การวิเคราะห์ปัจจัยที่มีผลต่อยอดขาย และการทำนายปริมาณการขายของร้านอาหาร เพื่อวางแผนกลยุทธ์กระตุ้นยอดขาย

DS512/513 Data Analytics และ DS514/515 Data Science

68199160253 ขนิษฐา ลีลาเทพินทร์

68199160273 ธารารัตน์ แก้วคาไสย์

68199160274 ธิติวุฒิ มลิวัลย์

10 ธันวาคม 2568

DATA ANALYTICS/SCIENCE PROJECT CANVAS

Designed by: ขนิษฐา, ธารารัตน์, ธิติวุฒิ

Date: 10 Dec 2025

Title: การวิเคราะห์ปัจจัยที่มีผลต่อยอดขายและการทำนายปริมาณการขายของร้านอาหารเพื่อวางแผนกลยุทธ์กระตุ้นยอดขาย

1. Background / Problem Statement

Background:

- ธุรกิจร้านอาหารต้องเผชิญความท้าทายในการบริหารจัดการยอดขาย การกำหนดราคา และการวางแผนโปรโมชั่นท่ามกลางปัจจัยภายนอกที่เปลี่ยนแปลง

Problem Statement:

- ขาดเครื่องมือคาดการณ์ยอดขายที่แม่นยำ
- การกำหนดราคาและโปรโมชั่นยังไม่อิงข้อมูล
- ไม่ทราบปัจจัยที่ส่งผลต่อยอดขายและพฤติกรรมลูกค้า

2. Questions/Hypothesis

Research Questions:

1. ปัจจัยใดบ้างที่มีผลกระทบต่อยอดขายของร้านอาหาร
- สภาพอากาศ โปรโมชั่น และเหตุการณ์พิเศษส่งผลต่อยอดขายอย่างไร
- ประเภทร้านอาหารและประเภทมื้ออาหารมีความสัมพันธ์กับยอดขายหรือไม่

2. กลยุทธ์การกำหนดราคาและโปรโมชั่นมีประสิทธิภาพเพียงใด
- ความแตกต่างระหว่างราคาตลาดกับราคาขายจริงส่งผลต่อยอดขายอย่างไร
- โปรโมชั่นช่วยเพิ่มยอดขายและผลกำไรได้จริงหรือไม่

3. Value Propositions

1. ผู้ประกอบการร้านอาหาร: เข้าใจปัจจัยที่ส่งผลต่อยอดขายและพฤติกรรมลูกค้า ช่วยในการวางแผนกลยุทธ์ราคา โปรโมชั่น และการบริหารเมนูอย่างมีประสิทธิภาพ
2. นักการตลาด: ได้ข้อมูลเชิงลึกเกี่ยวกับประสิทธิภาพของโปรโมชั่นและปัจจัยภายนอกที่ส่งผลต่อการตัดสินใจซื้อของลูกค้าเพื่อวางแผนแคมเปญที่ตรงเป้าหมาย

4. Data Sources/Attributes

- Data Sources: Kaggle (Link) Restaurant Sales Report 2024-2025
- Attributes: 13
- Records: 10,000

- Target: ปริมาณการขาย (quantity_sold)
- Features:
  ยอดขายวันก่อนหน้า (lag1), ราคาขาย (selling price), สภาพอากาศ, มีโปรโมชัน, เหตุการณ์พิเศษ (event)

5. Analysis/Model Development

1. Analysis:
- Descriptive Statistics
- Correlation Analysis

2. Results:
- Data Visualization: Tableau
- Summary Statistics

3. Modeling (Supervised learning)
- linear regression
- regularized linear regression
(ridge, lasso, elastic net)

4. Model evaluation: R^2, MAE, RMSE

6. Findings and Insights

7. Recommendation/Action and Impact

Explore the data

DATA DICTIONARY

Attribute	Description	Data Type	Valid Range/Example
date	วันที่ของการขาย (รูปแบบ dd/MM/yyyy)	Date/DateTime	01/01/2024 – 01/01/2025
restaurant_id	รหัสร้านอาหาร (ตัวเลขไม่ซ้ำ)	Number/Float	1 – 50
restaurant_type	ประเภทของร้าน (Food Stall / Casual / Fine Dining)	Text	Food Stall / Casual Dining / Fine Dining
menu_item_name	ชื่อเมนูอาหารที่ขาย	Text	Kaya Toast Set / Cendol / Teh Tarik
meal_type	ประเภทมื้ออาหาร (Breakfast / Lunch / Dinner)	Text	Dinner / Lunch / Breakfast
key_ingredients_tags	วัตถุดิบหลักของเมนู (คั่นด้วย comma)	Text	white bread, kaya, butter, soft-boiled eggs / rice flour jelly, coconut milk, palm sugar, red beans / black tea, condensed milk, evaporated milk
typical_ingredient_cost	ต้นทุนวัตถุดิบโดยประมาณ	Number/Float	0.8 – 9
observed_market_price	ราคาตลาดที่สังเกตได้ของเมนู	Number/Float	1.46 – 56.29
actual_selling_price	ราคาขายจริงให้ลูกค้า	Number/Float	1.36 – 83.09
quantity_sold	จำนวนที่ขายได้ (หน่วยเป็นจาน/หน่วย)	Number/Float	0 – 1668
has_promotion	มีโปรโมชันหรือไม่ (True/False)	Boolean	True / False
special_event	อยู่ในช่วงอีเวนต์พิเศษหรือไม่ (True/False)	Boolean	True / False
weather_condition	สภาพอากาศในวันขาย (เช่น Sunny/Rainy/Cloudy)	Text	Sunny / Cloudy / Rainy

Data:

- 10,000 records
- 13 Attributes

Types of Variables

- 11 Categorical
- 2 Continuous

DATA PREPARATION (Cleansing)

A10003 =COUNTA(A2:A10001)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	date	restaurant	restaurant_type	menu_item_name	meal_type	key_ingredients_tags	typical_ing	observed_	actual_sell	quantity_s	has_prom	special_ev	weather_co
9983	1/1/2025	17	Food Stall	Cendol	Lunch	rice flour jelly, coconut r	15.2	38.456	46.36	336	FALSE	FALSE	Sunny
9984	1/1/2025	29	Food Stall	Chicken Rice	Lunch	chicken, rice, ginger, ga	30.4	74.404	93.252	495	FALSE	FALSE	Sunny
9985	1/1/2025	41	Cafe	Cendol	Lunch	rice flour jelly, coconut r	15.2	41.496	53.504	371	FALSE	FALSE	Cloudy
9986	1/1/2025	18	Food Stall	Tandoori Chicken	Dinner	chicken, yogurt, tandoor	53.2	122.208	206.796	169	FALSE	FALSE	Sunny
9987	1/1/2025	12	Cafe	Mushroom Soup	Dinner	mushrooms, cream, onic	26.6	110.2	127.604	131	FALSE	FALSE	Rainy
9988	1/1/2025	6	Fine Dining	Chicken Chop	Dinner	chicken thigh, black pep	60.8	331.816	364.648	185	FALSE	FALSE	Sunny
9989	1/1/2025	25	Fine Dining	Mushroom Soup	Dinner	mushrooms, cream, onic	26.6	142.348	229.824	40	FALSE	FALSE	Cloudy
9990	1/1/2025	3	Cafe	Beef Rendang	Dinner	beef, coconut milk, gala	68.4	229.9	327.18	60	FALSE	FALSE	Rainy
9991	1/1/2025	31	Food Stall	Kaya Toast Set	Breakfast	white bread, kaya, butte	21.28	47.272	71.06	635	FALSE	FALSE	Sunny
9992	1/1/2025	3	Cafe	Spaghetti Carbonara	Dinner	spaghetti, eggs, cheese,	68.4	274.284	210.14	364	TRUE	FALSE	Rainy
9993	1/1/2025	35	Food Stall	Char Kway Teow	Lunch	flat rice noodles, prawns	38	71.44	115.824	304	FALSE	FALSE	Cloudy
9994	1/1/2025	49	Kopitiam	Teh Tarik	Breakfast	black tea, condensed mi	6.84	14.44	23.484	950	FALSE	FALSE	Cloudy
9995	1/1/2025	24	Casual Dining	Cendol	Dinner	rice flour jelly, coconut r	15.2	46.892	75.468	418	FALSE	FALSE	Rainy
9996	1/1/2025	41	Cafe	Kaya Toast Set	Breakfast	white bread, kaya, butte	21.28	64.904	88.844	660	FALSE	FALSE	Cloudy
9997	1/1/2025	45	Food Stall	Cendol	Dinner	rice flour jelly, coconut r	15.2	33.516	33.972	588	TRUE	FALSE	Sunny
9998	1/1/2025	44	Food Stall	Teh Tarik	Lunch	black tea, condensed mi	6.84	18.544	19.836	511	FALSE	FALSE	Rainy
9999	1/1/2025	38	Fine Dining	Iced Lemon Tea	Dinner	black tea, lemon, sugar	9.12	45.904	52.516	442	FALSE	FALSE	Sunny
10000	1/1/2025	3	Cafe	Cendol	Lunch	rice flour jelly, coconut r	15.2	50.16	65.056	242	FALSE	FALSE	Sunny
10001	1/1/2025	38	Fine Dining	Iced Lemon Tea	Dinner	black tea, lemon, sugar	9.12	40.888	51.072	276	FALSE	FALSE	Cloudy
10002													
10003	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000

