客戶個性分析 - 多元線性回歸

CRISP-DM 流程

- **Business Understanding**: 透過分析客戶的人口統計、消費行為和促銷活動反應，來預測客戶的總消費金額。目標是找出影響消費的關鍵因素，以制定更精準的行銷策略。
- **Data Understanding**: 探索 `marketing_campaign.csv` 資料集，了解各個欄位的意義、資料類型和基本統計特性。
- **Data Preparation**: 處理缺失值、進行特徵工程 (如計算年齡、家庭總人數、客戶資歷、總消費金額)、並對類別變數進行編碼。
- **Modeling**: 使用多元線性回歸模型進行預測。我將會進行特徵選擇，以找出對預測最有幫助的特徵。
- **Evaluation**: 使用 R-squared, MSE, RMSE, MAE 等指標評估模型效能，並透過視覺化方式呈現預測結果與殘差分析。
- **Deployment**: 在這個 Notebook 的最後，我會總結分析結果，並解釋模型係數的商業意涵。

1. 資料理解 (Data Understanding)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime

sns.set_style('whitegrid')
plt.rcParams['axes.unicode_minus'] = False

print("套件匯入完成")
```

```
套件匯入完成
```

```
try:
    df = pd.read_csv('marketing_campaign.csv', sep='\t')
    print("資料集 marketing_campaign.csv 載入成功")
except FileNotFoundError:
    print("錯誤：找不到 marketing_campaign.csv 檔案。請確認檔案已放置在正確的目錄下。")
```

```
資料集 marketing_campaign.csv 載入成功
```

```
if 'df' in locals():
    print("資料集基本資訊：")
    df.info()
    print("\n資料集描述性統計：")
    display(df.describe())
```

資料集基本資訊：

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

資料集描述性統計：

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	M
count	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	
mean	5592.159821	1968.805804	52247.251354	0.444196	0.506250	49.109375	303.935714	26.302232	166.950000	
std	3246.662198	11.984069	25173.076661	0.538398	0.544538	28.962453	336.597393	39.773434	225.715373	
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	2828.250000	1959.000000	35303.000000	0.000000	0.000000	24.000000	23.750000	1.000000	16.000000	
50%	5458.500000	1970.000000	51381.500000	0.000000	0.000000	49.000000	173.500000	8.000000	67.000000	
75%	8427.750000	1977.000000	68522.000000	1.000000	1.000000	74.000000	504.250000	33.000000	232.000000	
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	99.000000	1493.000000	199.000000	1725.000000	

8 rows × 26 columns

```
if 'df' in locals():
    print("缺失值檢查：")
    print(df.isnull().sum())
```

```
缺失值檢查：
ID                     0
Year_Birth             0
Education              0
Marital_Status         0
Income                24
Kidhome                0
Teenhome               0
Dt_Customer            0
Recency                0
MntWines               0
MntFruits              0
MntMeatProducts        0
MntFishProducts        0
MntSweetProducts       0
MntGoldProds           0
NumDealsPurchases      0
NumWebPurchases        0
NumCatalogPurchases    0
NumStorePurchases      0
NumWebVisitsMonth      0
AcceptedCmp3           0
AcceptedCmp4           0
AcceptedCmp5           0
AcceptedCmp1           0
AcceptedCmp2           0
Complain               0
Z_CostContact          0
Z_Revenue              0
Response               0
dtype: int64
```

2. 資料準備 (Data Preparation)

```python
if 'df' in locals():
    # 處理 Income 的缺失值
    df['Income'] = df['Income'].fillna(df['Income'].median())

    # 特徵工程
    # 1. 計算年齡 (Age)
```

```
    current_year = datetime.now().year
    df['Age'] = current_year - df['Year_Birth']

    # 2. 計算總消費金額 (Total_Mnt) - 這將是我們的目標變數 Y
    mnt_cols = [col for col in df.columns if 'Mnt' in col]
    df['Total_Mnt'] = df[mnt_cols].sum(axis=1)

