

Datalicious

Dokumen
Laporan Final
Project

Marketing Campaign



Datalicious - Kelompok 1

Members of Team :

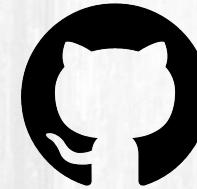
1. Nur Imam Masri
2. Astuti Rahmawati
3. Prasidya Bagaskara
4. Moh. Harwin Prayoga
5. Riskiyatul Hasanah
6. M Rayhan Azzindani
7. Siti Hajjah Mardiah
8. Christine
9. M. Ifzal Asril



Git & Github



[Link GDrive - Datalicious](#)



[Link Github - Datalicious](#)

Stage 0

Preparation



Latar Belakang Masalah

Problem Statement	Roles
<p>Lotto Mart adalah sebuah supermarket yang bergerak dibidang retail, menjual berbagai jenis produk seperti <i>Fish, Meat, Fruits, Sweet Products, Wines</i>, dan <i>Gold Products</i>. Selama 6 bulan terakhir, Marketing Team melakukan campaign berupa pemberian <i>discount vouchers</i> kepada semua customer melalui <i>Broadcast Message</i>. Namun, setelah campaign dilakukan, justru Lotto Mart menghadapi beberapa permasalahan sebagai berikut :</p> <ul style="list-style-type: none"> • Response rate dari marketing campaign yang dilakukan rendah yaitu sekitar 14.91% • Inefficient Cost dalam melakukan marketing campaign • Profit tidak sebanding dengan cost yang dikeluarkan 	<p>Berdasarkan hal tersebut, Marketing Team meminta tim data untuk menganalisis permasalahan yang terjadi. Selanjutnya perusahaan ingin membuat marketing campaign yang tepat sasaran sesuai dengan karakteristik customer. Strategi ini diharapkan mampu meningkatkan response rate, meminimalisasi cost, dan kemudian meningkatkan profit.</p> <p>Sebagai tim data di Supermarket Lotto Mart, untuk menganalisis keberhasilan marketing campaign selanjutnya. Berikut adalah beberapa roles beserta PIC yang berkontribusi dalam menyelesaikan permasalahan di Lotto Mart:</p> <ul style="list-style-type: none"> • Lead Data Science Sebagai koordinator project PIC: Nur Imam Masri • Machine Learning Engineer Membuat model dan evaluasi Machine Learning PIC: Prasidya Bagaskara dan Moh. Harwin Prayoga • Data Engineer Melakukan Data Preparation, Cleaning, dan Exploratory Data Analysis (EDA) PIC: M Rayhan Azzindani dan M. Ifzal Asril • Business Analyst Membuat insight business PIC: Riskiyatul Hasanah dan Christine • Data Analyst Membuat dashboard PIC: Siti Hajjah Mardiah dan Astuti Rahmawati

Latar Belakang Masalah

Goals	Objectives	Business Metrics
<p>Perusahaan ingin meningkatkan response rate dan meminimalisasi marketing campaign cost sehingga dapat memaksimalkan profit.</p>	<p>Membuat classification model untuk memprediksi kelompok customer yang akan merespon campaign agar dapat meminimalisasi biaya pemasaran dan memaksimalkan keuntungan pada campaign marketing berikutnya.</p>	<ul style="list-style-type: none"> Response Rate/RR Persentase total customer response terhadap total delivered campaign . <i>Indicator RR</i> : 30% is good (Efti 2018). Net Profit Margin /NPM Mengukur net profit dibanding penjualannya. Semakin besar NPM, maka kinerja marketing campaign semakin efektif dan efisien (Handayani, Winarningsih 2020). <i>Indicator NPM</i> : 5% is low, 10-19% is average, 20% is good (Jayathilaka 2020). Return of Investment /ROI CLV (customer lifetime value)) dibagi CAC((customer acquisition cost). Sebagai indikator kinerja perusahaan dan pembuatan keputusan para investor. <i>Indicator ROI</i> : 3:1 to 5:1 is good (Manzer 2017).

References

- Efti S. 2018. *Sales Benchmarks: The 30/50 rule for cold emailing & cold calling.* diakses 20 Mei 2023. <https://blog.close.com/sales-benchmarks/>
- Handayani N, Winarningsih S. 2020. The Effect of Net Profit Margin and Return on Equity Toward Profit Growth. *Moneter-Jurnal Akuntansi Dan Keuangan* 7(2): 198-204. <https://doi.org/10.31294/moneter.v7i2.8701>.
- Jayathilaka AK. 2020. Operating Profit and Net Profit: Measurements of Profitability. *Open Access Library Journal* 7(12): 1-11. <https://doi.org/10.4236/oalib.1107011>.
- Manzer D. 2017. *Five insights into measuring marketing ROI.* diakses 20 Mei 2023. <https://growswyft.com/five-insights-to-measure-your-roi/>.

Stage 1

EDA, Insights & Visualization



Descriptive Analysis



Marketing Campaign *Boost the profit of a marketing campaign*

- A response model can provide a significant boost to the efficiency of a marketing campaign by increasing responses or reducing expenses.
- The objective is to predict who will respond to an offer for a product or service

Acknowledgements

O. Parr-Rud. Business Analytics Using SAS Enterprise Guide and SAS Enterprise Miner. SAS Institute, 2014.

Link Datasets : <https://www.kaggle.com/datasets/rodsaldanha/marketing-campaign>

Descriptive Analysis

Dataset Description

The training dataset contains **2240 samples**, **29 features** and **1 target boolean variable "Response"**:

1. Accepted/Responses Campaign

- **AcceptedCmp1** - 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- **AcceptedCmp2** - 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- **AcceptedCmp3** - 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- **AcceptedCmp4** - 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- **AcceptedCmp5** - 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- **Response (target)** - 1 if customer accepted the offer in the last campaign, 0 otherwise
- **Complain** - 1 if customer complained in the previous 2 years

2. Customer Information

- **ID - Customer's id**
- **Year_Birth** - customer's year of birth
- **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
- **DtCustomer** - date of customer's enrolment with the company
- **Recency** - number of days since the last purchase

3. Sales Product Type

- **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 products in the last 2 years
- **MntSweetProducts** - amount spent on sweet products in the last 2 years
- **MntWines** - amount spent on wine products in the last 2 years
- **MntGoldProds** - amount spent on gold products in the last 2 years

4. Number of Purchases per Type

- **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

5. Customer Information

- **Z_CostContact** = 3 (Cost to contact a customer)
- **Z_Revenue** = 11 (Revenue after client accepting campaign)

Descriptive Analysis

Basic Datasets Information

- **Observations:**

- Di dalam dataset terdiri dari **29 columns (28 features and 1 target boolean)** dan **2240 rows** data
- Pada dataset terdapat 3 jenis tipe data yaitu : **int64, object, float64**
- **Kolom Income** memiliki 2216 nilai non-null, dan **24 nilai null / missing values**

Checking Duplicate Rows

- **Observations:** Data yang kita miliki tidak memiliki duplikat
- Ini dibuktikan pada **kolom ID** yang semua nilainya hanya muncul sekali

```
df[df.duplicated(keep=False)].sort_values(by=list(df.columns.values))

ID  Year_Birth  Education  Marital_Status  Income  Kidhome  Teenhome  Dt_Customer  Recency
<   >
df.duplicated().sum()

0
```

```
Shape of data : (2240, 29)
Number of rows : 2240
Number of columns : 29
```

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

Descriptive Analysis

Checking Missing Values

- **Observations:**

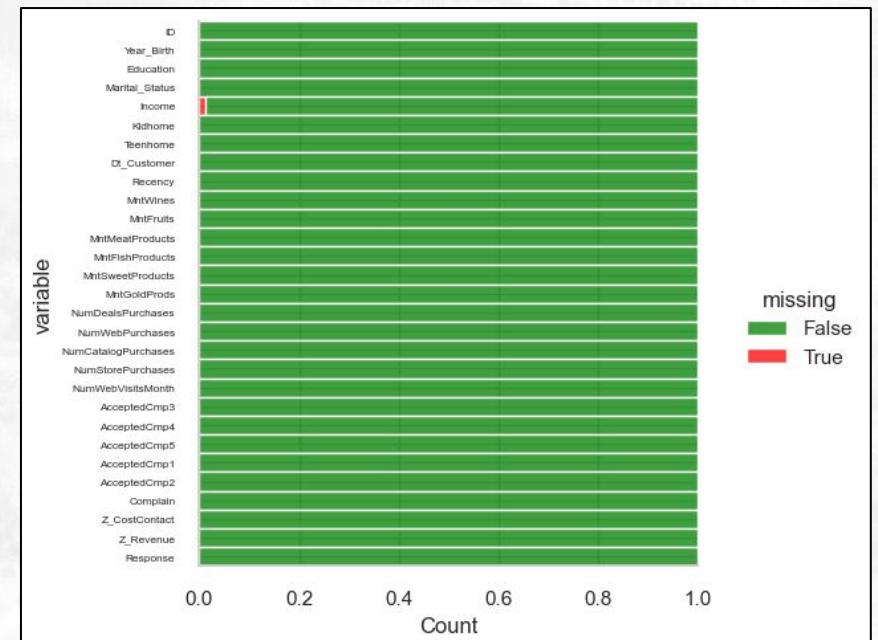
- Kolom **Income** memiliki **24 nilai null / missing values**, persentase sebesar **1.07%** dari jumlah data

- **Hal yang harus dilakukan saat Data Pre-Processing adalah:**

- **Karena data yang dimiliki tidak terlalu banyak**, sehingga untuk Missing Values pada Income akan dilakukan **Imputation** pada tahap Data Preprocessingnya :
 - *Imputation (Median)*, karena *Highly Positively Skewed*
 - *Multivariate Approach (MICE Imputation, KNN Imputer, dll)*

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

	Total	Null	Values	Percentage	Data Type
Income	24	1.071429			int64
ID	0	0.000000			int64
Z_CostContact	0	0.000000			int64
Complain	0	0.000000			object
AcceptedCmp2	0	0.000000			object
AcceptedCmp1	0	0.000000			float64
AcceptedCmp5	0	0.000000			int64
AcceptedCmp4	0	0.000000			int64
AcceptedCmp3	0	0.000000			object
NumWebVisitsMonth	0	0.000000			int64



Descriptive Analysis

Data Types Information

List of Column Types:

- **Date**
 - *Dt_Customer*
- **Categorical (10 Columns) :**
 - **ID** → Nominal
 - **Education** → Ordinal (Levels : Basic - Graduation - 2n Cycle - Master - PhD)
 - **Marital_Status** → Nominal
 - **AcceptedCmp1, AcceptedCmp2, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, Complain, Response** → Nominal (Binary 0 & 1)
- **Continuous (18 Columns):**
Year_Birth, Income, Kidhome, Teenhome, Recency, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, Z_CostContact, Z_Revenue
- **Di dalam dataset ditemukan tipe data yang kurang sesuai yaitu pada kolom berikut:**
 - **Dt-customer** : berbentuk **string/object**, tipe data kurang sesuai sehingga diubah menjadi **datetime (Memudahkan Extract)**
 - **Categorical (int64 dan object)** diubah ke **category** untuk **Business Insight**
 - **Nama kolom dan isi sudah sesuai**

#	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)

Descriptive Analysis

Descriptive Numerical & Date Features Columns

Terdapat beberapa kolom yang memiliki nilai summary yang aneh diantaranya:

- **Year_Birth**
 - **Tahun kelahiran tertua (min)** yaitu **1893**, hal ini yang kemungkinan adanya salah input data sehingga data harus diproses lebih lanjut pada **Outlier**
 - Kemungkinan akan dilakukan **Feature extraction untuk mengambil data Umur/Age** pada range tahun saat ini 2014 (sesuai pada data)

	Dt_Customer	Year_Birth	Income	Kidhome	Teenhome	Recency	
count	2240	2240.000000	2216.000000	2240.000000	2240.000000	2240.000000	
mean	2013-07-10 10:01:42.857142784	1968.805804	52247.251354	0.444196	0.506250	49.109375	
std		-	11.984069	25173.076661	0.538398	0.544538	28.962453
min	2012-07-30 00:00:00	1893.000000	1730.000000	0.000000	0.000000	0.000000	
25%	2013-01-16 00:00:00	1959.000000	35303.000000	0.000000	0.000000	24.000000	
50%	2013-07-08 12:00:00	1970.000000	51381.500000	0.000000	0.000000	49.000000	
75%	2013-12-30 06:00:00	1977.000000	68522.000000	1.000000	1.000000	74.000000	
max	2014-06-29 00:00:00	1996.000000	666666.000000	2.000000	2.000000	99.000000	

- **Income**
 - Memiliki nilai **Mean** yaitu sebesar **52247.25** dan **Median** sebesar **51381.5**, sehingga dapat disimpulkan bahwa data sedikit **Right-skewed Distribution** karena nilai **Mean > Median**
 - Mempunyai **Range 1730.0 (min)** ke **666666.0 (max)** yang sangat jauh, menandakan adanya **outliers**. sehingga perlu dilakukan **Log Transformation/Normalisasi atau Segmentasi** pada data income sebelum melanjutkan ke tahap pemodelan

- **Recency, Kidhome, Teenhome**
 - Memiliki nilai **mean** dan **median** yang sama artinya **kemungkinan** memiliki **normal-skewed distribution / bimodal** (akan **dicek pada univariate analysis**)
 - Untuk **Kidhome dan Teenhome** berpotensi untuk membuat **feature baru 'Dependents'** untuk lebih menggambarkan berapa jumlah anggota keluarga yg dependent

Descriptive Analysis

Descriptive Numerical & Date Features Columns

Terdapat beberapa kolom yang memiliki nilai summary yang aneh diantaranya:

- **Mount of Type Products**

Pada beberapa kolom terdapat summary nilai yang memiliki **nilai Mean** dan **Median** memiliki rentang nilai terlalu jauh seperti pada kolom-kolom berikut :

1. **MntWines**: memiliki nilai **mean** sebesar **303.9** dan **median** sebesar **173.5**
2. **MntFruits** : memiliki nilai **mean** sebesar **26.3** dan **median** sebesar **8**
3. **MntMeatProducts**: memiliki nilai **mean** sebesar **166.9** dan **median** sebesar **67**
4. **MntFishProducts** : memiliki nilai **mean** sebesar **37.5** dan **median** sebesar **12**
5. **MntSweetProducts** : memiliki nilai **mean** sebesar **27.06** dan **median** **8**
6. **MntGoldProds** : memiliki nilai **mean** sebesar **44.02** dan **median** sebesar **24**

	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
count	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000
mean	303.935714	26.302232	166.950000	37.525446	27.062946	44.021875
std	336.597393	39.773434	225.715373	54.628979	41.280498	52.167439
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	23.750000	1.000000	16.000000	3.000000	1.000000	9.000000
50%	173.500000	8.000000	67.000000	12.000000	8.000000	24.000000
75%	504.250000	33.000000	232.000000	50.000000	33.000000	56.000000
max	1493.000000	199.000000	1725.000000	259.000000	263.000000	362.000000

Jika dilihat dari beberapa nilai **mean** dan **median** yang memiliki **jarak agak aneh**, kemungkinan memiliki **jumlah outlier yang tinggi** dan **distribusi** yang **skewed**.

maka untuk Data Preprocessing perlu dilakukan **Log Transformation**, digunakan untuk mengubah data skewed mendekati / sesuai dengan normalitas.

Descriptive Analysis

Descriptive Numerical & Date Features Columns

Terdapat beberapa kolom yang memiliki nilai summary yang aneh diantaranya:

- **Moderately Positively Skewed** (Sedikit skew ke kanan) terdapat pada kolom berikut:
 - 1. NumDealsPurchases**
 - 2. NumWebPurchases**
 - 3. NumCatalogPurchases**
 - 4. NumStorePurchases**
 - 5. NumWebVisitsMonth**

Jika dilihat pada data ini nilai **mean** dan **median** ditemukan memiliki jarak yang tidak lumayan jauh, namun **kemungkinan** memiliki **distribusi** yang **skewed** ke kanan, maka **perlu di visualisasi lebih lanjut**, kemudian untuk Data Preprocessing perlu dilakukan **Log Transformation**, digunakan untuk mengubah data skewed mendekati / sesuai dengan normalitas.

	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	Z_CostContact	Z_Revenue
count	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000	2240.000000
mean	2.325000	4.084821	2.662054	5.790179	5.316518	3.000000	11.000000
std	1.932238	2.778714	2.923101	3.250958	2.426645	0.000000	0.000000
min	0.000000	0.000000	0.000000	0.000000	0.000000	3.000000	11.000000
25%	1.000000	2.000000	0.000000	3.000000	3.000000	3.000000	11.000000
50%	2.000000	4.000000	2.000000	5.000000	6.000000	3.000000	11.000000
75%	3.000000	6.000000	4.000000	8.000000	7.000000	3.000000	11.000000
max	15.000000	27.000000	28.000000	13.000000	20.000000	3.000000	11.000000

- **Z_CostContact, Z_Revenue**

Dapat diketahui bahwa nilai **cost (3)** dan **revenue (11)** ini hanya memiliki satu nilai, sehingga akan di **Drop pada step modelling nantinya** karena tidak memberikan informasi yang signifikan terhadap model prediksi

Descriptive Analysis

Descriptive Categorical Features Columns

	count	unique	top	freq
ID	2240	2240	0	1
Year_Birth	2240	59	1976	89
Education	2240	5	Graduation	1127
Marital_Status	2240	8	Married	864
AcceptedCmp1	2240	2	0	2096
AcceptedCmp2	2240	2	0	2210
AcceptedCmp3	2240	2	0	2077
AcceptedCmp4	2240	2	0	2073
AcceptedCmp5	2240	2	0	2077
Complain	2240	2	0	2219
Response	2240	2	0	1906

Observation Result based on Descriptive Analysis

1. Terlalu banyak kategori pada kolom **ID**
2. Customer **banyak yang lahir (Year Birth)** pada tahun **1976 (age = 38 years)** sebanyak **89 orang**
3. Kategori **Education, 2n Cycle** dan **Master** memiliki arti yang sama
4. Kategori **Education** pada customer yang mayoritas memiliki kategori pendidikan **Graduation** sebanyak **1127** orang, memiliki **nilai sangat besar** dibanding yang lain
5. Dalam kategori **Marital Status**, customer **majoritas sudah menikah (Married) 864 orang**
6. Dalam kategori **Marital Status, Single** dan **Alone** memiliki **arti yang sama**
7. Dalam kategori **Marital Status, Together** dan **Married** memiliki arti yang sama
8. Dalam kategori **Marital Status**, ada beberapa data yang tidak jelas apa yang dimaksud yaitu **Absurd** dan **YOLO** maka disarankan digabung dan diganti **Others**
9. Pada kategori **AcceptedCmp(1-5)**, customer mayoritas tidak merespon / accept dari campaign yang dilakukan
10. Pada kategori **Complain**, customer mayoritas tidak pernah complain dari campaign yang dilakukan
11. Target yang kita miliki terdapat pada kolom **Response** yang mana memiliki **ketimpangan yang sangat tinggi (Imbalanced Data)**
 - **Tidak merespon = 1906**
 - **Merespon = 334**

Descriptive Analysis

Descriptive Categorical Features Columns

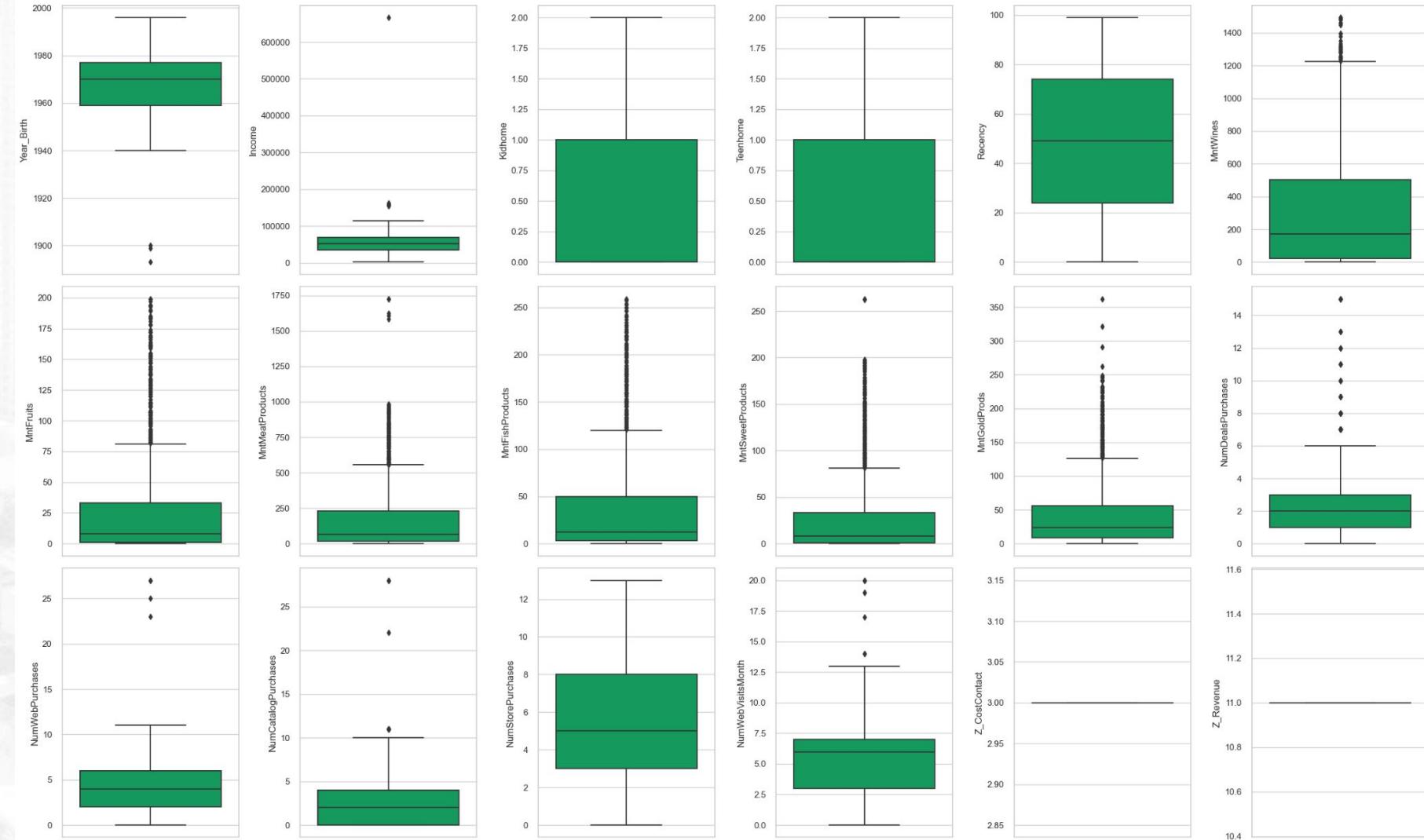
	count	unique	top	freq
ID	2240	2240	0	1
Year_Birth	2240	59	1976	89
Education	2240	5	Graduation	1127
Marital_Status	2240	8	Married	864
AcceptedCmp1	2240	2	0	2096
AcceptedCmp2	2240	2	0	2210
AcceptedCmp3	2240	2	0	2077
AcceptedCmp4	2240	2	0	2073
AcceptedCmp5	2240	2	0	2077
Complain	2240	2	0	2219
Response	2240	2	0	1906

Hal yang harus dilakukan saat Data Pre-Processing adalah:

1. Akan dilakukan **replace data / menyatukan yang memiliki arti yang sama** agar mengurangi jumlah dimensi maupun **redundansi pada data**
2. Pada kolom **Response**, Sebaran kategori yang timpang pada target. Pada target, menyebabkan proses Machine Learning gagal. Oleh karena itu, perlu dilakukan **Sampling Data (Undersampling / Oversampling / Combined / SMOTE / dll)**
3. Akan dilakukan **Feature Encoding** pada kolom **Education dan Marital_Status** untuk proses modelling, karena masih belum memiliki representasi nilai numerical

Univariate Analysis

DISTRIBUTION AND TABLE OF NUMERICAL VALUES



Univariate Analysis

Terdapat **outlier** pada beberapa kolom di dalam dataset, diantaranya:

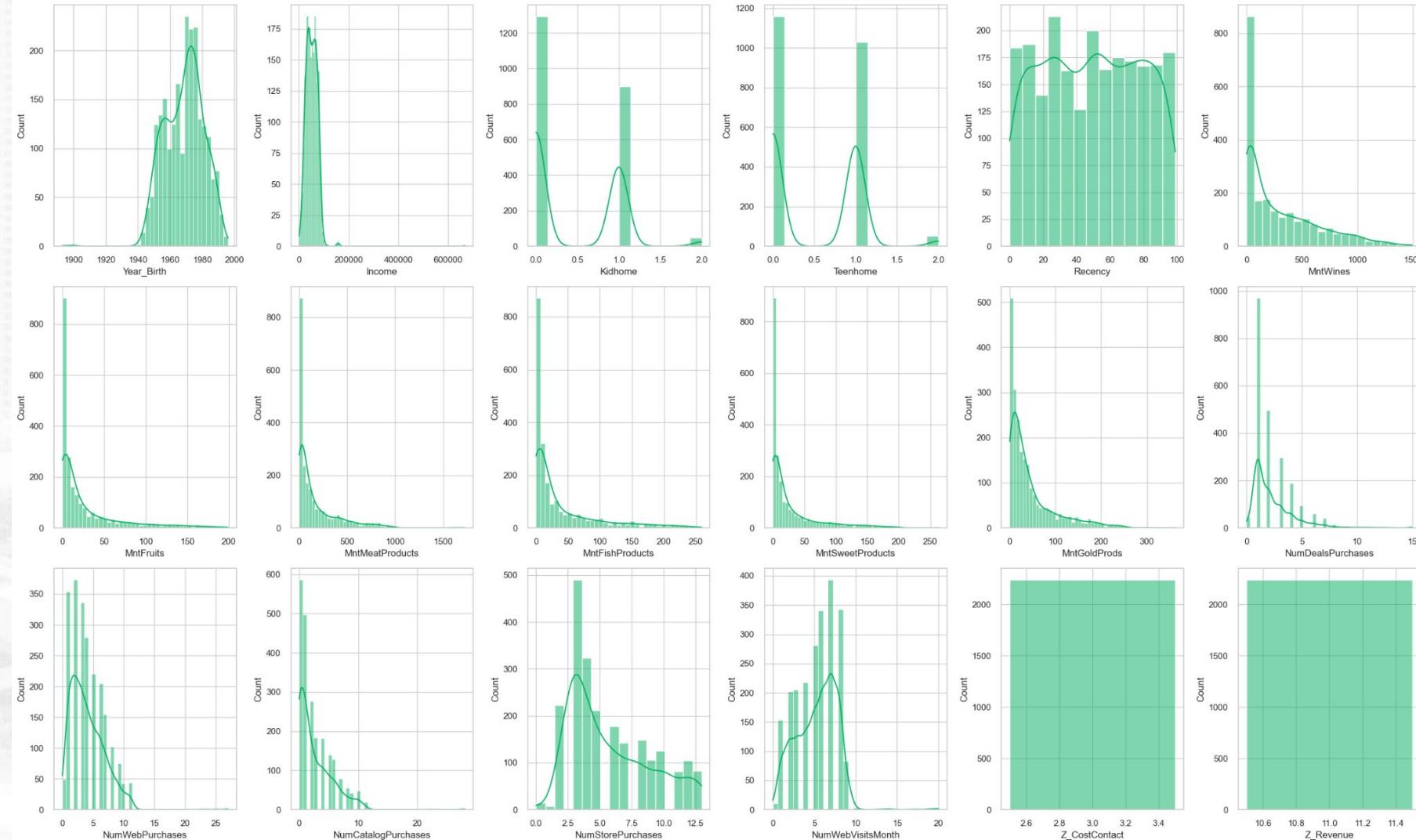
- **Year_Birth, outlier** terjauh adalah **di bawah 1900**
- **Income, outlier** terjauh adalah **di atas \$600,000**
- **MntWines, outlier** berada pada angka **1200 keatas**
- **MntFruits, outlier** berada di **sekitar angka 80** sampai **200**
- **MntMeatProducts, outlier** terjauh ada di **sekitar angka 1,750**
- **MntFishProducts, outlier** berada di **sekitar angka 125** sampai **diatas 250**
- **MntSweetProducts, outlier** terjauh berada di **sekitar angka 250**
- **MntGoldProds, outlier** terjauh berada di **sekitar angka 350**
- **NumDealsPurchases, outlier** terjauh berada di **angka 15**
- **NumWebPurchases, outlier** berada di sekitar **angka 25**
- **NumCatalogPurchases, outlier** terjauh berada di atas **angka 25**
- **NumWebVisitMonth, outlier** terjauh berada di **angka 20**

Hal yang harus dilakukan pada saat **Data Pre-Processing** adalah:

- Mengaplikasikan **Log Transformation** untuk **Feature Scaling** dan **Handling Outlier** yang mana transformasi ini **de-emphasizes / minimize outliers** dan dapat membantu untuk **potentially obtain a bell-shaped / normal distribution**. Hal ini dilakukan karena **jumlah data yang terbatas yaitu sebanyak 2240 baris** data saja, sehingga menjadi pilihan terbaik karena dilakukan tanpa menghapus baris data.
- Alternatif lainnya, membersihkan data dengan cara **menghapus outliers** berdasarkan **IQR** atau **Z-score**, akan tetapi hal ini akan mengurangi data yang dimiliki