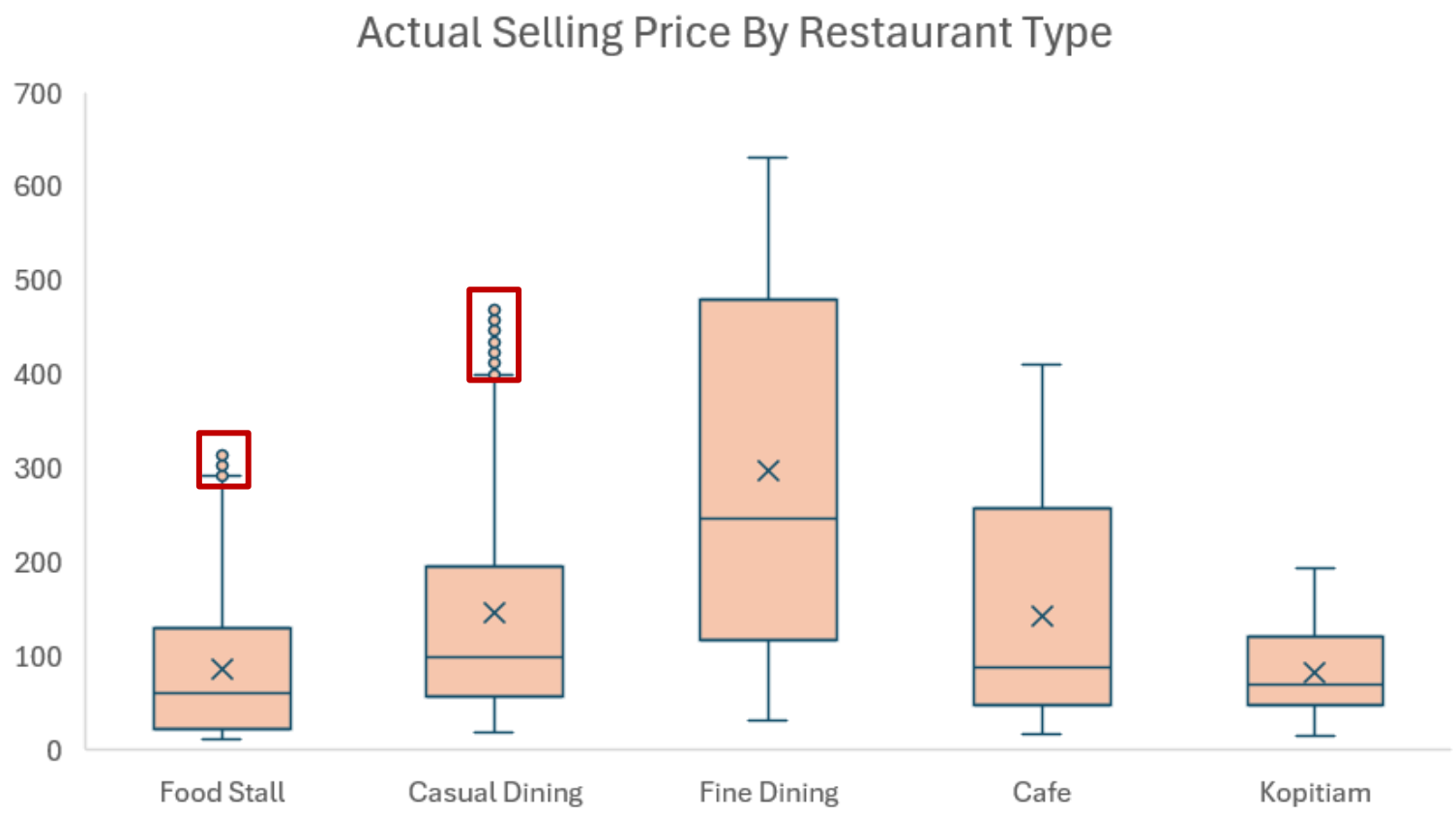
Data:

- 10,000 records
- 13 Attributes

Cleansing

- **Missing Value:** No
- **Duplicate:** No
- **Format:** Date

DATA PREPARATION (Outlier)



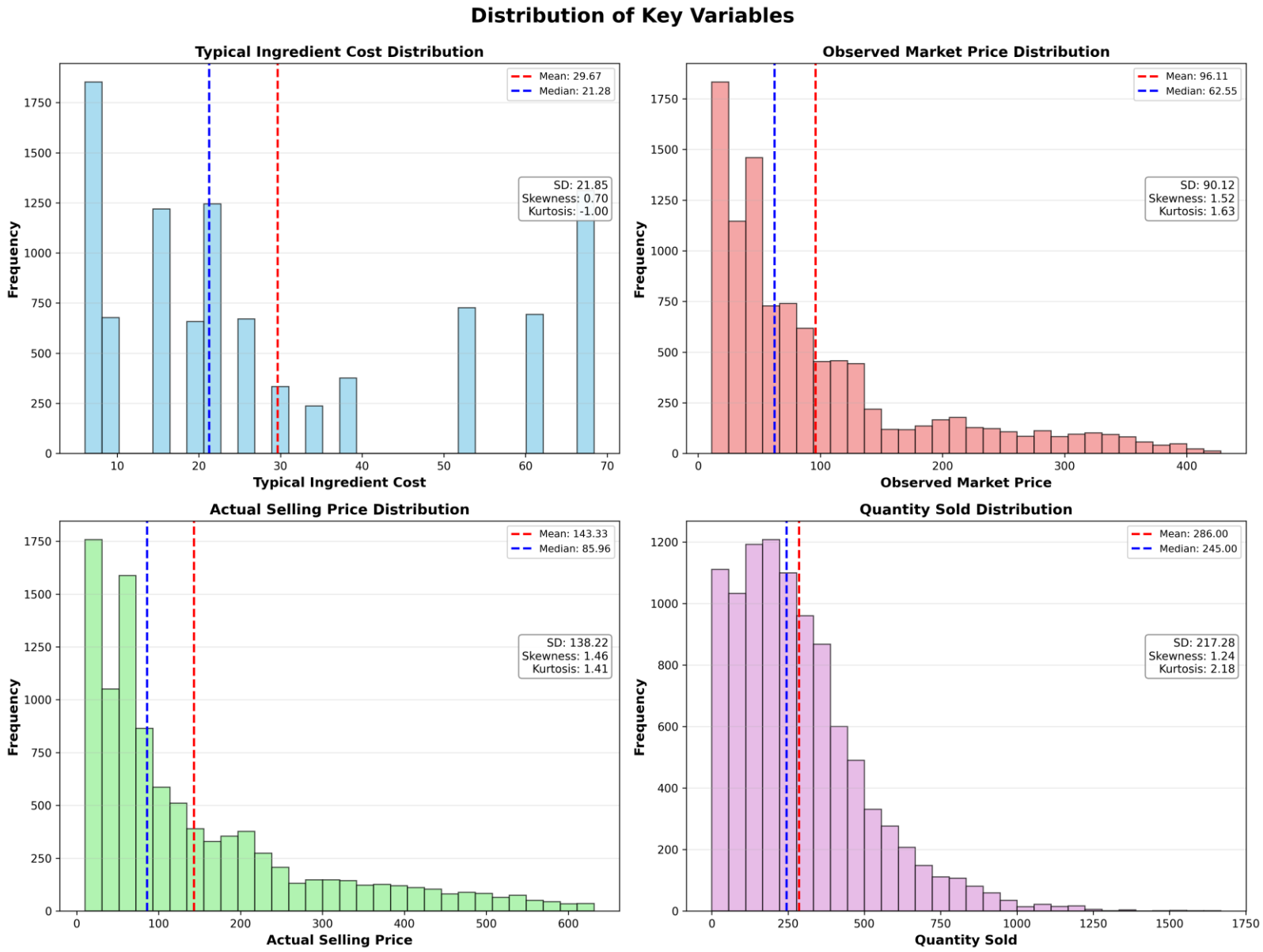
Data:

- 10,000 records
- 13 Attributes

Cleansing

- **Outlier:** Food Stall, Casual Dining (486)

EXPLORATORY DATA ANALYSIS (Histogram)



Data:

- 10,000 records
- 13 Attributes

Main Findings

- **Typical Ingredient Cost:**
  ค่าเฉลี่ย 29.67 บาท กระจุกตัวที่ 21.28 บาท
  มีการกระจายแบบ right-skewed
- **Observed Market Price:**
  ค่าเฉลี่ย 96.11 บาท กระจุกตัวที่ 62.55 บาท
  มีความแปรปรวนสูง (SD = 90.12)
- **Actual Selling Price:**
  ค่าเฉลี่ย 143.33 บาท กระจุกตัวที่ 85.96 บาท
  ราคาขายสูงกว่าราคาตลาดเฉลี่ย ~49%
- **Quantity Sold:**
  ค่าเฉลี่ย 286 หน่วย กระจุกตัวที่ 245 หน่วย มีบางวันที่ขายได้สูงสุดถึง 1,668 หน่วย

EXPLORATORY DATA ANALYSIS (Correlation)

	typical_ingredient_cost	observed_market_price	actual_selling_price	quantity_sold
typical_ingredient_cost	1.000			
observed_market_price	0.887	1.000		
actual_selling_price	0.888	0.952	1.000	
quantity_sold	-0.535	-0.464	-0.548	1.000

- **ข้อสังเกต:** ราคาทั้งหมดมีความสัมพันธ์เชิงลบกับยอดขาย แสดงว่าลูกค้ามีความอ่อนไหวต่อราคา (price sensitive)

Data:

- 10,000 records
- 13 Attributes

Main Findings

Strong Correlation

- **Observed Market Price ↔ Actual Selling Price** (r = 0.948) – ราคาตลาดและราคาขายจริงมีความสัมพันธ์สูงมาก
- **Typical Ingredient Cost ↔ Actual Selling Price** (r = 0.894) – ต้นทุนวัตถุดิบสัมพันธ์กับราคาขาย
- **Typical Ingredient Cost ↔ Observed Market Price** (r = 0.888) - ต้นทุนวัตถุดิบสัมพันธ์กับราคาตลาด