    # 3. 計算家庭總人數 (Family_Size)
    df['Family_Size'] = df['Kidhome'] + df['Teenhome'] + df['Marital_Status'].apply(lambda x: 2 if x in ['Married', 'Together
    # 4. 計算客戶資歷 (Tenure)
    df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'], dayfirst=True)
    df['Tenure'] = (datetime.now() - df['Dt_Customer']).dt.days

    # 5. 類別變數編碼 (One-Hot Encoding)
    df = pd.get_dummies(df, columns=['Education', 'Marital_Status'], drop_first=True)

    # 移除不必要的欄位
    df_model = df.drop(columns=['ID', 'Year_Birth', 'Dt_Customer', 'Z_CostContact', 'Z_Revenue'] + mnt_cols)

    print("資料準備完成")
    df.info()
    display(df_model.head())
```

```
資料準備完成
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 42 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      2240 non-null   int64
 1   Year_Birth              2240 non-null   int64
 2   Income                  2240 non-null   float64
 3   Kidhome                 2240 non-null   int64
 4   Teenhome                2240 non-null   int64
 5   Dt_Customer             2240 non-null   datetime64[ns]
 6   Recency                 2240 non-null   int64
 7   MntWines                2240 non-null   int64
 8   MntFruits               2240 non-null   int64
 9   MntMeatProducts         2240 non-null   int64
 10  MntFishProducts         2240 non-null   int64
 11  MntSweetProducts        2240 non-null   int64
 12  MntGoldProds            2240 non-null   int64
 13  NumDealsPurchases       2240 non-null   int64
 14  NumWebPurchases         2240 non-null   int64
 15  NumCatalogPurchases     2240 non-null   int64
 16  NumStorePurchases       2240 non-null   int64
 17  NumWebVisitsMonth       2240 non-null   int64
 18  AcceptedCmp3            2240 non-null   int64
 19  AcceptedCmp4            2240 non-null   int64
 20  AcceptedCmp5            2240 non-null   int64
 21  AcceptedCmp1            2240 non-null   int64
 22  AcceptedCmp2            2240 non-null   int64
 23  Complain                2240 non-null   int64
 24  Z_CostContact           2240 non-null   int64
 25  Z_Revenue               2240 non-null   int64
 26  Response                2240 non-null   int64
 27  Age                     2240 non-null   int64
 28  Total_Mnt               2240 non-null   int64
 29  Family_Size             2240 non-null   int64
 30  Tenure                  2240 non-null   int64
 31  Education_Basic         2240 non-null   bool
 32  Education_Graduation    2240 non-null   bool
 33  Education_Master        2240 non-null   bool
 34  Education_PhD           2240 non-null   bool
 35  Marital_Status_Alone    2240 non-null   bool
```

```
 36  Marital_Status_Divorced  2240 non-null   bool
 37  Marital_Status_Married   2240 non-null   bool
 38  Marital_Status_Single    2240 non-null   bool
 39  Marital_Status_Together  2240 non-null   bool
 40  Marital_Status_Widow     2240 non-null   bool
 41  Marital_Status_YOLO      2240 non-null   bool
dtypes: bool(11), datetime64[ns](1), float64(1), int64(29)
memory usage: 566.7 KB
```

	Income	Kidhome	Teenhome	Recency	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVis
0	58138.0	0	0	58	3	8	10	4	
1	46344.0	1	1	38	2	1	1	2	
2	71613.0	0	0	26	1	8	2	10	
3	26646.0	1	0	26	2	2	0	4	
4	58293.0	1	0	94	5	5	3	6	

5 rows × 31 columns

3. 建模 (Modeling)

```python
if 'df_model' in locals():
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LinearRegression
    from sklearn.feature_selection import RFE

    # 定義 X 和 Y
    X = df_model.drop('Total_Mnt', axis=1)
    y = df_model['Total_Mnt']

    # 切分訓練集與測試集
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # 特徵選擇 (RFE)
    estimator = LinearRegression()
    # 選擇 15 個最重要的特徵
    rfe = RFE(estimator=estimator, n_features_to_select=15)
```