Univariate Analysis

DISTRIBUTION AND TABLE OF NUMERICAL VALUES



Univariate Analysis

DISTRIBUTION AND TABLE

	Column Name	Skewness	Kurtosis	Type of Distribution
0	Year_Birth	-0.350000	0.713000	Moderately Normal Distribution (Symmetric)
1	Income	6.759000	159.274000	Highly Positively Skewed
2	Kidhome	0.635000	-0.781000	Bimodal Distribution
3	Teenhome	0.407000	-0.987000	Bimodal Distribution
4	Recency	-0.002000	-1.202000	Normal Distribution (Symmetric)
5	MntWines	1.175000	0.595000	Highly Positively Skewed
6	MntFruits	2.101000	4.039000	Highly Positively Skewed
7	MntMeatProducts	2.082000	5.502000	Highly Positively Skewed
8	MntFishProducts	1.918000	3.087000	Highly Positively Skewed
9	MntSweetProducts	2.135000	4.364000	Highly Positively Skewed
10	MntGoldProds	1.885000	3.541000	Highly Positively Skewed
11	NumDealsPurchases	2.417000	8.914000	Highly Positively Skewed
12	NumWebPurchases	1.382000	5.688000	Highly Positively Skewed
13	NumCatalogPurchases	1.880000	8.027000	Highly Positively Skewed
14	NumStorePurchases	0.702000	-0.623000	Moderately Positively Skewed
15	NumWebVisitsMonth	0.208000	1.815000	Moderately Normal Distribution (Symmetric)
16	Z_CostContact	NaN	NaN	Uniform Distribution
17	Z_Revenue	NaN	NaN	Uniform Distribution

Berdasarkan distribusi data di samping, dapat diketahui bahwa ada beberapa variabel yang memiliki *outlier* didalamnya dan beberapa memiliki *Skewed Distribution*. Berikut adalah beberapa variabel tersebut:

A. **Normal distribution**

- **Recency** Normal Distribution (Symmetric)
- **Year_Birth** Moderately Normal Distribution (Symmetric)
- **NumWebVisitsMonth** Moderately Normal Distribution (Symmetric)

B. **Uniform distribution**

- **Z_CostContact** Uniform Distribution - Memiliki satu nilai saja
- **Z_Revenue** Uniform Distribution - Memiliki satu nilai saja

C. **Positive skewed distribution**

- **Income** Income
- Amount of Wines Products **MntWines**
- Amount of Fruits Products **MntFruits**
- Amount of Meats Products **MntMeatProducts**
- Amount of Fish Products **MntMeatProducts**
- Amount of Sweet Products **MntSweetProducts**
- Amount of Golds Products **MntGoldProds**
- Number Deals Purchases **NumDealsPurchases**
- Number Web Purchases **NumWebPurchases**
- Number Catalog Purchases **NumCatalogPurchases**
- Number Store Purchases **NumStorePurchases**

D. **Bimodal distribution**

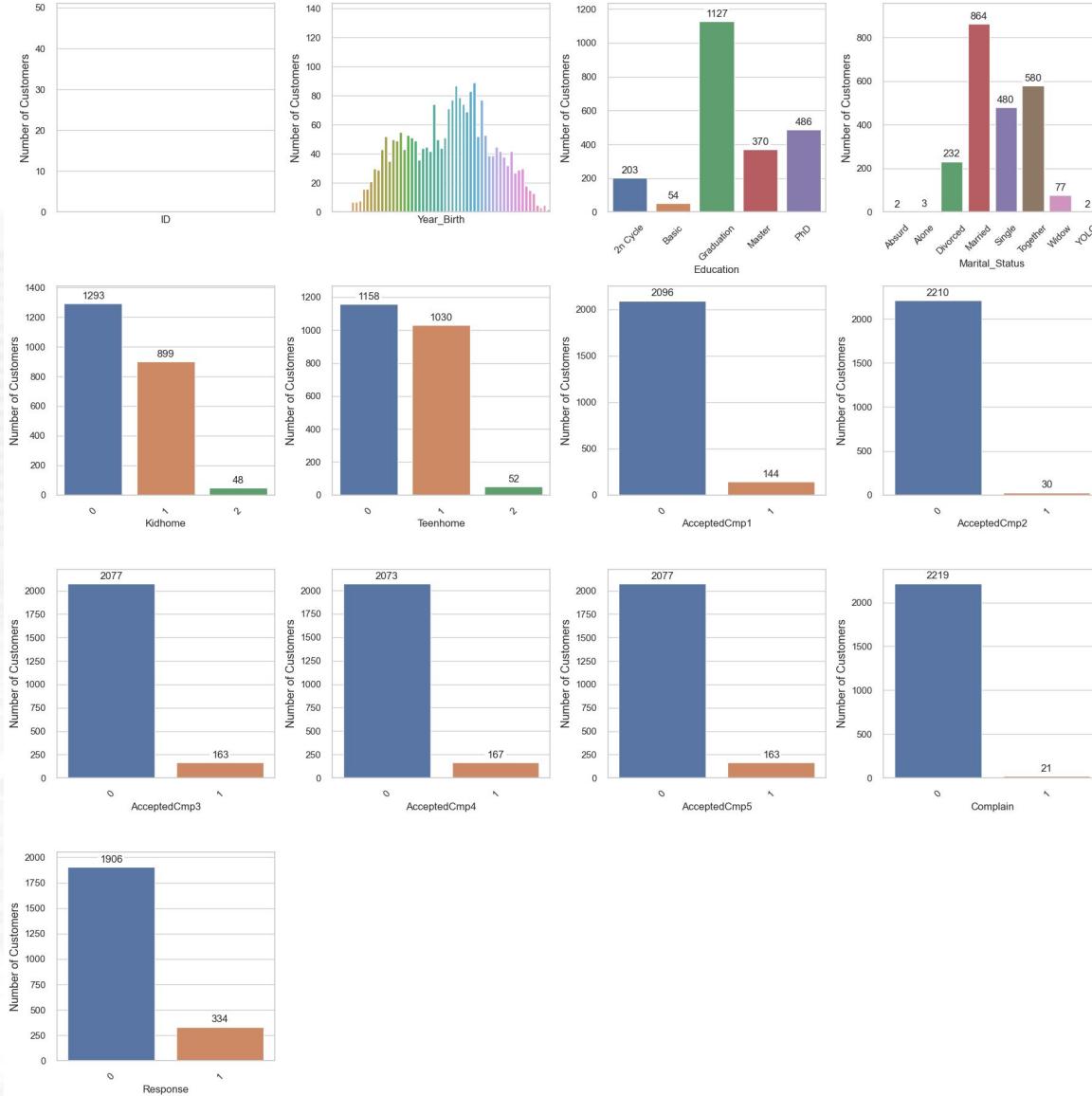
- Number of small children in customer's household **Kidhome**
- Number of teenagers in customer's household **Teenhome**

Rekomendasi pada data pre-processing:

Data yang mengalami **Positive Skewed** Distribution dilakukan **Log Transformation** sehingga data bisa menjadi **normal** distribution.

Univariate Analysis

Distribution of Categorical Values



1. Terlalu banyak kategori pada kolom **ID**
2. Kolom **Education** dan **Marital_Status** memiliki beberapa kategori **yang value-nya sama dan ambigu**.
 - a. Kategori **Education**, **2n-Cycle** dan **Master** memiliki arti yang sama
 - b. Kategori **Education** pada customer yang mayoritas memiliki kategori pendidikan **Graduation** sebanyak **1127** orang, memiliki **nilai sangat besar** dibanding yang lain
 - c. Dalam kategori **Marital Status**, customer **majoritas sudah menikah (Married)** **864** orang
 - d. Dalam kategori **Marital Status**, **Single** dan **Alone** memiliki **arti yang sama**
 - e. Dalam kategori **Marital Status**, **Together** dan **Married** memiliki arti yang sama
 - f. Dalam kategori **Marital Status**, ada beberapa data yang tidak jelas apa yang dimaksud yaitu **Absurd** dan **YOLO** maka disarankan digabung dan diganti **Others**
3. Kolom **Kidhome** dan **Teenhome** mayoritas customer tidak memiliki anak dan remaja (value 0)
4. Kolom **AcceptedCmp1**, **AcceptedCmp2**, **AcceptedCmp3**, **AcceptedCmp4**, **AcceptedCmp5**, **Complain**, dan **Response** value didominasi dengan value 0 (Tidak Response / Complain)
5. Target yang kita miliki terdapat pada kolom **Response** yang mana memiliki **ketimpangan yang sangat tinggi (Imbalanced Data)**
 - Tidak merespon = **1906**
 - Merespon = **334**

Univariate Analysis

Note for Data Pre-Processing Stage

Hal yang harus dilakukan saat *Data Pre-Processing*:

1. Kolom **ID** di drop untuk proses modelling
2. Dari kolom **Year_Birth** dibuat kolom baru yaitu kolom **Age** yang menunjukkan umur seorang customer.
3. Akan dilakukan **replace data / menyatukan yang memiliki arti yang sama** agar mengurangi jumlah dimensi maupun redundansi pada data
4. Melakukan **Label Encoding** pada kolom **Education**.
5. Melakukan **One Hot Encoding (OHE)** pada kolom **Marital_Status**.
6. Pada kolom **Response**, Sebaran kategori yang timpang pada feature mengindikasikan ketidakgunaan feature. Pada target, menyebabkan proses Machine Learning gagal. Oleh karena itu, perlu dilakukan **Sampling Data (Undersampling / Oversampling / Combined / SMOTE / dll)**

Multivariate Analysis

HEATMAP AND TABLE

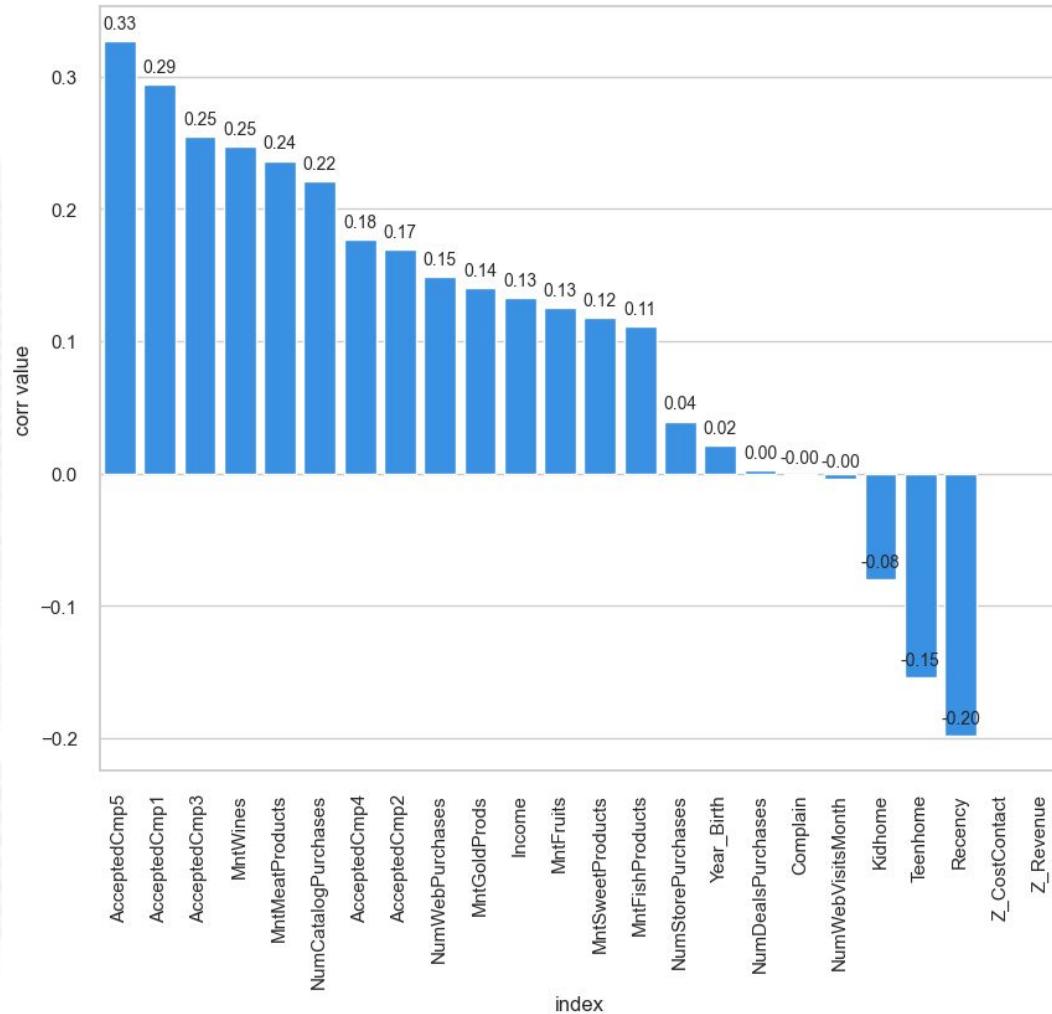


Year_Birth Income Kidhome Teenhome Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases NumStorePurchases NumWebVisitsMonth AcceptedCmp1 AcceptedCmp2 AcceptedCmp3 AcceptedCmp4 AcceptedCmp5 Complain Response

	index	corr value	Corr Type
0	Response	1.000000	Positif
1	AcceptedCmp5	0.326634	Positif
2	AcceptedCmp1	0.293982	Positif
3	AcceptedCmp3	0.254258	Positif
4	MntWines	0.247254	Positif
5	MntMeatProducts	0.236335	Positif
6	NumCatalogPurchases	0.220810	Positif
7	Recency	0.198437	Negatif
8	AcceptedCmp4	0.177019	Positif
9	AcceptedCmp2	0.169293	Positif
10	Teenhome	0.154446	Negatif
11	NumWebPurchases	0.148730	Positif
12	MntGoldProds	0.139850	Positif
13	Income	0.133047	Positif
14	MntFruits	0.125289	Positif
15	MntSweetProducts	0.117372	Positif
16	MntFishProducts	0.111331	Positif
17	Kidhome	0.080008	Negatif
18	NumStorePurchases	0.039363	Positif
19	Year_Birth	0.021325	Positif
20	NumWebVisitsMonth	0.003987	Negatif
21	NumDealsPurchases	0.002238	Positif
22	Complain	0.001707	Negatif

Multivariate Analysis

HEATMAP AND TABLE



Korelasi Feature dengan Response :

- A. Top 10 Yang berkorelasi tinggi ke target sebagai berikut, Kemungkinan besar top ini bisa menjadi feature yang paling relevan dan harus dipertahankan:
- **AcceptedCmp5** - 0.32 – Positif
 - **AcceptedCmp1** - 0.29 – Positif
 - **AcceptedCmp3** - 0.25 – Positif
 - **MntWines** - 0.24 – Positif
 - **MntMeatProducts** - 0.23 – Positif
 - **NumCatalogPurchases** - 0.22 – Positif
 - **Recency** - 0.19 – Negatif
 - **AcceptedCmp4** - 0.17 – Positif
 - **AcceptedCmp2** - 0.16 – Positif
 - **Teenhome** - 0.15 – Negatif
- B. Korelasi kolom **Response** dengan kolom lainnya cenderung **rendah**. Dari seluruh korelasi antara feature-target berada di range **0.00** sampai **0.33**. Oleh karena itu, kami memutuskan untuk membuat nilai threshold di angka **0.15**. Feature-feature di atas yang kemungkinan kami pertahankan adalah feature yang memiliki nilai korelasi **>0.15**.

Multivariate Analysis

Selain itu, pada korelasi antar-feature, terdapat pola yang menarik sebagai berikut:

- A. Untuk kolom **Complain**, **Z_CostContact** dan **Z_Revenue** berpotensi untuk dihapus, karena tidak memiliki korelasi dibanding kolom lainnya
- B. **Year Birth**
 - **Year_Birth** berkorelasi positif dengan **Kidhome**, bisa dikatakan semakin muda maka semakin banyak pula anak kecilnya sedangkan untuk kolom **Teenhome** berkorelasi negatif atau semakin tua customer maka jumlah anak remaja semakin banyak.
- C. **Kidhome & Teenhome**
 - Customer yang (**Kidhome**) memiliki anak kecil **Income** nya cenderung rendah.
 - Customer yang (**Kidhome**) memiliki anak kecil cenderung lebih sering mengunjungi **web** dan melakukan pembelian ketika sedang diskon.
 - Customer yang (**Teenhome**) memiliki anak remaja cenderung lebih banyak melakukan pembelian ketika sedang **diskon**.
- D. **Income**
 - Customer yang memiliki **Income** yang tinggi cenderung banyak melakukan pembelian.
 - Kolom **Income** berkorelasi **positif** cukup besar dengan **MnWines**, **MntFruits**, **MntMeatProduct**, **MntFishProduct**, **MntSweetProduct** dan **MntGoldProduct**.
 - Kolom **Income** berkorelasi **positif** cukup besar dengan **NumWebPurchases**, **NumCatalogPurchases**, dan **NumStorePurchases** sedangkan dengan kolom **NumWebVisitsPurchases** berkorelasi **negatif** cukup besar, bisa dikatakan bahwa semakin besar **income** customer maka mayoritas tempat yang dipilih untuk melakukan pembelian adalah **Web**, **Catalog** dan **Store**, sedangkan customer yang memiliki **income** rendah cenderung lebih banyak melakukan pembelian melalui **web** atau lebih sering mengunjungi web.

Multivariate Analysis

Lanjutan korelasi antar-feature :

D. Product

- Kolom **MntWines** berkorelasi **positif** cukup besar dengan **MntMeatProduct**, kemungkinan customer membeli wine juga membeli meat.
- Kolom **MntFruits** berkorelasi **positif** cukup besar dengan **MntMeatProduct**, **MntFishProduct** dan **MntSweetProduct**, kemungkinan ketika customer membeli product tersebut bersamaan.
- **MntMeatProduct** berkorelasi positif cukup besar dengan **NumCatalogPurchases**, bisa dikatakan sebagian besar pembelian meat product dilakukan melalui catalog.
- **MntWines** paling banyak dibeli melalui **katalog** dari pada metode pembelian lainnya.

E. Purchases

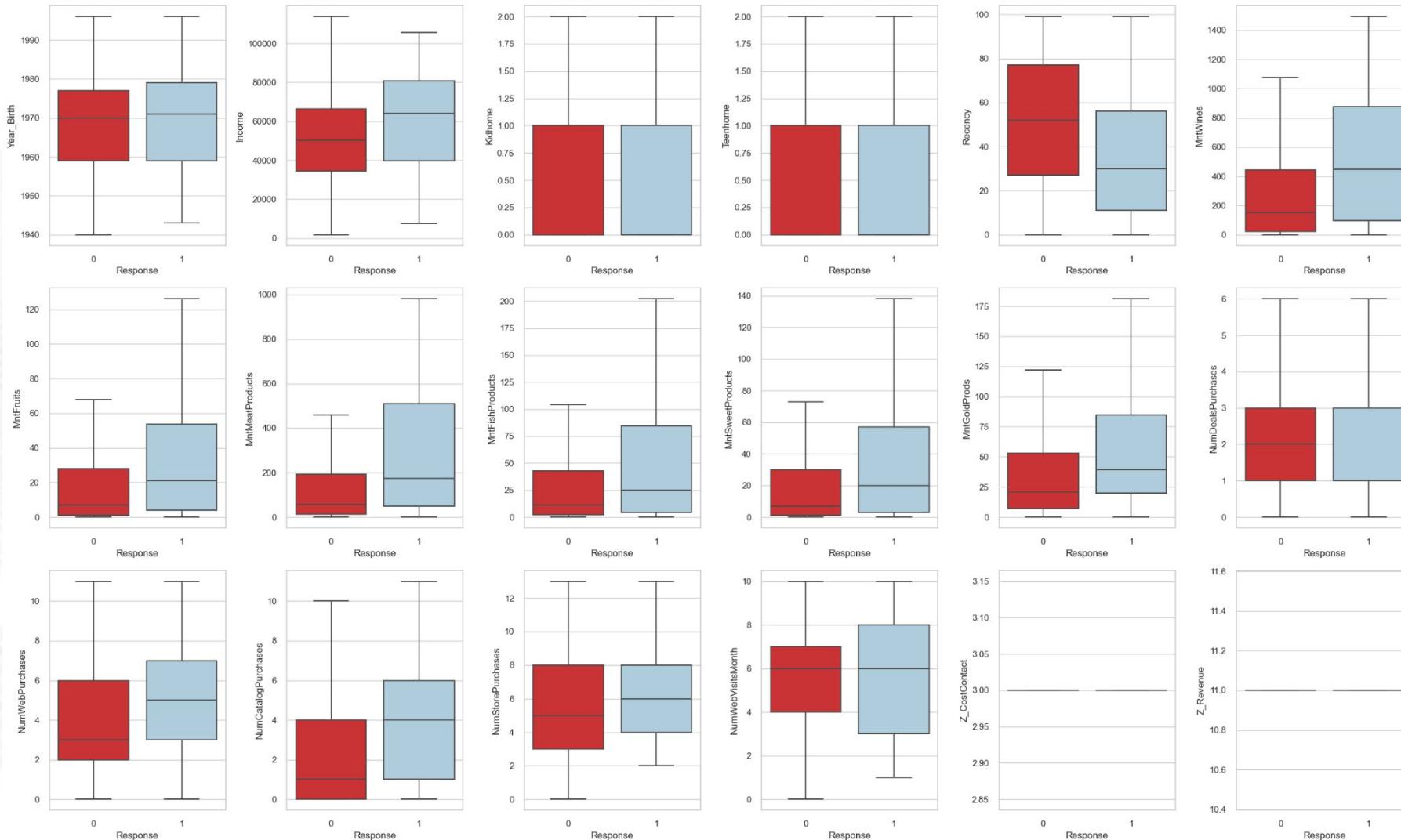
- Kolom **NumDealsPurchases** berkorelasi **positif** dengan **NumWebVisitMonth**, ketika sedang diskon customer yang mengunjungi web meningkat.
- Kolom **NumWebVisitMonth** berkorelasi **negatif** cukup besar dengan **NumCatalogPurchases** dan **NumStorePurchases**, ketika customer lebih sering mengunjungi web maka pembelian melalui catalog dan store menurun.

F. Additional

- Kombinasi 5 Kolom product cukup tinggi nilai korelasinya. Oleh karena itu, customer cenderung suka membeli lebih dari 1 product dalam sekali berbelanja.
- Produk **Wines** dan **Gold** lebih banyak dibeli menggunakan **website**. Sedangkan **Fruits**, **Meat**, **Fish** dan **Sweet** dibeli melalui **Store** maupun **Catalog**.
- Produk yang ditawarkan menggunakan **Deals** (potongan harga) belum terlalu menarik minta customer karena korelasi dengan kolom produk apapun sangat **rendah**.
- Customer yang menggunakan **Deals** lebih banyak customer yang menggunakan **Website** untuk membeli barang/produk.

Multivariate Analysis

BOXPLOT VS RESPON

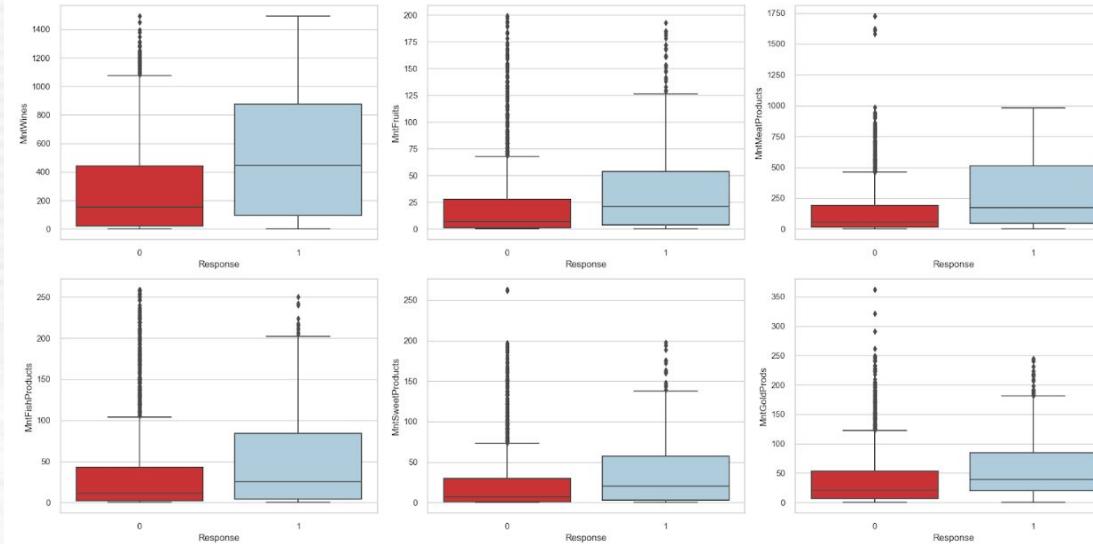


- **Income**, Customer yang merespon campaign cenderung memiliki income yang lebih tinggi dengan rata-rata income yang dimiliki customer sekitar 65000, dibanding customer yang tidak merespon campaign memiliki income rata-rata hanya sekitar 50000.
- **Recency**, Customer yang merespon campaign cenderung lebih aktif untuk berbelanja dengan rata-rata hari terakhir pembelian produk sekitar 30 hari, dibanding customer yang tidak merespon campaign memiliki rata-rata hari terakhir pembelian produk sekitar lebih dari 50 hari

Multivariate Analysis

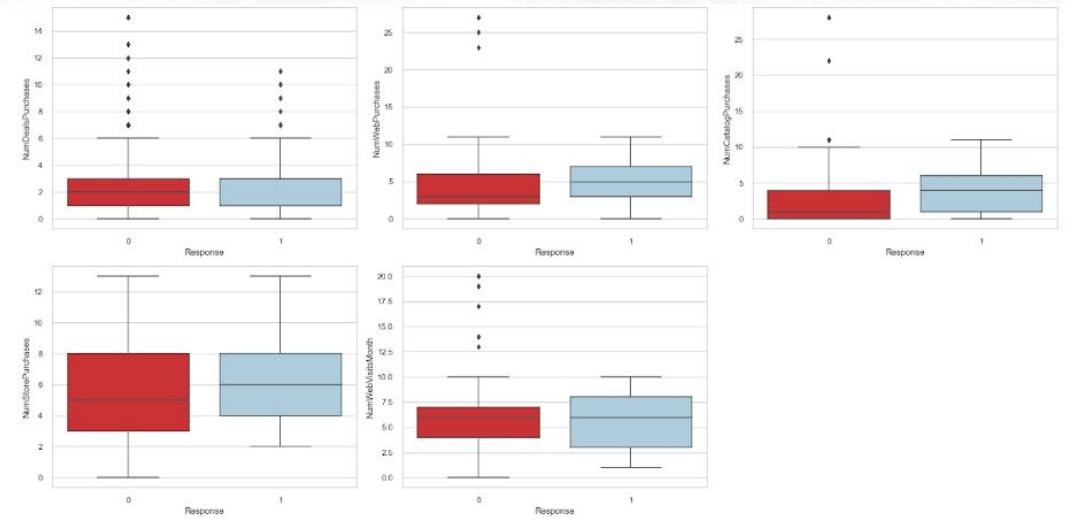
SIMPLIFIED BOXPLOT VS RESPON

Tim melakukan 'Simplified Analysis based on MntProducts'. Pada dataset ini terdapat banyak jenis produk yang ditawarkan seperti **MntWines**, **MntFruits**, **MntMeatProducts**, **MntFishProducts**, **MntSweetProducts**, **MntGoldProducts**.