Medium Correlation

- **Actual Selling Price ↔ Quantity Sold** (r = -0.537) - ราคาขายสูง ยอดขายลดลง
- **Typical Ingredient Cost ↔ Quantity Sold** (r = -0.534) - ต้นทุนสูง ยอดขายลดลง
- **Observed Market Price ↔ Quantity Sold** (r = -0.460) - ราคาตลาดสูง ยอดขายลดลง

Dashboard

Restaurant Dashboard

REVENUE	COST	QUANTITY SOLD	GROSS PROFIT
1,036K	219K	6K	817K
(Baht)	(Baht)	(Items)	(Baht)

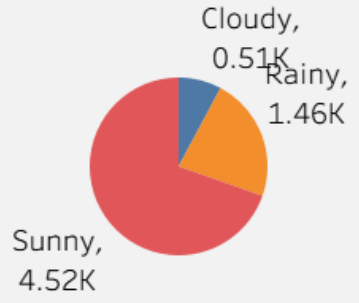
Month of Date: February

Restaurant Type: Fine Dining

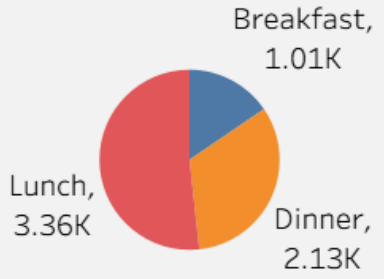
Promotion: Has Promotiton

Event: Hasn't Special Event

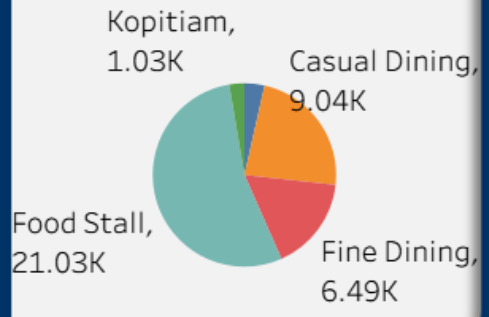
Quantity Sold By Weather



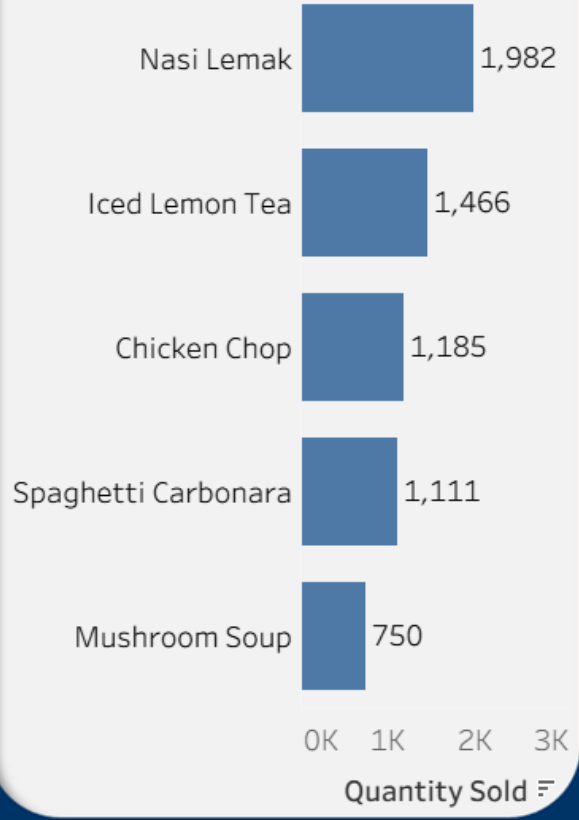
Quantity Sold By Meal Type



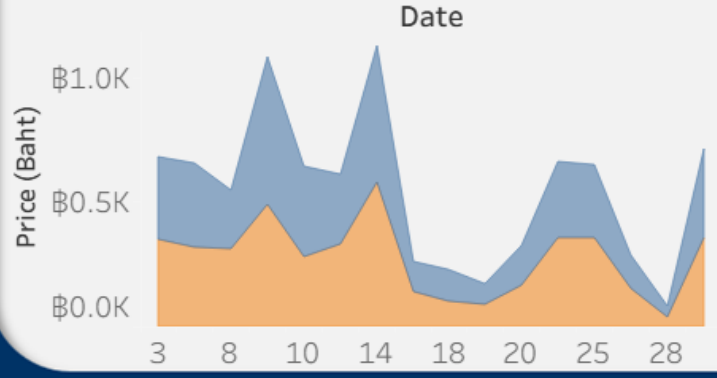
Quantity by Restaurant Type



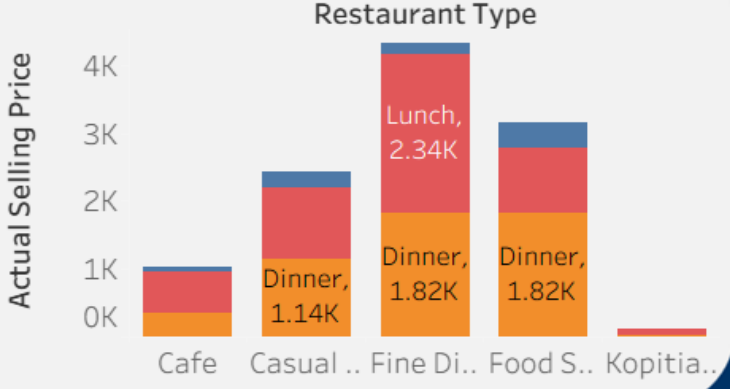
Menu



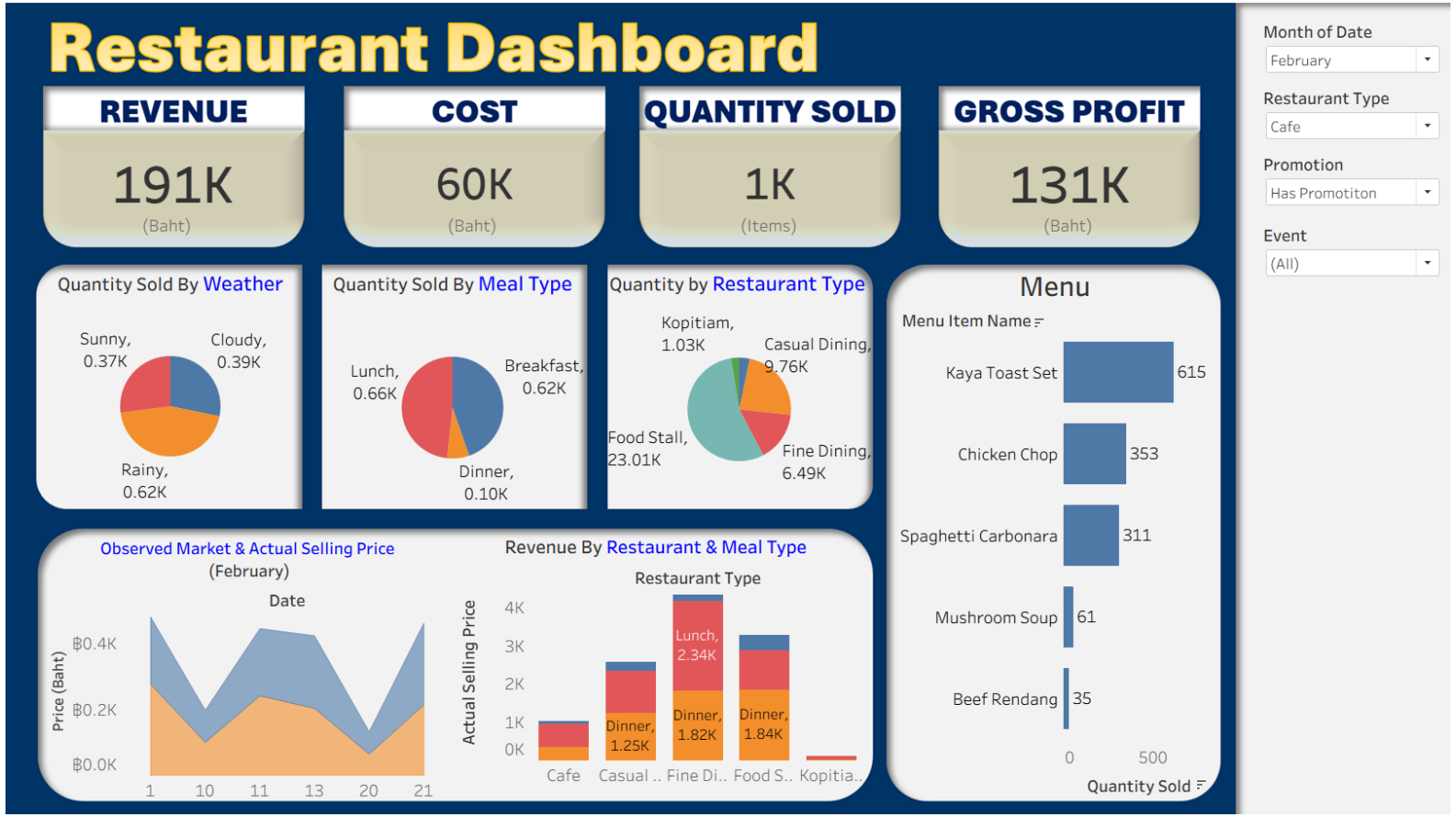
Observed Market & Actual Selling Price (February)



