  
 10  Package_Status 16597 non-null  object  
 11  Sales_Channel 16597 non-null  object  
 12  Customer_Type 16597 non-null  object  
 13  Package_Type  16597 non-null  object  
 14  Supplier_Type 16597 non-null  object  
 15  Receipt_Date  16597 non-null  datetime64[ns]
 16  Receipt_Week  16597 non-null  object  
 17  Receipt_Month 16597 non-null  object  
 18  Receipt_Time  16597 non-null  object  
 19  Price        16597 non-null  float64
 20  Discount      732 non-null   object
```

- เปลี่ยนประเภทของข้อมูล
- Spilt คอลัมน์ข้อมูลทีอยู่
- ลบช่องว่างของข้อมูล



Data Preprocessing

```
▶ #เพิ่มข้อมูลอายุ จากการนำ year มีจุดนั้น ลงกับ column Year_Birthday  
from datetime import datetime  
df2['Age'] = datetime.now().year - (df2['Year_Birthday']-543)  
df2.head()
```

[ing-a-view-versus-a-copy](#)

cription_Type	...	Receipt_Date	Receipt_Week	Receipt_Month	Receipt_Time	Price	Discount	Price_After_Discount	Payment_Method	Supplier_Name	Age
OneTime	...	2024-01-01	1	January	07:41 น.	800.0	NaN	800.0	SCB_EASY	Maintenance team	28
OneTime	...	2024-01-01	1	January	07:48 น.	1600.0	NaN	1600.0	2C2P	AirMate	68
SUBSCRIBE	...	2024-01-01	1	January	15:23 น.	565.0	NaN	565.0	2C2P	บริษัท บุญรอดเท รดลัง จำกัด	43
SUBSCRIBE	...	2024-01-01	1	January	16:03 น.	1399.0	NaN	1399.0	2C2P	Greenmania	48
SUBSCRIBE	...	2024-01-01	1	January	16:37 น.	1700.0	NaN	1700.0	2C2P	BeNeat	37

- เพิ่มคอลัมน์อายุ
- ตัดข้อมูลที่ไม่ถูกต้องออก (ผู้ที่อายุน้อยกว่า 14 ปี)

```
▶ #ลบแค่ Age ที่มีค่าน้อยกว่า 14  
df3 = df2.drop(df2[df2['Age'] < 14].index)  
df3.head()
```

cription_Type	...	Receipt_Date	Receipt_Week	Receipt_Month	Receipt_Time	Price	Discount	Price_After_Discount	Payment_Method	Supplier_Name	Age
OneTime	...	2024-01-01	1	January	07:41 น.	800.0	NaN	800.0	SCB_EASY	Maintenance team	28
OneTime	...	2024-01-01	1	January	07:48 น.	1600.0	NaN	1600.0	2C2P	AirMate	68
SUBSCRIBE	...	2024-01-01	1	January	15:23 น.	565.0	NaN	565.0	2C2P	บริษัท บุญรอดเท รดลัง จำกัด	43
SUBSCRIBE	...	2024-01-01	1	January	16:03 น.	1399.0	NaN	1399.0	2C2P	Greenmania	48
SUBSCRIBE	...	2024-01-01	1	January	16:37 น.	1700.0	NaN	1700.0	2C2P	BeNeat	37



Imbalance Data and Encoding

▶ # ลองดูการเก็บข้อมูล balanced โดยข้อมูลในช่องทาง direct sales มีข้อมูลน้อยสุดอยู่ที่ 387 transaction จะจัดทำการตัดข้อมูล ทั้ง 3 กลุ่มให้เท่ากันอยู่ที่ 380 transaction

```
balanced_df = df3.groupby('Sales_Channel').sample(n=380, random_state=42)
```

