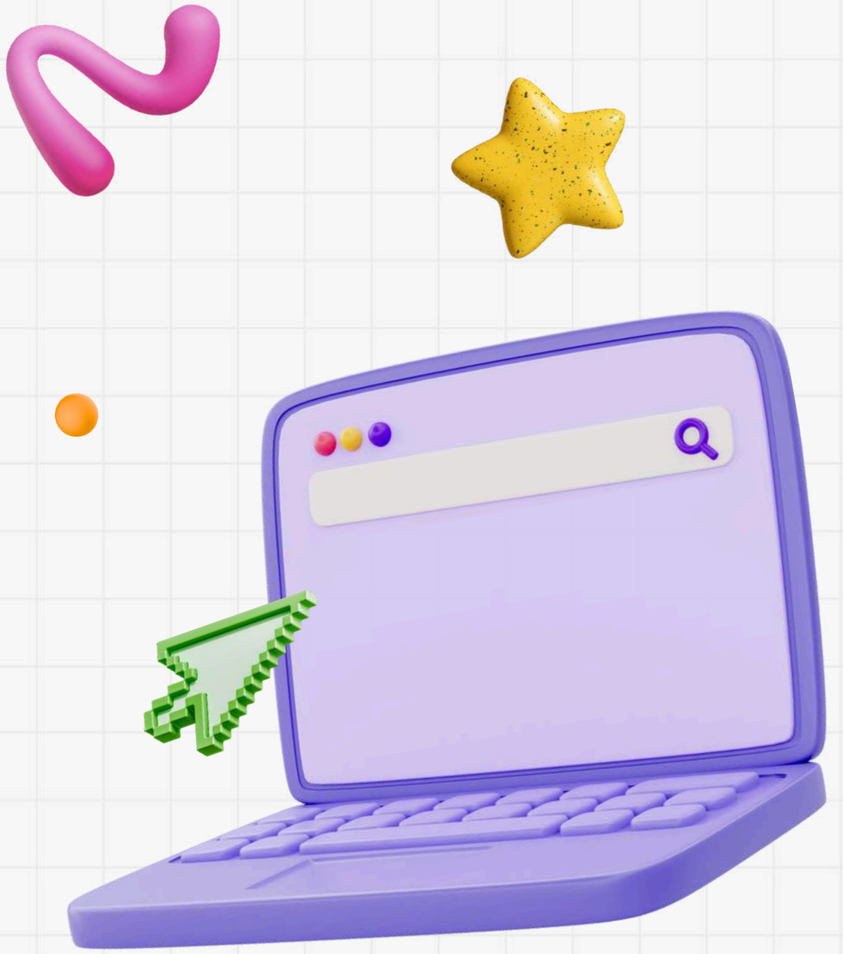
Marketing Analytics

Customer Segmentation
(To Create Targeted Advertising Campaigns)

Ana Farida

Agenda

1. Background
2. Goals and Objectives
3. Data Insights
4. Modelling



Background

- Sebuah perusahaan retail menjual produk regular dan gold berupa *wines*, *fruits*, *meat*, *fish* dan *sweet products*.
- Perusahaan ini memiliki 3 *sales channels* yaitu *catalogs*, *physical stores* dan *company website*.
- *Perusahaan ini memiliki ratusan ribu registered customer dan melayani hampir 1 juta pelanggan per tahun.*
- *Dataset yang digunakan berisi data 2.240 customer dengan fitur socio-demographic firmographicnya.*
- Perusahaan ini akan melakukan *direct marketing campaign* kepada registered customer untuk produk baru yang akan dirilis bulan depan.

Goals & Objectives

- Meningkatkan pendapatan (*revenue*) dan keuntungan (*profit*) dari penjualan (*sales*) produk serta mengurangi biaya (*cost*) marketing.

- Optimalisasi *direct marketing campaign* untuk penjualan produk baru yang akan dirilis bulan depan.
- Membuat *predictive model* dengan cara memahami karakteristik customer yang berpotensi untuk membeli produk baru tersebut. (*customer segmentation to create targeted advertising campaigns*)
- Predictive model dapat diaplikasikan ke dalam data customer lain. (di luar splitting dataset yang digunakan (train, validation, test)).