Berdasarkan boxplot diatas dapat disimpulkan bahwa, pada **Response yang menerima (values 1)**, memiliki nilai **Mount** (Jumlah Pembelian) di tiap product **lebih tinggi** daripada yang tidak merespon (values 0)

Adapun 'Simplified Analysis based on Type Purchases' yang terdiri dari **NumDealsPurchases**, **NumWebPurchases**, **NumCatalogPurchases**, **NumStorePurchases**, **NumWebVisitsMonth**



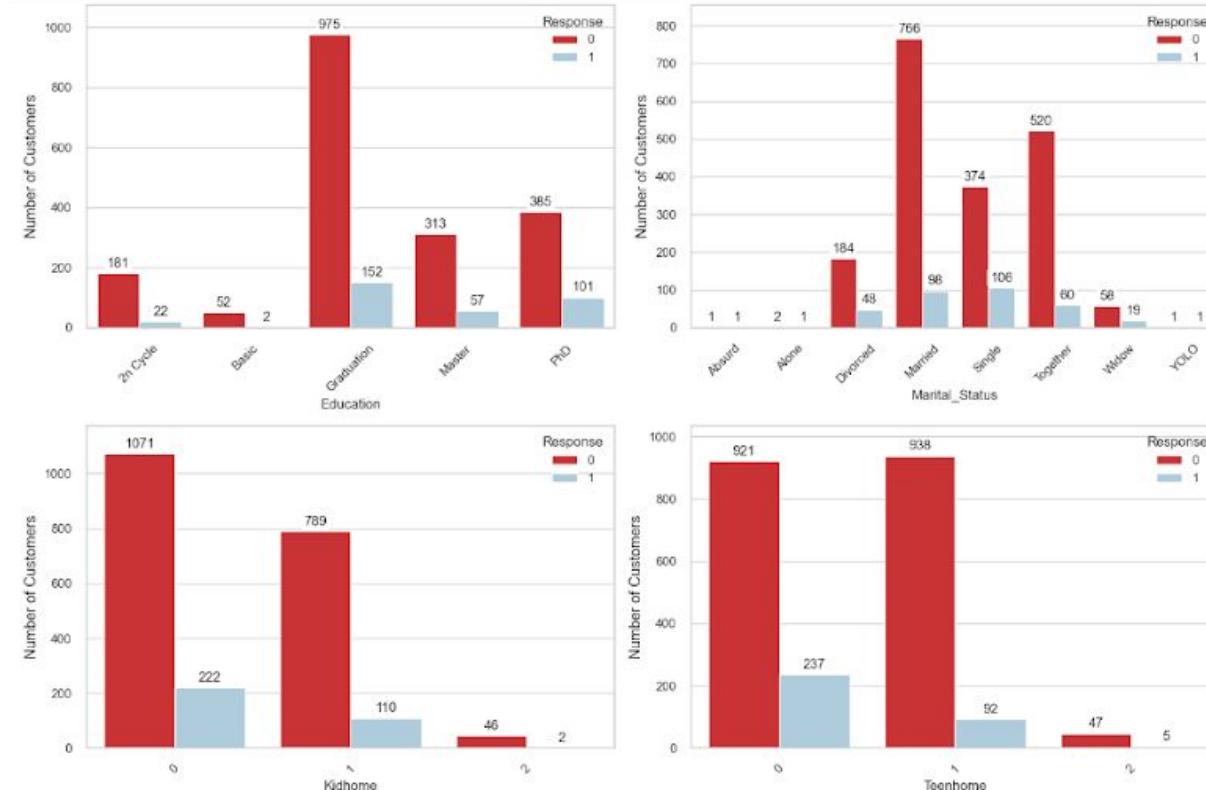
1. Untuk pembelian melalui **Web**, **Catalog**, **Web**, Pada **Response yang menerima (values 1)**, memiliki nilai **Purchases** (Jumlah Pembelian) lumayan sedikit **lebih tinggi** daripada yang tidak merespon (values 0)
2. Pada pembelian melalui **Deals / Discount** dan **Store** memiliki nilai yang **tidak terlalu berbeda**

Multivariate Analysis

SIMPLIFIED COUNTPLOT VS RESPON

Pada bagian ini, Tim melakukan analisis berdasarkan **Status Customer dan Campaign/Complain terhadap Response**.

'Simplified based on Status Customer'



Berdasarkan countplot disamping, dapat disimpulkan bahwa:

1. Education

Graduation, PhD dan Master memiliki jumlah respon yang tinggi, masing-masing ada pada ratio perbandingan respon > 13.4% (maks 20%)

2. Marital_Status

Single, Married, Together, dan Divorced memiliki jumlah respon yang tinggi, namun ratio perbandingan respon vs no responnya < 22.4% (maks 50%)

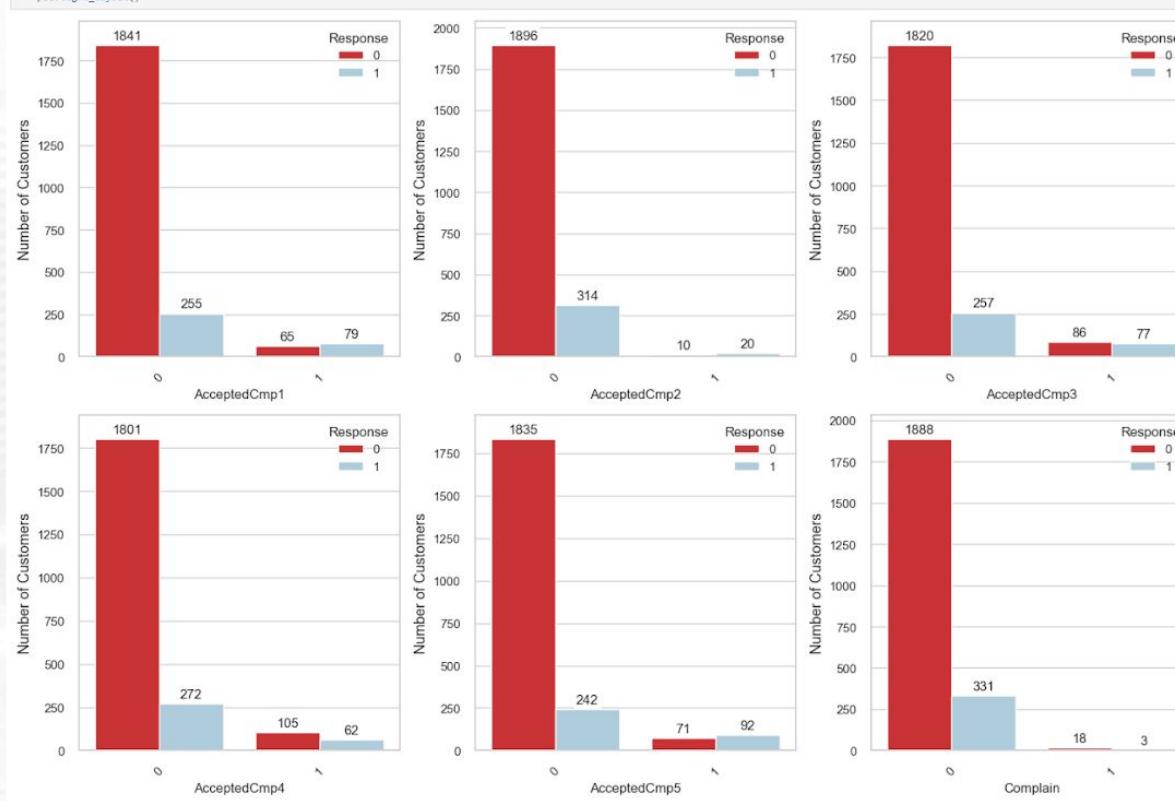
3. Kidhome & Teenhome

Semakin tinggi jumlah anak/remaja yang dimiliki customer, maka semakin kecil kemungkinan customer menerima Response (marketing campaign terakhir), sehingga lebih baik perusahaan menargetkan campaign kepada customer yang tidak memiliki anak/remaja. Begitupun pada ratio perbandingan respon vs no responnya menurun.

Multivariate Analysis

SIMPLIFIED BOXPLOT VS RESPON

'Simplified based on Campaign/Complain'



Berdasarkan countplot disamping dapat disimpulkan bahwa:

1. AcceptedCmp columns

- Dari segi Response tertinggi cenderung pada yang tidak accept campaign.
- Dari yang Accept Campaign, hanya pada 1, 3, 4, 5 yang memiliki nilai yang kebih besar. Bisa juga dilihat pada ratio perbandingan respon vs no responnya 33-56%, berarti tidak berbeda signifikan dan perbandingannya lumayan sama.
- Pada Acccept Campaign 2 Jumlahnya sangat sedikit, namun ratio perbandingan respon vs no responnya yang paling tinggi 66%.

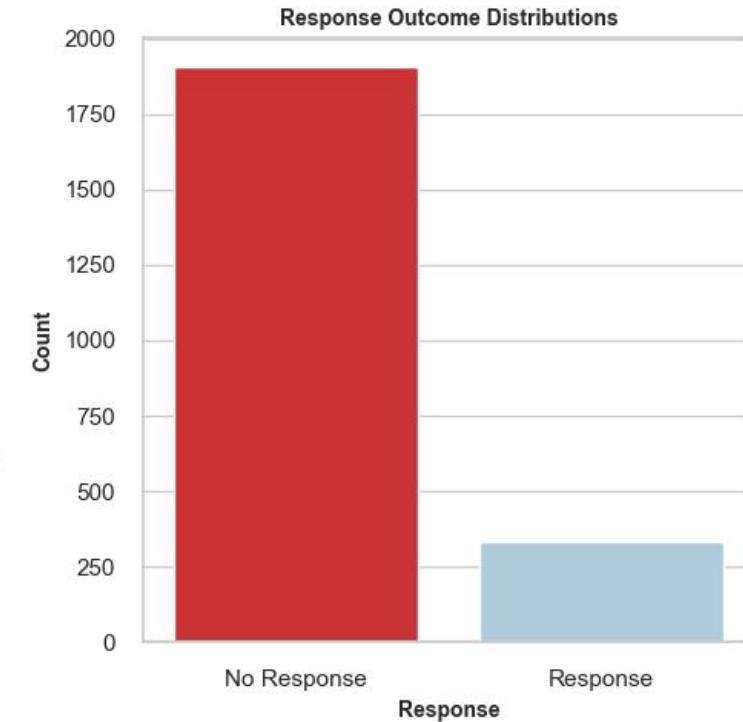
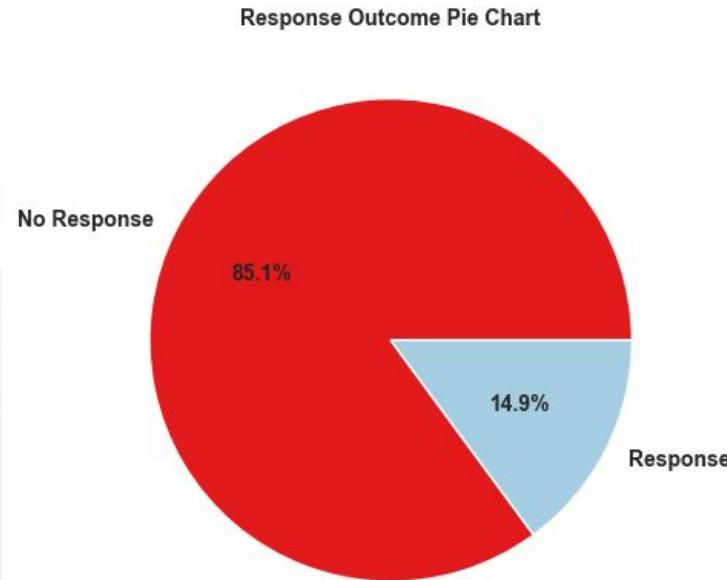
2. Complain

- Dari segi Response tertinggi cenderung pada yang tidak ada complain pada campaign

Business Insight

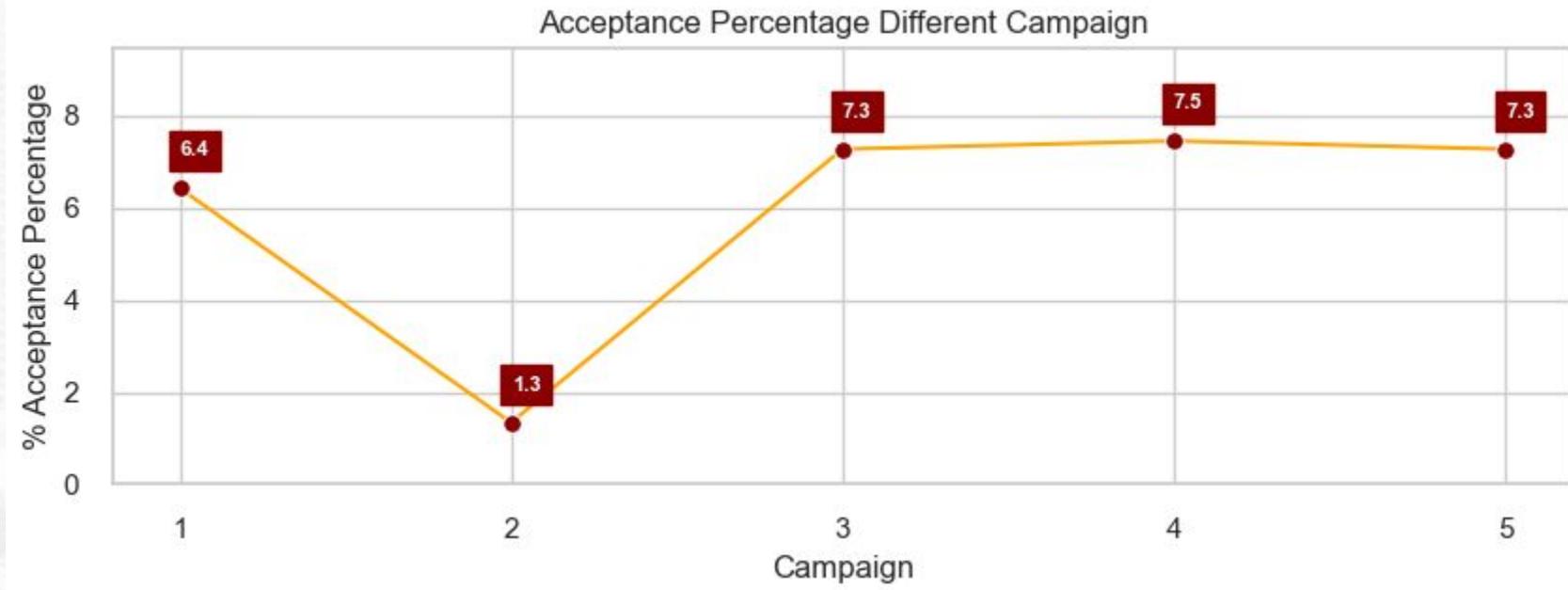
Response Ratio

Response	total	%
0	0	1906 85.090000
1	1	334 14.910000



- Jumlah yang **Response** signifikan lebih kecil dibandingkan yang Tidak Merespon **No Response**, dengan ratio **14.9%**
- Ini berarti adanya **data imbalance** pada campaign terakhir (ke-6) perusahaan
- Sehingga bisa dilakukan **upaya peningkatan proses marketing campaign** agar lebih banyak customer yang merespon campaign

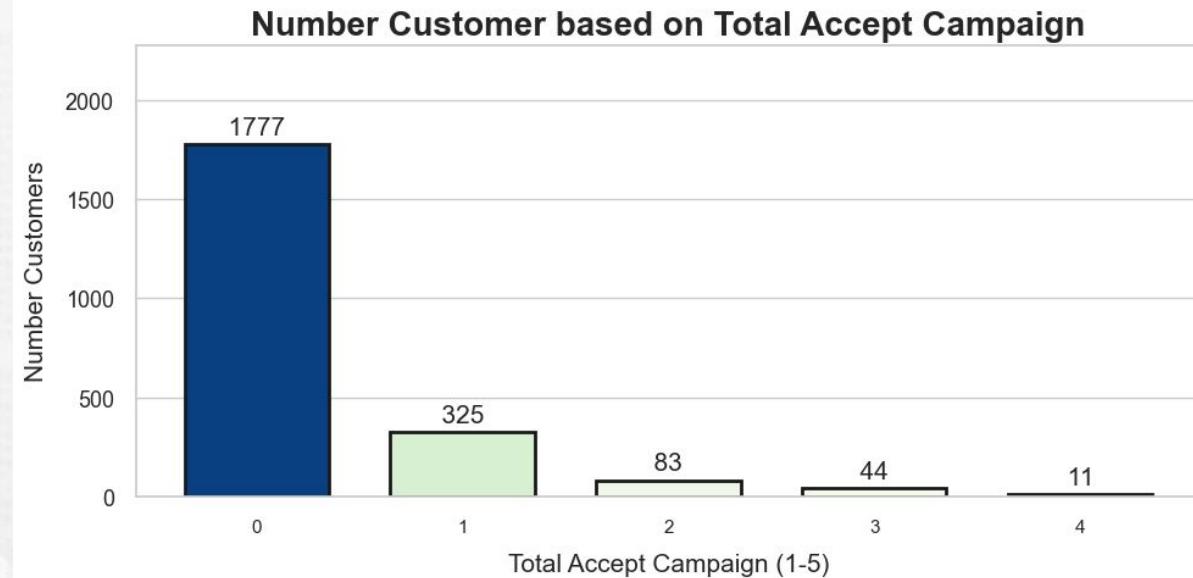
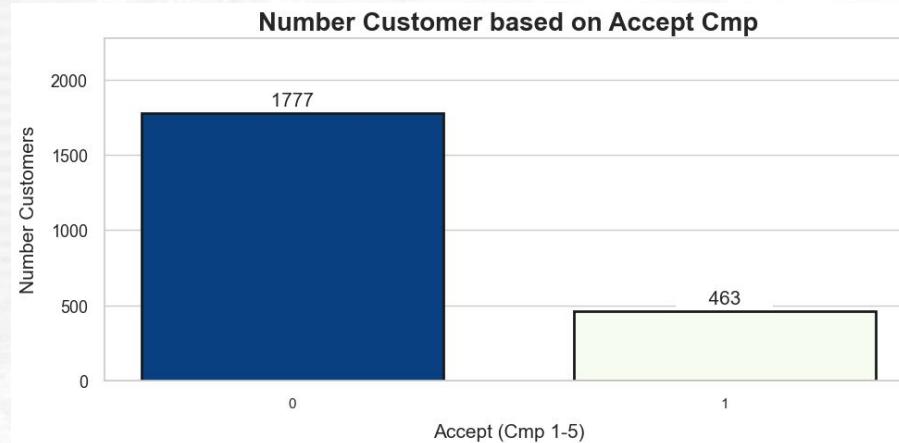
Acceptance Rate for each Campaign (1-5)



Dari **Campaign Rate** yang telah kita lakukan, terdapat **perubahan** yang terjadi dari **Campaign 1 ke Campaign 2 (menurun drastis) dan 3 (naik signifikan)**. Adapun dari **Campaign 3-5** Semuanya memiliki rate yang kurang lebih **sama** disekitar ~7%. Sehingga dari perusahaan dan tim perlu melakukan **identifikasi lanjutan untuk mengetahui apakah ada pola dari pembelian customer**

Business Insight

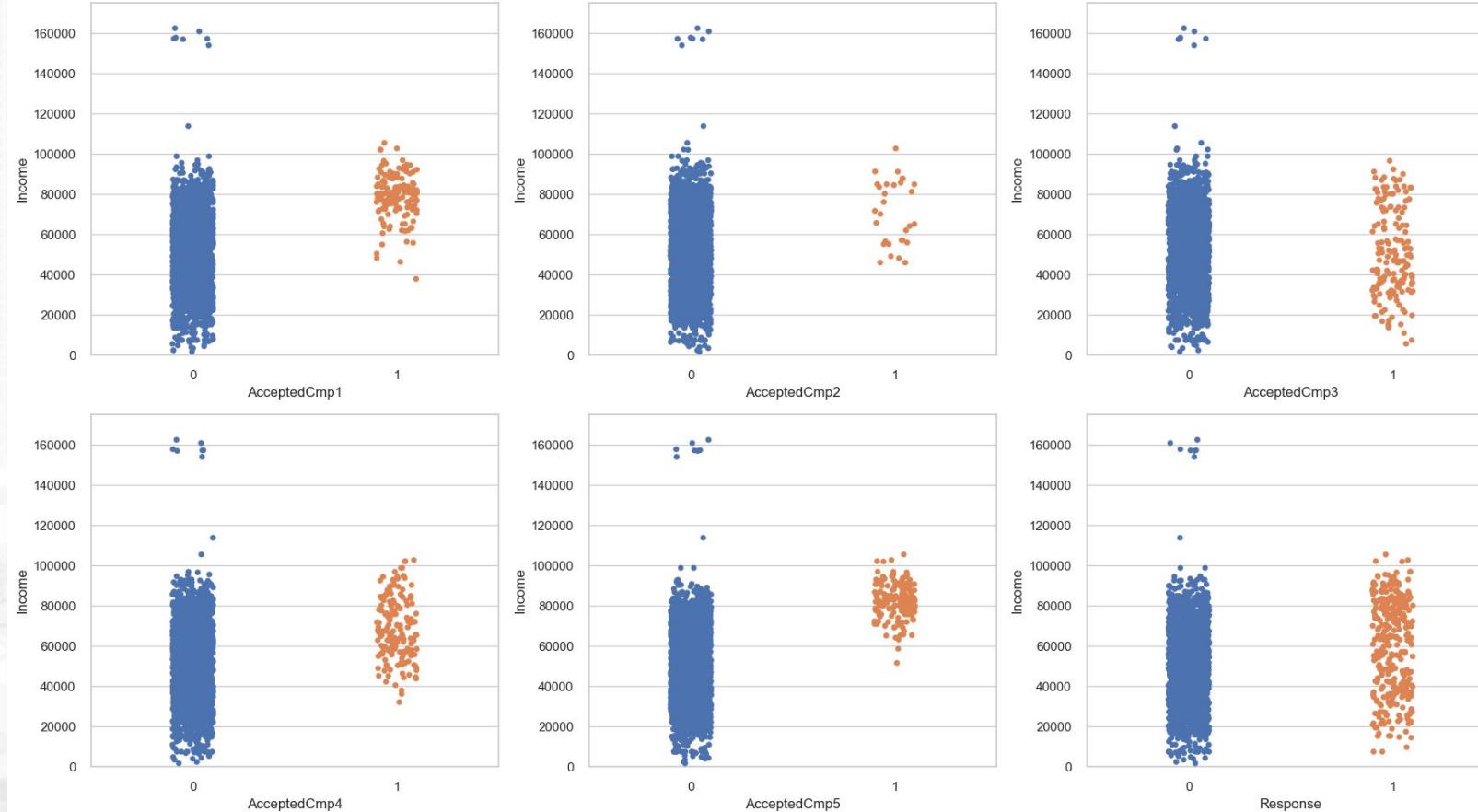
Total Accept Campaign (1-5)



Paling banyak pada **Lima Campaign** kita adalah **0 (tidak pernah merespon)**, namun ada yang sedikit **berpotensi** pada, hanya **sekali (1) atau dua kali (2) merespon** masing-masing **325 dan 83 Customers**

Business Insight

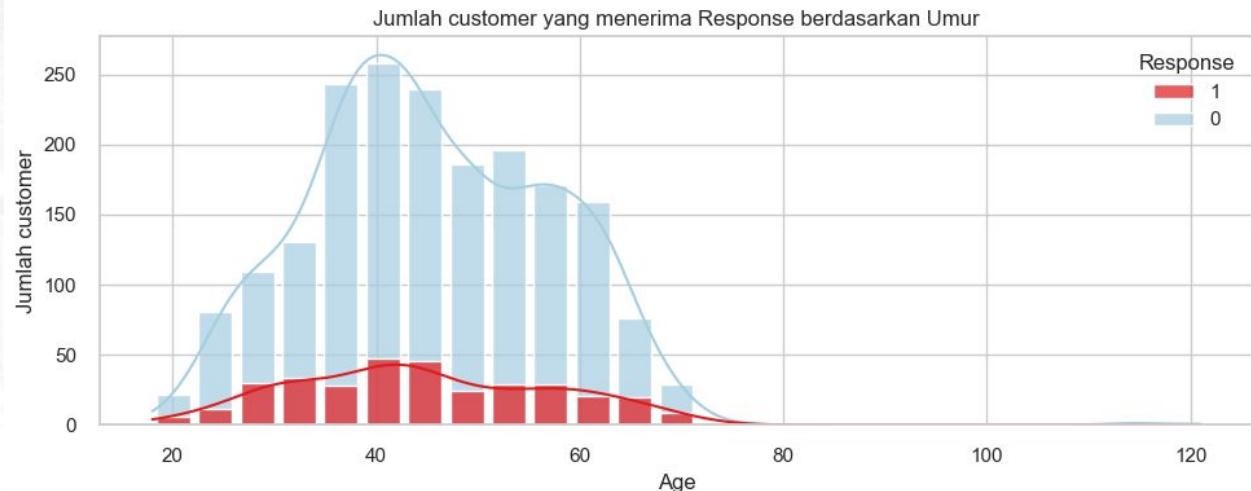
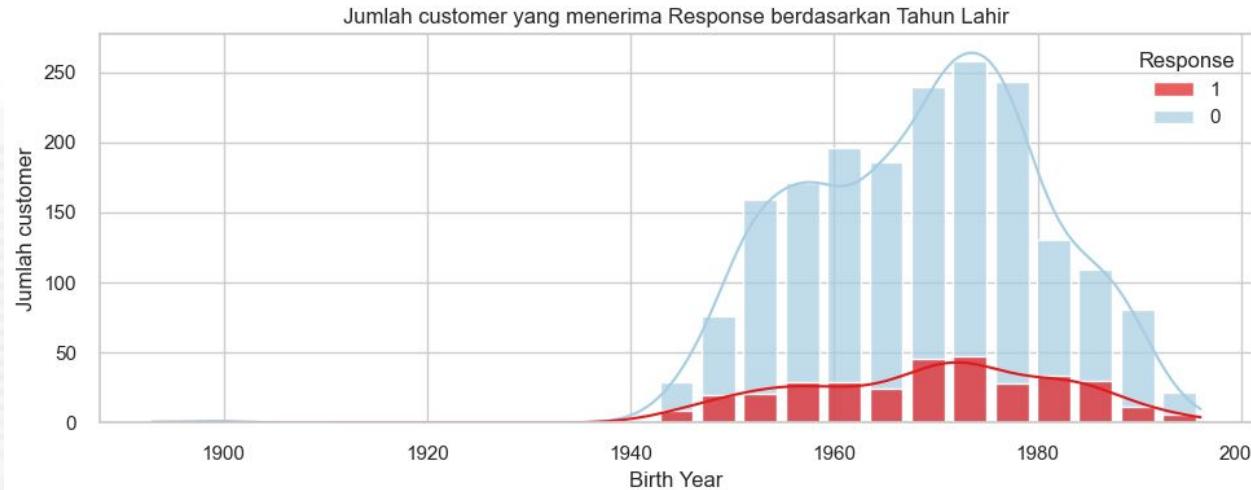
Income vs. Campaign



Customer dengan **income diatas > \$120.000** tidak ada yang menerima/respon campaign perusahaan. Jadi sebaiknya perusahaan **fokus melakukan campaign kepada customer dengan income dibawah < \$120000**.

Business Insight

Birth Year / Age vs. Response



Kategori customer yang menerima Response (marketing campaign terakhir) terbanyak berasal dari **tahun lahir 1970-1975 (39-44 years old)**, dan **1980-1990 (24-34 years old)**.

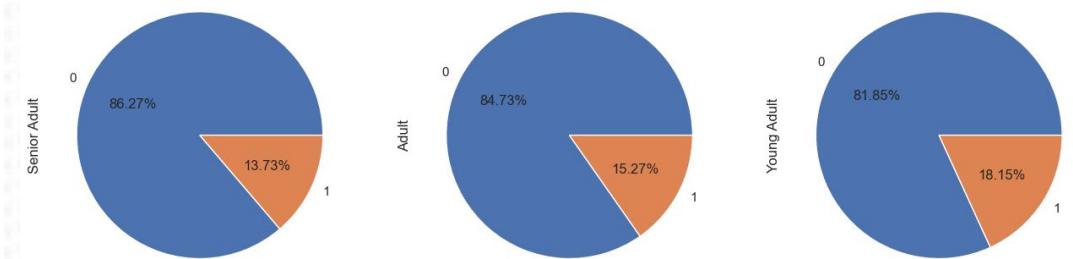
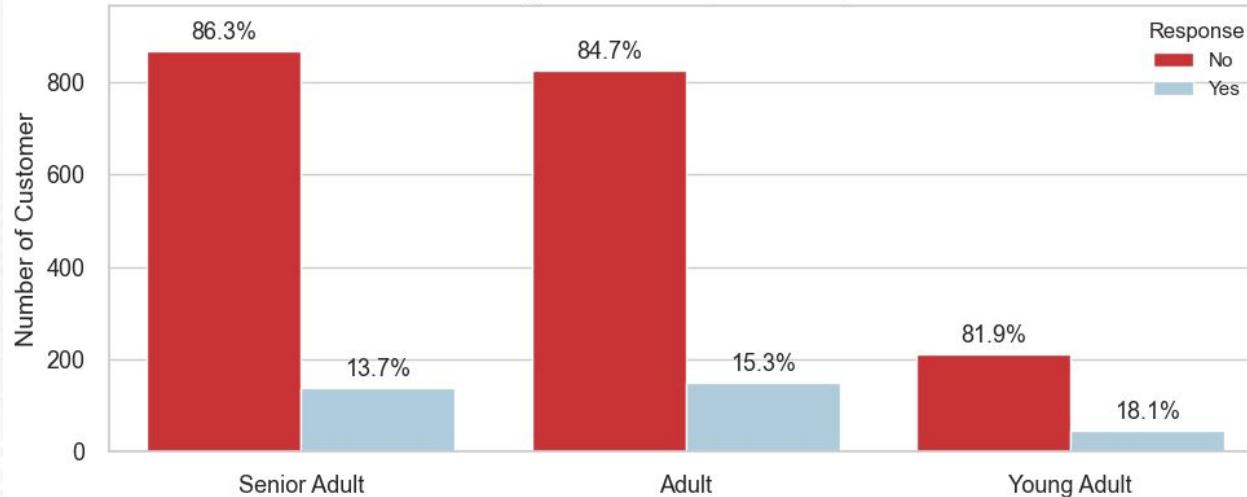
Jika perusahaan harus memprioritaskan beberapa customer saja, maka perusahaan dapat memilih customer yang lahir pada tahun tersebut untuk menawarkan sebuah campaign.

Namun tetap memperhatikan juga **No Response (Tidak Menerima)** karena pada area **tahun lahir 1970-1975 juga sangat tinggi**.