```
balanced_df.head()
```

...	UserID	Gender	Year_Birthday	Subdistrict	District	Province	Postal_Code	Package_ID	Package_Group	Subscription_Type	...	Receipt_Week	Rec...
10074	46157	F	2535	ตำบลบางแพ้ว	อำเภอ บางแพ	สมุทรปราการ	10540.0	815	ค่าบริการสำราญ	OneTime	...	6	
8201	33272	M	2538	ตำบลตลาด คลอง	อำเภอ บางกรวย	นนทบุรี	11130.0	5370	งานสุขาภิบาล	OneTime	...	49	
7966	6564	M	2504	แขวงบางแส	เขตบางแส	กรุงเทพมหานคร	10160.0	5348	Fix-to-Order	OneTime	...	47	
2659	2038	M	2529	แขวงบางแส	เขตบางแส	กรุงเทพมหานคร	10160.0	4581	Fix-to-Order	OneTime	...	16	
6549	19100	F	2527	แขวงตลาด บางเขน	เขตหลักสี่	กรุงเทพมหานคร	10210.0	5001	Fix-to-Order	OneTime	...	40	

5 rows × 26 columns

balanced_df.shape
(1140, 26)

- จัดการข้อมูล Imbalance โดยการสุ่มข้อมูล (เลือกจำนวนข้อมูลจากจำนวนกลุ่มที่น้อยที่สุด)
- Encode ข้อมูลที่เป็น Category

▶ # Encode ข้อมูลอีกรอบด้วยข้อมูลชุดใหม่

```
target = 'Sales_Channel'
```

#แยก Features / Target

```
X_New = balanced_df.drop(columns=[target])
y_New = balanced_df[target]
```

#คัดลอกข้อมูลเพื่อเตรียมแปลง data type

```
X_copy = X_New.copy()
```

#แปลง datetime ให้เป็นตัวเลข timestamp

```
for col in X_copy.columns:
    if np.issubdtype(X_copy[col].dtype, np.datetime64):
        X_copy[col] = X_copy[col].astype('int64') // 10**9 # แปลงเป็น seconds
```

#แปลง object เป็นตัวเลขด้วย LabelEncoder

```
le = LabelEncoder()
```

for col in X_copy.columns:

```
    if X_copy[col].dtype == 'object':
        X_copy[col] = le.fit_transform(X_copy[col].astype(str))
```

#target ค่าเป็น object ก็แปลงด้วย

```
if y_New.dtype == 'object':
    y_New = le.fit_transform(y_New.astype(str))
```



Feature Selection and Scaling

```
#เลือก feature ที่เกี่ยวข้องต้องการทดสอบด้วยข้อมูลใหม่  
selected_features = [  
    'Package_Type',  
    'Price_After_Discount',  
    'Price',  
    'Subscription_Type',  
    'Package_Group',  
    'Age'  
]
```

```
X_selected2 = X_copy[selected_features]  
y_selected2 = y_New  
  
# train-test split data  
  
X_train2, X_test2, y_train2, y_test2 = train_test_split(  
    X_selected2, y_selected2, test_size=0.2, random_state=0  
)
```

```
#ปรับข้อมูล  
scaler = StandardScaler()  
X_train_scaled2 = scaler.fit_transform(X_train2)  
X_test_scaled2 = scaler.transform(X_test2)
```

```
print(X_train_scaled2.shape)  
print(y_train2.shape)  
print(X_test_scaled2.shape)  
print(y_test2.shape)
```

```
(912, 6)  
(912,)  
(228, 6)  
(228,)
```

- กำหนด Feature ที่ใช้
- Scaling ข้อมูล



Model and Grid Search

```
▶ #ลองทดสอบ model ในนี้
from sklearn.neighbors import KNeighborsClassifier
knn2 = KNeighborsClassifier()

from sklearn.model_selection import GridSearchCV

# Define the parameter grid
param_grid = {'n_neighbors': [3,5,7,9,11,13,15,17,19,21,25]}

# Set up the GridSearchCV
grid_search = GridSearchCV(knn2, param_grid, cv=5, scoring='accuracy')

# Fit the grid search
grid_search.fit(X_train_scaled2, y_train2)

# แสดงผล best parameter
print("Best parameters : ", grid_search.best_params_)

# ใช้ best estimator ท่านาย test set
best_knn = grid_search.best_estimator_
y_predict_train2 = best_knn.predict(X_train_scaled2)
y_predict_test2 = best_knn.predict(X_test_scaled2)

...
... Best parameters : {'n_neighbors': 21}
```

- ทดสอบด้วยโมเดล K-Nearest Neighbors และ Logistic Regression
- Grid Search เพื่อเลือกพารามิเตอร์ที่ดีที่สุด

```
▶ #model LogisticRegression

from sklearn.linear_model import LogisticRegression
logreg2 = LogisticRegression()