Exploratory Data Analysis (EDA)

Feature	Description
AcceptedCmp1	1 if costumer accepted the offer in the 1st campaign, 0 otherwise
AcceptedCmp2	1 if costumer accepted the offer in the 2nd campaign, 0 otherwise
AcceptedCmp3	1 if costumer accepted the offer in the 3rd campaign, 0 otherwise
AcceptedCmp4	1 if costumer accepted the offer in the 4th campaign, 0 otherwise
AcceptedCmp5	1 if costumer accepted the offer in the 5th campaign, 0 otherwise
Response (target)	1 if costumer accepted the offer in the last campaign, 0 otherwise
Complain	1 if costumer complained in the last 2 years
DtCustomer	date of customer's enrollment with the company
Education	customer's level of education
Marital	customer's marital status
Kidhome	number of small children in customer's household
Teenhome	number of teenagers in customer's household
Income	customer's yearly household income
MntFishProducts	amount spent on fish products in the last 2 years
MntMeatProducts	amount spent on meat products in the last 2 years
MntFruits	amount spent on fruits in the last 2 years
MntSweetProducts	amount spent on sweet products in the last 2 years
MntWines	amount spent on wines in the last 2 years
MntGoldProds	amount spent on *gold* products in the last 2 years
NumDealsPurchases	number of purchases made with discount
NumCatalogPurchases	number of purchases made using catalogue
NumStorePurchases	number of purchases made directly in stores
NumWebPurchases	number of purchases made through company's web site
NumWebVisitsMonth	number of visits to company's web site in the last month
Recency	number of days since the last purchase

RangeIndex: 2240 entries, 0 to 2239
Data columns (total 28 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	object
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	Response	2240 non-null	int64
26	Complain	2240 non-null	int64
27	Country	2240 non-null	object

Peprocessing Data (Cleaning, Integration, Transformation, Reduction)

Data terdiri dari 2240 rows dan 28 columns.

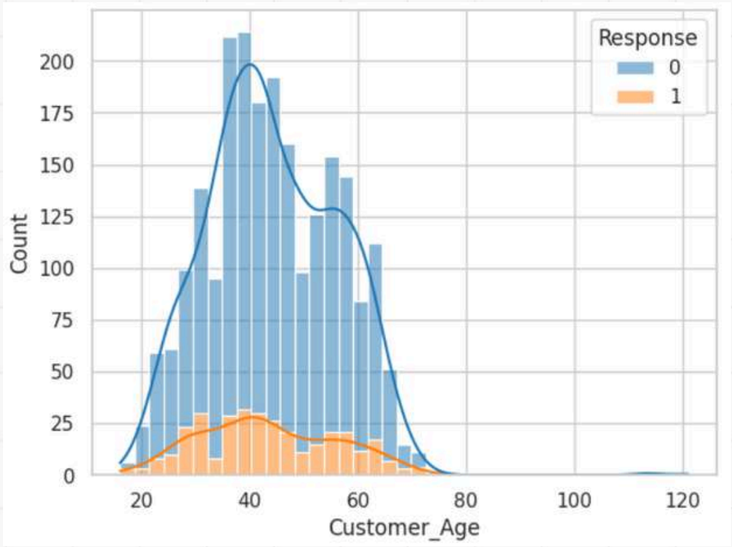
a. Apakah data type tidak sesuai?
b. Apakah ada duplicates, missing values, outliers?
c. Apakah distribusi tidak sesuai?

a. Semua data type sesuai, kecuali:
1. Income = data type berupa object harus diubah menjadi float (terdapat whitespace pada nama kolom).
2. Dt_Customer = data type berupa object harus diubah menjadi date.

