Marketing Analytics

Customer Segmentation
(To Create Targeted Advertising Campaigns)

Ana Farida

Agenda

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

Background

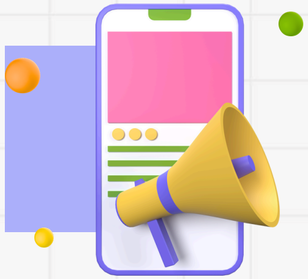
Sebuah perusahaan retail menjual produk regular dan gold berupa *wines*, *fruits*, *meat*, *fish* dan *sweet products*.

Perusahaan ini memiliki 3 *sales channels* yaitu *catalogs*, *physical stores* dan *company website*.

Perusahaan ini akan melakukan *marketing campaign* untuk produk baru yang akan dirilis bulan depan.

Goals & Objectives

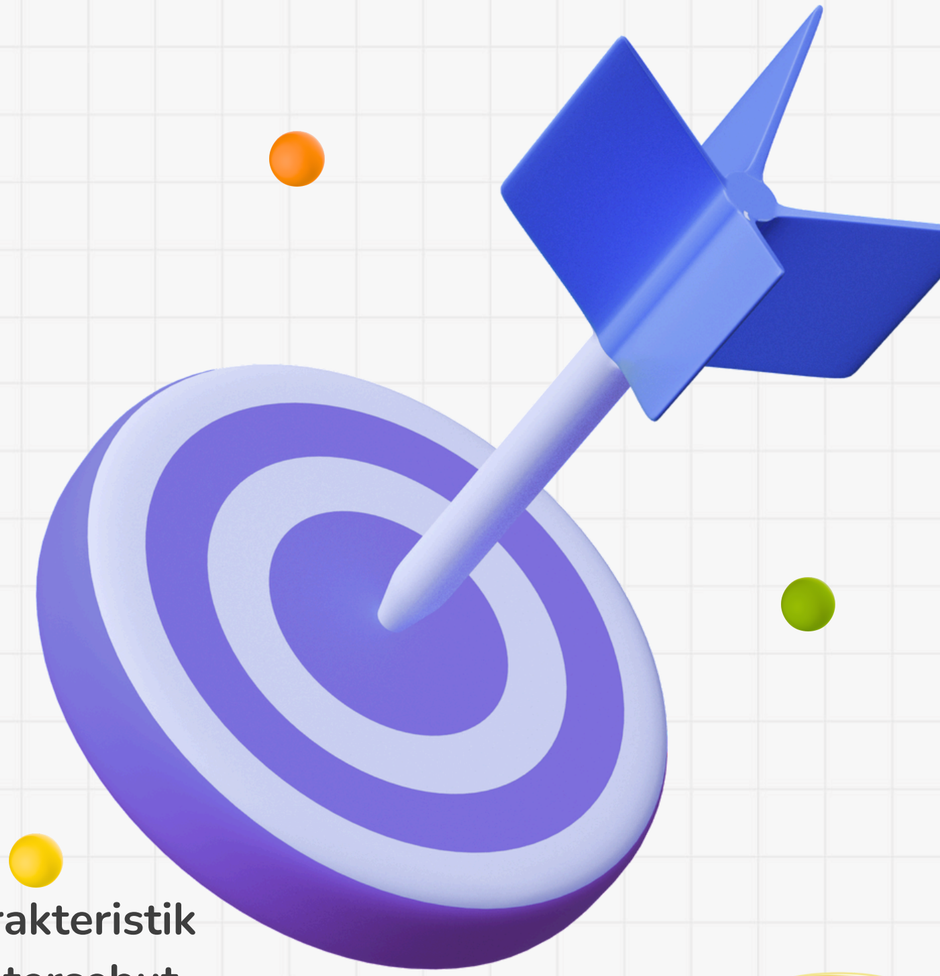
Meningkatkan pendapatan (*revenue*) dan keuntungan (*profit*) dari penjualan (*sales*) produk serta mengurangi biaya (*cost*) marketing.



Optimalisasi *marketing campaign* untuk penjualan produk baru yang akan dirilis bulan depan.