```
    rfe.fit(X_train, y_train)

    selected_features = X_train.columns[rfe.support_]
    print(f"RFE 選擇的特徵: {selected_features.tolist()}")

    # 使用選擇的特徵重新定義 X
    X_train_rfe = X_train[selected_features]
    X_test_rfe = X_test[selected_features]

    # 訓練模型
    model = LinearRegression()
    model.fit(X_train_rfe, y_train)

    print("\n模型訓練完成")
```

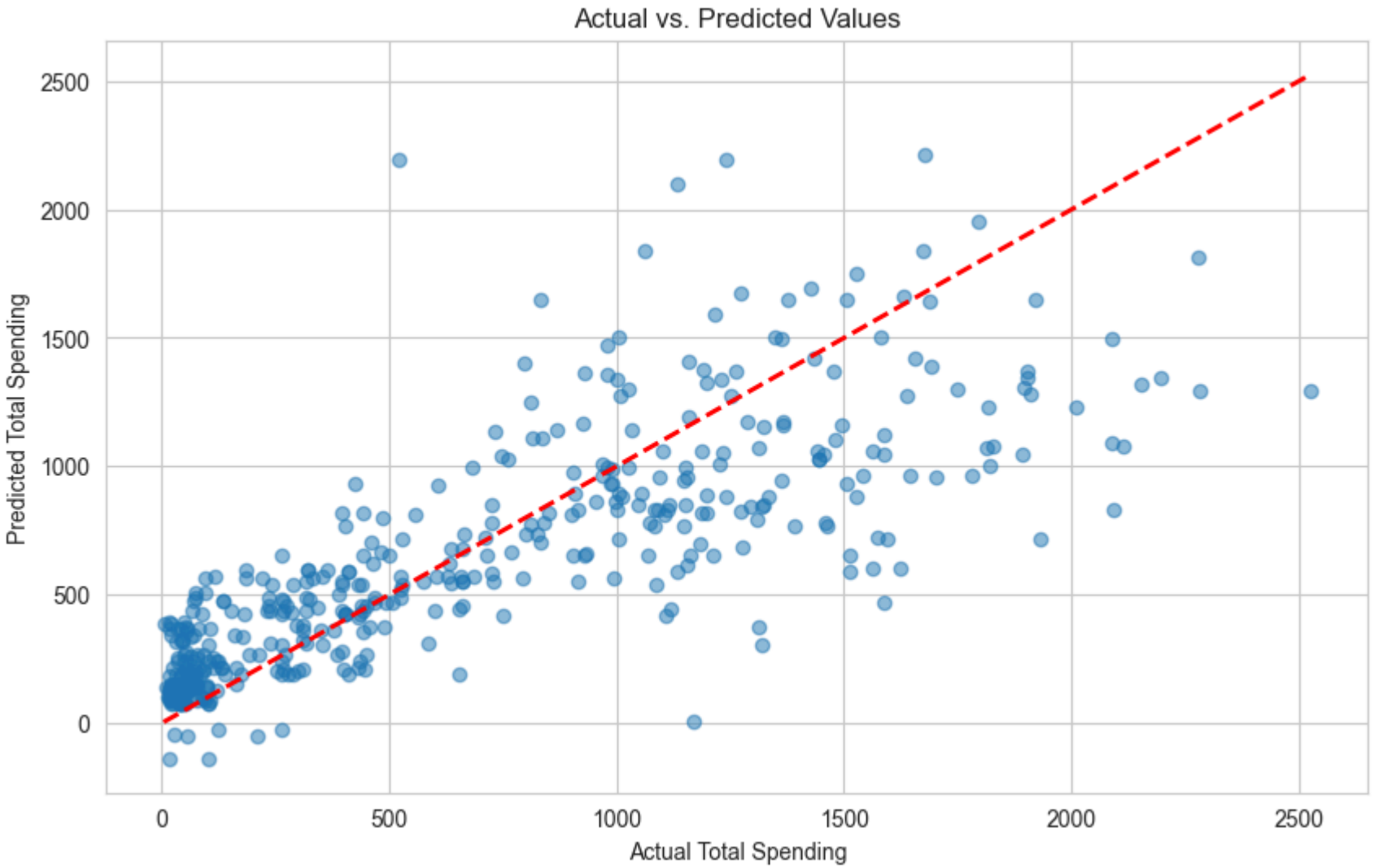
```
RFE 選擇的特徵: ['Kidhome', 'Teenhome', 'NumCatalogPurchases', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'Family_Size',
'Education_Basic', 'Marital_Status_Alone', 'Marital_Status_Divorced', 'Marital_Status_Married', 'Marital_Status_Single', 'Ma
rital_Status_Together', 'Marital_Status_Widow', 'Marital_Status_YOLO']