Revenue By Restaurant & Meal Type



DASHBOARD



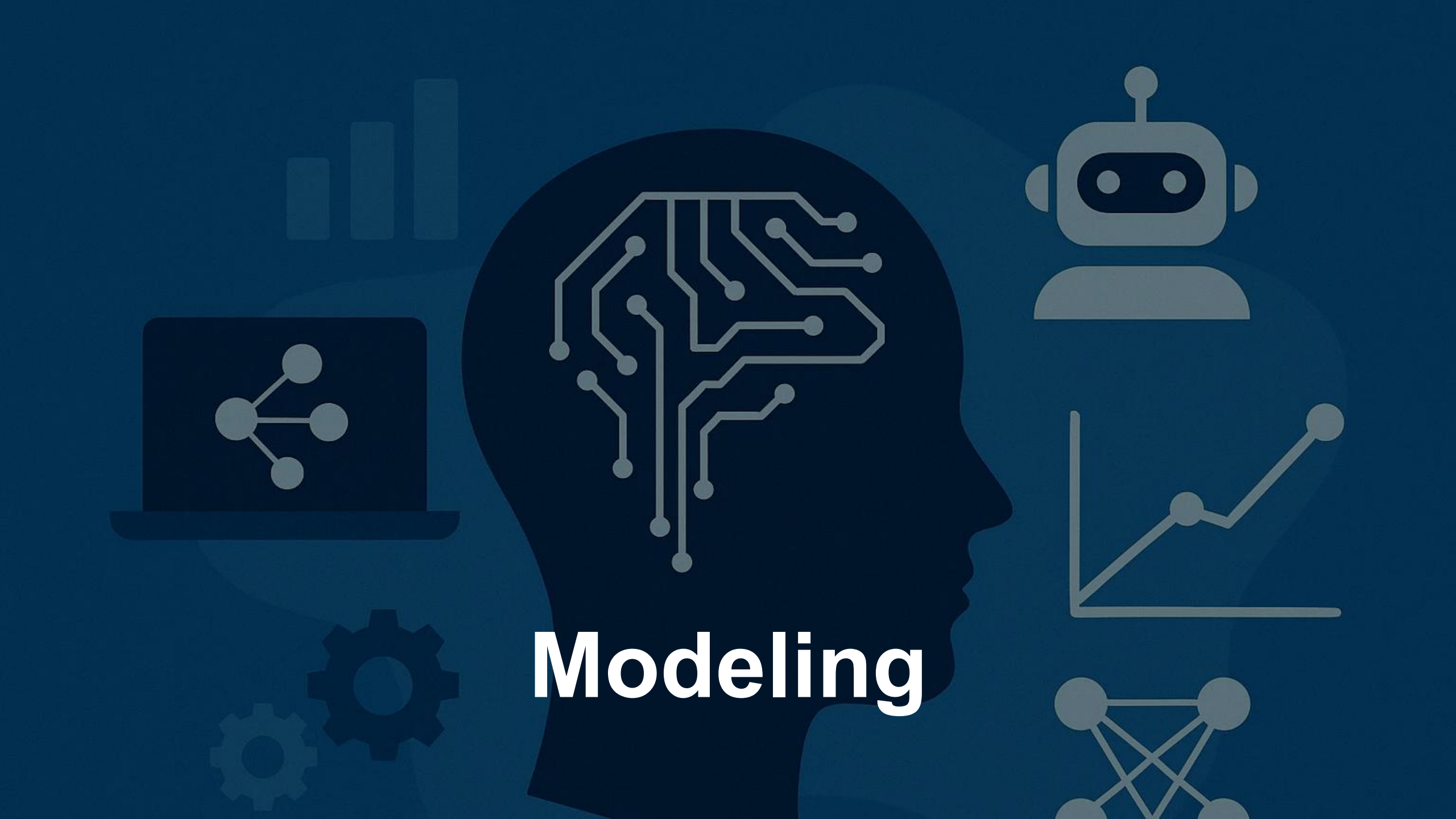
Data:

- 10,000 records
- 13 Attributes

Main Findings

จาก **Dashboard** จะพบว่า

- **สภาพอากาศ:** มีผลต่อยอดขาย โดยในวันที่อากาศ Sunny จะขายได้มากที่สุด
- **ประเภทร้านอาหารและประเภทมื้ออาหาร:** ประเภทร้าน Finding มียอดขายดีที่สุด และในแต่ละประเภทร้านมื้อกลางวันเป็นมื้อที่ทำยอดขายได้ดีที่สุด
- **โปรโมชั่น:** จะพบว่าวันที่มีโปรโมชั่นจะช่วยเพิ่มยอดขายได้มากขึ้นกว่าวันปกติ



Modeling

Data Cleaning

Import data and check “Missing Data”

```python
# import numpy, pandas, matplotlib, seaborn
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```python
# read csv file
data = pd.read_csv('/content/drive/MyDrive/DS514.csv')
data.head()
```

	date	restaurant_id	restaurant_type	menu_item_name	meal_type	key_ingredients_tags	typical_ingredient_cost	observed_market_price	actual_selling_price	quantity_sol
0	1/1/2024	11	Food Stall	Laksa	Lunch	rice noodles, fish broth, tamarind, shrimp pas...	34.20	80.484	94.848	36
1	1/1/2024	7	Casual Dining	Spaghetti Carbonara	Dinner	spaghetti, eggs, cheese, beef bacon, black pepper	68.40	202.464	459.496	10
2	1/1/2024	23	Fine Dining	Beef Rendang	Dinner	beef, coconut milk, galangal, lemongrass, spic...	68.40	375.592	609.140	3
3	1/1/2024	22	Food Stall	Roti Canai	Dinner	flour, ghee, egg, water, curry	6.08	15.504	18.848	50
4	1/1/2024	32	Fine Dining	Spaghetti Carbonara	Lunch	spaghetti, eggs, cheese, beef bacon, black pepper	68.40	306.280	422.104	26

```python
# check data.info
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   date                    10000 non-null  object
 1   restaurant_id           10000 non-null  int64
 2   restaurant_type         10000 non-null  object
 3   menu_item_name          10000 non-null  object
 4   meal_type               10000 non-null  object
 5   key_ingredients_tags    10000 non-null  object
 6   typical_ingredient_cost 10000 non-null  float64
 7   observed_market_price   10000 non-null  float64
 8   actual_selling_price    10000 non-null  float64
 9   quantity_sold           10000 non-null  int64
 10  has_promotion           10000 non-null  bool
 11  special_event           10000 non-null  bool
 12  weather_condition       10000 non-null  object
dtypes: bool(2), float64(3), int64(2), object(6)
memory usage: 879.0+ KB
```

```python
# check data is NULL
data.isnull().sum()
```

	0
date	0
restaurant_id	0
restaurant_type	0
menu_item_name	0
meal_type	0
key_ingredients_tags	0
typical_ingredient_cost	0
observed_market_price	0
actual_selling_price	0
quantity_sold	0
has_promotion	0
special_event	0
weather_condition	0

dtype: int64

- Import important library
- Import data from csv file
- Check “Missing Data” → No “NULL” value
- Data: 10,000 records / 13 columns

Data Cleaning

Check and change data type

```python
# change date from 'object' to 'date'
data['date'] = pd.to_datetime(data['date'])
data.info()
```