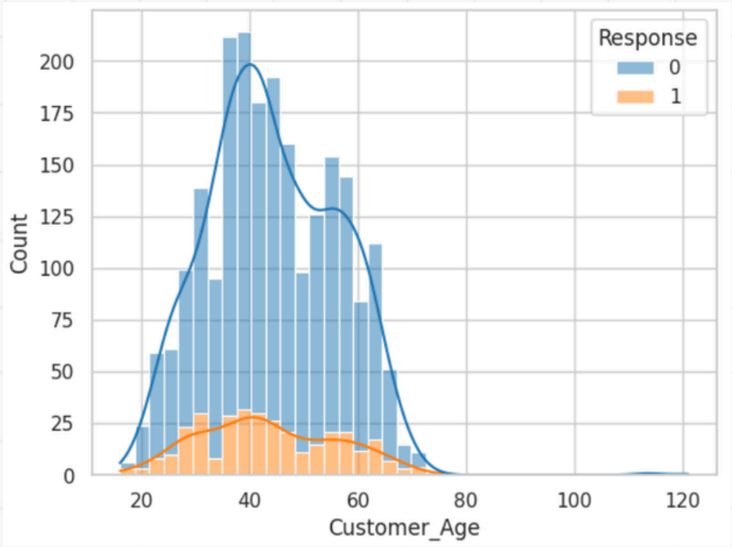
Membuat *predictive model* dengan cara memahami karakteristik customer yang berpotensi untuk membeli produk baru tersebut. (*customer segmentation to create targeted advertising campaigns*)

Predictive model dapat diaplikasikan ke dalam data customer lain. (di luar splitting dataset yang digunakan (train, validation, test)).

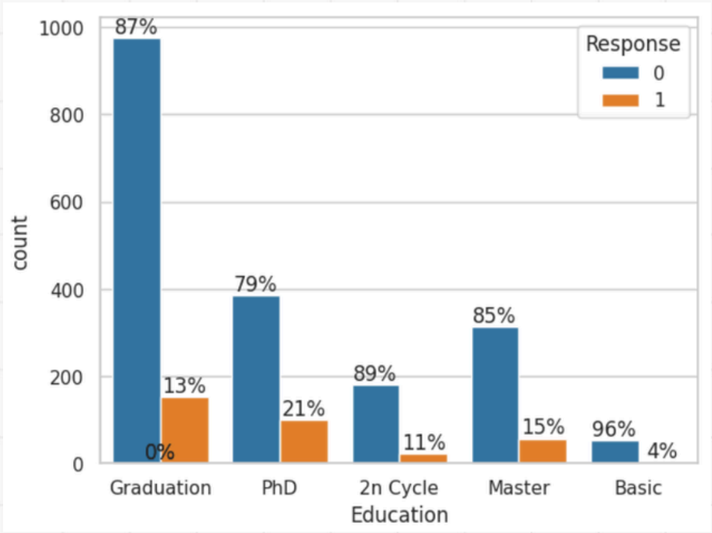


Data Insights (1)

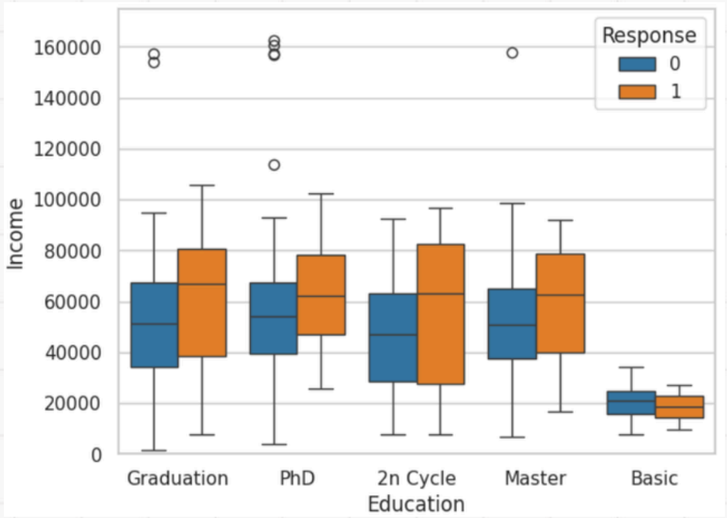
Descriptive analysis dengan menggunakan data historis customer (masa lampau) untuk mengidentifikasi karakteristik customer yang cenderung menerima / merespon positif campaign.