# Define the parameter grid
param_grid2 = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

# Set up the GridSearchCV
grid_search = GridSearchCV(logreg2, param_grid2, cv=5)

# Fit the grid search
grid_search.fit(X_train_scaled2, y_train2)

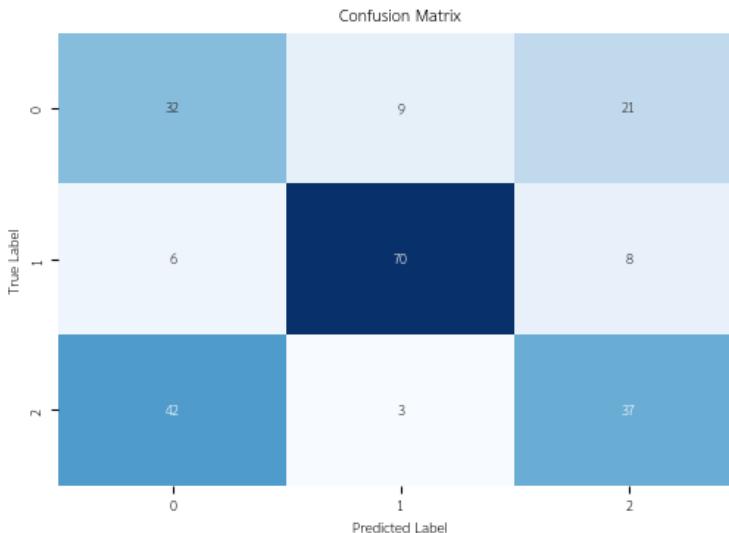
# แสดงผล best parameter
print("Best parameters : ", grid_search.best_params_)