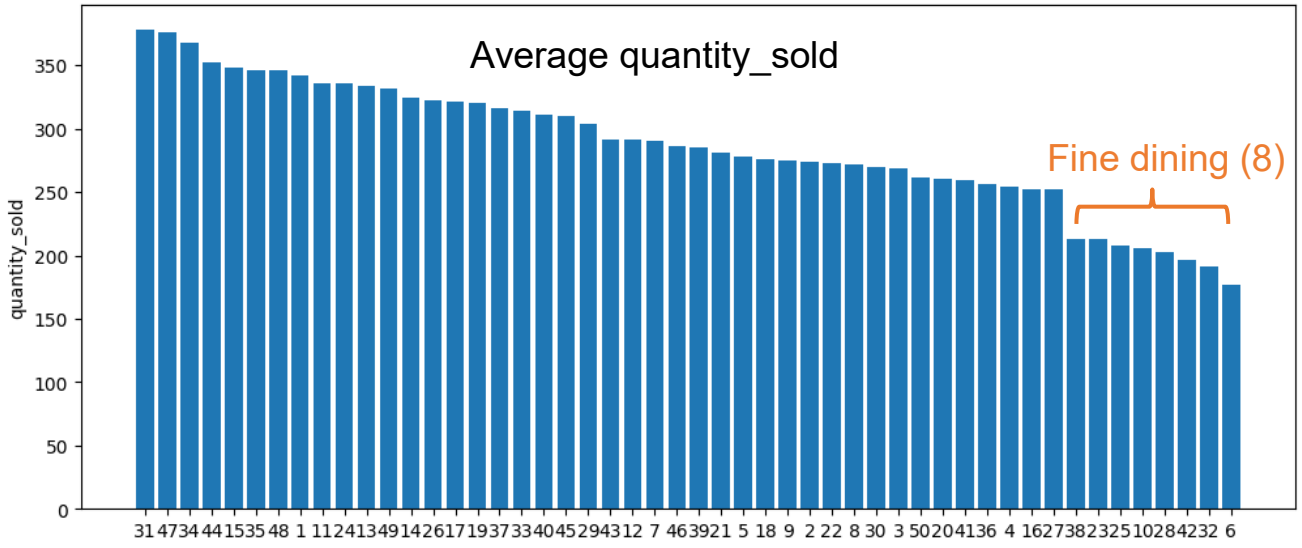
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   date                     10000 non-null  datetime64[ns]
 1   restaurant_id            10000 non-null  int64
 2   restaurant_type          10000 non-null  object
 3   menu_item_name           10000 non-null  object
 4   meal_type                10000 non-null  object
 5   key_ingredients_tags     10000 non-null  object
 6   typical_ingredient_cost  10000 non-null  float64
 7   observed_market_price    10000 non-null  float64
 8   actual_selling_price     10000 non-null  float64
 9   quantity_sold            10000 non-null  int64
 10  has_promotion            10000 non-null  bool
 11  special_event            10000 non-null  bool
 12  weather_condition        10000 non-null  object
dtypes: bool(2), datetime64[ns](1), float64(3), int64(2), object(5)
memory usage: 879.0+ KB
```

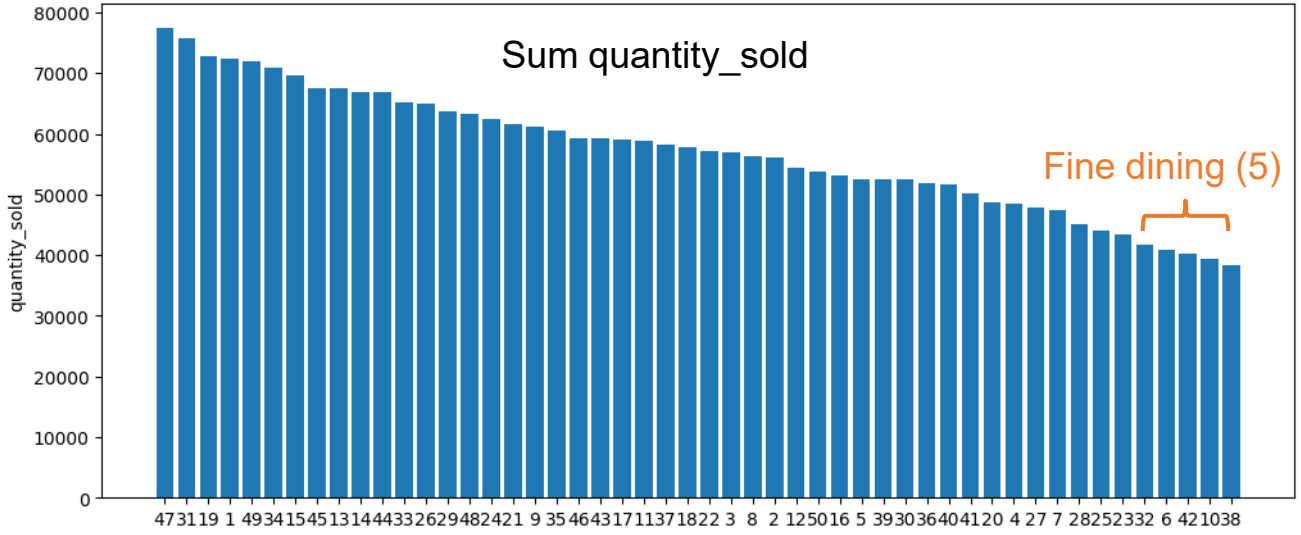
```python
# change restaurant_id from 'int' to 'object'
data['restaurant_id'] = data['restaurant_id'].astype(str)
data.info()
```

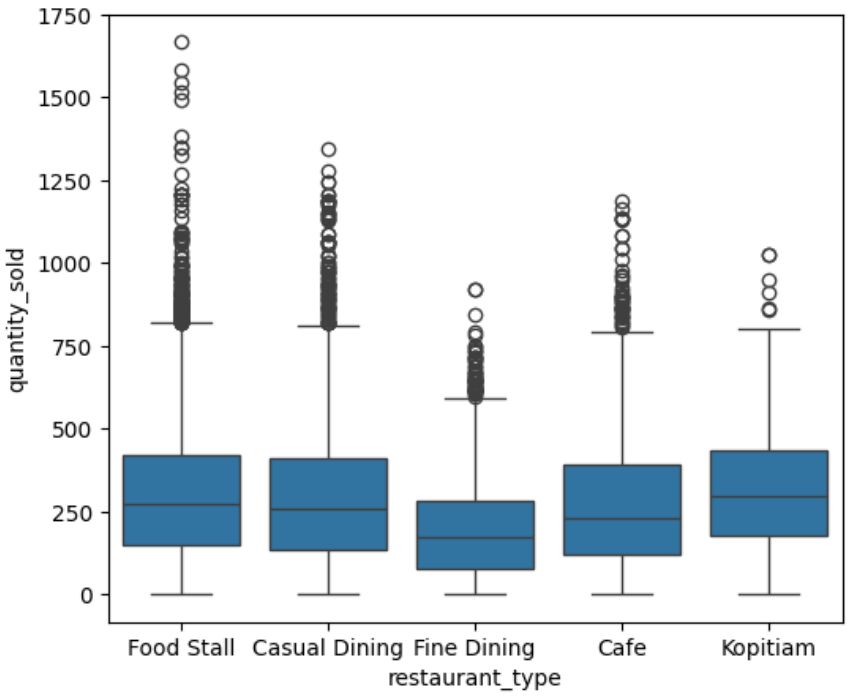
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   date                     10000 non-null  datetime64[ns]
 1   restaurant_id            10000 non-null  object
 2   restaurant_type          10000 non-null  object
 3   menu_item_name           10000 non-null  object
 4   meal_type                10000 non-null  object
 5   key_ingredients_tags     10000 non-null  object
 6   typical_ingredient_cost  10000 non-null  float64
 7   observed_market_price    10000 non-null  float64
 8   actual_selling_price     10000 non-null  float64
 9   quantity_sold            10000 non-null  int64
 10  has_promotion            10000 non-null  bool
 11  special_event            10000 non-null  bool
 12  weather_condition        10000 non-null  object
dtypes: bool(2), datetime64[ns](1), float64(3), int64(1), object(6)
memory usage: 879.0+ KB
```

- Change date from “object” to “datetime64[ns]”
- Change restaurant_id from “int64” to “object”