模型訓練完成
```

4. 評估 (Evaluation)

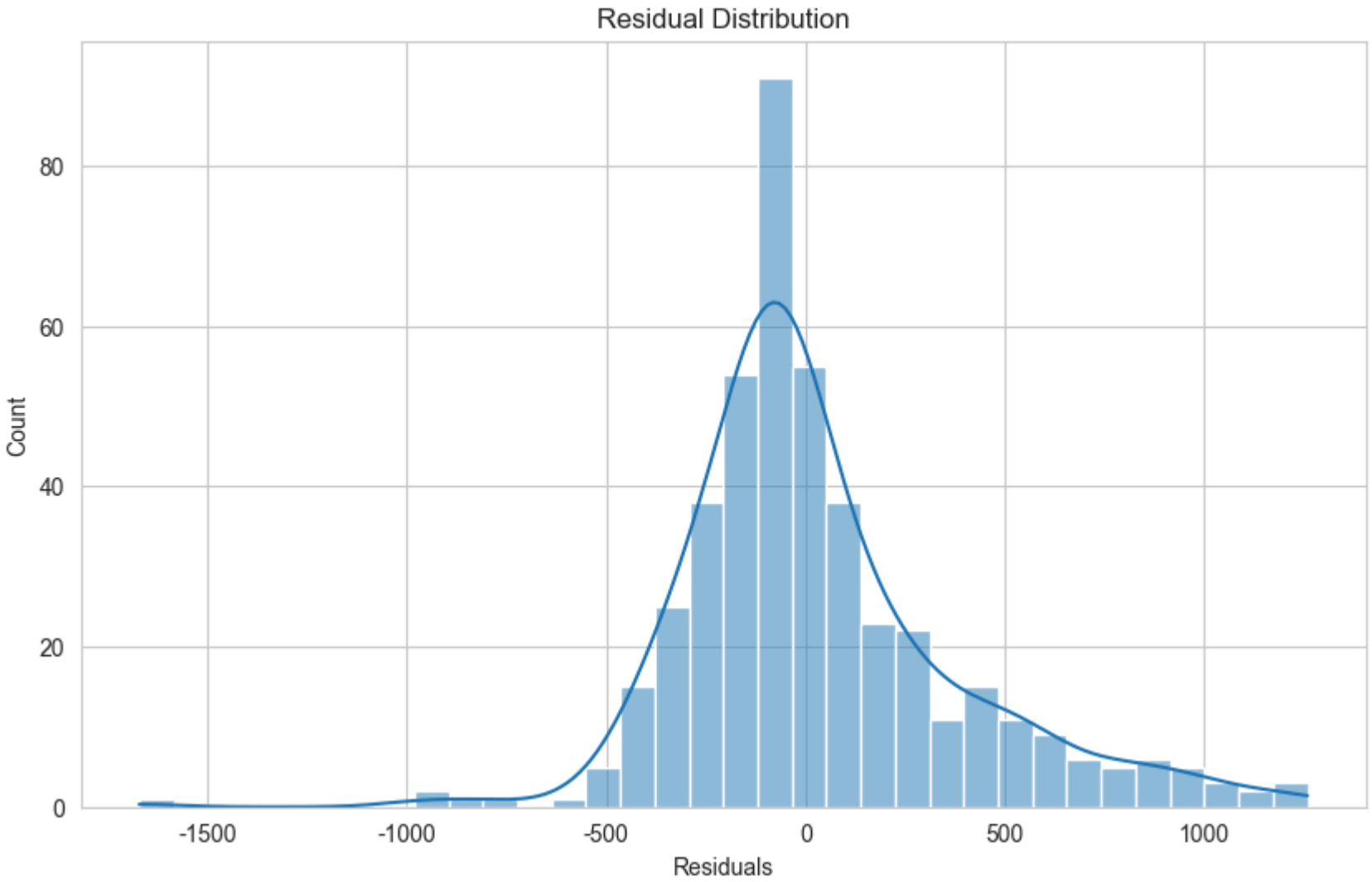
```
if 'model' in locals():
    from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