# ใช้ best estimator ท่านาย test set
best_logreg = grid_search.best_estimator_
...
... Best parameters : {'C': 10}
```

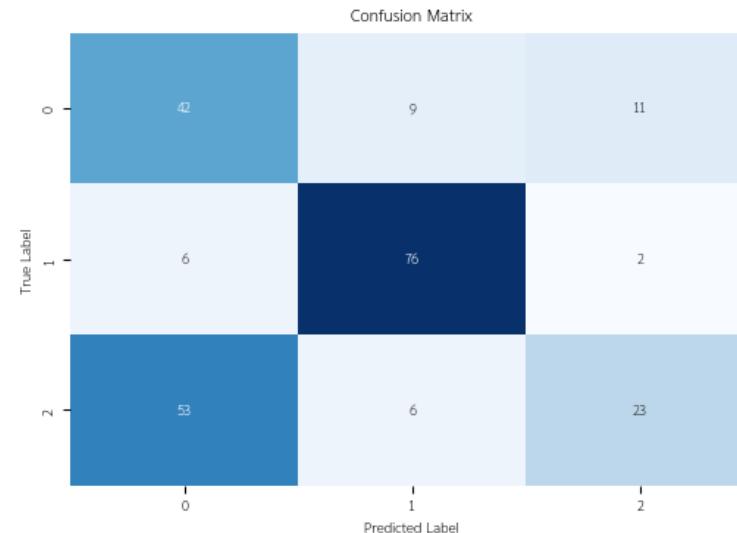


Results from model

K-Nearest Neighbors



Logistic Regression



Class 0 : DirecSales

Class 1 : Organic

Class 2 : TeleSales

Classification Report:

	precision	recall	f1-score	support
0	0.40	0.52	0.45	62
1	0.85	0.83	0.84	84
2	0.56	0.45	0.50	82
accuracy			0.61	228
macro avg	0.60	0.60	0.60	228
weighted avg	0.62	0.61	0.61	228

Classification Report:

	precision	recall	f1-score	support
0	0.42	0.68	0.52	62
1	0.84	0.90	0.87	84
2	0.64	0.28	0.39	82
accuracy			0.62	228
macro avg	0.63	0.62	0.59	228
weighted avg	0.65	0.62	0.60	228



Conclusion

Summary of Analytics Findings

- Performance Overview ยอดขายเพิ่ม 15% YoY แต่จำนวน Transaction และจำนวน Buyer ลดลง นั่นหมายความว่า ยอดขายเติบโตจากสินค้า/บริการที่มีมูลค่าสูงกว่าเดิม
- TeleSales ยังคงเป็นช่องทางทำรายได้หลักและเติบโตอย่างต่อเนื่อง โดยส่วนใหญ่เป็นสินค้า/บริการ กลุ่มงาน Home Expert
- ลูกค้าเกินครึ่งเป็นผู้หญิง และกลุ่มใหญ่สุดอยู่ใน Gen Y โดยอาศัยอยู่กรุงเทพมหานครและปริมณฑล

Model Performance & Insights

- ทั้งสองโมเดล มีความแม่นยำเฉลี่ยใกล้เคียงกัน (~60%)
- Class 0 (DirectSales) Logistic Regression ให้ Recall สูงถึง 68% ดีกว่า KNN
- Class 1 (Organic) ถูกทำนายได้ดีที่สุดในทั้งสองโมเดล โดยมี Recall 83–90%
- Class 2 (TeleSales) คือ class ที่ไม่ได้ทำนายได้แย่ที่สุด (ไม่ถึง 40–50%)
- โมเดล Logistic Regression ให้ค่าความแม่นยำดีกว่า และค่าแต่ละ class ทำนายถูกได้มากกว่า KNN
- โมเดลสามารถทำนาย Organic ได้ดี แต่ยังไม่แม่นยำสำหรับ TeleSales อาจจะเพราะข้อมูลซับซ้อนเกินไป

Title: ยอดขายสินค้า/บริการของบริษัท A

<h2>1. Problem Statement/Background</h2> <ul style="list-style-type: none"> - บริษัทมีหลายสินค้า/บริการที่ขายผ่านช่องทางต่าง ๆ เช่น Direct Sales, Tele Sales, Organic (ลูกค้าซื้อเอง) ข้อมูลการขายที่มีข้อมูลเก็บอยู่คือ Package Group, Package Type, Solution Type, และ Price After Discount เป็นต้น - ต้องการทำนายช่องทางการขายที่เหมาะสม เพื่อช่วยให้ทีมขายเลือกช่องทางได้ตรงกับประเภทสินค้าและราคาที่เสนอ - ปัจจุบันยังไม่มีข้อมูลที่ช่วยในการตัดสินใจเลือกช่องทางการขาย ส่งผลให้โอกาสลดลง และเสียทรัพยากรของทีมขาย 	<h2>2. SMART Objectives/ Value Propositions</h2> <ul style="list-style-type: none"> - Specific : พัฒนาโมเดลเพื่อทำนายช่องทางการขาย - Measurable : ให้ได้ Accuracy $\geq 70\%$ - Achievable : ในข้อมูลการขายที่มีอยู่ในการสร้างและปรับแต่งโมเดลให้เหมาะสม - Relevant: ช่วยทีมขายนำผลทำนายไปใช้วางกลยุทธ์ช่องทางใหม่ประสิทธิภาพมากขึ้น - Time-bound: ภายใน 6 สัปดาห์ - พัฒนาโมเดลเพื่อทำนายช่องทางการขายให้ได้ Accuracy $\geq 70\%$ ภายใน 6 สัปดาห์ เพื่อช่วยทีมขายใช้ผลทำนายในการวางแผนกลยุทธ์ช่องทางใหม่ประสิทธิภาพมากขึ้น 	<h2>3. Questions/Hypothesis</h2> <ul style="list-style-type: none"> • Analytical Questions <ul style="list-style-type: none"> - performance ปีนี้เป็นอย่างไร? - ลูกค้าส่วนใหญ่ซื้อสินค้า/บริการผ่านช่องทางใดมากที่สุด? - สัดส่วนการขายแต่ละช่องทางเป็นอย่างไร? - ลูกค้ามีลักษณะเป็นอย่างไร? • Predictive Hypothesis <ul style="list-style-type: none"> - สามารถทำนายช่องทางการขายที่เหมาะสม