EDA 1 – Overview of Quantity Sold

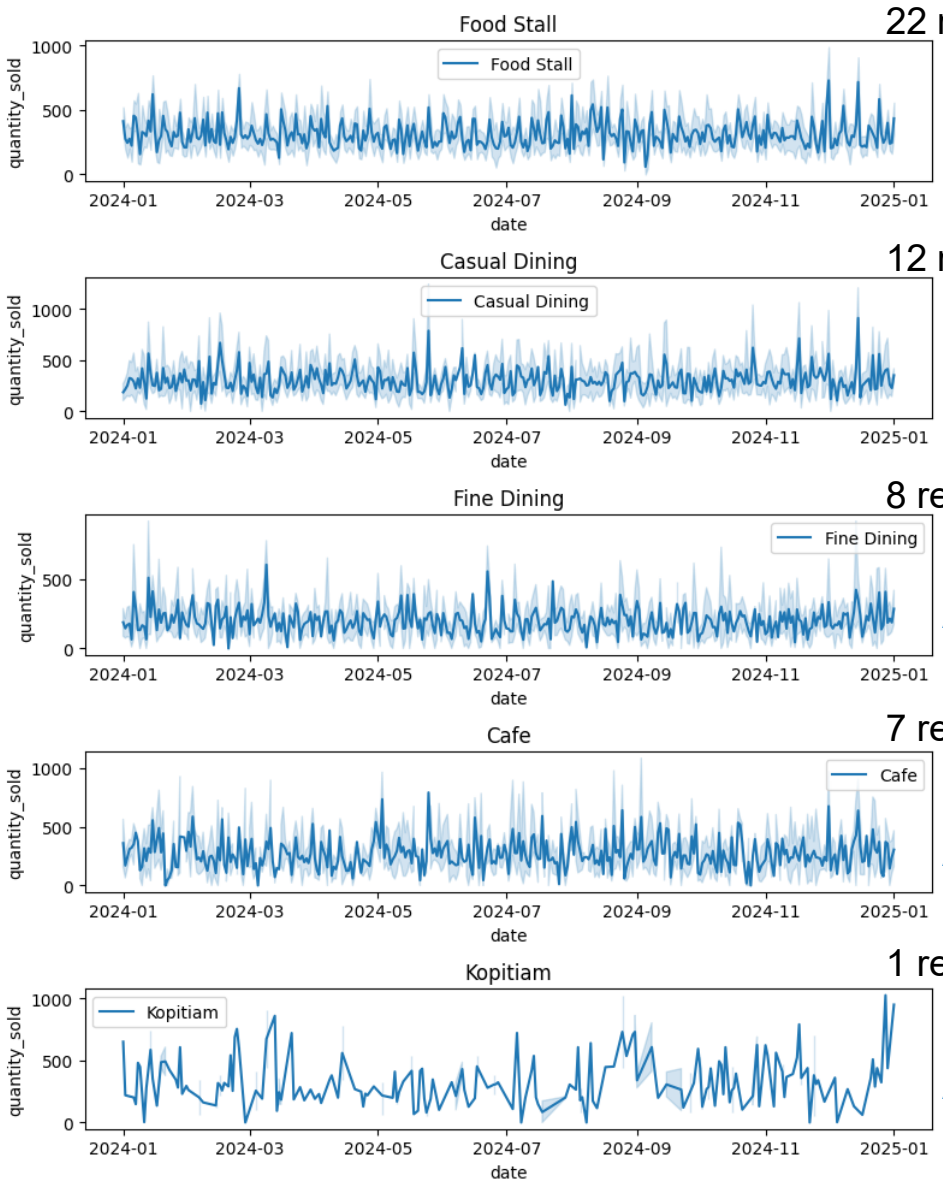


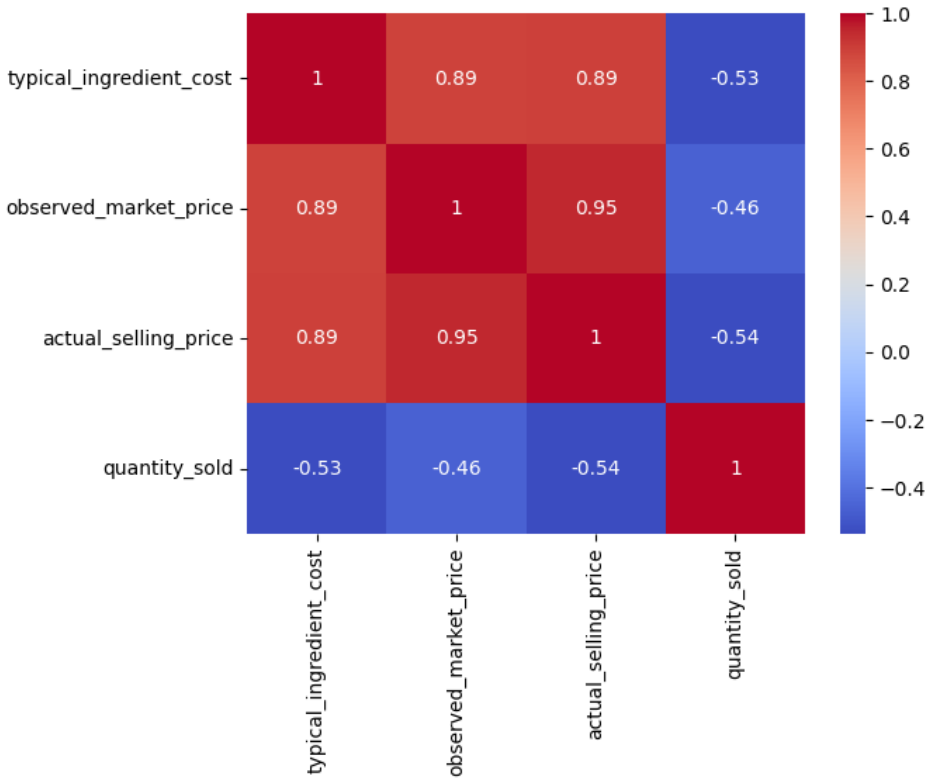




- Average and sum quantity_sold show that fine dining ranks the lowest among the 50 restaurants.
- Box-whisker plot shows that the median value for fine dining is lower than the other four restaurant types.

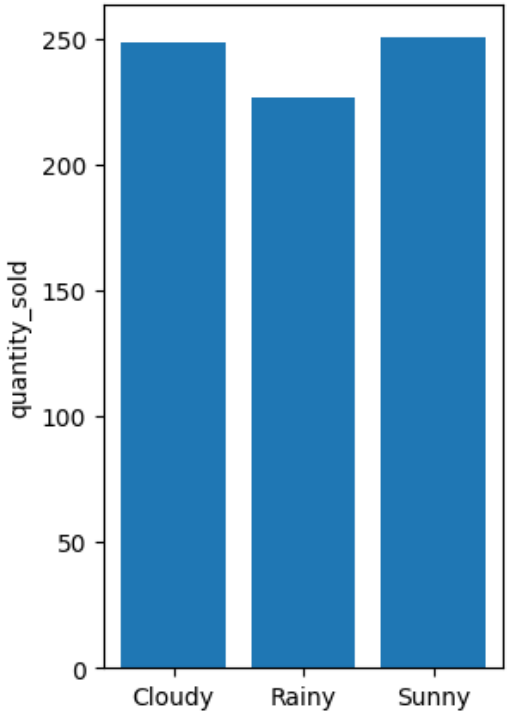
EDA 2 – Analysis of 1-year Trend and Correlation

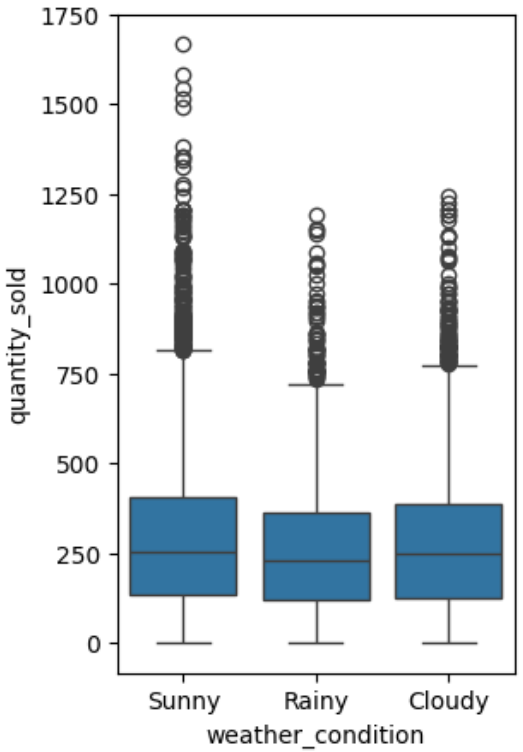


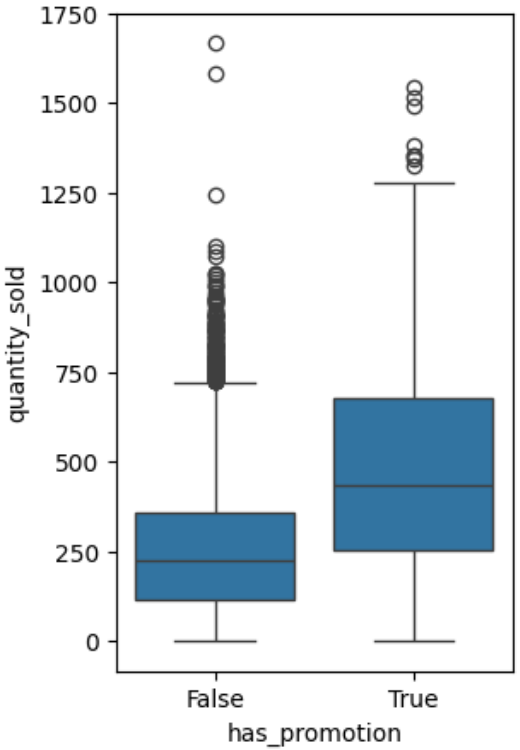


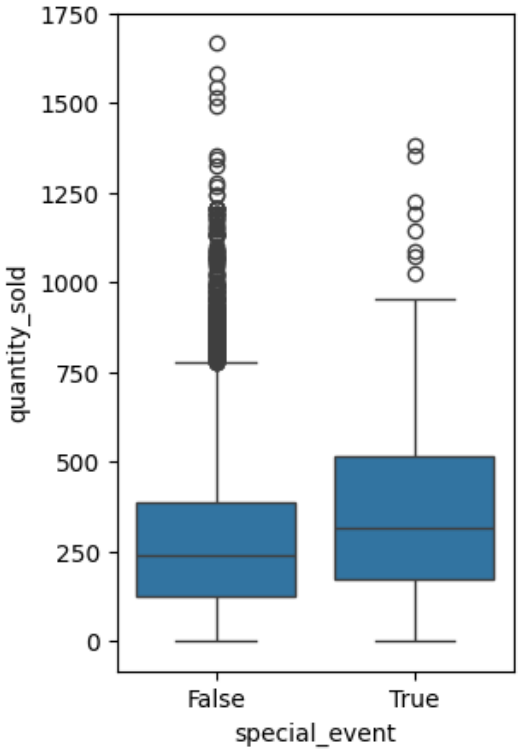
- No obvious trend or pattern is observed in the time-series plot, but the annual average quantity_sold for fine dining is significantly lower than that of others.
- Correlation analysis indicates that cost, market_price, and selling_price are strongly positively correlated. However, quantity_sold shows a moderate negative correlation with others

EDA 3 – Effects of Weather/Promotion/Event









- The average quantity_sold on sunny (251) and cloudy (249) days is higher than on rainy days (227).
- However, the difference is very small, which makes it difficult to observe in the box-whisker plot.

- The average quantity_sold on promotion days (482) is higher than on non-promotion days (251).
- Similarly, the average quantity_sold on days with special events (363) is also higher than on days without events (282).

Data and Feature Selection

Select only the last five restaurants (restaurant_id = ‘32’ ‘6’ ‘42’ ‘10’ ‘38’) to build the model for predicting quantity_sold, and then use the model to identify ways to improve sales performance.

```
# Prepare more data for modeling
# focus on restaurant_id '32' '6' '42' '10 '38'
model_data = data[data['restaurant_id'].isin(['32', '6', '42', '10', '38'])]
model_data.head()
```

	date	restaurant_id	restaurant_type	menu_item_name	meal_type	key_ingredients_tags	typical_ingredient_cost	observed_market_price	actual_selling_price	quantity_sold	has_promotion	specia
4	2024-01-01	32	Fine Dining	Spaghetti Carbonara	Lunch	spaghetti, eggs, cheese, beef bacon, black pepper	68.4	306.280	422.104	262	False	
23	2024-01-01	42	Fine Dining	Spaghetti Carbonara	Dinner	spaghetti, eggs, cheese, beef bacon, black pepper	68.4	308.864	431.832	143	False	
34	2024-01-02	6	Fine Dining	Mushroom Soup	Dinner	mushrooms, cream, onion, garlic, vegetable broth	26.6	114.608	222.528	30	False	
36	2024-01-02	42	Fine Dining	Nasi Lemak	Lunch	rice, coconut milk, sambal, anchovies, egg, pe...	19.0	102.372	82.764	238	True	
40	2024-01-02	42	Fine Dining	Beef Rendang	Dinner	beef, coconut milk, galangal, lemongrass, spic...	68.4	390.488	508.288	126	False	

restaurant_id	0
10	192
32	218
38	180
42	205
6	232

Total 1,027 records

numeric

categorical

Quantity_sold = f(lag1, actual_selling_price, weather_condition, has_promotion, special_event)