Age Group vs. Response

Comparison of Response Users in Age Group

The older a person is, the higher the response value

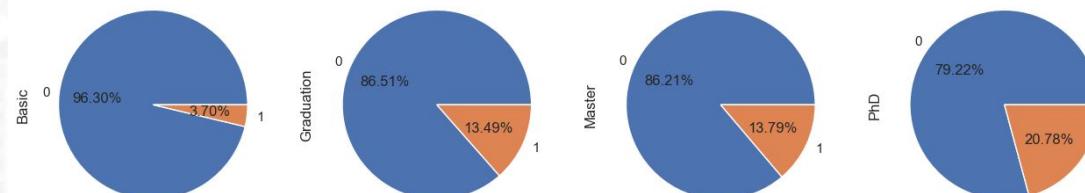
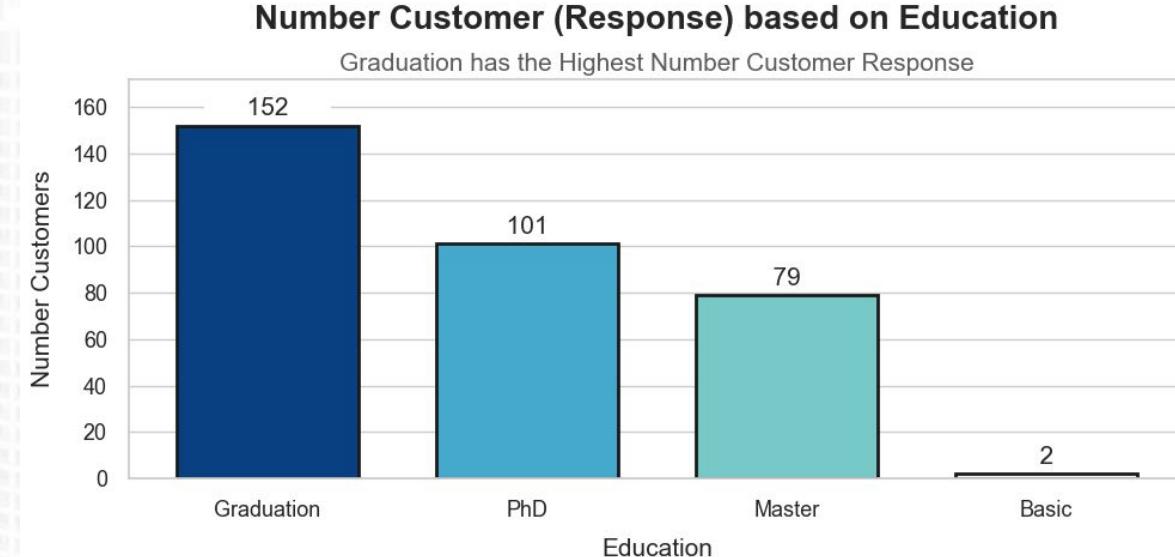
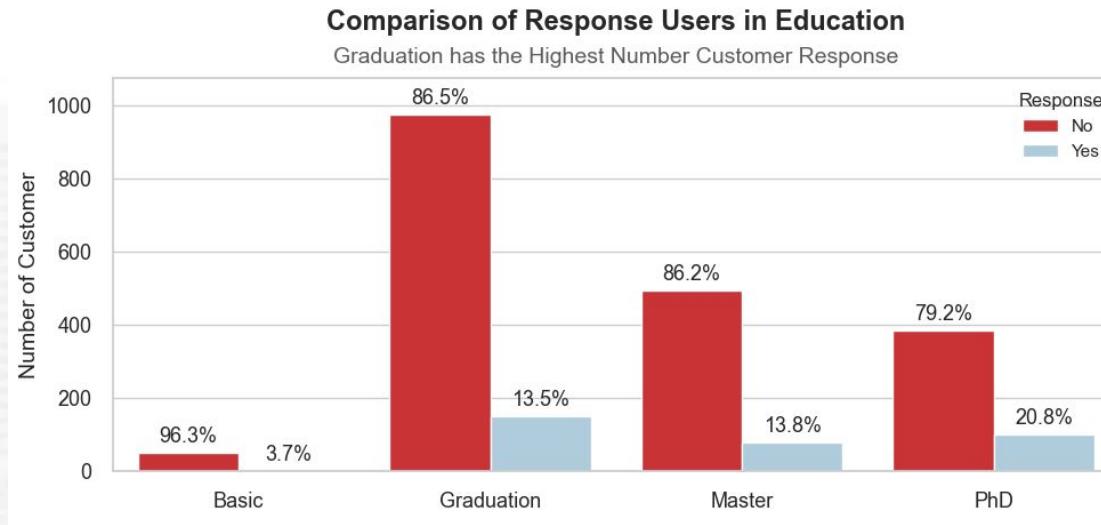


Berdasarkan chart diatas dapat dilihat bahwa kelompok umur **yang paling banyak** merespon campaign adalah **Adult dan Senior Adult** dan **yang paling rendah** adalah **Young Adult**.

Artinya semakin Tua seseorang maka jumlah response juga meningkat

Business Insight

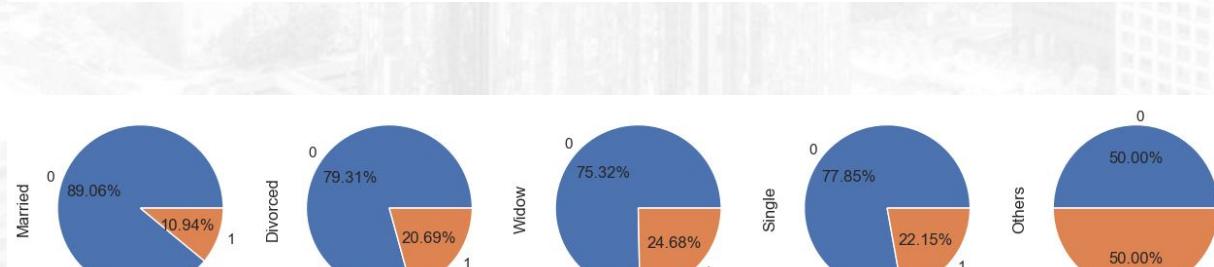
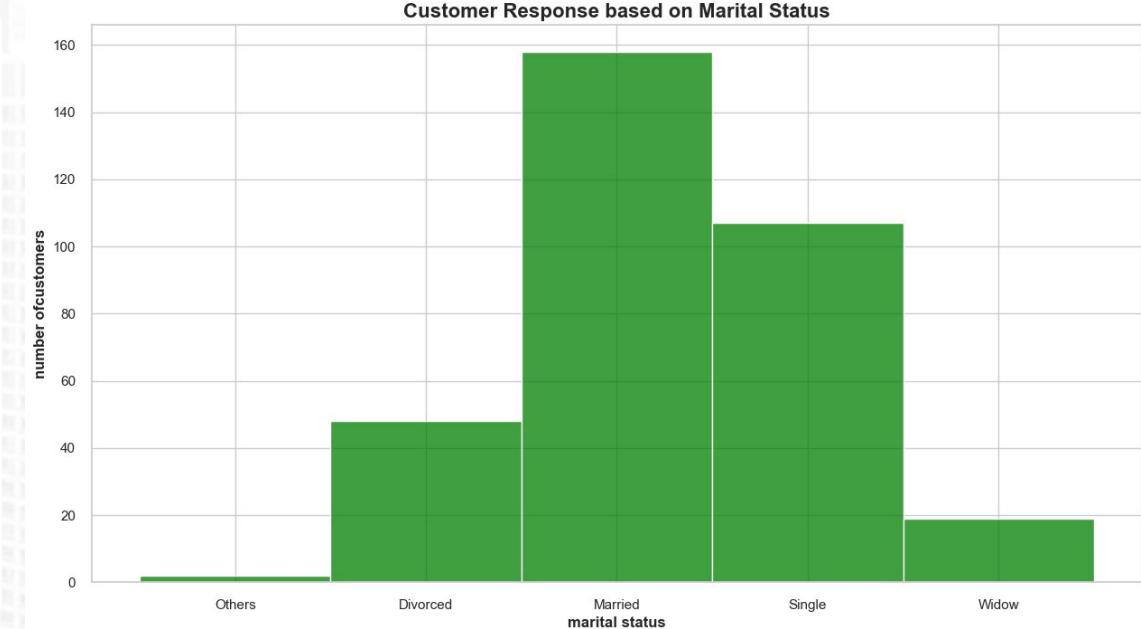
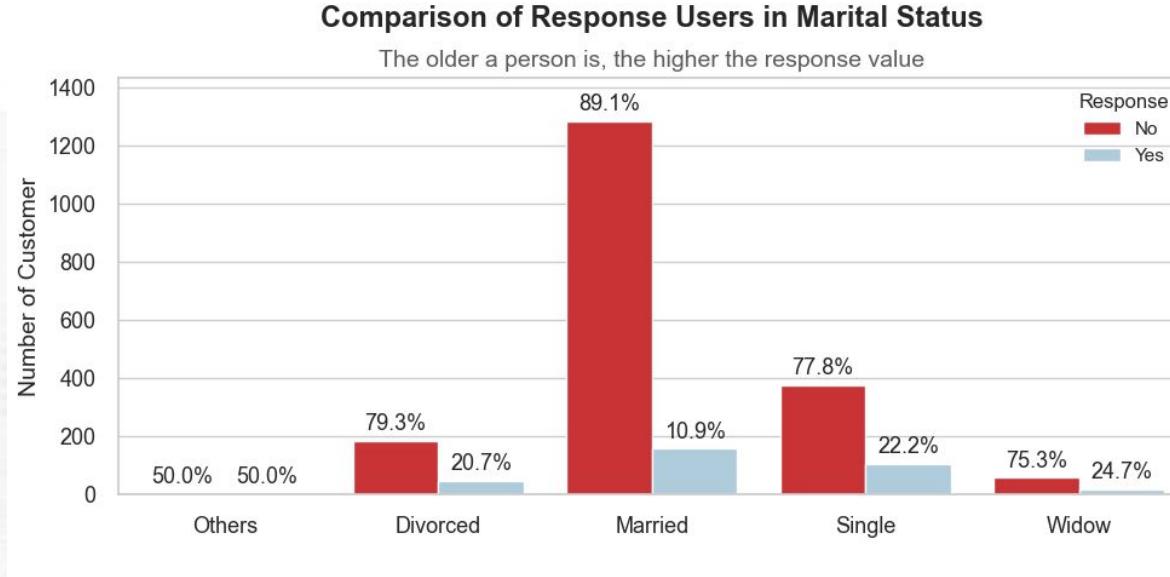
Education vs. Response



Dari visualisasi **Education**, dapat dilihat bahwa customer **yang merespon terbanyak** berasal dari customer yang memiliki edukasi **Graduation dan PhD**, sehingga marketing team dapat memfokuskan campaign ke customer yang beredukasi Graduation.

Business Insight

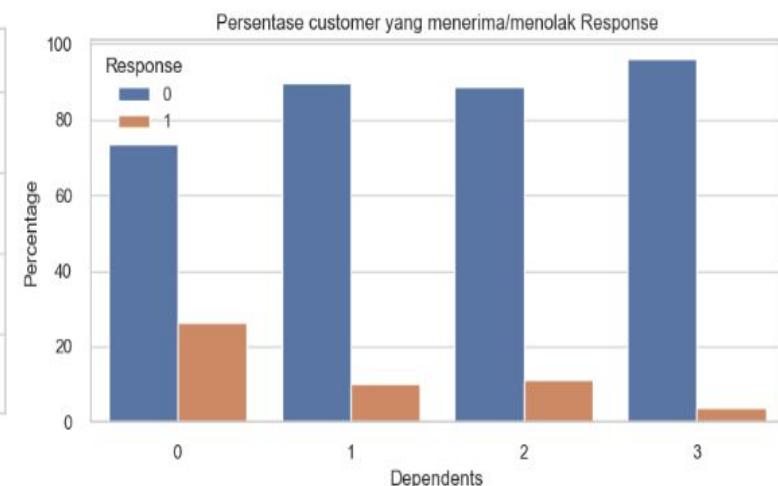
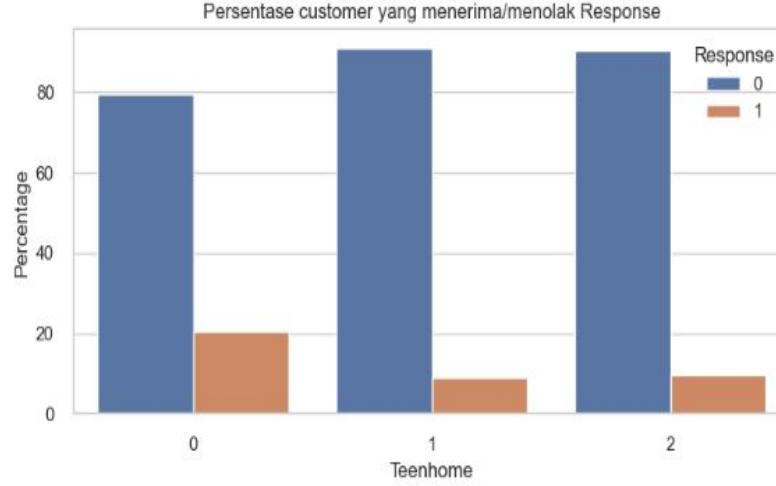
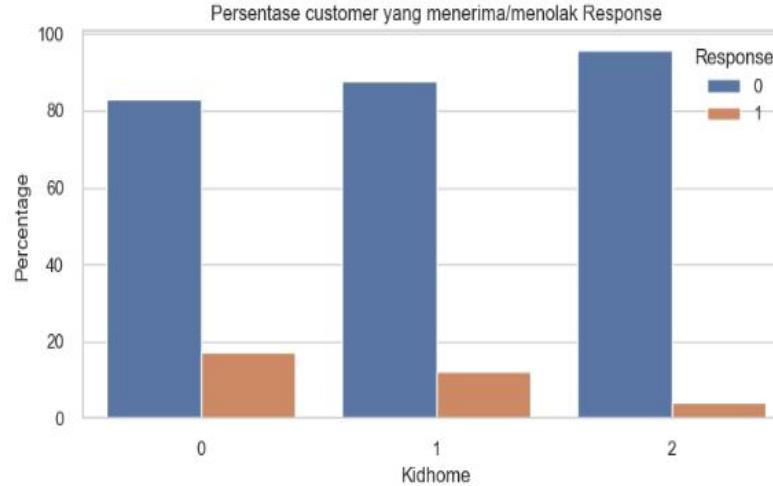
Marital Status vs. Response



Dari visualisasi **Marital Status**, dapat dilihat bahwa customer **yang merespon terbanyak** berasal dari customer dengan status pernikahan yaitu **Married dan Single**, sehingga marketing team dapat memfokuskan campaign ke customer yang telah menikah (Married). **Sedangkan** untuk "**Absurd**" dan "**Yolo**" pada data "**Others**" sangat sedikit

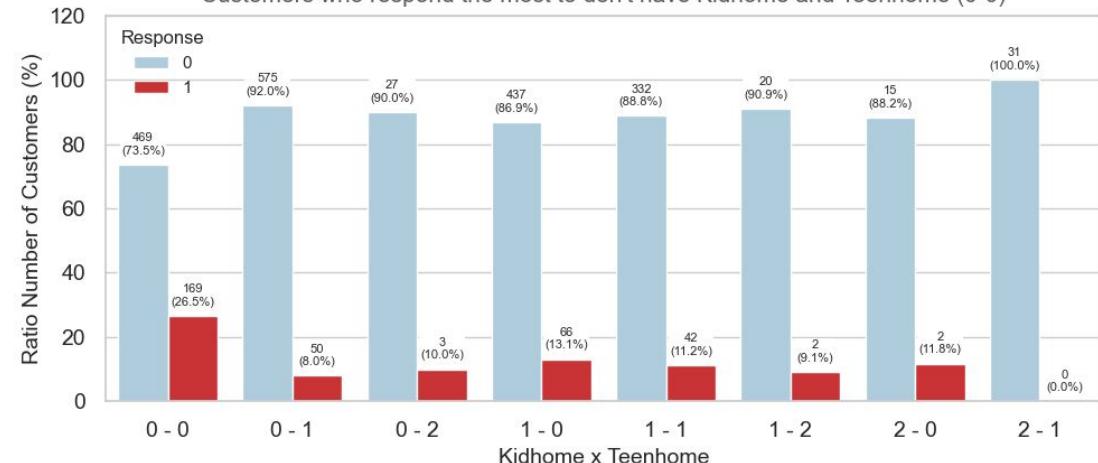
Business Insight

Kids and Teens (Dependents) vs. Response



Comparison of Response Customers in Kidhome and Teenhome

Customers who respond the most to don't have Kidhome and Teenhome (0-0)

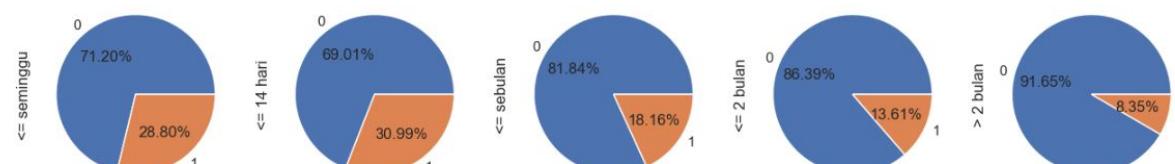
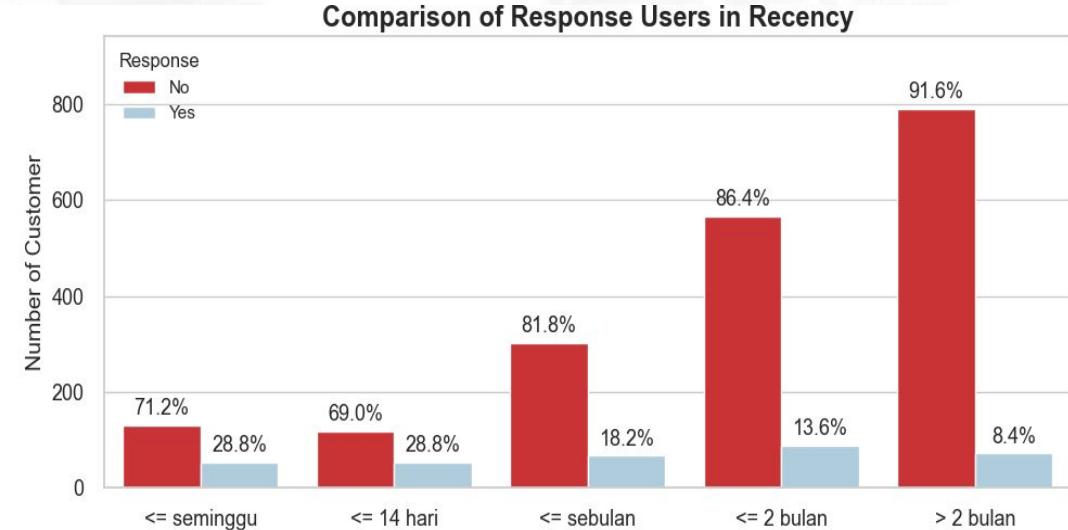
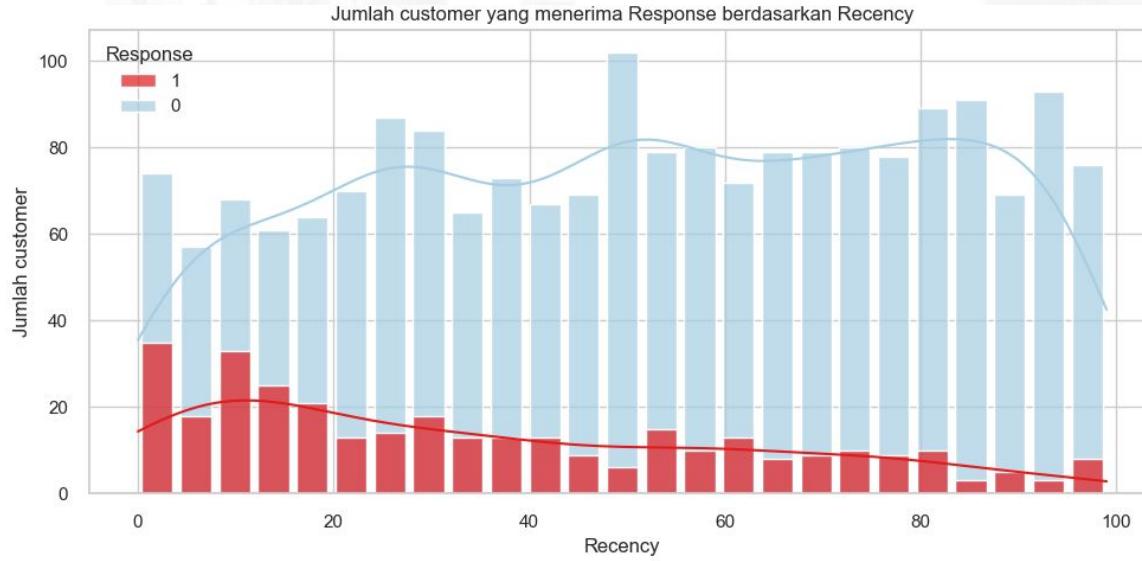


Semakin tinggi jumlah anak/remaja yang dimiliki customer, maka semakin kecil kemungkinan customer menerima Response

Dari visualisasi **Kids and Teens (Dependents)**, dapat dilihat bahwa customer **yang merespon terbanyak** berasal dari customer yang **tidak memiliki anak dan remaja**, sehingga marketing team dapat memfokuskan campaign ke customer yang tidak memiliki anak atau remaja.

Business Insight

Recency vs. Response

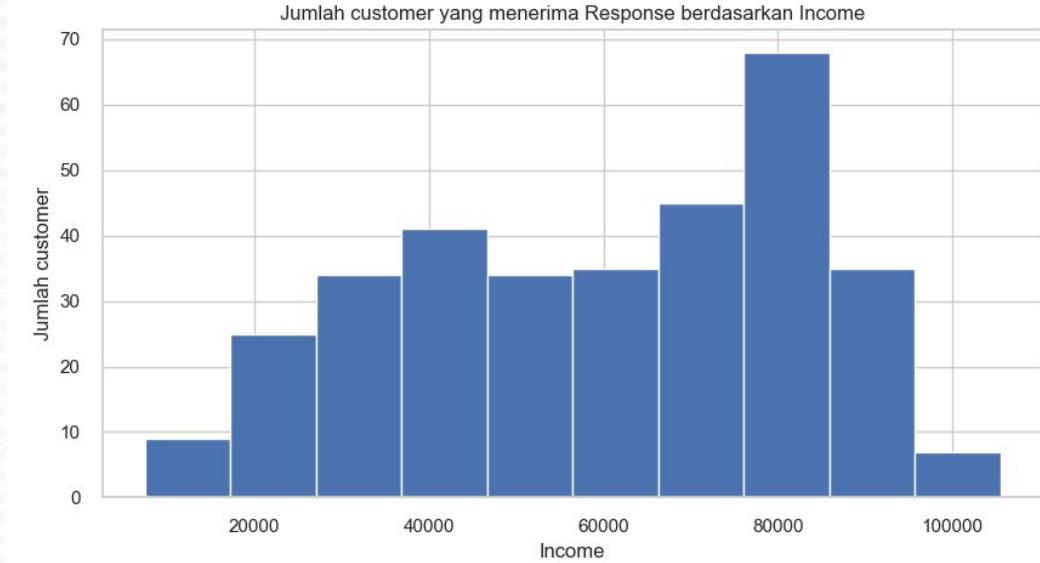
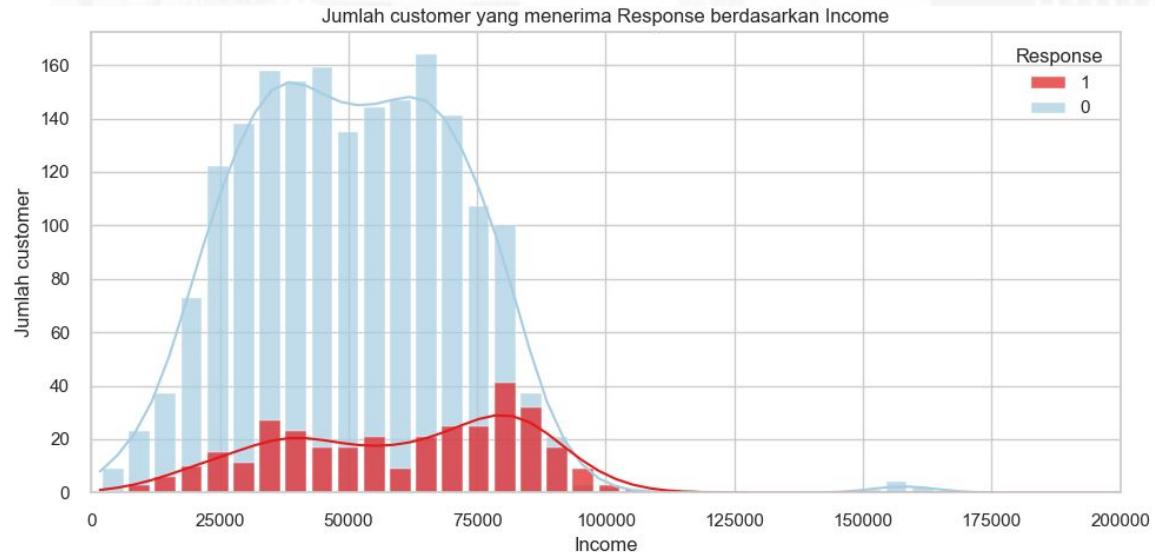


Dari visualisasi **Recency**, dapat dilihat bahwa customer **yang merespon terbanyak** berasal dari customer dengan **Recency yang rendah**, sehingga marketing team dapat memfokuskan campaign ke customer yang memiliki Recency rendah.

* **Recency rendah** : waktu purchase terakhir pelanggan dengan produk belum terlalu lama

Business Insight

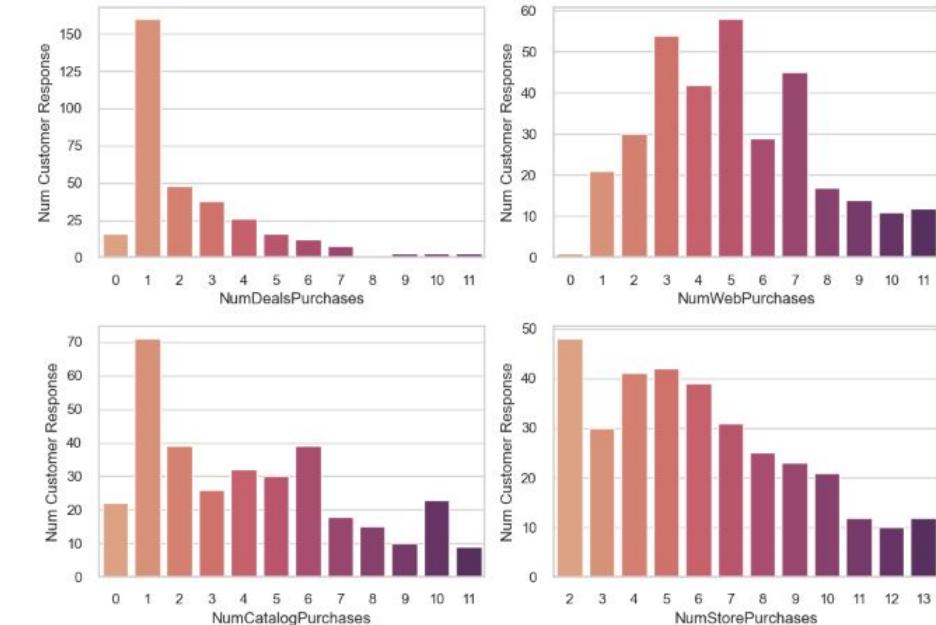
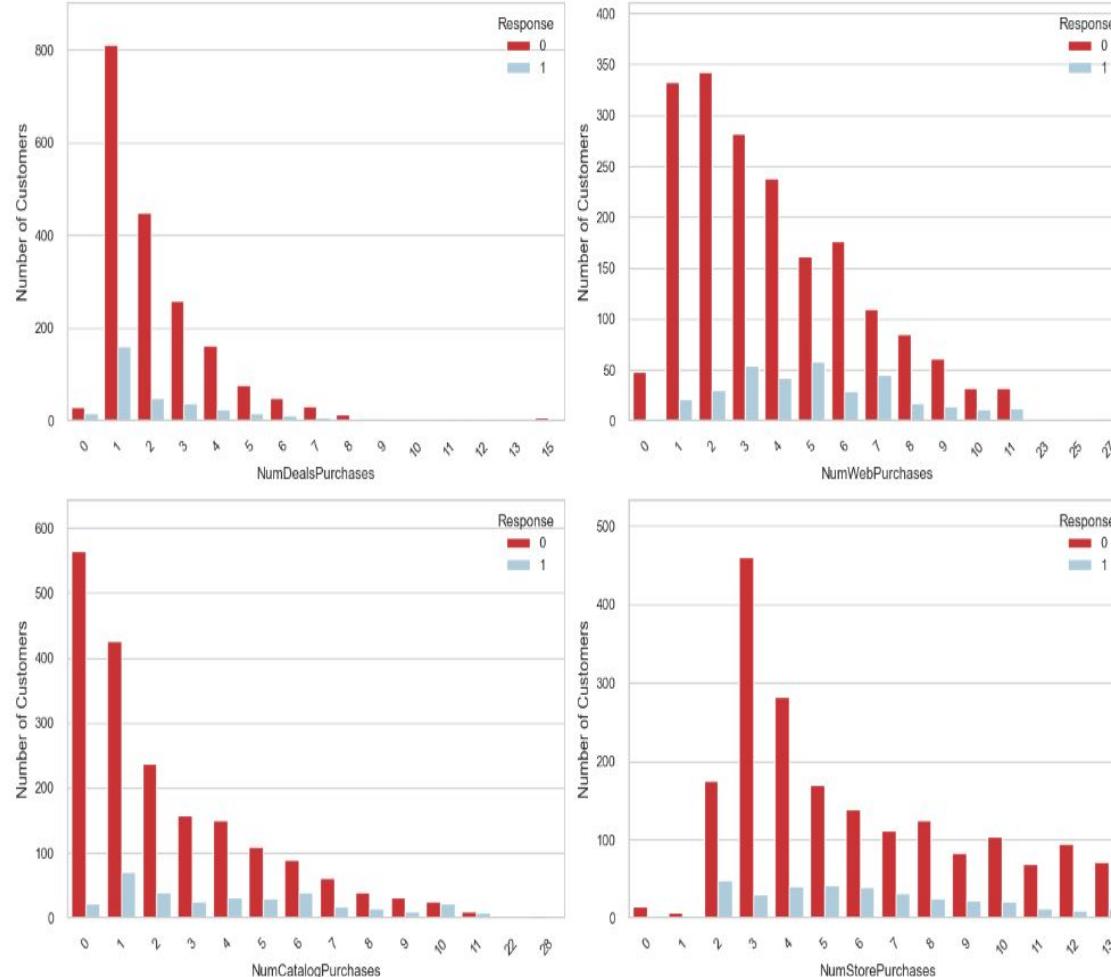
Income vs. Response



Dari visualisasi **Income**, dapat dilihat bahwa customer **yang merespon terbanyak** berasal dari customer dengan pendapatan **> \$75000**, sehingga marketing team dapat memfokuskan campaign ke customer yang memiliki pendapatan diatas 75000.