เพื่อช่วยให้ทีมขายเลือกช่องทางได้ตรงกับประเภทสินค้าและราคาที่เสนอได้ 	<h2>4. Data Sources/Attributes</h2> <ul style="list-style-type: none"> • Data sources & collection <ul style="list-style-type: none"> - ข้อมูลบริษัท A จากไฟล์ Excel • Data cleaning & preprocessing <ul style="list-style-type: none"> - แยกคอลัมน์ที่ข้อมูลถูกรวมกันไว้ เช่น อายุ ตำบล จังหวัด รหัสไปรษณีย์ - ลบแถวที่มีค่าว่าง - ลบข้อมูลที่อาจจะผิดพลาด เช่น คนที่อายุต่ำกว่า 14 ปี • Target variables & feature <ul style="list-style-type: none"> - Target : Sales_Channel - Feature : <ul style="list-style-type: none"> Package_Group, Subscription_Type, Package_Type, Price_After_Discount, Price, Age • Encoding & scaling strategies ** <ul style="list-style-type: none"> - Label Encoding สำหรับ categorical features (เนื่องจากมีกลุ่มข้อมูลเยอะ) - scaling ข้อมูล
<h2>5. Analysis/Model Development</h2> <ul style="list-style-type: none"> • Analytics Methodology & Exploratory Data Analysis (EDA) <ul style="list-style-type: none"> - เปรียบเทียบจำนวนและยอดขายแต่ละช่องทาง - หาความสัมพันธ์ของข้อมูล ปัจจัย อายุ ราคา ราคารหงส์หักส่วนลด - เปรียบเทียบยอดขายรายปี รายเดือน - เปรียบเทียบยอดขายตามกลุ่มลูกค้า/บริการ - ดูข้อมูล demographics เพศ อายุ จังหวัด • Modeling Methodology <ul style="list-style-type: none"> - K-Nearest Neighbors, Logistic Regression (Classification) - Training & hyperparameter tuning: Grid Search - Evaluation metrics : Accuracy ,Precision ,Recall ,F1-Score, Confusion Matrix, Classification Report 	<h2>6. Findings and Insights</h2> <ul style="list-style-type: none"> • Business Insights <ul style="list-style-type: none"> - ยอดขายเพิ่มขึ้น แต่จำนวนขายและผู้ซื้อลดลงจากปีก่อนหน้า นั่นหมายความว่าขายสินค้า/บริการที่มีราคาสูงได้เพิ่มขึ้น - ช่องทางที่ทำยอดขายได้มากที่สุดยังคงเป็นช่องทาง TeleSales และ DirectSales ซึ่งมีการเดินทางผ่านช่องทาง TeleSales มากขึ้น - สินค้า/บริการส่วนใหญ่มาจากกลุ่ม Home Expert และที่ทำให้ยอดขายเพิ่มขึ้นมากที่สุดได้แก่ Fix-to-Order (การซ่อมบ้าน) - ทำส่วน และ น้ำดื่ม เป็นสินค้า/บริการที่มีการซื้อผ่านช่องทาง Organic เป็นส่วนใหญ่ - ลูกค้าส่วนใหญ่เป็นเพศหญิงมากกว่า ,Gen Y เป็นกลุ่มลูกค้ารายใหญ่, ลูกค้าอยู่ในกทม.มากที่สุด • Predictive Results <ul style="list-style-type: none"> Model performance : Logistic Regression (หลังตัดข้อมูลข้อมูล) prediction accuracy : Accuracy หลังตัดข้อมูล โดยรวมลดลงแต่ผลลัพธ์มีความสมดุลมากขึ้นทั้ง 3 class (Accuracy รวม : 0.62) <ul style="list-style-type: none"> - Class 0 (DirectSales) : โมเดลจับได้ดีขึ้น แต่ยังไม่แม่นยำมาก (precision : 0.42 , recall : 0.68) - Class 1 (Organic) : จำแนกได้ดีที่สุด แม่นยำ (precision : 0.84 , recall : 0.90) - Class 2 (TeleSales) : โมเดลจับได้น้อยแม่นยำไม่นักต้องพัฒนาเพิ่ม (precision : 0.64 , recall : 0.28) 	<h2>7. Recommendation/Action and Impact</h2> <ul style="list-style-type: none"> - เพิ่มประสิทธิภาพช่องทางการขายผ่าน TeleSales ให้สูงขึ้น (อาจจะเกี่ยวข้องกับ feature เพิ่มที่นำมาปรับปรุง model) - วางแผนกลยุทธ์การตลาดแต่ละช่องทางการขาย - จัดโปรโมชั่นกับสินค้า/บริการที่ขายดี เช่น Fix-to-Order เน้นการ TeleSales มากขึ้น - ปรับปรุงหน้า Application ให้สินค้า/บริการ ที่ลูกค้านิยมซื้อ เช่น ทำส่วน และ น้ำดื่ม 	