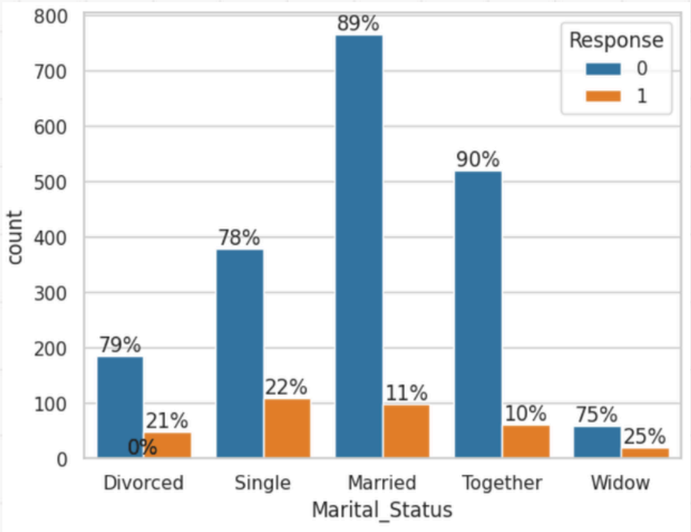
Berusia 30 hingga 45



Berpendidikan Graduation (S1) dan PhD (S3)



Berpenghasilan $40,000 - $80,000



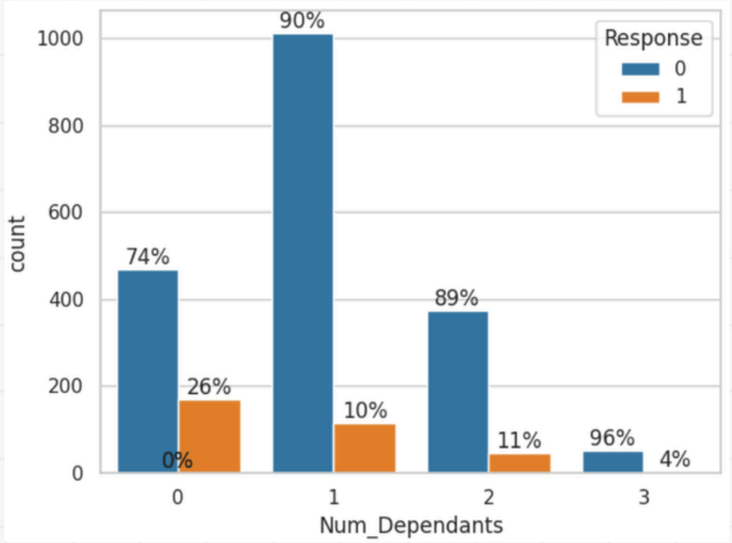
Berstatus Single dan Married

Response

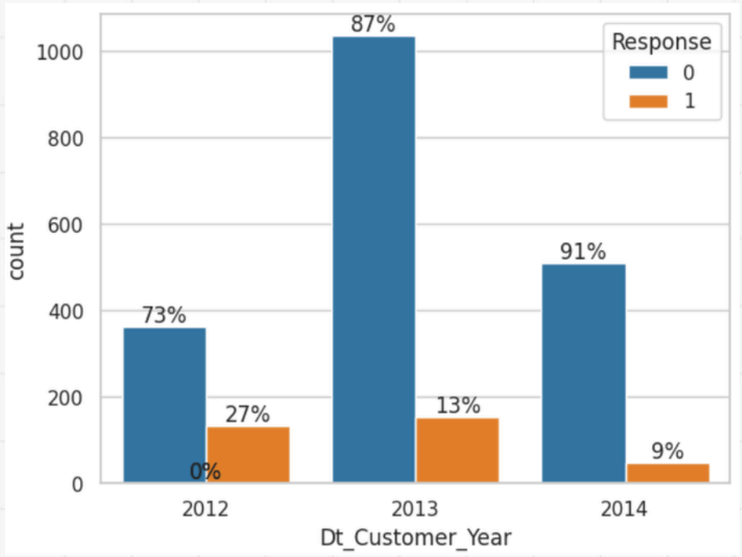
0 : Negative (Tidak Tertarik)

1 : Positive (Tertarik)

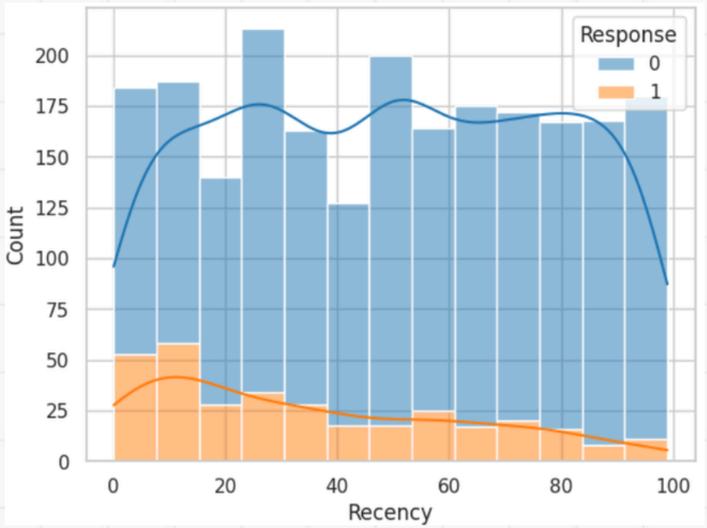
Data Insights (2)