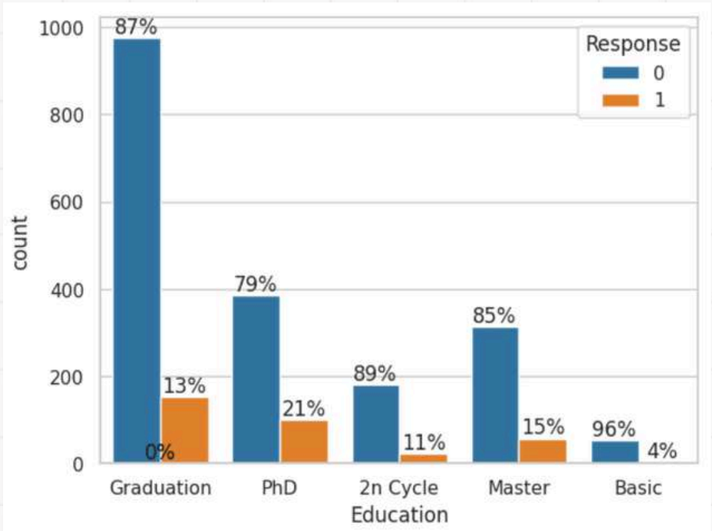
b.
1. Tidak ada duplicate data.
2. Missing values pada column Income sebanyak 24. Adanya outliers menyebabkan distribusinya tidak normal, sehingga missing values diisi dengan nilai mediannya bukan meannya.
3. Mencari kecenderungan distribusi data dan menemukan outliers pada data numerik dengan cara extract, transform serta visualisasi data

Data Insights (1)

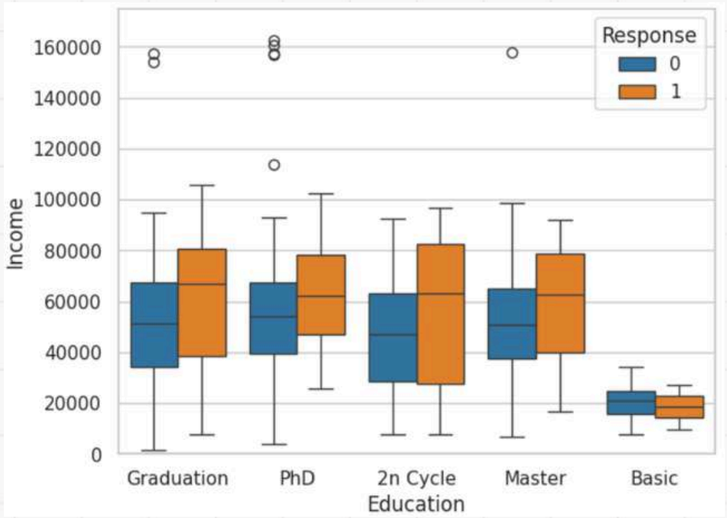
Descriptive analysis dengan menggunakan data historis customer (masa lampau) untuk mengidentifikasi karakteristik customer yang cenderung menerima / merespon positif campaign.



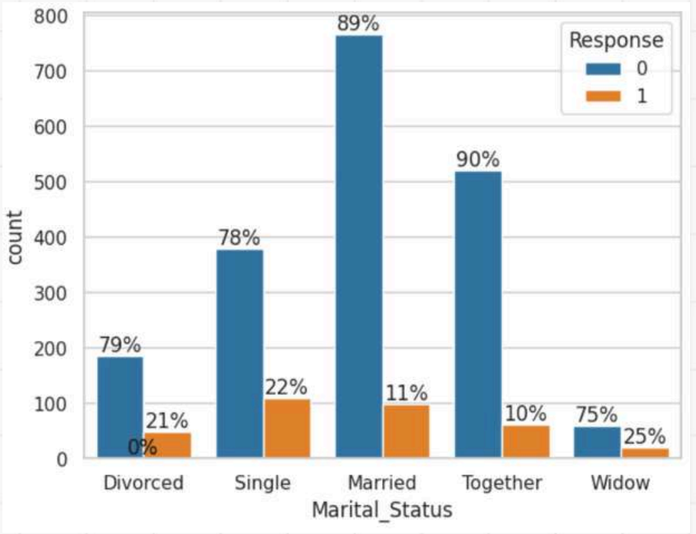
Berusia 30 hingga 45



Berpendidikan Graduation (S1) dan PhD (S3)