Business Insight

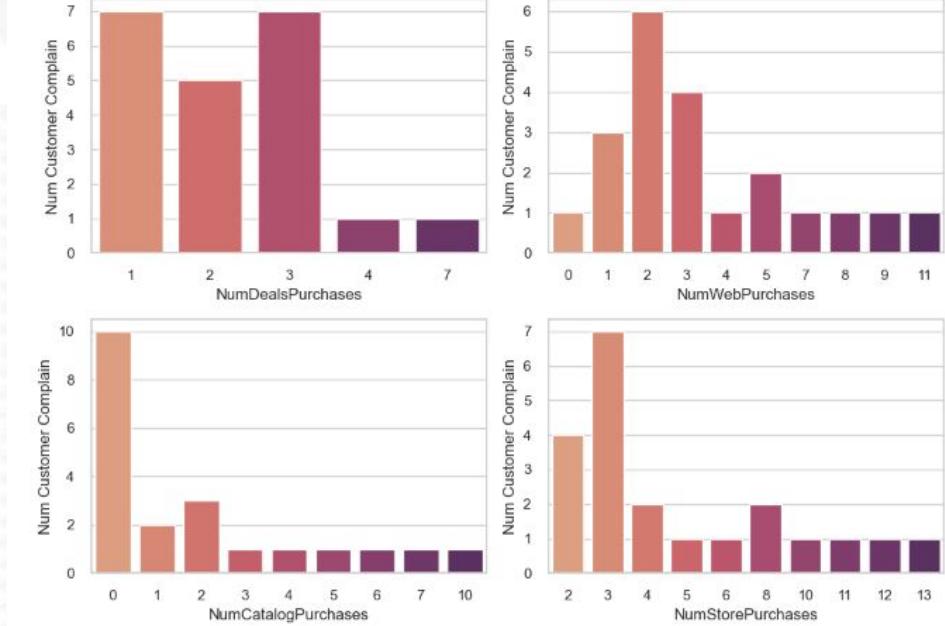
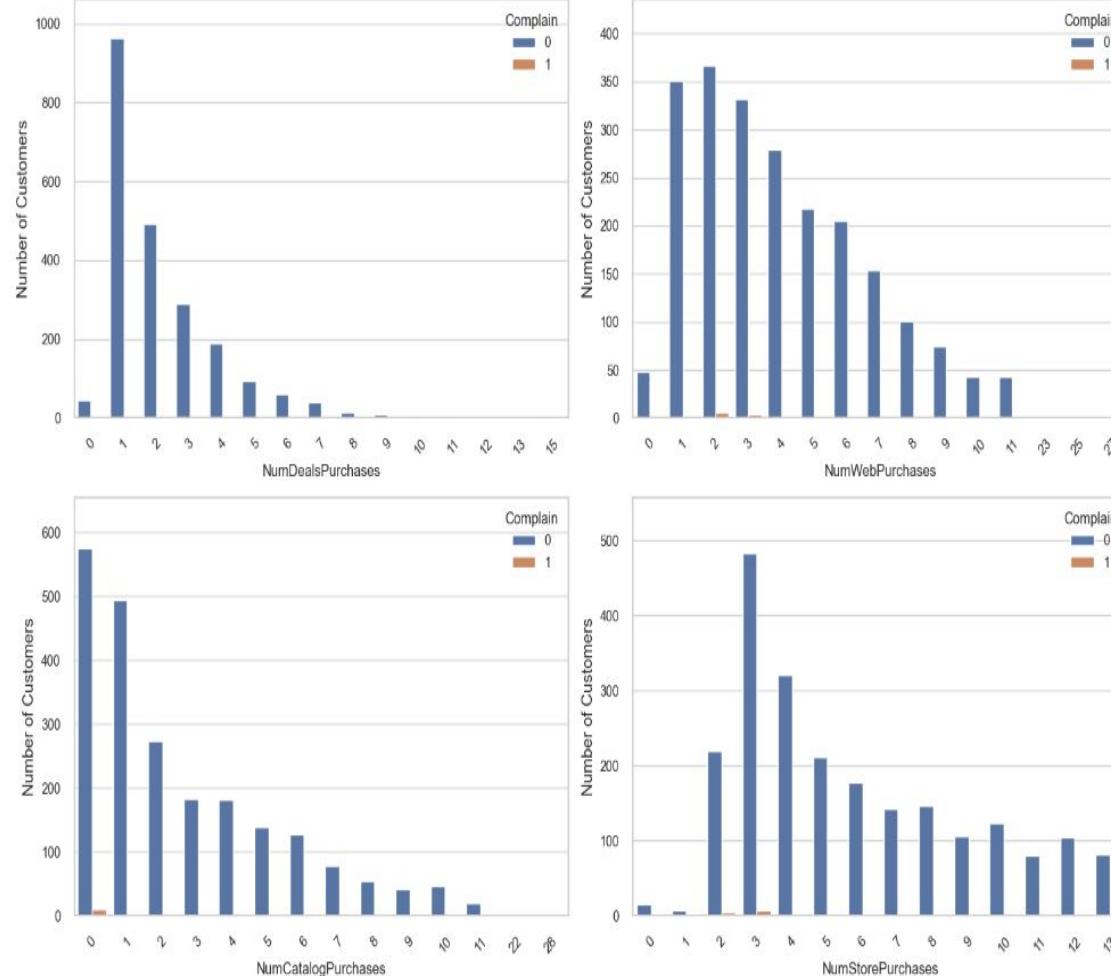
Purchase Type vs. Response



Dari visualisasi **Purchase Type**, dapat dilihat bahwa semakin sedikit pembelian yang dilakukan (baik yang menggunakan diskon ataupun yang melalui web, catalog, store), maka semakin besar kemungkinan customer untuk menerima Response, sehingga marketing team dapat memfokuskan campaign ke customer dengan jumlah pembelian yang masih sedikit.

Business Insight

Purchase Type vs. Complain



Dari visualisasi **Purchase Type**, dapat dilihat bahwa customer **yang complain terbanyak** berasal dari customer dengan Purchase rendah, sehingga perusahaan sebaiknya meningkatkan kualitas pelayanan untuk keseluruhan customer dengan Purchase yang rendah maupun tinggi.

Business Insight

Income x Spending for Response

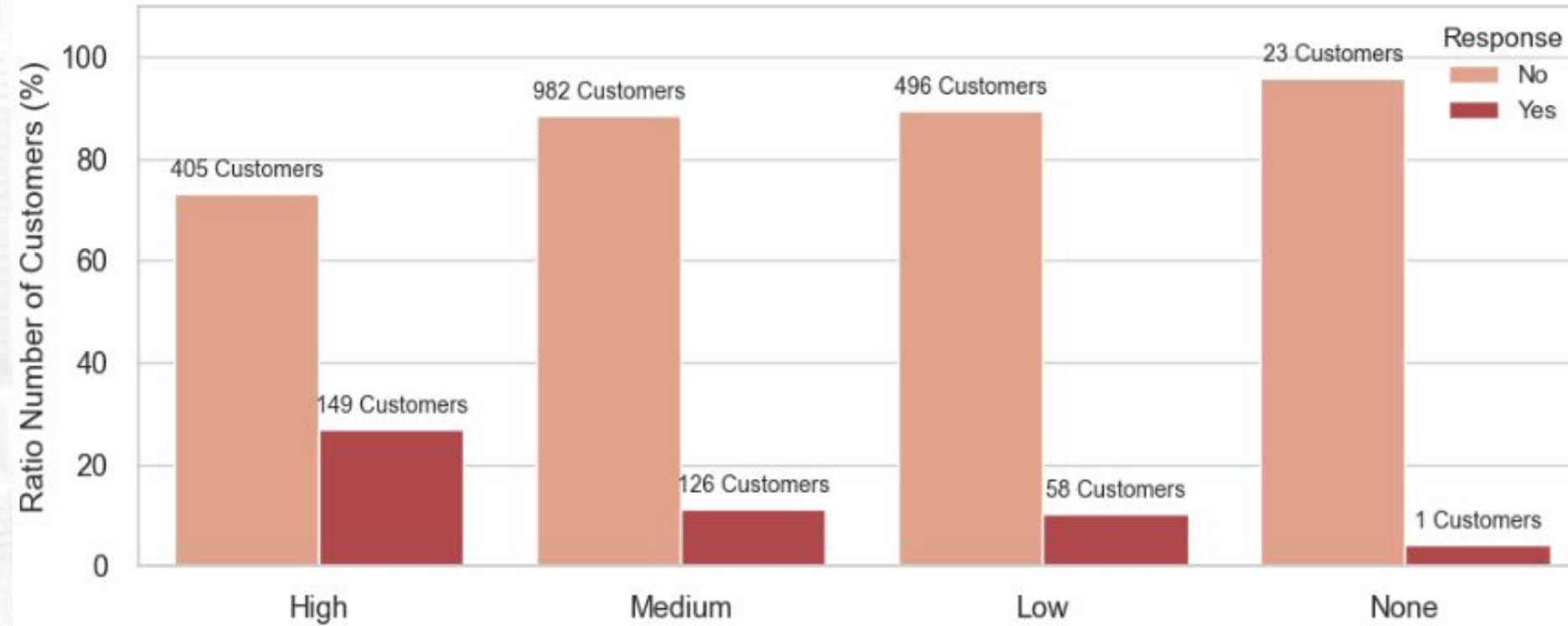


Income dan Spending memiliki korelasi positif pada response, dimana **semakin tinggi nilai income dan spending semakin besar tingkat respon** sehingga fitur income dan spending perlu dipertahankan

Income Segmentation x Response

Comparison of Response Users in Income Segmentation

Higher the income, more likely to give response for campaign

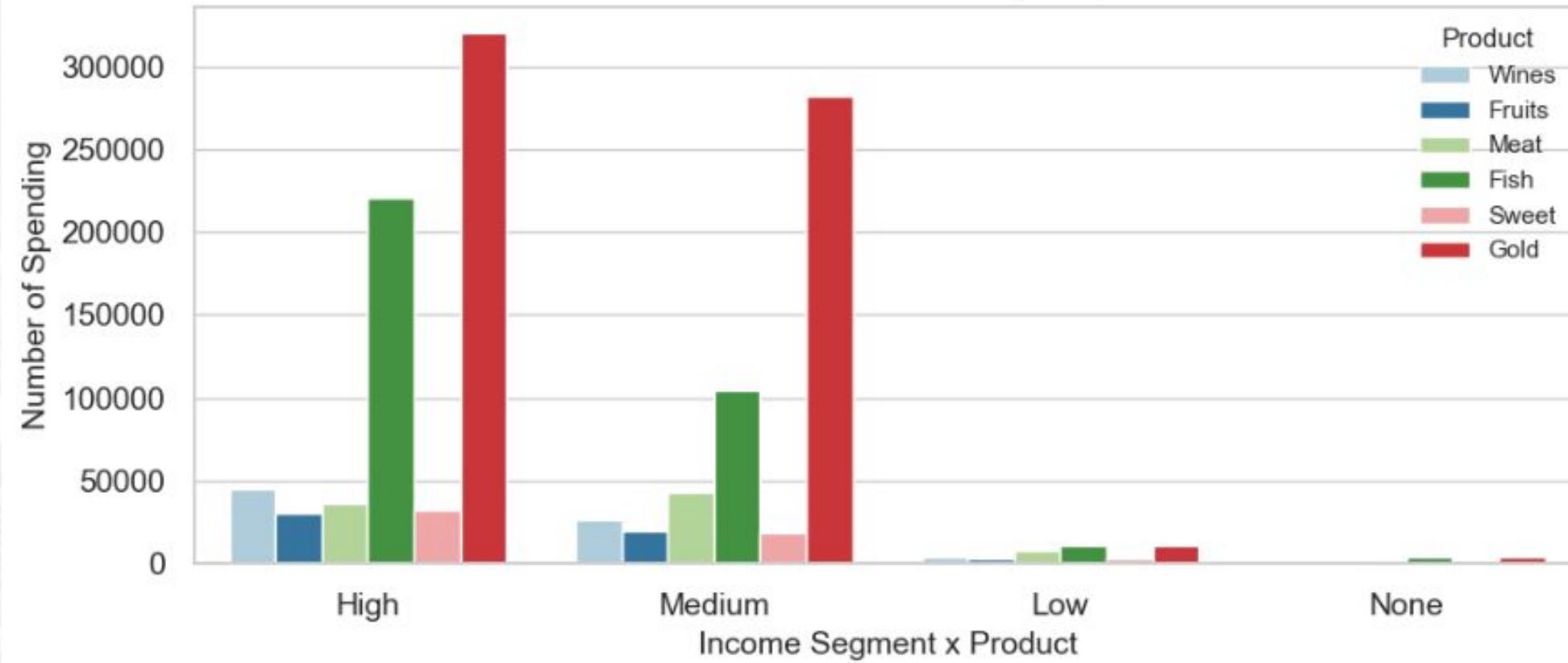


Customer yang merespon campaign cenderung memiliki **income yang lebih tinggi**, terbukti dari customer yang memiliki **income “High” level** merespon campaign lebih banyak.

Income Segmentation x Product Type

Comparison of Income Segment in Product

Customers who have in High on average buy Gold and Fish

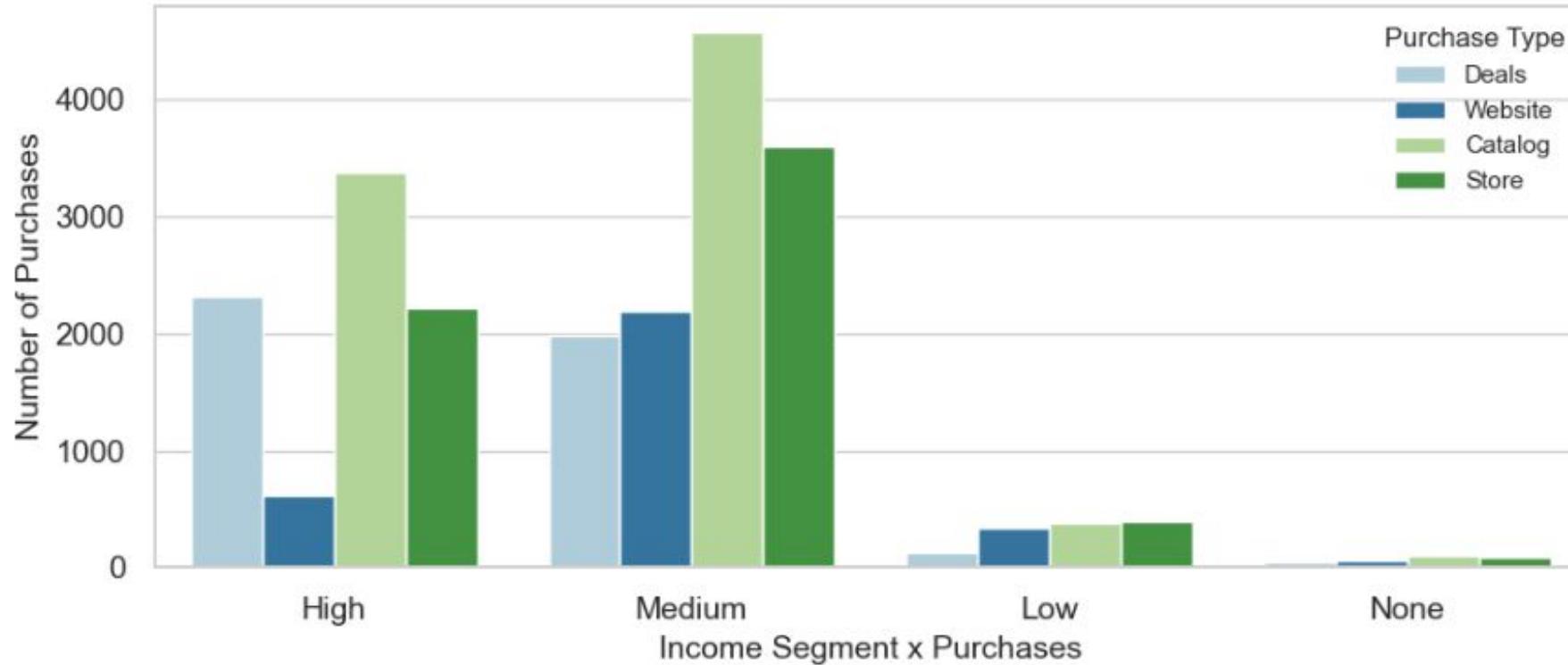


Customer dengan **income High dan Medium** lebih suka dengan produk **Gold dan Fish**. Oleh karena itu, jika ingin membuat campaign disarankan untuk memberikan campaign produk Gold dan Fish untuk customer dengan income tersebut.

Income Segmentation x Purchase Type

Comparison of Income Segment in Purchase Type

Customers who have in High on average use Purchases Discount and Catalog

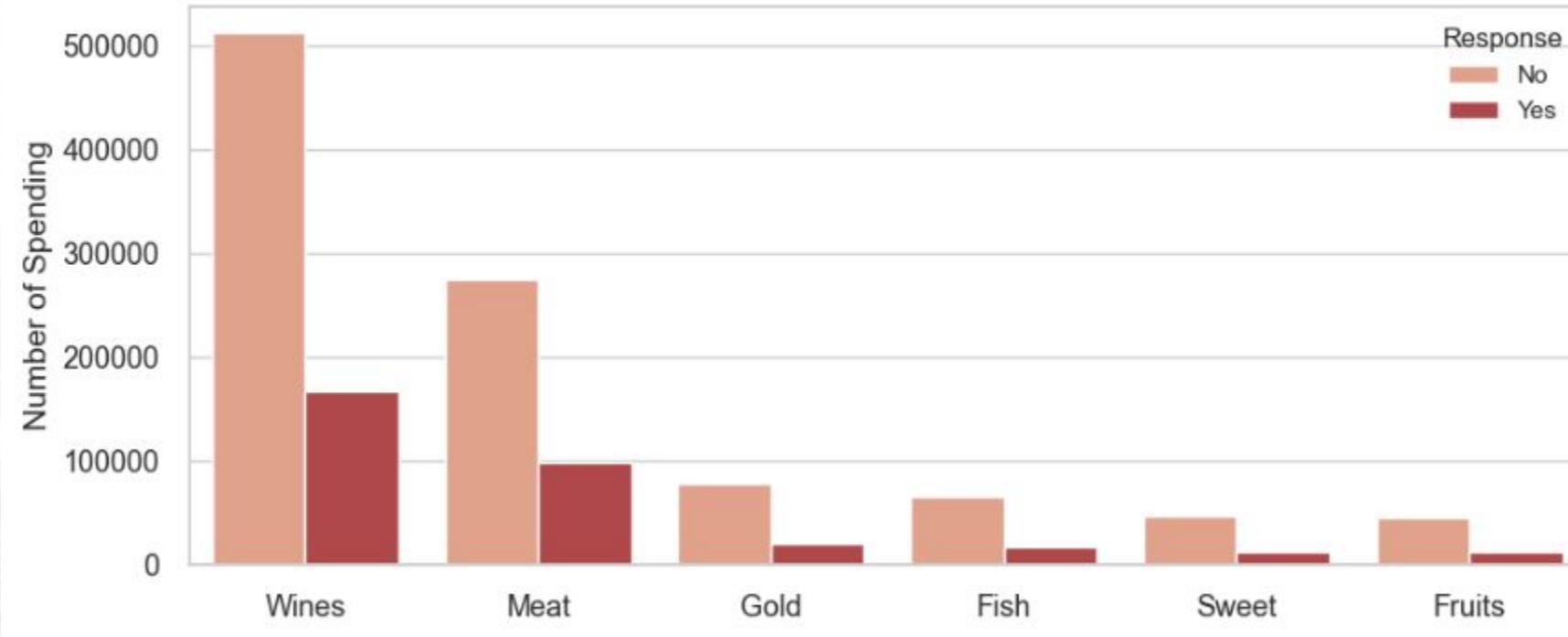


Untuk pembuatan campaign, disarankan berbentuk **Catalog atau Diskon** dan lebih banyak diarahkan ke customer dengan income **High / Medium**. Untuk opsi kedua, bisa langsung dibuat banner/booth pada Store.

Product Type x Response

Comparison of Response Users in Spending Product

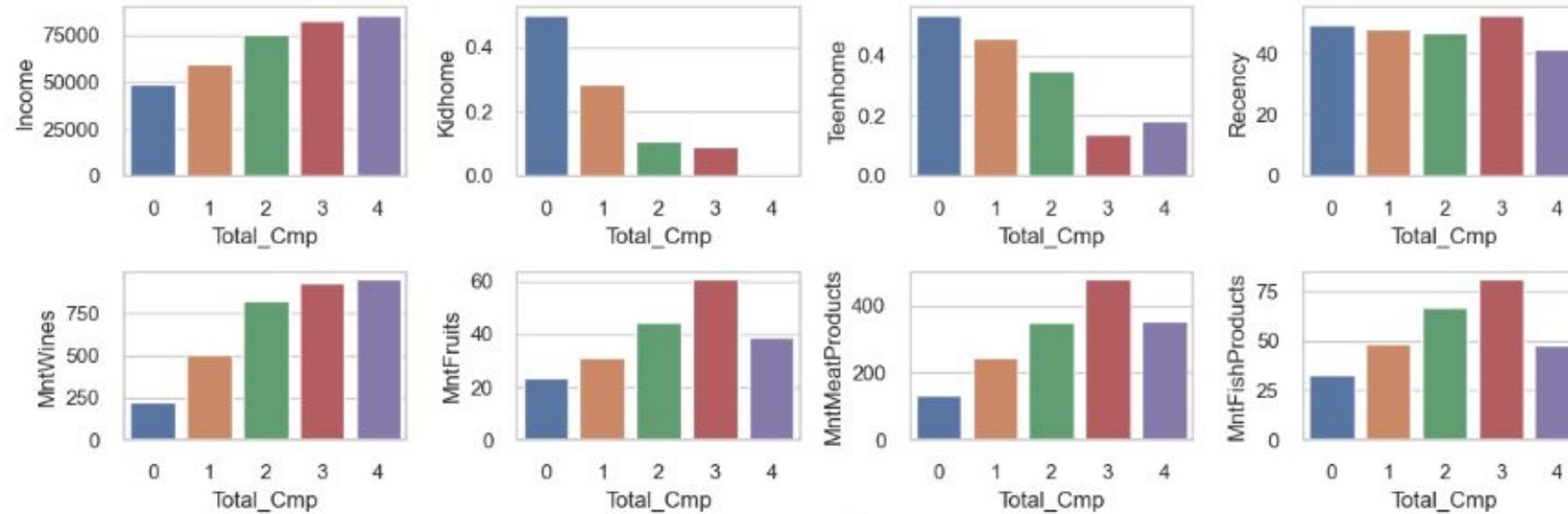
Response customers mostly use Wines and Meat Product



Customer yang merespon campaign cenderung lebih banyak membeli **Wines dan Meat products**. Sehingga untuk campaign selanjutnya produk Wines dan Meat menjadi rekomendasi produk utama untuk customer yang merespon.

Business Insight

Total Campaign vs Variable



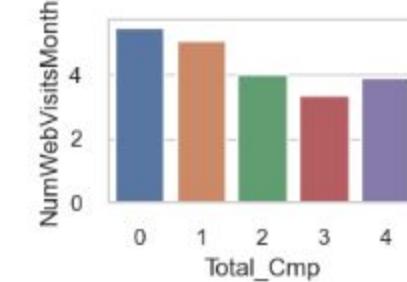
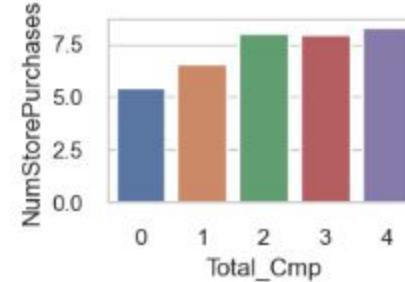
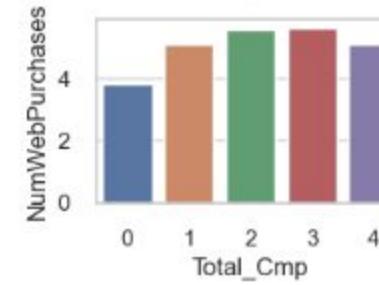
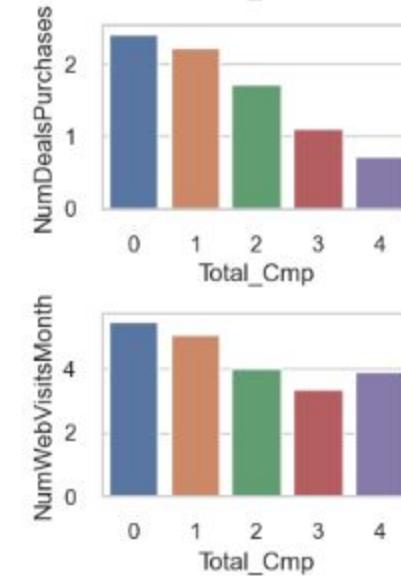
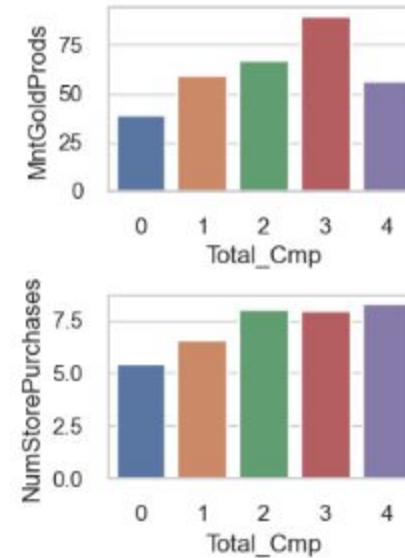
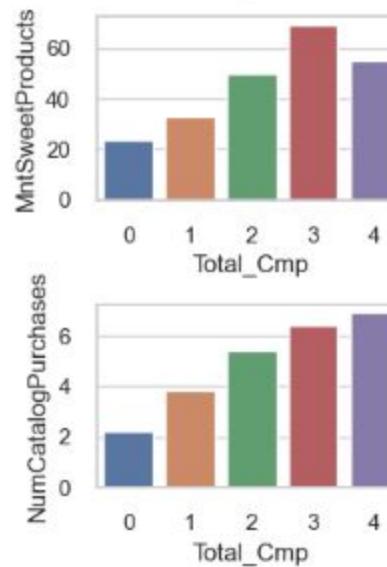
Berikut adalah insights yang diperoleh berdasarkan *Total Campaign vs Variable*:

- Income:**
Customer yang menerima total 4 campaign cenderung memiliki income yang lebih tinggi.
- Kidhome:**
Customer yang tidak menerima campaign sama sekali memiliki kidhome lebih banyak.
- Teenhome:**
Customer yang tidak menerima campaign sama sekali memiliki Teenhome lebih banyak.
- Recency:**
Customer yang menerima total 3 campaign cenderung memiliki recency (total hari terakhir berbelanja) yang lebih lama.

- MntWines:**
Customer yang menerima total 4 campaign cenderung membeli produk Wines lebih banyak.
- MntFruits:**
Customer yang menerima total 3 campaign cenderung membeli produk Fruits lebih banyak.
- MntMeatProducts:**
Customer yang menerima total 3 campaign cenderung membeli produk Meat lebih banyak.
- MntFishProducts:**
Customer yang menerima total 3 campaign cenderung membeli produk Fish lebih banyak.

Business Insight

Total Campaign vs Variable



Berikut adalah insights yang diperoleh berdasarkan *Total Campaign vs Variable*:

- MntSweetProducts:**
Customer yang menerima total 3 campaign cenderung membeli produk Sweet lebih banyak.
- MntGoldProducts:**
Customer yang menerima total 3 campaign cenderung membeli produk Gold lebih banyak.
- NumDealsPurchases:**
Customer yang tidak menerima campaign sama sekali lebih sering membeli produk menggunakan diskon.
- NumWebPurchases:**
Customer yang menerima total 2 dan 3 campaign cenderung lebih sering membeli produk melalui web.

- NumCatalogPurchases:**
Customer yang menerima total 4 campaign cenderung lebih sering membeli produk melalui katalog.
- NumStorePurchases:**
Customer yang menerima total 4 campaign cenderung lebih sering membeli produk melalui toko.
- NumWebVisitsMonths:**
Customer yang tidak menerima campaign sama sekali lebih sering melakukan web visit pada bulan terakhir.