Target

Feature

Data and Feature Selection

Create ‘lag1’ as another feature for modeling

```
# create lag1 of quantity_sold and diff(quantity_sold and lag1) of model_data
model_data['lag1'] = model_data.groupby(['restaurant_id','menu_item_name'])['quantity_sold'].shift(1)
model_data['diff'] = model_data['quantity_sold'] - model_data['lag1']
model_data.tail()
```

```
gWithCopyWarning:
a slice from a DataFrame.
 value instead

ps://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(['restaurant_id','menu_item_name'])['quantity_sold'].shift(1)
gWithCopyWarning:
a slice from a DataFrame.
 value instead

ps://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
ty_sold'] - model_data['lag1']
```

pe	menu_item_name	meal_type	key_ingredients_tags	typical_ingredient_cost	observed_market_price	actual_selling_price	quantity_sold	has_promotion	special_event	weather_condition	lag1
ng	Mushroom Soup	Lunch	mushrooms, cream, onion, garlic, vegetable broth	26.60	119.852	235.904	74	False	False	Sunny	171.0
ng	Chicken Chop	Dinner	chicken thigh, black pepper sauce, fries, cole...	60.80	301.796	364.800	186	False	False	Cloudy	190.0
ng	Chicken Chop	Dinner	chicken thigh, black pepper sauce, fries, cole...	60.80	331.816	364.648	185	False	False	Sunny	381.0
ng	Iced Lemon Tea	Dinner	black tea, lemon, sugar syrup	9.12	45.904	52.516	442	False	False	Sunny	460.0
ng	Iced Lemon Tea	Dinner	black tea, lemon, sugar syrup	9.12	40.888	51.072	276	False	False	Cloudy	442.0

Data and Feature Selection

```python
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.metrics import r2_score, mean_absolute_error, root_mean_squared_error
```

Import library

```python
# --- 1. Feature and Target Definition ---

# Define the target variable (y) and the predictor variables (X)
X = model_data[['lag1', 'actual_selling_price', 'weather_condition', 'has_promotion', 'special_event']]
y = model_data['quantity_sold']

# Remove NULL value from lag1
X = X.dropna()
y = y[X.index]

# Split data into training and testing sets (80/20 split)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

# shuffle=False is crucial for time-series data to maintain chronological order

# Define feature types for the ColumnTransformer
numerical_features = ['lag1', 'actual_selling_price']
categorical_features = ['weather_condition', 'has_promotion', 'special_event']
```

Set “features” and “target”

Remove “NULL” value of lag1

Train/test split (80/20)

Define numerical features / categorical features

```python
# summary data_size of X_train, X_test, y_train, y_test
print("X_train size:", X_train.shape)
print("X_test size:", X_test.shape)
print("y_train size:", y_train.shape)
print("y_test size:", y_test.shape)
```

```
X_train size: (800, 5)
X_test size: (201, 5)
y_train size: (800,)
y_test size: (201,)
```

Check size of data

Data Pre-processing / Linear Regression

```python
# --- 2. Data Preprocessing Pipeline (Scaling and Encoding) ---

# Create the preprocessing pipeline using ColumnTransformer
# Apply Standard Scaling to numerical features
# Apply One-Hot Encoding to categorical features (handle_unknown='ignore' prevents error on unseen category)

preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False, drop='first'), categorical_features)
    ],
    remainder='passthrough' # Keep any columns not specified (if any)
)
```

Pre-processing
Numerical features -> Standard Scaler
Categorical features -> One-Hot Encoder

```python
# --- 3. Model 1: Baseline Linear Regression ---

# Create the full pipeline for Linear Regression
lr_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                              ('regressor', LinearRegression())])

print("--- Baseline Linear Regression ---")
lr_pipeline.fit(X_train, y_train)
lr_predictions = lr_pipeline.predict(X_test)
print('Coefficients:', lr_pipeline.named_steps['regressor'].coef_)
print('Intercept:', lr_pipeline.named_steps['regressor'].intercept_)
print("R2 Score:", r2_score(y_test, lr_predictions))
print("MAE:", mean_absolute_error(y_test, lr_predictions))
print("RMSE:", root_mean_squared_error(y_test, lr_predictions))
```

```
--- Baseline Linear Regression ---
Coefficients: [ 37.0161553  -62.20757124  14.57398616  13.10039712  86.55420265
  97.70211191]
Intercept: 163.90107045745583
R2 Score: 0.4315164663470321
MAE: 96.96116652820474
RMSE: 127.86589616291008
```

Pipeline = preprocessor + linear regression

	lag1	selling_price	weather		promotion	event
Linear Regression Coefficients:	[37.02	-62.21	14.57	13.10	86.55	97.70]
Intercept:	163.90					
R2 Score:	0.4315					
MAE:	96.96					
RMSE:	127.86					

Regularized Linear Regression (polynomial features)

```python
# --- 4. Model 2 Ridge Regression with polynomial feature ---

from sklearn.preprocessing import PolynomialFeatures

polynomial_ridge_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('polynomialfeatures', PolynomialFeatures()), # Placeholder, degree will be tuned
    ('regressor', Ridge())
])

param_grid = {
    # Alpha values for Ridge regularization (log scale for better searching)
    'regressor__alpha': np.logspace(-6, 3, 10),
    # Polynomial degrees to test (1 to 10)
    'polynomialfeatures__degree': np.arange(1, 11)
}

print("--- Ridge Regression with Polynomial Features Tuning ---")
grid_search = GridSearchCV(
    polynomial_ridge_pipeline,
    param_grid=param_grid,
    cv=5,
    scoring='r2'
)

grid_search.fit(X_train, y_train)

Ridge_best_estimator = grid_search.best_estimator_
Ridge_predictions = Ridge_best_estimator.predict(X_test)

best_params = grid_search.best_params_
best_degree = best_params['polynomialfeatures__degree']
best_alpha = best_params['regressor__alpha']

# Evaluation
r2_train = r2_score(y_train, Ridge_best_estimator.predict(X_train))
r2_test = r2_score(y_test, Ridge_predictions)
mae_train = mean_absolute_error(y_train, Ridge_best_estimator.predict(X_train))
mae_test = mean_absolute_error(y_test, Ridge_predictions)
rmse_train = root_mean_squared_error(y_train, Ridge_best_estimator.predict(X_train))
rmse_test = root_mean_squared_error(y_test, Ridge_predictions)
```

Pipeline = preprocessor + polynomial feature + Ridge/Lasso/Elastic net

Hyperparameter tuning (param_grid)

Ridge/Lasso/Elastic net	polynomial degree = 1 to 10
Ridge/Lasso/Elastic net	α = 10^-6 to 1000
Elastic net	l1 ratio = 0 to 1

GridSearchCV = pipeline + param_grid + cv = 5 + scoring ‘r2’

Find the best hyperparameters (polynomial degree, α, l1 ratio)

Evaluation

R2, MAE, RMSE

Results from Models

--- Ridge ---

Best Polynomial Degree:	3
Best Alpha:	100
R2 Score on Training Set:	0.5365
R2 Score on Test Set:	0.5444
MAE on Training Set:	79.17
MAE on Test Set:	87.38
RMSE on Training Set:	101.38
RMSE on Test Set:	114.47

--- Lasso ---

Best Polynomial Degree:	3
Best Alpha:	1
R2 Score on Training Set:	0.5470
R2 Score on Test Set:	**0.5639**
MAE on Training Set:	78.91
MAE on Test Set:	**86.09**
RMSE on Training Set:	100.23
RMSE on Test Set:	**111.99**

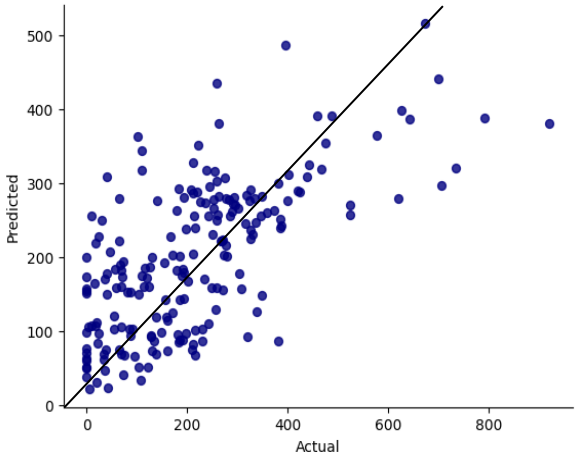
--- Elastic Net ---

Best Polynomial Degree:	4
Best Alpha:	0.01
Best L1 Ratio:	0.33
R2 Score on Training Set:	0.6192
R2 Score on Test Set:	0.5285
MAE on Training Set:	72.14
MAE on Test Set:	88.20
RMSE on Training Set:	91.89
RMSE on Test Set:	116.45

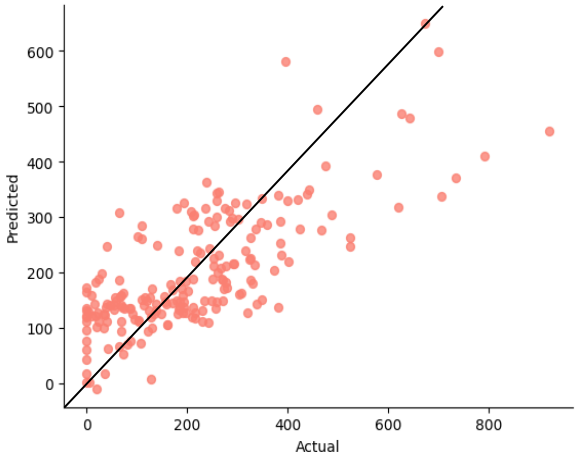
Lasso regression achieved the best performance on the test set, with the highest R^2 of 0.5639 using an alpha of 1 and a polynomial degree of 3.