Berpenghasilan $40,000 - $80,000



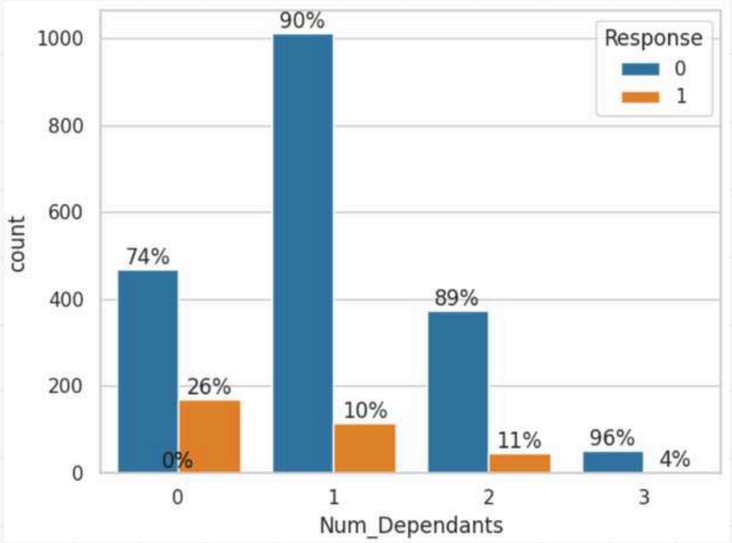
Berstatus Single dan Married

Response

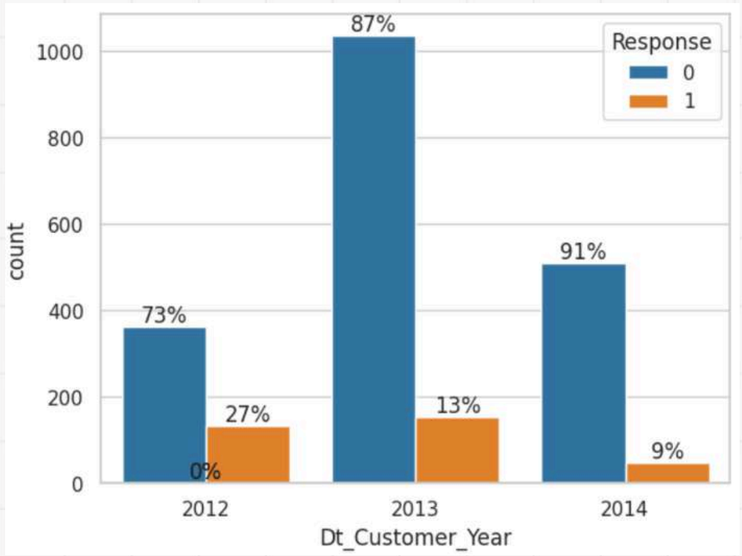
0 : Negative (Tidak Tertarik)

1 : Positive (Tertarik)

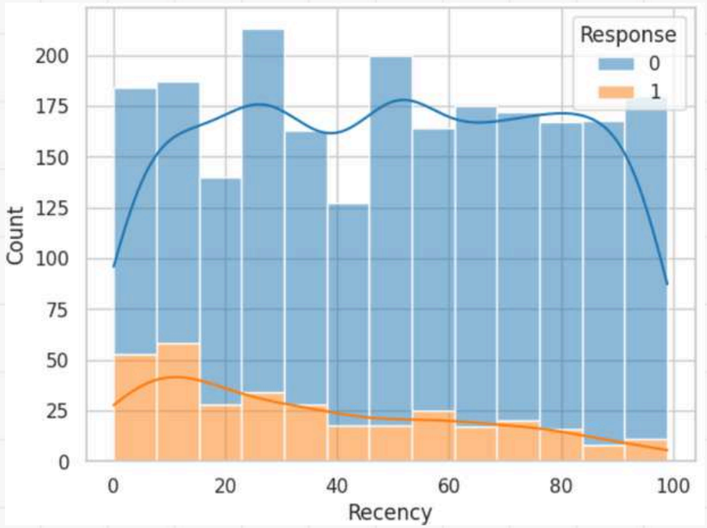
Data Insights (2)