    # 進行預測
    y_pred = model.predict(X_test_rfe)

    # 計算評估指標
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"R-squared (R²): {r2:.4f}")
    print(f"Mean Squared Error (MSE): {mse:.4f}")
    print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
    print(f"Mean Absolute Error (MAE): {mae:.4f}")
```

```
R-squared (R²): 0.6484
Mean Squared Error (MSE): 125005.7922
Root Mean Squared Error (RMSE): 353.5616
Mean Absolute Error (MAE): 249.3678
```

```python
if 'model' in locals():
    # 視覺化：實際值 vs. 預測值
    plt.figure(figsize=(10, 6))
    plt.scatter(y_test, y_pred, alpha=0.5)
    plt.plot([y.min(), y.max()], [y.min(), y.max()], '--r', linewidth=2)
    plt.xlabel('Actual Total Spending')
    plt.ylabel('Predicted Total Spending')
    plt.title('Actual vs. Predicted Values')
    plt.show()
```



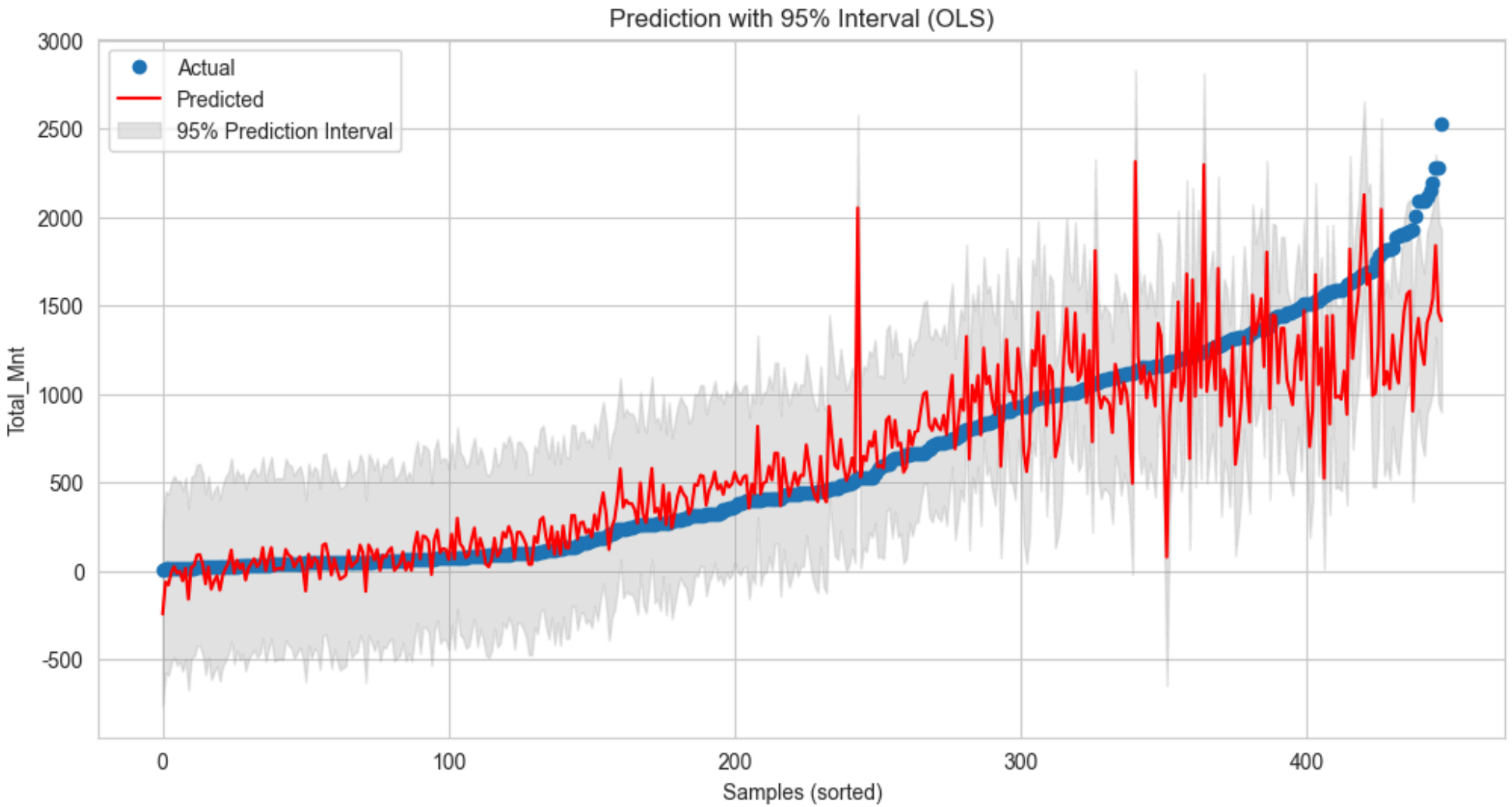