Stage 2

Data Pre-processing



Handling Missing Values

Pada proses handling missing values untuk kolom **Income** ada beberapa metode yang dapat dilakukan diantaranya:

- **Drop Rows Missing Values**

Drop rows pada missing values tidak kita dilakukan karena data yang dimiliki terbatas

- **Imputation Median**

Proses imputasi dilakukan dengan menggunakan **median**, karena distribusi data pada kolom **Income** berbentuk **Highly Positively Skewed** dengan menggunakan fungsi **Fillna** or **SimpleImputer**

- **Multivariate Approach**

- Melakukan **transform** beberapa **kolom object/string**, karena penggunaan **multivariate approach** memerlukan semua **kolom numeric (Feature Encoding/Drop Character)**
 - **Metode yang digunakan** diantaranya :

- a. **KNNImputer** or **K-Nearest Neighbor**
- b. **MICE** or **Multiple Imputation by Chained Equation**

Menggunakan **IterativeImputer** or **LightGBM**

- **Choice Determination:**

Pada proses handling missing values ini kita menggunakan **Imputation using MICE with LightGBM**

Imputation using MICE with LightGBM

```
import miceforest as mf

# Create kernel.
kds = mf.ImputationKernel(
    df_ma,
    save_all_iterations=True,
    random_state=100
)

# Run the MICE algorithm
kds.mice(iterations=5, n_estimators=50)

# Return the completed dataset.
df_imputed = kds.complete_data()
df["Income"] = df_imputed["Income"].copy()
df.head()
```

Kesimpulan

Pada proses handling missing values terdapat **missing values** sebanyak **24 rows (1,07%)** pada kolom **Income**. Dikarenakan keterbatasan data yang ada, pada prosesnya kita **tidak melakukan penghapusan baris (Drop Rows)**, melainkan **dilakukan proses Imputation**. Proses handling missing values yang dilakukan menggunakan **Imputation using MICE with LightGBM**. Imputasi **MICE** dapat **lebih efisien** ketika menggunakan **miceforest** karena diharapkan **kinerjanya jauh lebih baik** saat **mengimplementasikan algortima lightgbm di backend** untuk melakukan imputasi. **LightGBM** dikenal dengan akurasi prediksi yang tinggi ketika digabungkan dengan algortima MICE sehingga menjadikannya algortima yang kuat untuk imputasi.

Handling Duplicate Rows

```
df.duplicated().sum()  
0  
  
print(f"Data Frame Dimension Before Duplicate Removal: {df.shape}")  
df = df.drop_duplicates().reset_index(drop=True)  
print(f"Data Frame Dimension After Duplicate Removal: {df.shape}")  
  
Data Frame Dimension Before Duplicate Removal: (2240, 29)  
Data Frame Dimension After Duplicate Removal: (2240, 29)  
  
df.duplicated(subset=["ID"]).sum()  
0
```

Kesimpulan

- Berdasarkan hasil pengecekan pada dataset tidak ditemukan baris data yang memiliki duplikat, sehingga kita tidak perlu melakukan handling duplicated data
- Pada pengecekan duplikat subset untuk ID tidak ditemukan ada nya ID customer yang sama

Handling Invalid Values

Beberapa hal yang dilakukan pada proses Handling Invalid Values diantaranya:

- Melakukan konversi data Date

```
df["Dt_Customer"] = pd.to_datetime(df["Dt_Customer"])
```

Untuk mempermudah dalam **proses feature extraction/engineering** maka untuk data yang mengandung datetime akan dilakukan **konversi ke format datetime pandas**

- Melakukan penyederhanaan Marital_Status

```
# Mengganti kategori 'Widow', 'Alone', 'Absurd', 'YOLO'  
df['Marital_Status'] = df['Marital_Status'].replace(['Widow', 'Alone', 'Absurd', 'YOLO'], 'Single')  
# Mengganti kategori 'Together' menjadi 'Married'  
df['Marital_Status'] = df['Marital_Status'].replace(['Together'], 'Married')
```

```
df['Marital_Status'].unique()
```

```
array(['Single', 'Married', 'Divorced'], dtype=object)
```

Perlu dilakukan replace data / menyatukan yang memiliki arti yang sama agar mengurangi jumlah dimensi maupun redundansi pada data yang dirincikan sebagai berikut:

- Mengganti kategori **Widow**, **Alone**, **Absurd**, **YOLO** menjadi **Single**
- Mengganti kategori **Together** menjadi **Married**
- Mempertahankan kategori **Divorced**

Handling Invalid Values

Beberapa hal yang dilakukan pada proses Handling Invalid Values diantaranya:

- Melakukan penyederhanaan Education

```
# Levels : Basic - Graduation - 2n Cycle - Master - PhD
df['Education'] = df['Education'].replace(['2n Cycle'], 'Master')
```

Pada kategori **2n Cycle** dan **Master** akan dilakukan replace data / menyatukan yang memiliki arti yang sama juga, sehingga untuk baris yang memiliki kategori **2n Cycle** akan dihapus dan digantikan dengan kategori **Master**

Marital_status dan Education setelah dilakukan penyederhanaan

	ID	Year_Birth	Education	Marital_Status
0	5524	1957	Graduation	Single
1	2174	1954	Graduation	Single
2	4141	1965	Graduation	Married
3	6182	1984	Graduation	Married
4	5324	1981	PhD	Married

Handling Outliers

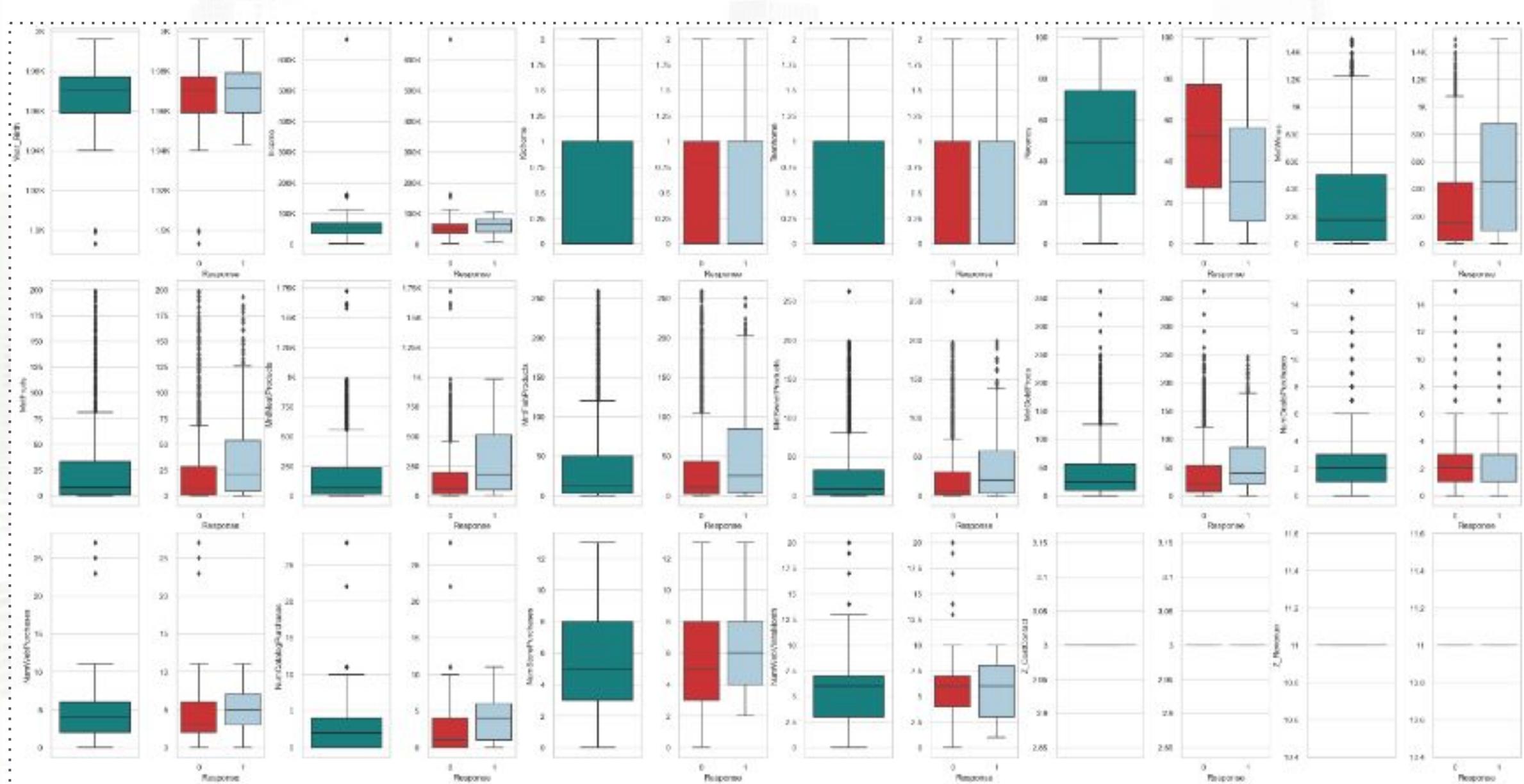
Jumlah baris: 2240

Outlier All Data : 706

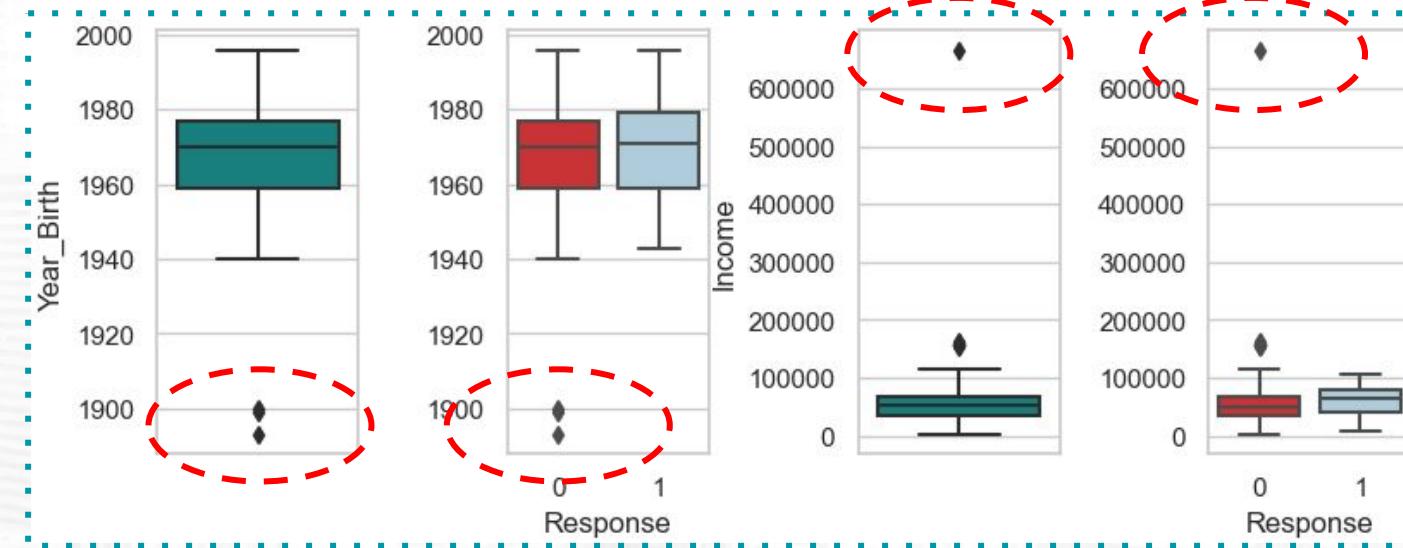
Not Outlier All Data : 1534

	Column Name	is Outlier	Lower Limit	Upper Limit	Outlier	No Outlier
0	Year_Birth	True	1932.0	2004.0	3	2237
1	Income	True	-14290.5	118153.5	8	2232
2	Kidhome	False	-1.5	2.5	0	2240
3	Teenhome	False	-1.5	2.5	0	2240
4	Recency	False	-51.0	149.0	0	2240
5	MntWines	True	-697.0	1225.0	35	2205
6	MntFruits	True	-47.0	81.0	227	2013
7	MntMeatProducts	True	-308.0	556.0	175	2065
8	MntFishProducts	True	-67.5	120.5	223	2017
9	MntSweetProducts	True	-47.0	81.0	248	1992
10	MntGoldProds	True	-61.5	126.5	207	2033
11	NumDealsPurchases	True	-2.0	6.0	86	2154
12	NumWebPurchases	True	-4.0	12.0	4	2236
13	NumCatalogPurchases	True	-6.0	10.0	23	2217
14	NumStorePurchases	False	-4.5	15.5	0	2240
15	NumWebVisitsMonth	True	-3.0	13.0	8	2232
16	Z_CostContact	False	3.0	3.0	0	2240
17	Z_Revenue	False	11.0	11.0	0	2240

Handling Outliers



Handling Outliers



Karena pada kolom **Year_Birth** memiliki nilai min yang sangat jauh di tahun **1893-1900** dan **Income** memiliki nilai max yang sangat tinggi sebesar **\$666.666** , maka akan dilakukan penghapusan rows pada nilai ini agar tidak ada ketimpangan nilai. Ada beberapa metode yang dapat kita lakukan :

- **Handling Oulier**
 - IQR (Interquartile Range)
 - Z-Score
- **Manually Trimmed**

Choice Determination:

- Untuk kasus saat ini, akan digunakan metode **Manually Trimmed**, agar menghindari penghapusan data yang terlalu banyak jika menggunakan Handling Outlier
- Adapun pada kolom lainnya selain **Year_Birth** dan **Income** yang terdapat outlier tidak kita handle karena akan melalui proses **Normal Distribution Transformation** nantinya yang akan **mereduksi** outliernya.

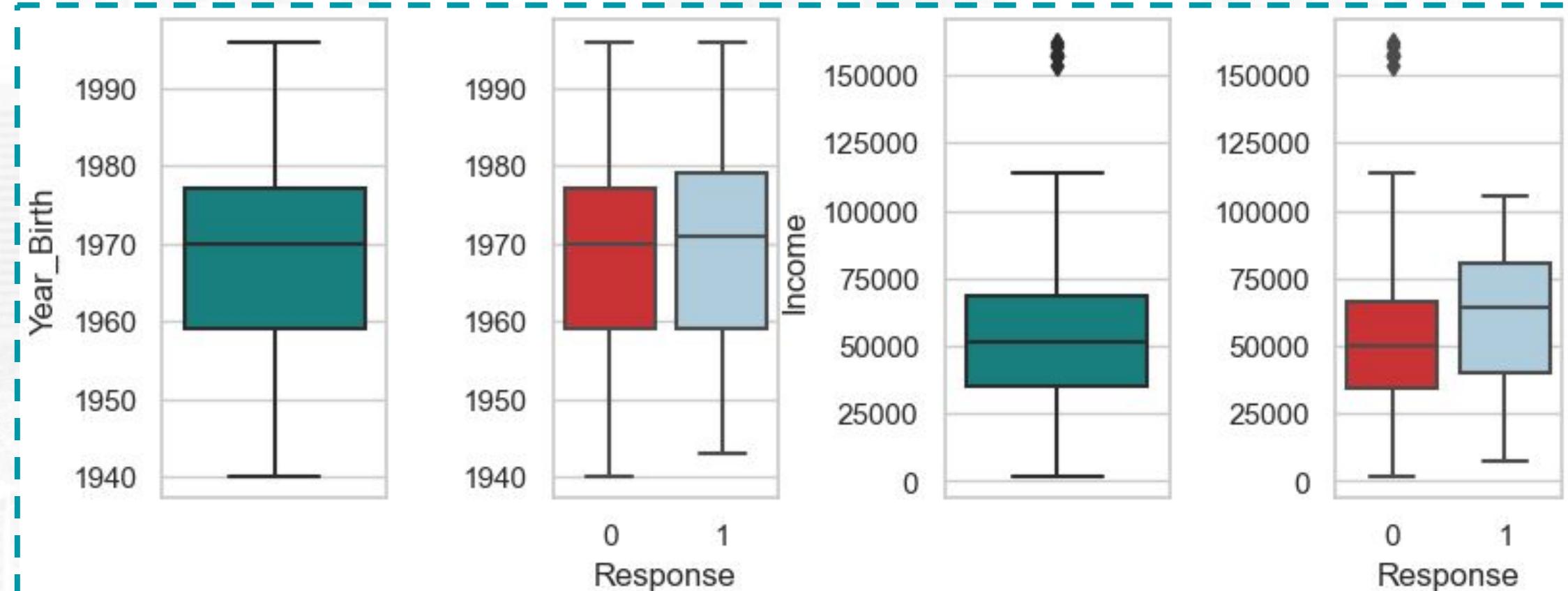
Handling Outliers

	Year_Birth	Income
count	2236.000000	2236.000000
mean	1968.898032	51980.909213
std	11.703281	21518.486256
min	1940.000000	1730.000000
25%	1959.000000	35335.500000
50%	1970.000000	51371.000000
75%	1977.000000	68468.250000
max	1996.000000	162397.000000

Manually Trimmed

- Kolom Year_Birth, menghapus nilai yang sangat jauh di tahun 1893-1900
- Kolom Income menghapus nilai yang sangat tinggi sebesar \$666.666

Mengecek hasil Trimmed / Drop Outliers



Handling Outliers

Berdasarkan hasil perhitungan menggunakan Z-score dan juga IQR, dapat diketahui bahwa jumlah baris yang dihapus dari Year_Birth dan Income berdasarkan IQR untuk kolom tidak jauh berbeda dibandingkan dengan Z-score, yaitu :

- **IQR :**

- Jumlah data sebelum handling outliers : 2240
- Jumlah data setelah handling outliers (Year_Birth) : 2237
- Jumlah data setelah handling outliers (Income) : 2229

- **Z_Score :**

- Jumlah data sebelum handling outliers : 2240
- Jumlah data setelah handling outliers (Year_Birth) : 2237
- Jumlah data setelah handling outliers (Income) : 2229

Namun, karena kita ingin meminimalisasi penghapusan data maka untuk proses ini kita memiliki Manually Trimmed agar tidak terlalu banyak data yang dihapus, jadi hanya berfokus pada data yang memiliki jauh yang sangat tinggi

- ❖ Jumlah data sebelum handling outliers : 2240
- ❖ Jumlah data setelah handling outliers (Year_Birth) : 2237
- ❖ Jumlah data setelah handling outliers (Income) : 2236

Feature Engineering / Extraction

Kita akan melakukan Calculation, Extraction, dan Binning features :

1. Kolom **Umur / Age Customer**

Berdasarkan data diketahui basis tahunnya : **SAS Institute, 2014**. Jadi umur customer didapatkan dari tahun 2014 dikurangi dengan tahun kelahirannya.

```
# currentYear = datetime.now().year
currentYear = 2014 # based on data
df['Age'] = currentYear - df['Year_Birth']
```

2. Kolom **Age Group**

Dari hasil umur customer akan disederhanakan menjadi 3 group : **Young Adult < 30, Adult 30-45 Tahun dan Senior Adult > 45 tahun.**

```
def age_group(x):
    if x > 45:
        grup = 'Senior Adult'
    elif x > 30:
        grup = 'Adult'
    else:
        grup = 'Young Adult'

    return grup

df['Age_group'] = df["Age"].apply(lambda x: age_group(x))
```

Feature Engineering / Extraction

3. Kolom **Has Child**

Menggabungkan **Kidhome** dan **Teenhome** menjadi feature **Has_child**, yang mana hasil penjumlahannya yang **memiliki anak minimal 1**.

```
df['Has_child'] = np.where(df["Kidhome"]+df["Teenhome"] > 0, 1, 0)
```

4. Kolom **Dependents**

Jumlah tanggungan dari customer, dari **penjumlahan Kidhome dan Teenhome**.

```
df['Dependents'] = df['Kidhome'] + df['Teenhome']
```

5. Kolom **Lifetime**

Sudah berapa bulan customer sejak pembelian pertama di supermarket.

```
df['Lifetime'] = (2014 - df["Dt_Customer"].dt.year)*12 + df["Dt_Customer"].dt.month
```

Feature Engineering / Extraction

6. Kolom **Spending**

Jumlah pembelian tiap customer pada keseluruhan product.

```
df['Spending']=df['MntWines']+ df['MntFruits']+ df['MntMeatProducts']+ df['MntFishProducts']+ df['MntSweetProducts']+ df['MntGoldProds']
```

7. Kolom **Primer and Tersier product**

Jumlah pembelian tiap customer pada kelompok primer dan tersier product.

```
df['Primer_purchase'] = df['MntFruits']+df['MntMeatProducts']+df['MntFishProducts']
df['Tersier_purchase'] = df['MntWines']+df['MntSweetProducts']+df['MntGoldProds']
```

8. Kolom **Total of Purchases**

Jumlah pembelian tiap customer pada keseluruhan metode pembelian.

```
df['Total_Purchases'] = df['NumDealsPurchases'] + df['NumWebPurchases'] + df['NumCatalogPurchases'] + df['NumStorePurchases']
```

9. Kolom **Total_Cmp**

Jumlah tiap customer merespon ke 5 campaign yang dilaksanakan (AcceptedCmp 1 - 5).

```
df['Total_Cmp']=df['AcceptedCmp1'].astype("int64")+ df['AcceptedCmp2'].astype("int64")+ df['AcceptedCmp3'].astype("int64")+
df['AcceptedCmp4'].astype("int64")+ df['AcceptedCmp5'].astype("int64")
```


Feature Engineering / Extraction

13. Kolom **Conversion Rate Web**

Perbandingan Total Purchases dengan Jumlah Pengunjung Website.

```
df['Conversion_rate_web'] = np.round(df['Total_Purchases'] / df['NumWebVisitsMonth'], 2)
df['Conversion_rate_web'].fillna(0, inplace=True)
df['Conversion_rate_web'].replace([np.inf, -np.inf], 0, inplace=True)
```

14. Kolom **Month Joined**

Membuat kolom extraction month dari tanggal Customer pertama kali berbelanja.

```
df['Month_joined'] = df['Dt_Customer'].dt.month
```

15. Kolom **Recency_sgmt**

Perkiraan pembagian dengan rentang 19 Hari : 4 score -> setengah bulan, 3 score -> 1 bulan, 2 score -> 1 setengah bulan, 1 score -> 2 bulan dan 0 score -> 3 bulan.

	Recency		
	min	max	count
Recency_sgmt			
0	80	99	446
1	60	79	437
2	40	59	450
3	20	39	447
4	0	19	456

```
divided = {5: 19, 4: 39, 3: 59, 2: 79}

def RScore(x,d):
    if x <= d[5]:
        return 4
    elif x <= d[4]:
        return 3
    elif x <= d[3]:
        return 2
    elif x <= d[2]:
        return 1
    else:
        return 0

df['Recency_sgmt'] = df['Recency'].apply(lambda x: RScore(x, divided))
df.groupby(['Recency_sgmt'])[['Recency']].agg(['min', 'max', 'count']).style.background_gradient(cmap='Reds')
```

Feature Engineering / Extraction

Hasil Feature Engineering / Extraction

	ID	Age	Age_group	Has_child	Dependents	Lifetime	Spending	Primer_purchase	Tersier_purchase
413	6504	39	Adult	1	1	23	78	42	36
581	5756	31	Adult	1	1	4	55	7	48
1225	8210	39	Adult	1	1	20	405	24	381
1324	839	39	Adult	1	1	21	170	54	116
1000	5527	27	Young Adult	1	1	14	58	26	32

	ID	Total_Purchases	Total_Cmp	Ever_Accept	Total_revenue	Income_sgmt	Conversion_rate_web	Month_joined	Recency_sgmt
413	6504	9	1	1	11	Low	1.80	11	3
581	5756	7	0	0	0	Medium	1.17	4	1
1225	8210	18	0	0	0	Medium	4.50	8	1
1324	839	9	0	0	0	Medium	1.80	9	2
1000	5527	7	0	0	0	Low	0.88	2	3

Feature Engineering / Extraction

Hasil Keseluruhan Feature

Categorical (String)

- Education - Basic, Graduation, Master, PhD
- Marital_Status - Single, Married, Divorced
- Age_group - Young Adult, Adult, Senior Adult
- Income_sgmt - High, Medium, Low

Categorical (Int)

- ID
- Kidhome - 0, 1, 2
- Teenhome - 0, 1, 2
- AcceptedCmp1 - 0, 1
- AcceptedCmp2 - 0, 1
- AcceptedCmp3 - 0, 1
- AcceptedCmp4 - 0, 1
- AcceptedCmp5 - 0, 1
- Ever_Accept - 0, 1
- Complain - 0, 1
- Response - 0, 1
- Has_child - 0, 1
- Recency_sgmt - 0, 1, 2, 3, 4

Numericals

- Year_Birth = 1940 - 1996
- Income = 1730.0 - 162397.0
- Kidhome = 0 - 2
- Teenhome = 0 - 2
- Recency = 0 - 99
- Age = 18 - 74
- Dependents = 0 - 3
- Lifetime = 1 - 36
- Spending = 5 - 2525
- Primer_purchase = 1 - 1727
- Tersier_purchase = 3 - 1689
- Total_Purchases = 0 - 44
- NumWebVisitsMonth = 0 - 20
- Conversion_rate_web = 0.0 - 43.0
- Total_Cmp = 0 - 4
- Total_revenue = 0 - 44
- Month_joined = 1 - 12

Numericals (one)

- Z_CostContact = 3
- Z_Revenue = 11

Numericals (Product)

- MntWines = 0 - 1493
- MntFruits = 0 - 199
- MntMeatProducts = 0 - 1725
- MntFishProducts = 0 - 259
- MntSweetProducts = 0 - 263
- MntGoldProds = 0 - 362

Numericals (Purchases)

- NumDealsPurchases = 0 - 15
- NumWebPurchases = 0 - 27
- NumCatalogPurchases = 0 - 28
- NumStorePurchases = 0 - 13

Timestamp

- Dt_Customer = 2012-07-30 - 2014-06-29

Feature Transformation (Numeric)

Mengecek Skewness di tiap kolom untuk menentukan jenis transformation

	Column Name	Skewness	Kurtosis	Type of Distribution					
0	Kidhome	0.636	-0.779	Bimodal Distribution	21	Tersier_purchase	0.972	0.045	Moderately Positively Skewed
1	Teenhome	0.406	-0.987	Bimodal Distribution	22	Age	0.092	-0.797	Normal Distribution (Symmetric)
2	Conversion_rate_web	2.332	6.442	Highly Positively Skewed	23	Lifetime	0.061	-1.003	Normal Distribution (Symmetric)
3	MntFishProducts	1.919	3.086	Highly Positively Skewed	24	Month_joined	0.001	-1.277	Normal Distribution (Symmetric)
4	MntFruits	2.103	4.058	Highly Positively Skewed	25	Recency	-0.004	-1.202	Normal Distribution (Symmetric)
5	MntGoldProds	1.884	3.548	Highly Positively Skewed	26	Year_Birth	-0.092	-0.797	Normal Distribution (Symmetric)
6	MntMeatProducts	2.084	5.515	Highly Positively Skewed	27	Z_CostContact	NaN	NaN	Uniform Distribution
7	MntSweetProducts	2.134	4.360	Highly Positively Skewed	28	Z_Revenue	NaN	NaN	Uniform Distribution
8	MntWines	1.175	0.597	Highly Positively Skewed					
9	NumCatalogPurchases	1.880	8.030	Highly Positively Skewed					
10	NumDealsPurchases	2.417	8.910	Highly Positively Skewed					
11	NumWebPurchases	1.380	5.681	Highly Positively Skewed					
12	Primer_purchase	1.588	2.214	Highly Positively Skewed					
13	Total_Cmp	2.721	7.971	Highly Positively Skewed					
14	Total_revenue	2.721	7.971	Highly Positively Skewed					
15	Dependents	0.415	-0.252	Moderately Normal Distribution (Symmetric)					
16	Income	0.352	0.709	Moderately Normal Distribution (Symmetric)					
17	NumWebVisitsMonth	0.208	1.816	Moderately Normal Distribution (Symmetric)					
18	Total_Purchases	0.250	-0.895	Moderately Normal Distribution (Symmetric)					
19	NumStorePurchases	0.700	-0.626	Moderately Positively Skewed					
20	Spending	0.860	-0.343	Moderately Positively Skewed					

Feature Transformation (Numeric)

Scaling and Converting to a Normal Distribution :

- Log Transformation
- Box-Cox Transformation
- Yeo-Johnson Transformation

Column yang akan di transform :

- Conversion_rate_web
- MntFishProducts
- MntFruits
- MntGoldProds
- MntMeatProducts
- MntSweetProducts
- MntWines
- NumCatalogPurchases
- NumDealsPurchases
- NumStorePurchases
- NumWebPurchases
- Primer_purchase
- Spending
- Tersier_purchase
- Total_revenue

Just Scaling

- Normalization
- Standardization

Column yang akan di transform

- Age
- Income
- Lifetime
- Month_joined
- NumWebVisitsMonth
- Recency
- Total_Purchases
- Year_Birth

Kolom yang *tidak perlu melakukan Transformasi* karena rentang nilai yang masih wajar sebagai berikut :