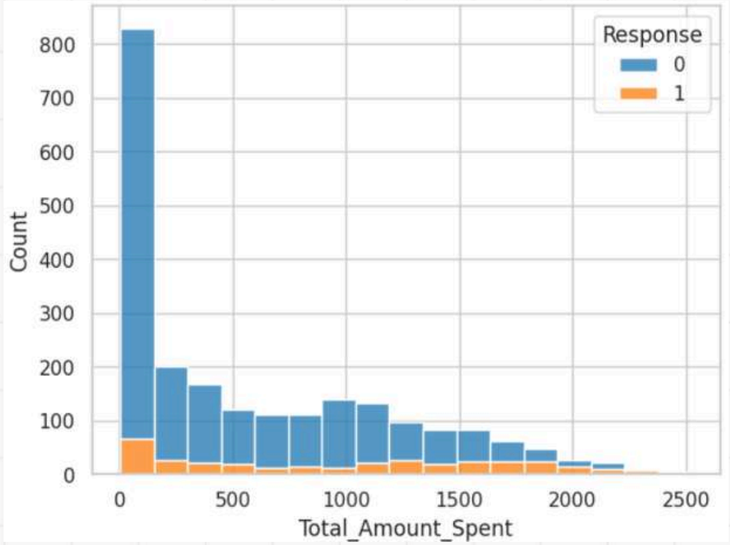
Tidak memiliki anak (0) / sedikit anak (1 anak)



Tahun Membership (2012)



Last transaction / tanggal pembelian terakhir (kurang dari 30 hari)



Transaksi Pembelian $1,000-$2,000

Response

0 : Negative (Tidak Tertarik)

1 : Positive (Tertarik)

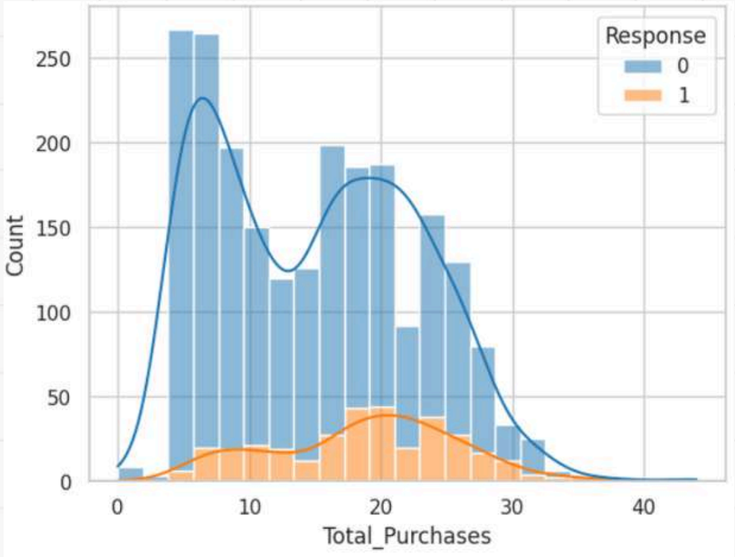
Data Insights (3)

	Response
Response	1.000000
MntWines	0.247254
MntMeatProducts	0.236335
MntGoldProds	0.139850
MntFruits	0.125289
MntSweetProducts	0.117372
MntFishProducts	0.111331