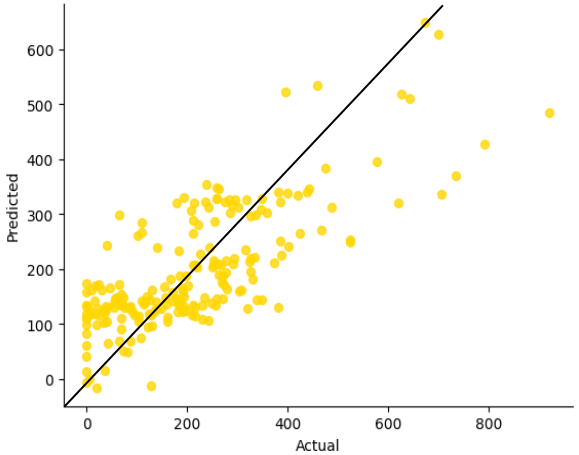
Results from Models

Linear regression
R^2 = 0.4315

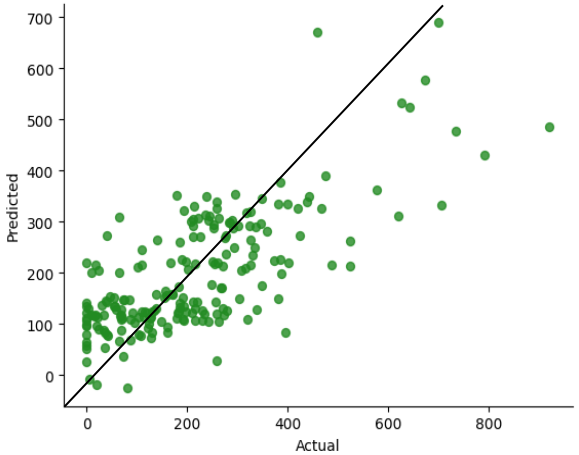


Ridge regression
R^2 = 0.5444





Lasso regression
R^2 = 0.5639



Elastic net regression
R^2 = 0.5285

- Lasso regression เป็นโมเดลทำนายปริมาณการขายที่ให้ค่า R^2 สูงสุด รวมถึง MAE และ RMSE น้อยที่สุด

ข้อเสนอแนะสำหรับการพัฒนาโมเดล

- อาจพิจารณาใช้ data ที่เป็น numerical แทน categorical เพื่อความแม่นยำในการทำนาย เช่น อุณหภูมิ ปริมาณฝน และ เปอร์เซ็นต์ส่วนลดค่าอาหาร
- ใช้การ split data ตามเวลา เนื่องจากข้อมูลเป็น time-series แทน random split
- พิจารณาโมเดล regression อื่น ๆ ร่วม เช่น kNN, SVM, และ อื่นๆ
- พิจารณาใช้การทำนายแบบ regression ร่วมกับการตีความปัจจัยที่ส่งผลต่อยอดขายจากโมเดลด้วย SHAP

DATA ANALYTICS/SCIENCE PROJECT CANVAS

Designed by: ขนิษฐา, ธารารัตน์, ธิติวุฒิ

Date: 10 Dec 2025

Title: การวิเคราะห์ปัจจัยที่มีผลต่อยอดขายและการทำนายปริมาณการขายของร้านอาหารเพื่อวางแผนกลยุทธ์กระตุ้นยอดขาย

1. Background / Problem Statement

Background:

- ธุรกิจร้านอาหารต้องเผชิญความท้าทายในการบริหารจัดการยอดขาย การกำหนดราคา และการวางแผนโปรโมชั่นท่ามกลางปัจจัยภายนอกที่เปลี่ยนแปลง

Problem Statement:

- ขาดเครื่องมือคาดการณ์ยอดขายที่แม่นยำ
- การกำหนดราคาและโปรโมชั่นยังไม่อิงข้อมูล
- ไม่ทราบปัจจัยที่ส่งผลต่อยอดขายและพฤติกรรมลูกค้า

2. Questions/Hypothesis

Research Questions:

1. ปัจจัยใดบ้างที่มีผลกระทบต่อยอดขายของร้านอาหาร
   - สภาพอากาศ โปรโมชั่น และเหตุการณ์พิเศษส่งผลต่อยอดขายอย่างไร
   - ประเภทร้านอาหารและประเภทมื้ออาหารมีความสัมพันธ์กับยอดขายหรือไม่
2. กลยุทธ์การกำหนดราคาและโปรโมชั่นมีประสิทธิภาพเพียงใด
   - ความแตกต่างระหว่างราคาตลาดกับราคาขายจริงส่งผลต่อยอดขายอย่างไร
   - โปรโมชั่นช่วยเพิ่มยอดขายและผลกำไรได้จริงหรือไม่

3. Value Propositions

1. **ผู้ประกอบการร้านอาหาร**: เข้าใจปัจจัยที่ส่งผลต่อยอดขายและพฤติกรรมลูกค้า ช่วยในการวางแผนกลยุทธ์ราคา โปรโมชั่น และการบริหารเมนูอย่างมีประสิทธิภาพ
2. **นักการตลาด**: ได้ข้อมูลเชิงลึกเกี่ยวกับประสิทธิภาพของโปรโมชั่นและปัจจัยภายนอกที่ส่งผลต่อการตัดสินใจซื้อของลูกค้า เพื่อวางแผนแคมเปญที่ตรงเป้าหมาย

4. Data Sources/Attributes

- Data Sources: Kaggle (Link) Restaurant Sales Report 2024-2025
- Attributes: 13
- Records: 10,000

- Target: ปริมาณการขาย (quantity_sold)
- Features:
  ยอดขายวันก่อนหน้า (lag1), ราคาขาย (selling price), สภาพอากาศ, มีโปรโมชัน, เหตุการณ์พิเศษ (event)

5. Analysis/Model Development

1. Analysis:
- Descriptive Statistics
- Correlation Analysis

2. Results:
- Data Visualization: Tableau
- Summary Statistics

3. Modeling (Supervised learning)
- linear regression
- regularized linear regression (ridge, lasso, elastic net)

4. Model evaluation: R^2, MAE, RMSE

6. Findings and Insights

ปัจจัยที่ส่งผลต่อยอดขาย

สภาพอากาศแดดให้ยอดขายสูงสุด 298 หน่วย (+13% จากวันฝน) โปรโมชั่นเพิ่มยอดขาย +92% และเหตุการณ์พิเศษเพิ่ม +29% Fine Dining มีกำไรสูงสุด 255 บาท แต่ขายน้อย ขณะที่ Food Stall ขายมาก 311 หน่วย แต่กำไรต่ำ 61 บาท อาหารเช้าขายได้มากกว่าเที่ยง-เย็น 85-119%

ประสิทธิภาพการกำหนดราคาและโปรโมชั่น

ราคามีความสัมพันธ์เชิงลบกับยอดขาย ($r = -0.54$) การให้ส่วนลดเพิ่มยอดขาย +97% แต่ลดกำไรต่อหน่วย -55% โปรโมชั่นเพิ่มยอดขาย +92% แต่ลดกำไรต่อหน่วย -39% (จาก 121 เหลือ 74 บาท) อย่างไรก็ตามกำไรรวมยังเพิ่มขึ้นเนื่องจากปริมาณขายที่สูงขึ้นมาก

การทำนายปริมาณการขายของร้านอาหาร

Lasso regression เป็นโมเดลทำนายปริมาณาการขายที่ให้ค่า R^2 สูงสุด รวมถึง MAE และ RMSE น้อยที่สุด แต่ค่า R^2 ที่ได้ยังต่ำ มีค่าเพียง 0.5639 ซึ่งอาจจะต้องใช้ SHAP มาช่วยดู Feature Importance ที่ส่งผลต่อโมเดลประกอบ และ อาจพัฒนาโมเดลด้วยเทคนิค regression อื่น ๆ รวมถึงการแบ่ง data แบบ Time-series แทนการ random split เพื่อเพิ่มความแม่นยำของโมเดล

7. Recommendation/Action and Impact

- ปรับกลยุทธ์โปรโมชั่นให้มีประสิทธิภาพ
- บริหารจัดการตามสภาพอากาศและเหตุการณ์พิเศษ
- Menu Engineering และ Dynamic Pricing
- กระจายยอดขายตลอดทั้งวัน

Link Github [Click]

ThitiwutM / Restaurant-data-insights-and-prediction

Type / to search

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

Files

main

Go to file t

- DS514 Quantity_sold prediction.i...
- README.md
- data distribution.jpg
- model results.jpg
- project scope.jpg
- restaurant_dataset.csv

Restaurant-data-insights-and-prediction / README.md

ThitiwutM add picture — 7d50603 · 12 minutes ago — History

Preview | Code | Blame — 56 lines (31 loc) · 7.4 KB — Raw

Restaurant-data-insights-and-prediction

This is a term project for the DS512/513/514/515 Data Analytics and Data Science course at SWU, Thailand.

The project aims to explore insights from restaurant sales data and identify the key factors influencing sales volume. In addition, we developed predictive models for quantity sold using linear regression and regularized linear regression techniques (Ridge, Lasso, and Elastic Net) to support restaurants in improving their performance and developing effective strategies to enhance sales.

The process includes data type checking, data cleaning, exploratory data analysis (EDA), data visualization and dashboard creation, feature selection, data pre-processing before modeling, data pipeline development, model building, and model evaluation.

โครงงานนี้เป็นโครงงานของรายวิชา DS512/513/514/515 Data Analytics and Data Science ระดับปริญญาโท มหาวิทยาลัยศรีนครินทรวิโรฒ

โครงงานนี้มีวัตถุประสงค์เพื่อสำรวจข้อมูลยอดขายของร้านอาหารและหาปัจจัยสำคัญที่มีผลต่อปริมาณการขาย นอกจากนี้ยังได้พัฒนาโมเดลสำหรับการทำนายปริมาณการขาย (Quantity sold) โดยใช้เทคนิค Linear Regression และ Regularized Linear Regression (Ridge, Lasso และ Elastic Net) เพื่อสนับสนุนร้านอาหารในการปรับปรุงประสิทธิภาพและพัฒนากลยุทธ์ที่เหมาะสมในการเพิ่มยอดขาย

กระบวนการทำงานประกอบด้วย การตรวจสอบประเภทข้อมูล การทำความสะอาดข้อมูล การวิเคราะห์ข้อมูลเบื้องต้น (EDA) การสร้างภาพข้อมูลและแดชบอร์ด การคัดเลือกตัวแปรสำหรับโมเดล การเตรียมข้อมูลก่อนสร้างโมเดล การพัฒนา data pipeline การสร้างโมเดล และการประเมินผลโมเดล