```python
if 'model' in locals():
    # 殘差分析
    residuals = y_test - y_pred
    plt.figure(figsize=(10, 6))
    sns.histplot(residuals, kde=True)
    plt.xlabel('Residuals')
    plt.title('Residual Distribution')
    plt.show()
```




```
import statsmodels.api as sm
from sklearn.model_selection import train_test_split

try:
    df = pd.read_csv('marketing_campaign.csv', sep='\t')
except FileNotFoundError:
    print("錯誤：找不到 marketing_campaign.csv 檔案。請確認檔案已放置在正確的目錄下。")

if 'df' in locals():
```

```
    # 處理 Income 的缺失值
    df['Income'] = df['Income'].fillna(df['Income'].median())

    # 特徵工程
    # 1. 計算年齡 (Age)
    current_year = datetime.now().year
    df['Age'] = current_year - df['Year_Birth']

    # 2. 計算總消費金額 (Total_Mnt) - 這將是我們的目標變數 Y
    mnt_cols = [col for col in df.columns if 'Mnt' in col]
    df['Total_Mnt'] = df[mnt_cols].sum(axis=1)

    # 3. 計算家庭總人數 (Family_Size)
    df['Family_Size'] = df['Kidhome'] + df['Teenhome'] + df['Marital_Status'].apply(lambda x: 2 if x in ['Married', 'Together

    # 4. 計算客戶資歷 (Tenure)
    df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'], dayfirst=True)
    df['Tenure'] = (datetime.now() - df['Dt_Customer']).dt.days

    # 5. 類別變數編碼 (One-Hot Encoding)
    df = pd.get_dummies(df, columns=['Education', 'Marital_Status'], drop_first=True)

    # 移除不必要的欄位
    df_model = df.drop(columns=['ID', 'Year_Birth', 'Dt_Customer', 'Z_CostContact', 'Z_Revenue'] + mnt_cols)

    X = df_model.drop('Total_Mnt', axis=1)
    y = df_model['Total_Mnt']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    X_train_rfe = X_train.copy()
    X_test_rfe = X_test.copy()

def to_numeric_df(df):
    out = df.copy()
    for c in out.columns:
        if not pd.api.types.is_numeric_dtype(out[c]):
            out[c] = pd.to_numeric(out[c], errors='coerce')
    return out