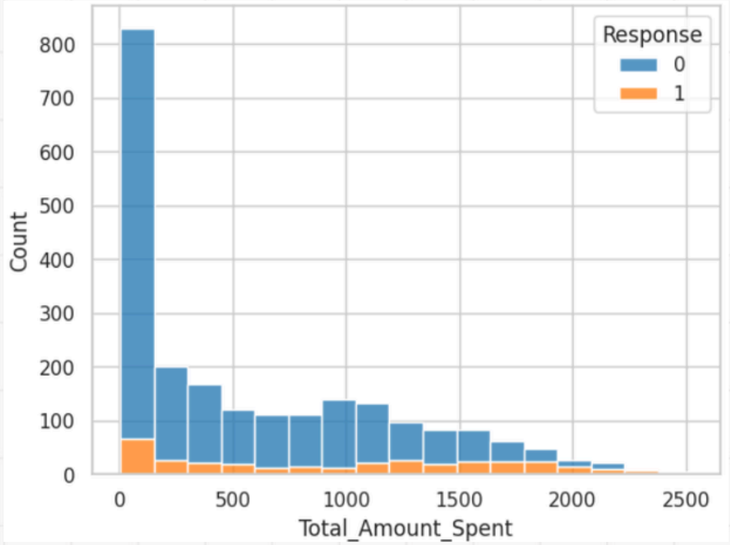
Tidak memiliki anak (0) / sedikit anak (1 anak)



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)

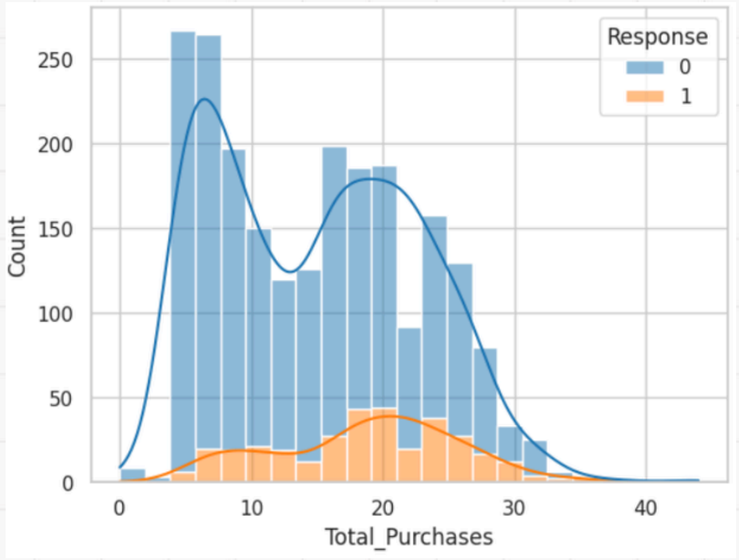
Data Insights (3)

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

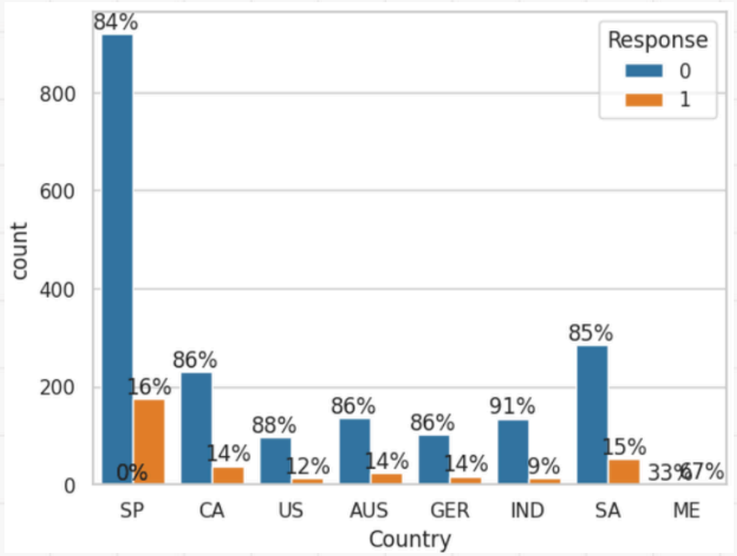
Membeli Wines dan MeatProducts

	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 diterima 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.

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
```