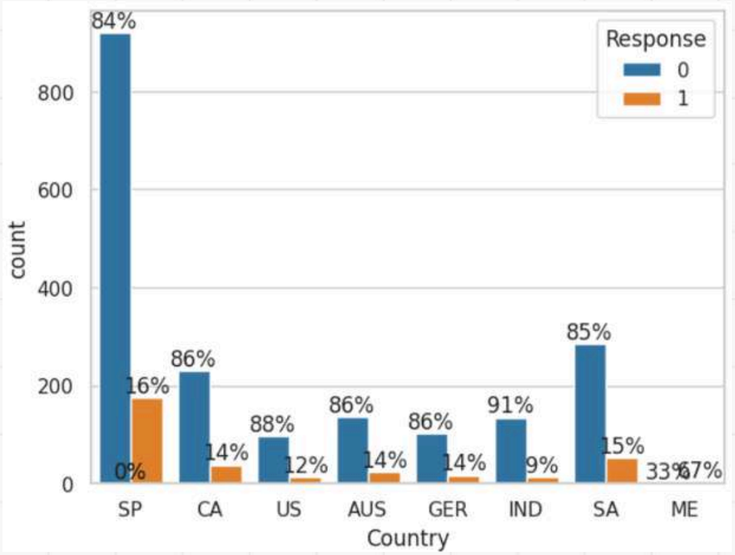
Membeli Wines dan MeatProducts

	Response
Response	1.000000
NumCatalogPurchases	0.220810
NumWebPurchases	0.148730
NumStorePurchases	0.039363
NumDealsPurchases	0.002238

Membeli lewat Catalog dan Web



Jumlah transaksi pembelian 10-25 kali



Negara asal SP (Spain) dan ME(Mexico)

	Response
Response	1.000000
AcceptedCmp5	0.326634
AcceptedCmp1	0.293982
AcceptedCmp3	0.254258
AcceptedCmp4	0.177019
AcceptedCmp2	0.169293

Menerima Campaign 5, 1, 3

Response
0 : Negative (Tidak Tertarik)
1 : Positive (Tertarik)

Modelling

-Predictive analysis dengan menggunakan model dari data historis customer untuk memprediksi performance model.

-Modelling terdiri dari
x = independent variable (karakteristik customer) terhadap
y = dependent variable / target (customer response, kemungkinan campaign direspon positif atau tidak).

-Modelling menggunakan Logistic Regression (probabilitas akurasi 0.746 atau 74.6%) dan Random Forest (probabilitas akurasi 0.882 atau 88.2%).

-Performance Stability Check dengan menggunakan Model Random Forest karena probabilitas akurasi lebih tinggi dibandingkan Logistic Regression.

-Model Random Forest diterapkan pada data lain, dihasilkan probabilitas akurasi 0.878 atau 87.8%. Perbedaan probabilitas akurasi yang tidak jauh mengindikasikan bahwa performance model cukup stabil untuk dapat diaplikasikan untuk memprediksi probabilitas respon ratusan ribu registered customer lainnya.

d. Performance Stability Check

```
x_full_train_final=pd.concat([x_train_final, x_valid_final])
y_full_train_final=pd.concat([y_train_final, y_valid_final])
```

```
model=RandomForestClassifier(random_state=42)
model.fit(x_full_train_final, y_full_train_final)
```

```
RandomForestClassifier(random_state=42)
```

```
y_test_pred=model.predict_proba(x_test_final)[:, 1]
print('RandomForest ROCAUC Result: ', roc_auc_score(y_test_final, y_test_pred).round(3))
```

```
RandomForest ROCAUC Result:  0.878
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.metrics import roc_auc_score
```

```
model=LogisticRegression(random_state=42)
model.fit(x_train_final, y_train_final)
model.predict_proba(x_valid_final)[:, 1]
y_valid_pred=model.predict_proba(x_valid_final)[:, 1]
```

```
print('Logistic Regression ROCAUC Result: ', roc_auc_score(y_valid_final, y_valid_pred).round(3))
```

```
Logistic Regression ROCAUC Result:  0.746
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
```

```
model=RandomForestClassifier(random_state=42)
model.fit(x_train_final, y_train_final)
```

```
RandomForestClassifier(random_state=42)
```

```
y_valid_pred=model.predict_proba(x_valid_final)[:, 1]
print('Random Forest ROCAUC Result: ', roc_auc_score(y_valid_final, y_valid_pred).round(3))
```

```
Random Forest ROCAUC Result:  0.882
```