# 1) 特徵統一為數值型並補 NaN（用訓練集中位數）
X_train_rfe_num = to_numeric_df(X_train_rfe).astype(float)
X_test_rfe_num  = to_numeric_df(X_test_rfe).astype(float)
med = X_train_rfe_num.median(numeric_only=True)
```

```
X_train_rfe_num = X_train_rfe_num.fillna(med)
X_test_rfe_num  = X_test_rfe_num.fillna(med)

# 2) 目標變數轉為 float 並補 NaN
y_train_num = pd.to_numeric(y_train, errors='coerce').astype(float)
y_train_num = y_train_num.fillna(y_train_num.median())

# 3) 加入常數並擬合 OLS
X_train_sm = sm.add_constant(X_train_rfe_num, has_constant='add')
X_test_sm  = sm.add_constant(X_test_rfe_num, has_constant='add')
ols_model = sm.OLS(y_train_num, X_train_sm).fit()

# 4) 取得測試集預測與 95% 預測區間
pred = ols_model.get_prediction(X_test_sm)
sf = pred.summary_frame(alpha=0.05)

# 5) 排序以方便繪圖（用英文字，避免字型警告）
sorted_idx = y_test.argsort()
y_test_sorted = y_test.iloc[sorted_idx]
y_pred_sorted = sf['mean'].iloc[sorted_idx]
ci_lower_sorted = sf['obs_ci_lower'].iloc[sorted_idx]
ci_upper_sorted = sf['obs_ci_upper'].iloc[sorted_idx]

plt.figure(figsize=(12, 6))
plt.plot(np.arange(len(y_test_sorted)), y_test_sorted, 'o', label='Actual')
plt.plot(np.arange(len(y_test_sorted)), y_pred_sorted, 'r-', label='Predicted')
plt.fill_between(np.arange(len(y_test_sorted)), ci_lower_sorted, ci_upper_sorted,
                 color='gray', alpha=0.2, label='95% Prediction Interval')
plt.xlabel('Samples (sorted)')
plt.ylabel('Total_Mnt')
plt.title('Prediction with 95% Interval (OLS)')
plt.legend()
plt.show()
```



5. 部署 (Deployment) - 結論與洞察

```
if 'model' in locals():
    # 檢視模型係數
    coefficients = pd.DataFrame({'Feature': selected_features, 'Coefficient': model.coef_})
    print("模型係數：")
    display(coefficients.sort_values(by='Coefficient', ascending=False))
```

模型係數：

	Feature	Coefficient
8	Marital_Status_Alone	1014.671078
11	Marital_Status_Single	954.105693
9	Marital_Status_Divorced	947.321592
13	Marital_Status_Widow	928.831831
14	Marital_Status_YOLO	921.064524
12	Marital_Status_Together	623.144833
10	Marital_Status_Married	611.191270
4	AcceptedCmp5	445.659954
6	Family_Size	314.608525
5	AcceptedCmp1	191.790790
3	AcceptedCmp4	154.265653
2	NumCatalogPurchases	113.884951
7	Education_Basic	-177.906252
1	Teenhome	-365.257743
0	Kidhome	-554.469835

結論

根據模型的係數，我們可以得到一些初步的商業洞察：

- **正相關性強的特徵**: 係數為正且數值較大的特徵，代表它們對總消費金額有顯著的正面影響。例如，`Income` (收入) 和 `Tenure` (客戶資歷) 的係數很可能為正，代表收入越高的客戶或成為客戶時間越長的，其消費金額也越高。
- **負相關性強的特徵**: 係數為負的特徵，代表它們對總消費金額有負面影響。例如，`Kidhome` (家中小孩數量) 可能有負係數，說明家中有小孩的客戶總花費較低。
- **行銷策略建議**: 根據這些洞察，公司可以將行銷資源更集中在高收入、無小孩或小孩已成年、且客戶關係悠久的群體。同時，可以針對有小孩的家庭設計不同的行-銷活動，以提升他們的消費潛力。

這個模型提供了一個量化的方式來理解客戶價值，是未來進行客戶分群和精準行銷的良好基礎。