- Kidhome
- Teenhome
- Dependents
- Total_Cmp

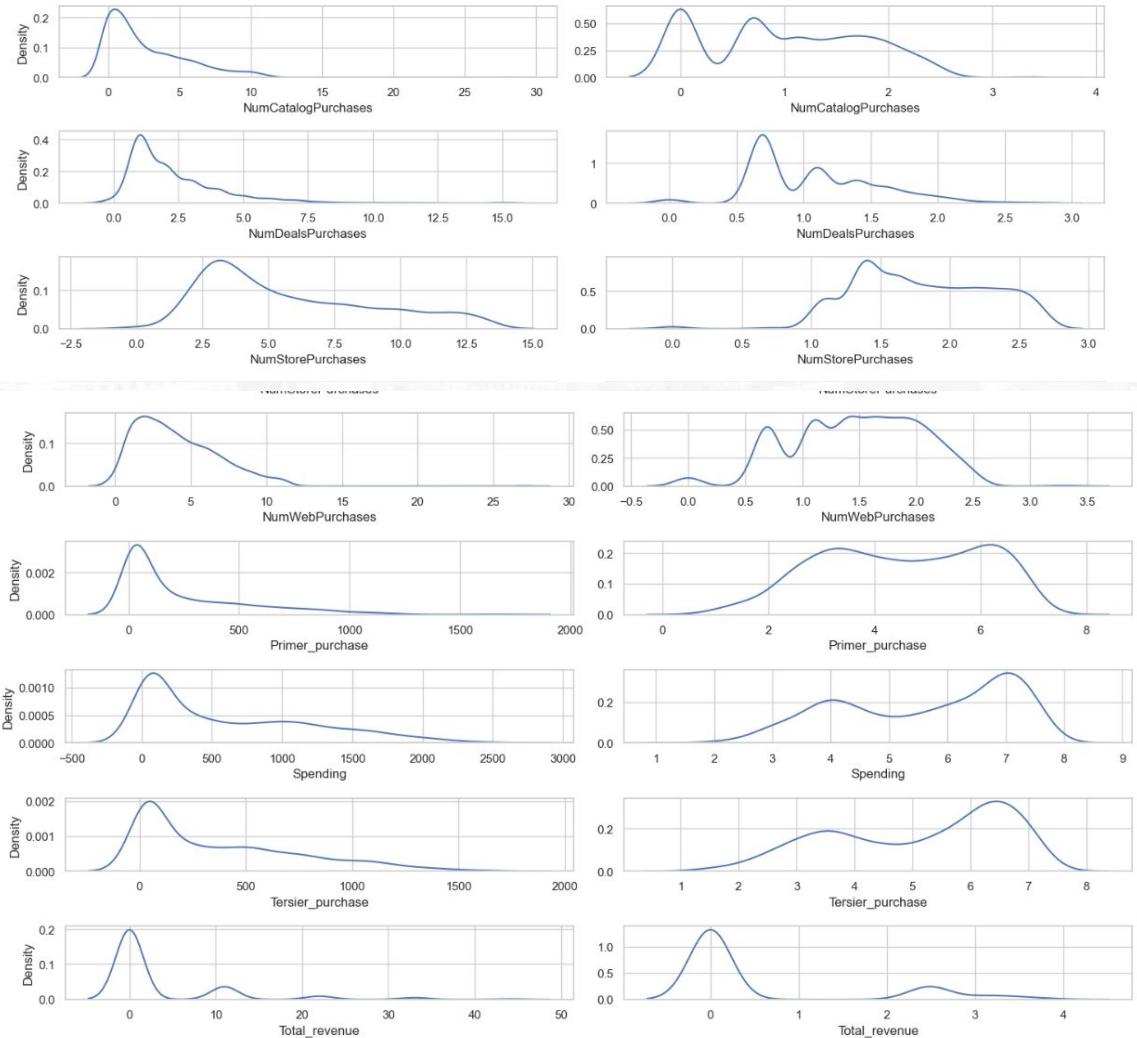
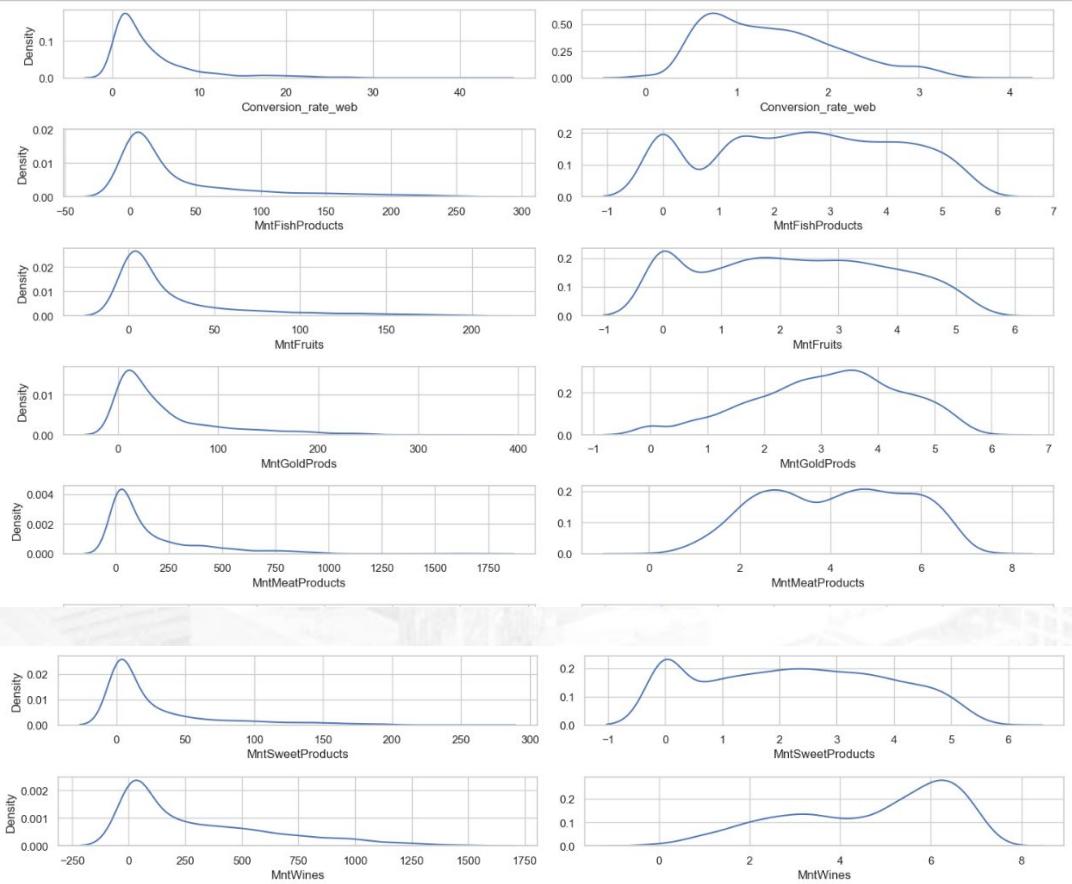
Choice Determination:

- Pada proses `Scaling and Converting to a Normal Distribution` ini kita menggunakan `Yeo-Johnson Transformation`, karena dari hasilnya kita bisa melihat hasil bentuk curve yang lebih Normal Distribusi
- Pada proses `Just Scaling` ini kita menggunakan `Normalization` karena lebih robust untuk algoritma yang akan kita gunakan

Feature Transformation (Numeric)

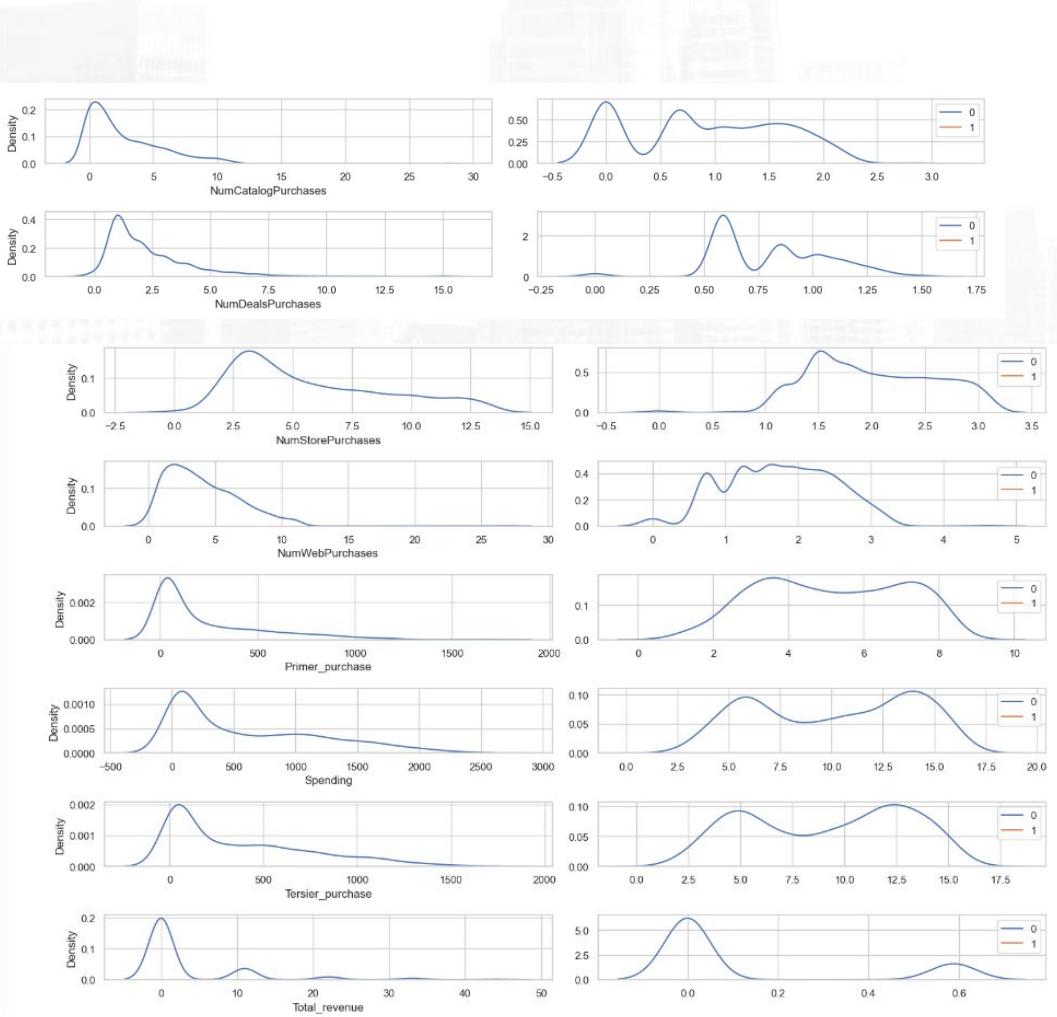
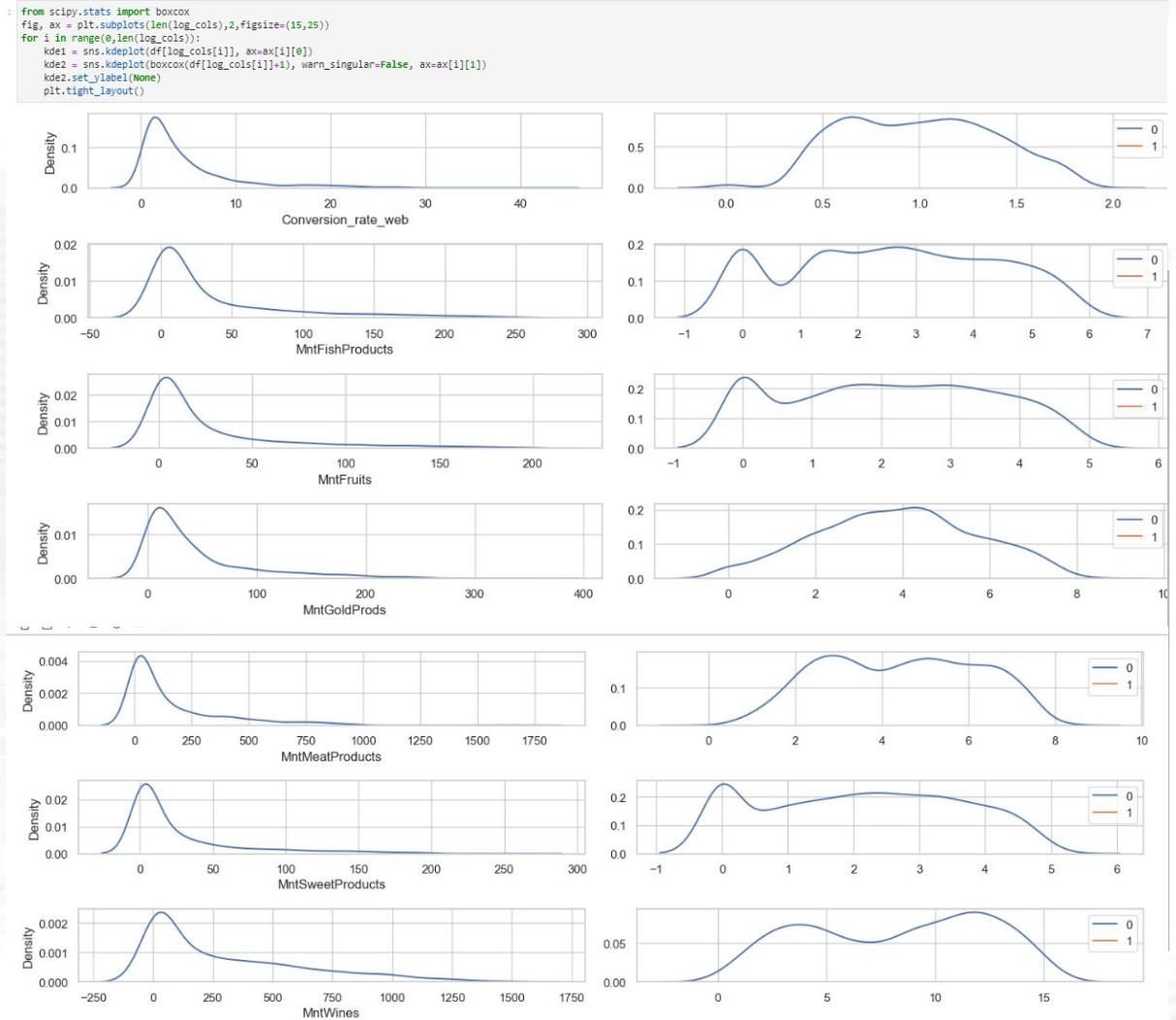
Log Transformation

```
[74]: fig, ax = plt.subplots(len(log_cols),2,figsize=(15,25))
for i in range(0,len(log_cols)):
    kde1 = sns.kdeplot(df[log_cols[i]], ax=ax[i][0])
    kde2 = sns.kdeplot(np.log(df[log_cols[i]]+1), ax=ax[i][1])
    kde2.set_ylabel(None)
    plt.tight_layout()
```



Feature Transformation (Numeric)

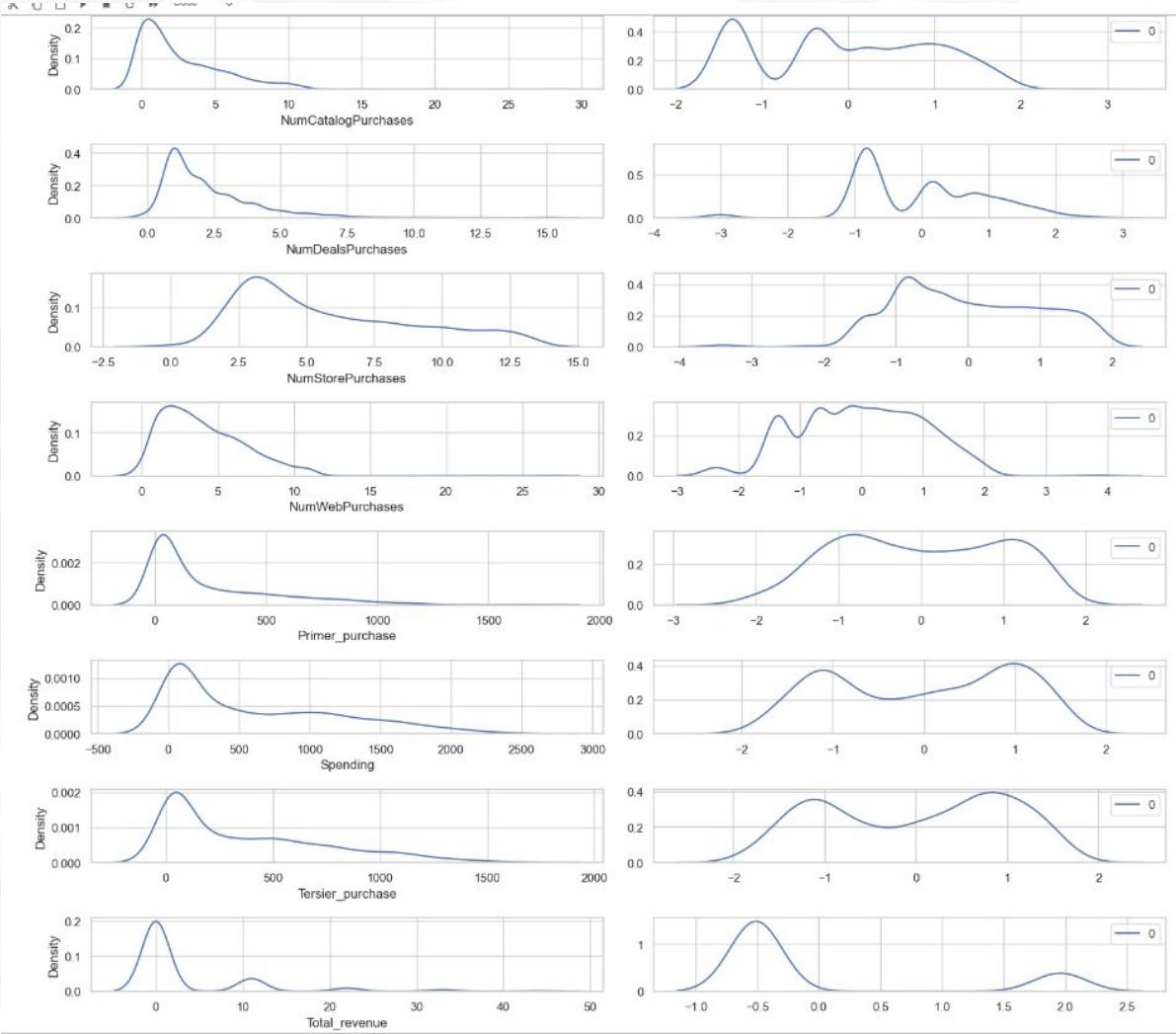
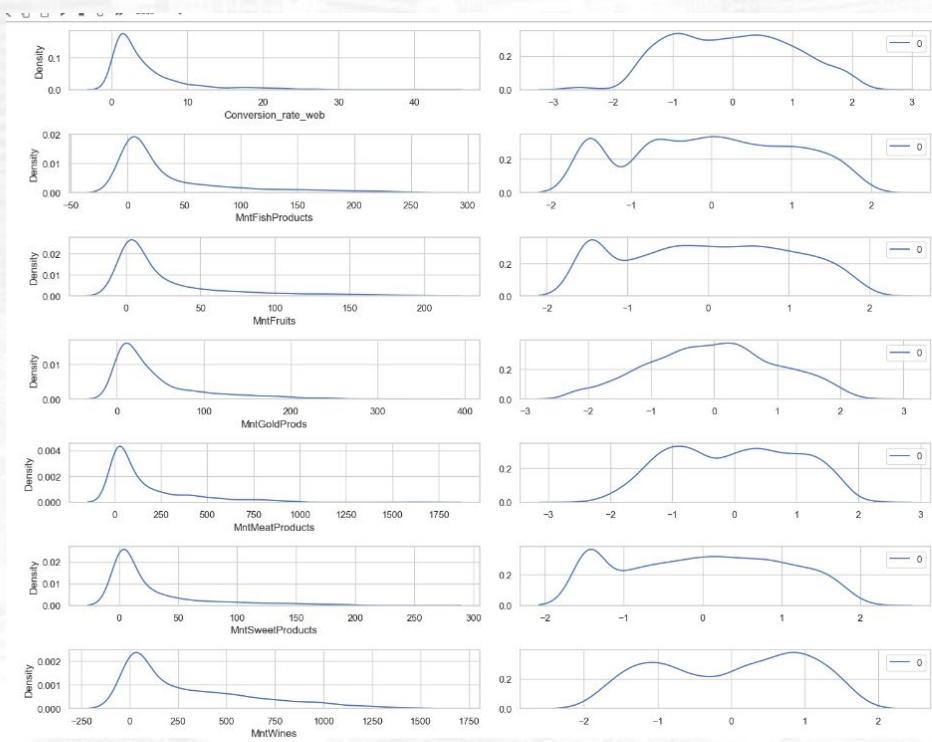
Box-Cox Transformation (with Scipy)



Feature Transformation (Numeric)

Box-Cox Transformation (with Sklearn)

```
#8]: from sklearn.preprocessing import PowerTransformer
fig, ax = plt.subplots(len(log_cols),2,figsize=(15,25))
for i in range(0,len(log_cols)):
    pt = PowerTransformer(method='box-cox')
    data = pt.fit_transform(df[[log_cols[i]]]+1)
    kde1 = sns.kdeplot(df[log_cols[i]], ax=ax[i][0])
    kde2 = sns.kdeplot(data, ax=ax[i][1])
    kde2.set_ylabel(None)
    plt.tight_layout()
```

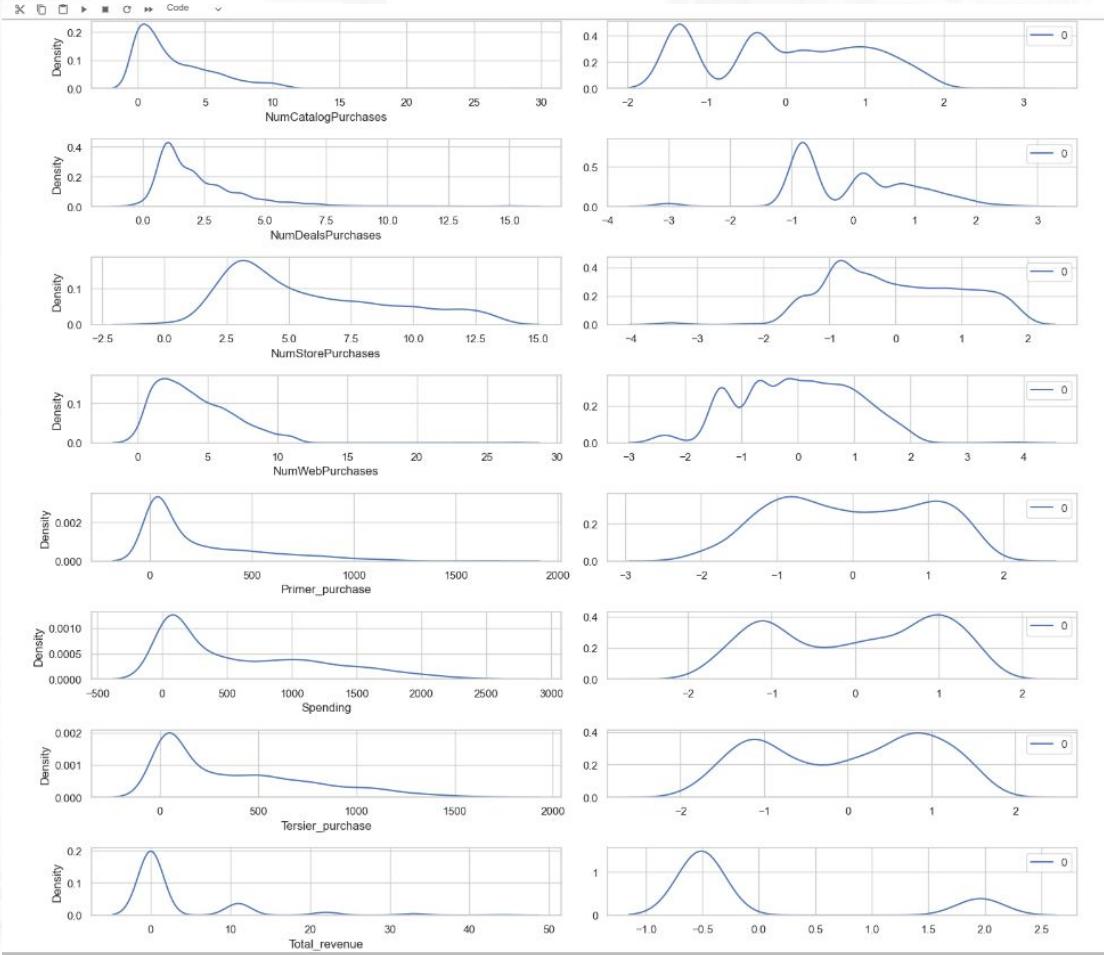
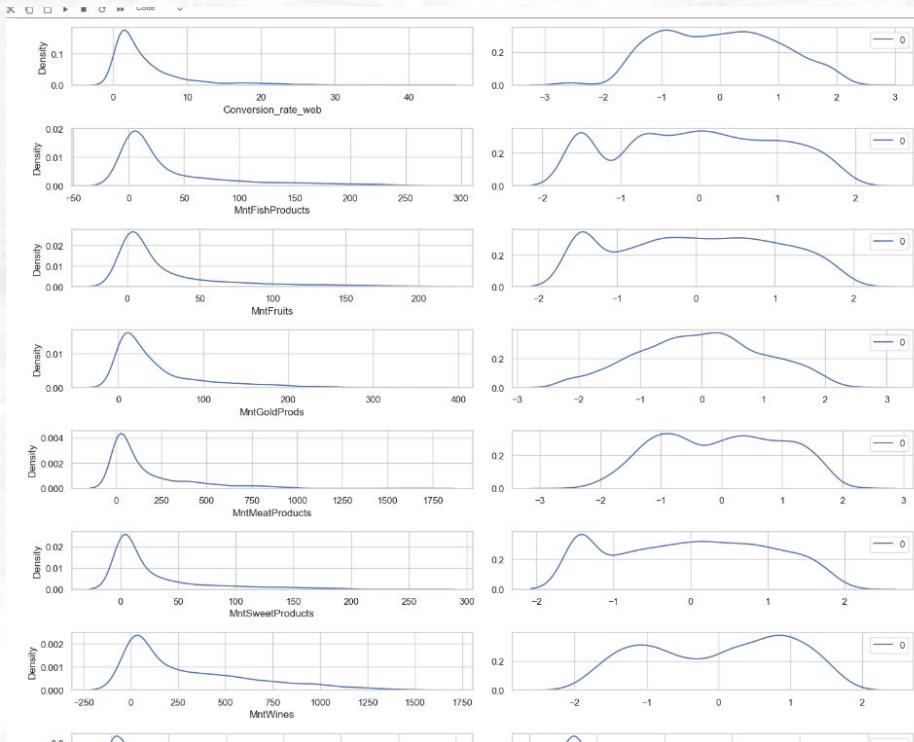


Feature Transformation (Numeric)

Yeo-Johnson Transformation (with Scipy)

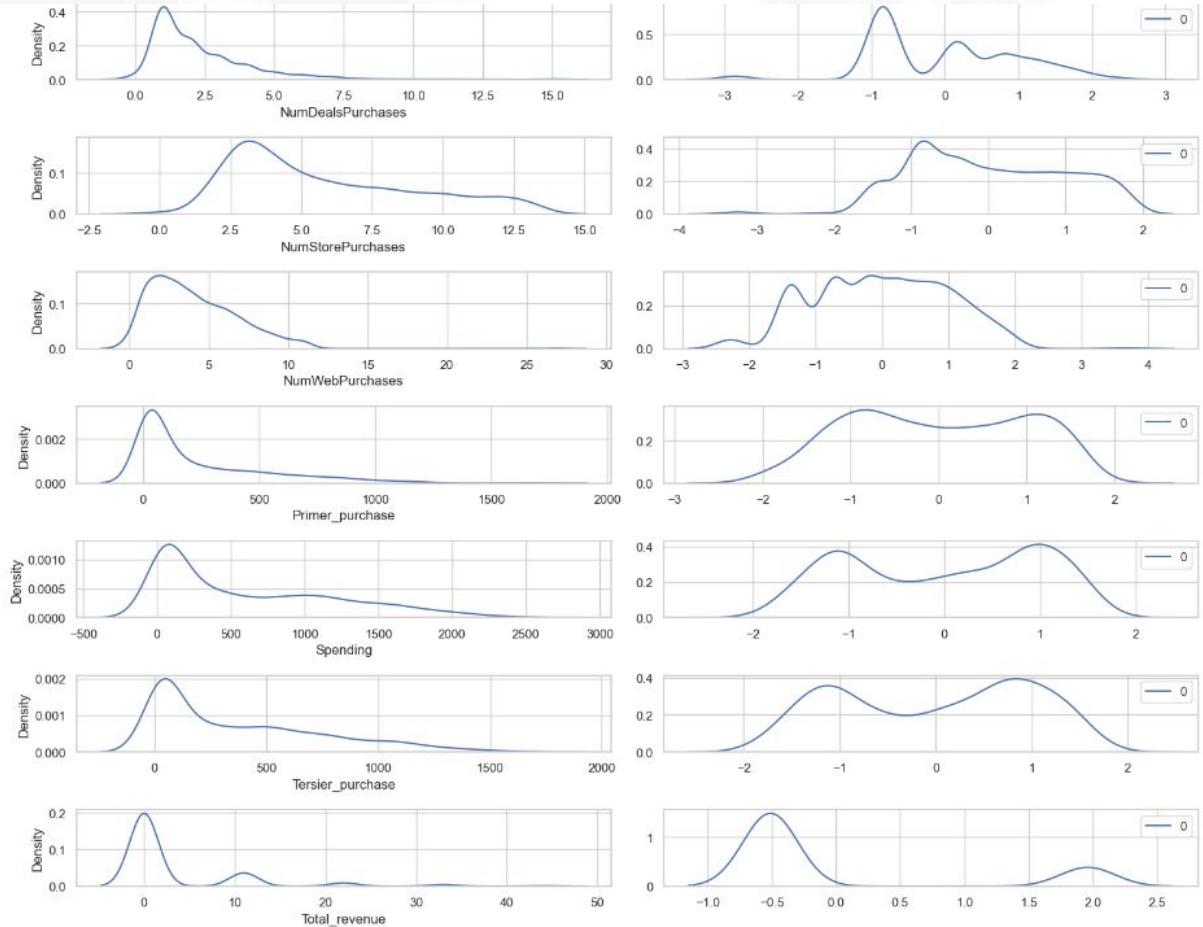
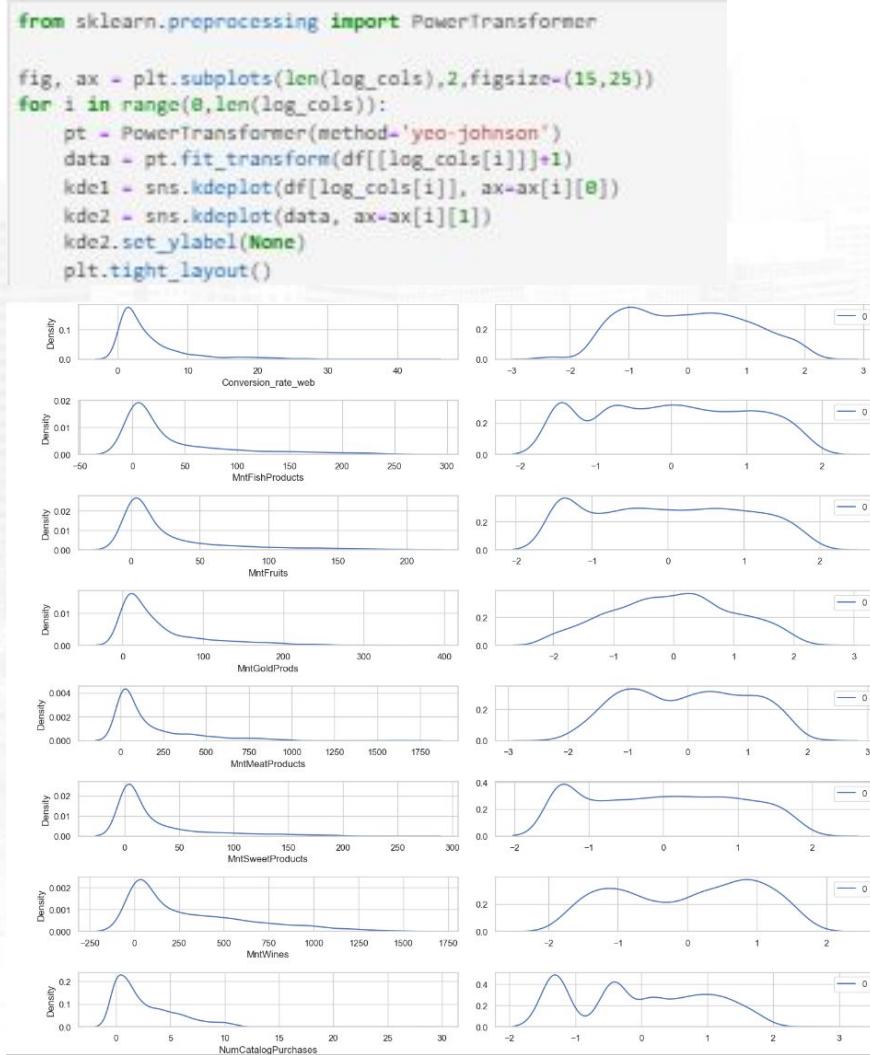
Unlike the Box-Cox transform, it does not require the values for each input variable to be strictly positive. It supports zero values and negative values. This means we can apply it to our dataset without scaling it first.

```
from scipy.stats import yeojohnson
fig, ax = plt.subplots(len(log_cols),2,figsize=(15,25))
for i in range(0,len(log_cols)):
    data, fitted_lambda = yeojohnson(df[log_cols[i]],lmbda=None)
    kde1 = sns.kdeplot(df[log_cols[i]], ax=ax[i][0])
    kde2 = sns.kdeplot(data, ax=ax[i][1])
    kde2.set_ylabel(None)
    plt.tight_layout()
```



Feature Transformation (Numeric)

Yeo-Johnson Transformation (with Sklearn)



Feature Transformation (Numeric)

Yeo-Johnson Transformation (with Sklearn)

	Conversion_rate_web	MntFishProducts	MntFruits	MntGoldProds	MntMeatProducts	MntSweetProducts	MntWines	I
count	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	
mean	2.955298e-16	1.112209e-17	-1.763646e-16	-8.818229e-17	1.747757e-17	2.812300e-16	3.495514e-17	
std	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	
min	-2.602504e+00	-1.509682e+00	-1.454849e+00	-2.138575e+00	-2.511714e+00	-1.434003e+00	-1.970872e+00	
25%	-8.424338e-01	-7.007989e-01	-9.853113e-01	-6.882051e-01	-8.385940e-01	-9.711568e-01	-9.104855e-01	
50%	-2.079832e-02	3.254224e-03	-1.431804e-03	3.731807e-03	3.471239e-02	-1.681003e-03	1.261694e-01	
75%	7.479659e-01	8.389835e-01	8.296137e-01	6.910780e-01	8.423659e-01	8.167801e-01	8.746546e-01	
max	2.363106e+00	1.862577e+00	1.884177e+00	2.492608e+00	2.222716e+00	2.012128e+00	1.812958e+00	

	NumCatalogPurchases	NumDealsPurchases	NumStorePurchases	NumWebPurchases	Primer_purchase	Spending	Tersier_purchase	Total_revenue
	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03	2.236000e+03
	-5.878820e-17	-4.448836e-16	-3.860954e-16	1.382317e-16	-7.467690e-17	2.351528e-16	-9.930439e-17	7.944351e-18
	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00	1.000224e+00
	-1.338023e+00	-3.011224e+00	-3.392032e+00	-2.362005e+00	-2.314924e+00	-2.069143e+00	-1.976478e+00	-5.103128e-01
	-1.338023e+00	-8.201550e-01	-8.627790e-01	-7.093807e-01	-8.349123e-01	-9.800995e-01	-9.755613e-01	-5.103128e-01
	1.589299e-01	1.581372e-01	-4.692181e-02	1.851832e-01	-3.838851e-02	1.312892e-01	1.334070e-01	-5.103128e-01
	7.999743e-01	7.441576e-01	8.062471e-01	8.238423e-01	8.913990e-01	9.161492e-01	8.693208e-01	-5.103128e-01
	2.767807e+00	2.658365e+00	1.780425e+00	3.933317e+00	2.046915e+00	1.757460e+00	1.840735e+00	1.985443e+00

Feature Transformation (Numeric)

Normalization

```
from sklearn.preprocessing import MinMaxScaler
# create a scaler object
scaler = MinMaxScaler()
# fit and transform the data
df[norm_cols] = pd.DataFrame(scaler.fit_transform(df[norm_cols]), columns=df[norm_cols].columns)

df[norm_cols].describe()
```

	Age	Income	Lifetime	Month_joined	NumWebVisitsMonth	Recency	Total_Purchases	Year_Birth
count	2236.000000	2236.000000	2236.000000	2236.000000	2236.000000	2236.000000	2236.000000	2236.000000
mean	0.483964	0.312764	0.489522	0.496910	0.265944	0.496124	0.338012	0.516036
std	0.208987	0.133932	0.306805	0.317167	0.121344	0.292498	0.174497	0.208987
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.339286	0.209162	0.342857	0.181818	0.150000	0.242424	0.181818	0.339286
50%	0.464286	0.308968	0.485714	0.454545	0.300000	0.494949	0.340909	0.535714
75%	0.660714	0.415382	0.657143	0.818182	0.350000	0.747475	0.477273	0.660714
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Berdasarkan hasil pengecekan pada beberapa fitur yang telah diproses menggunakan transformation sebelumnya, dapat diketahui bahwa keseluruhan nilai skewnessnya sudah memiliki rentang yang lebih seragam (tidak jauh dan tidak terlalu bervariasi). Sehingga dapat disimpulkan bahwa teknik fitur transformation yang telah kami lakukan sudah valid dan kami.

Feature Encoding (Categoric)

Melihat feature categorical yang masih memiliki nilai betype string/object

```
#cek unique value
for x in cat_str:
    unq = list(df[x].unique())
    sorted(unq)

    print(f'===== {x} =====')
    if len(unq) >= 10:
        unq = list(unq[:10])+['.....']
        print(f'{unq}')
    else:
        print(f'{unq}')
    print()

===== Education =====
['Graduation', 'PhD', 'Master', 'Basic']

===== Marital_Status =====
['Single', 'Married', 'Divorced']

===== Age_group =====
['Senior Adult', 'Young Adult', 'Adult']

===== Income_sgmt =====
['Medium', 'High', 'Low']
```

cat_str

```
['Education', 'Marital_Status', 'Age_group', 'Income_sgmt']
```

Berdasarkan hasil analisis, berikut beberapa encoding yang dilakukan:

1. Label Encoding

- **Label Encoder**
- **Manually Mapped**

Adapun daftar column yang akan diproses :

- **Education** => Basic (0), Graduation (1), Master (2), PhD (3)
- **Age_group** => Young Adult (0), Adult (1), Senior Adult (2)
- **Income_sgmt** => Low (0), Medium (1), High (2)

2. One Hot Encoding

- **get_dummies**
- **OneHotEncoder**

Adapun daftar column yang akan diproses :

- **Marital_Status** => Single, Married, Divorced

Choice Determination

- Pada proses **Label Encoding** ini kita menggunakan **Manually Mapped**, karena kita bisa menentukan secara fleksibel urutan/order dari categorical feature
- Pada proses **One Hot Encoding** ini kita menggunakan **OneHotEncoder**, karena hasil encodingnya lebih rapi dan lebih mudah untuk dilakukan adjust

Feature Encoding (Categoric)

1. Label Encoding

a) Menggunakan **Label Encoder**

```
# from sklearn.preprocessing import LabelEncoder

# cat = cat_str.copy()
# cat.remove("Marital_Status")

# le = LabelEncoder()

# for i in cat_str:
#     le.fit(df[i])
#     df[i] = le.transform(df[i])
#     print(le.classes_)
```

b) Menggunakan Metode **Mapping**

```
map_edu = {
    'Basic' : 0,
    'Graduation' : 1,
    'Master' : 2,
    'PhD' : 3
}

df['Education'] = df['Education'].map(map_edu)
df['Education'].unique()

array([1, 3, 2, 0], dtype=int64)

map_age = {
    'Young Adult' : 0,
    'Adult' : 1,
    'Senior Adult' : 2
}

df['Age_group'] = df['Age_group'].map(map_age)
df['Age_group'].unique()

array([2, 0, 1], dtype=int64)

map_income = {
    'Low' : 0,
    'Medium' : 1,
    'High' : 2
}

df['Income_sgmt'] = df['Income_sgmt'].map(map_income)
df['Income_sgmt'].unique()

array([1, 2, 0], dtype=int64)
```

Feature Encoding (Categoric)

2. One Hot Encoding

a) Menggunakan `get_dummies`

```
# # Not Auto Drop Columns
# ohe = pd.get_dummies(df["Marital_Status"])
# df.drop("Marital_Status", axis=1, inplace=True)
# df = df.join(ohe)

# # cara 2 : Auto drop Column
# # df = pd.get_dummies(data=df, columns=["Marital_Status"])

# df.head()
```

b) Menggunakan `OneHotEncoder`

```
oh.fit_transform(df[["Marital_Status"]])
<2236x3 sparse matrix of type '<class 'numpy.float64'>'  
with 2236 stored elements in Compressed Sparse Row format>

from sklearn.preprocessing import OneHotEncoder

oh = OneHotEncoder()
df_ohe = pd.DataFrame(
    oh.fit_transform(df[["Marital_Status"]]).toarray(),
    columns=list(oh.categories_[0])
)
print(oh.categories_[0])

df.drop("Marital_Status", axis=1, inplace=True)

df = pd.concat([df, df_ohe], axis=1)
['Divorced' 'Married' 'Single']
```

Kesimpulan

Berdasarkan hasil pengecekan pada beberapa fitur yang telah diproses menggunakan encoding sebelumnya, dapat diketahui bahwa keseluruhan nilai telah bertipe numeric sesuai dengan nilai yang kita assign. Sehingga dapat disimpulkan bahwa teknik fitur encoding yang telah kami lakukan sudah valid dan kami.

Feature Selection

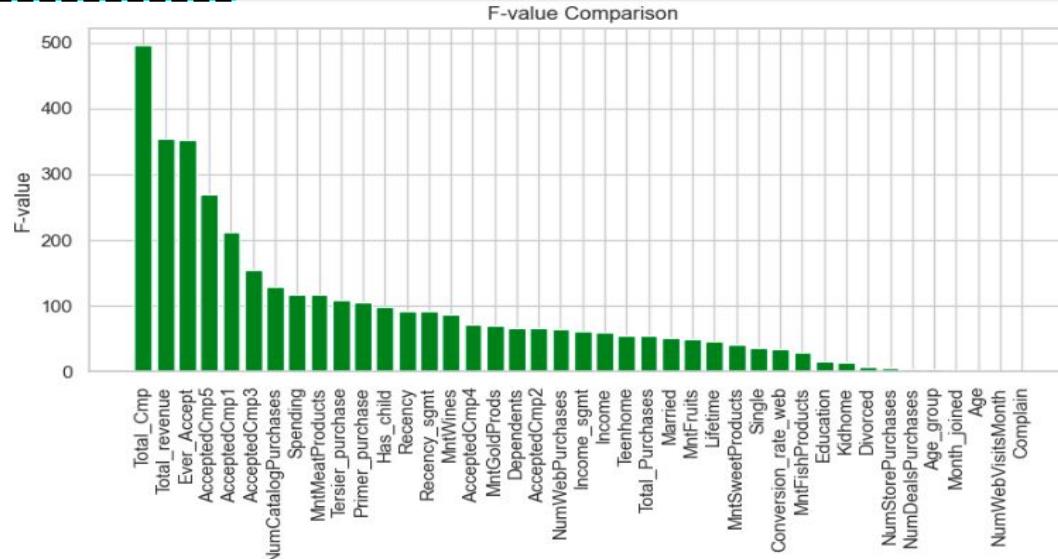
1. Drop Unnecessary Features

```
df.drop(['ID', 'Year_Birth', 'Dt_Customer', 'Z_CostContact', 'Z_Revenue'], inplace=True, axis=1)
```

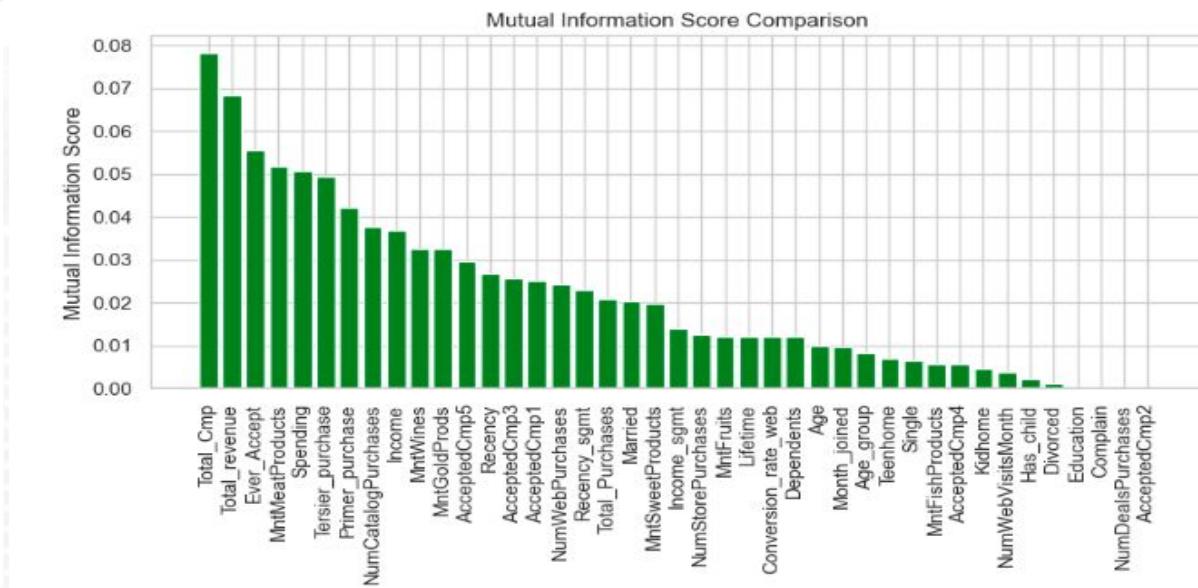
- **ID** >> memiliki banyak kategori
- **Year_Birth** >> Feature extraction untuk mengambil data **Age** pada tahun 2014
- **Dt_Customer** >> tidak terlalu mempengaruhi model
- **Z_CostContact** dan **Z_Revenue** >> tidak memberikan informasi yang signifikan terhadap model prediksi

2. Univariate Selection

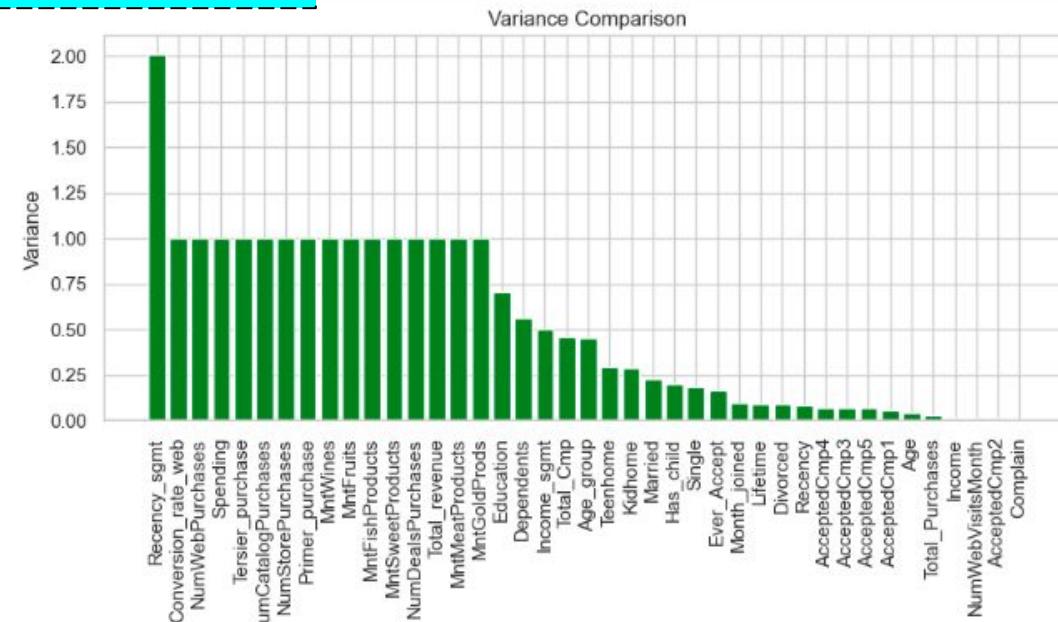
ANOVA F-value



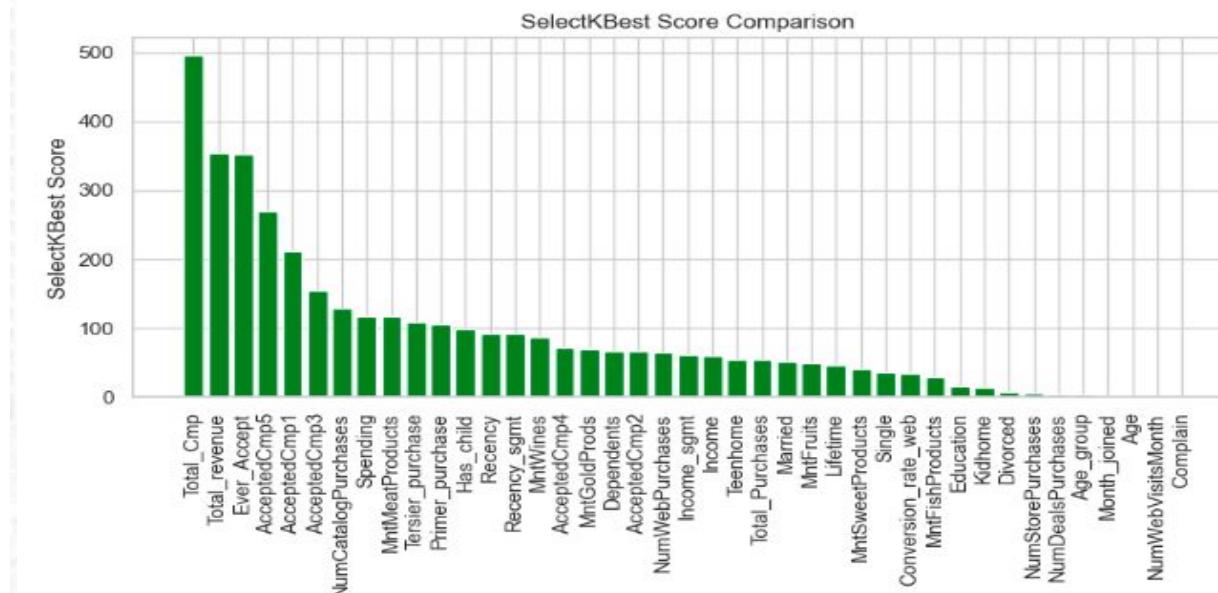
Mutual information



Variance Threshold

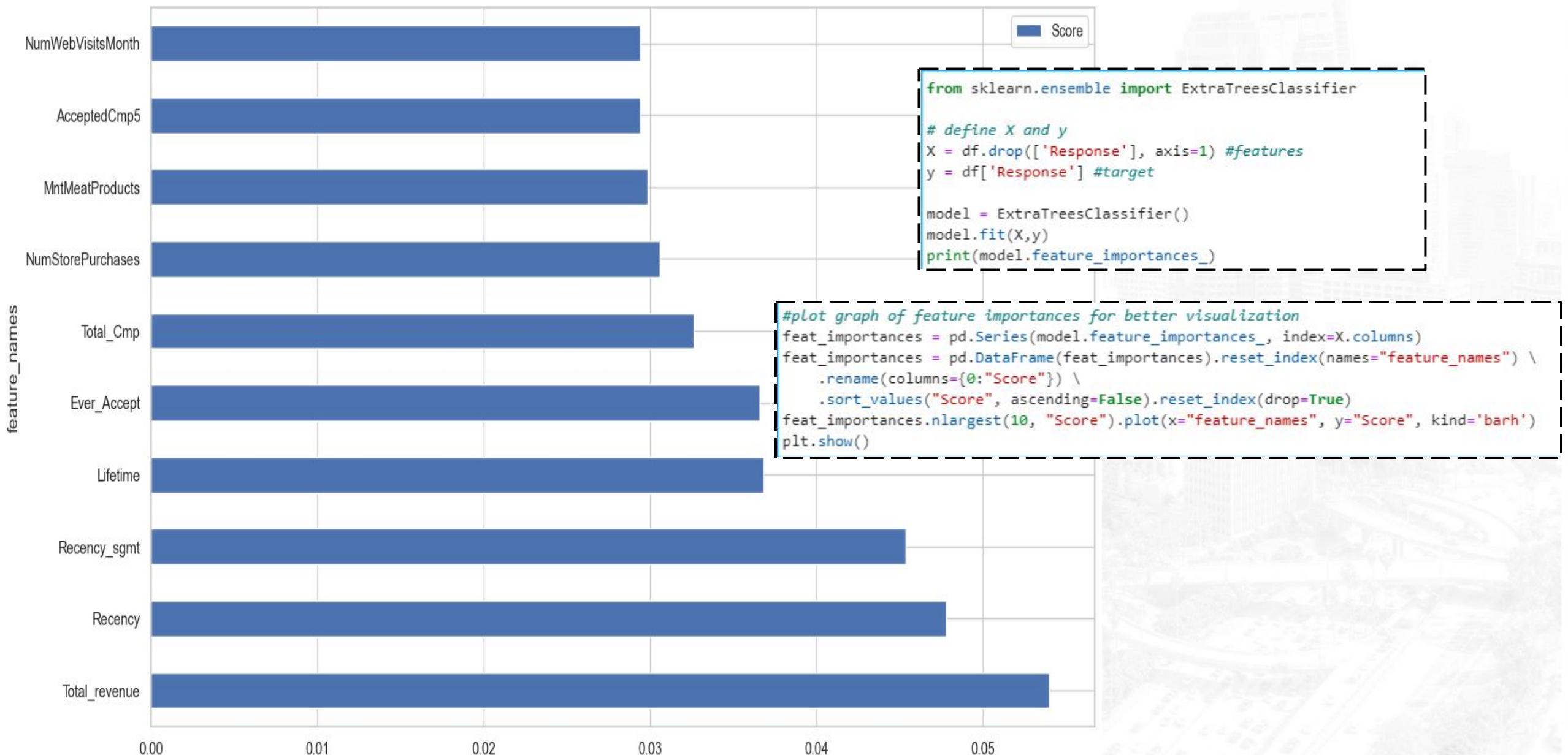


Scikit-learn's SelectKBest



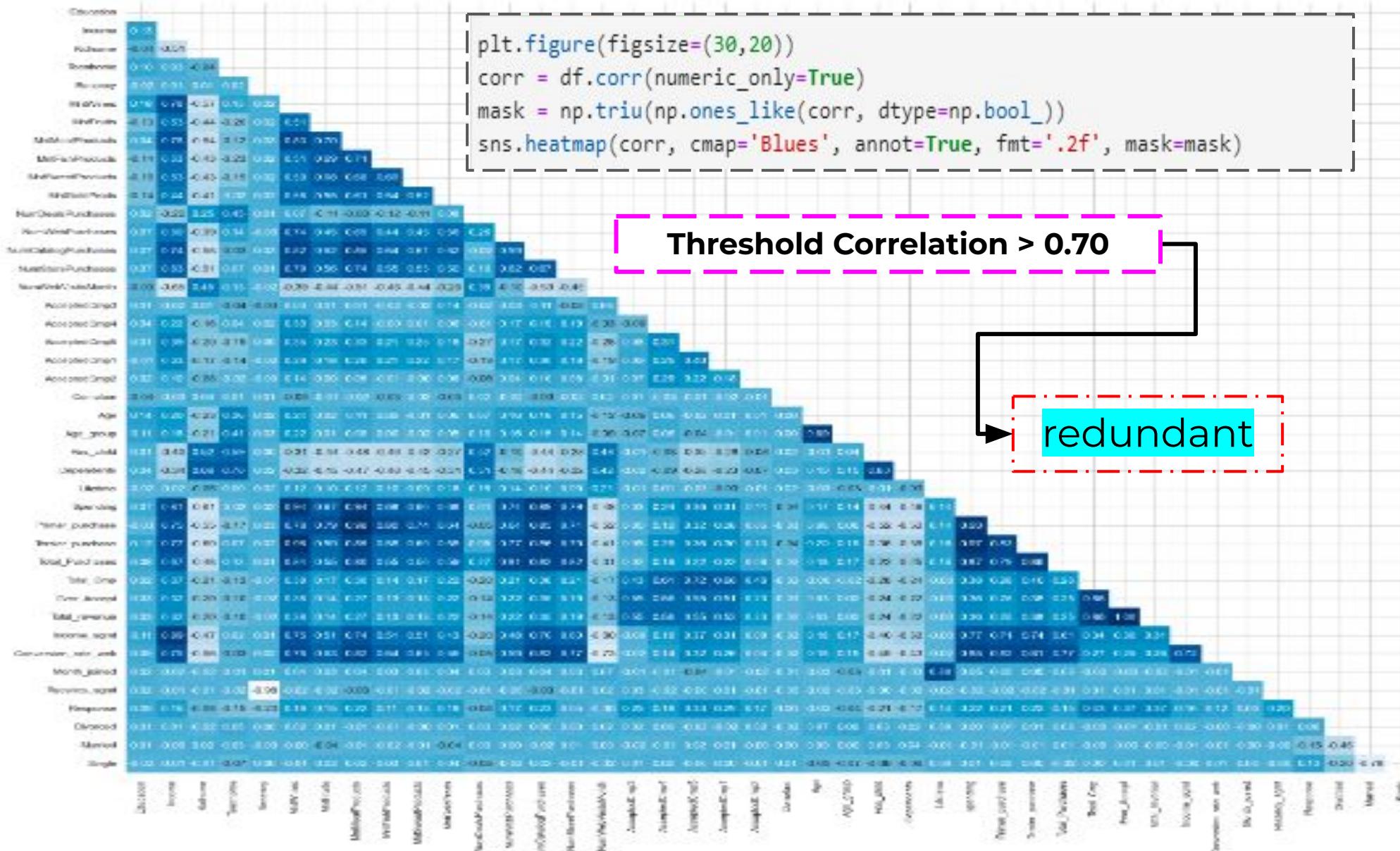
Feature Selection

3. Feature Importance

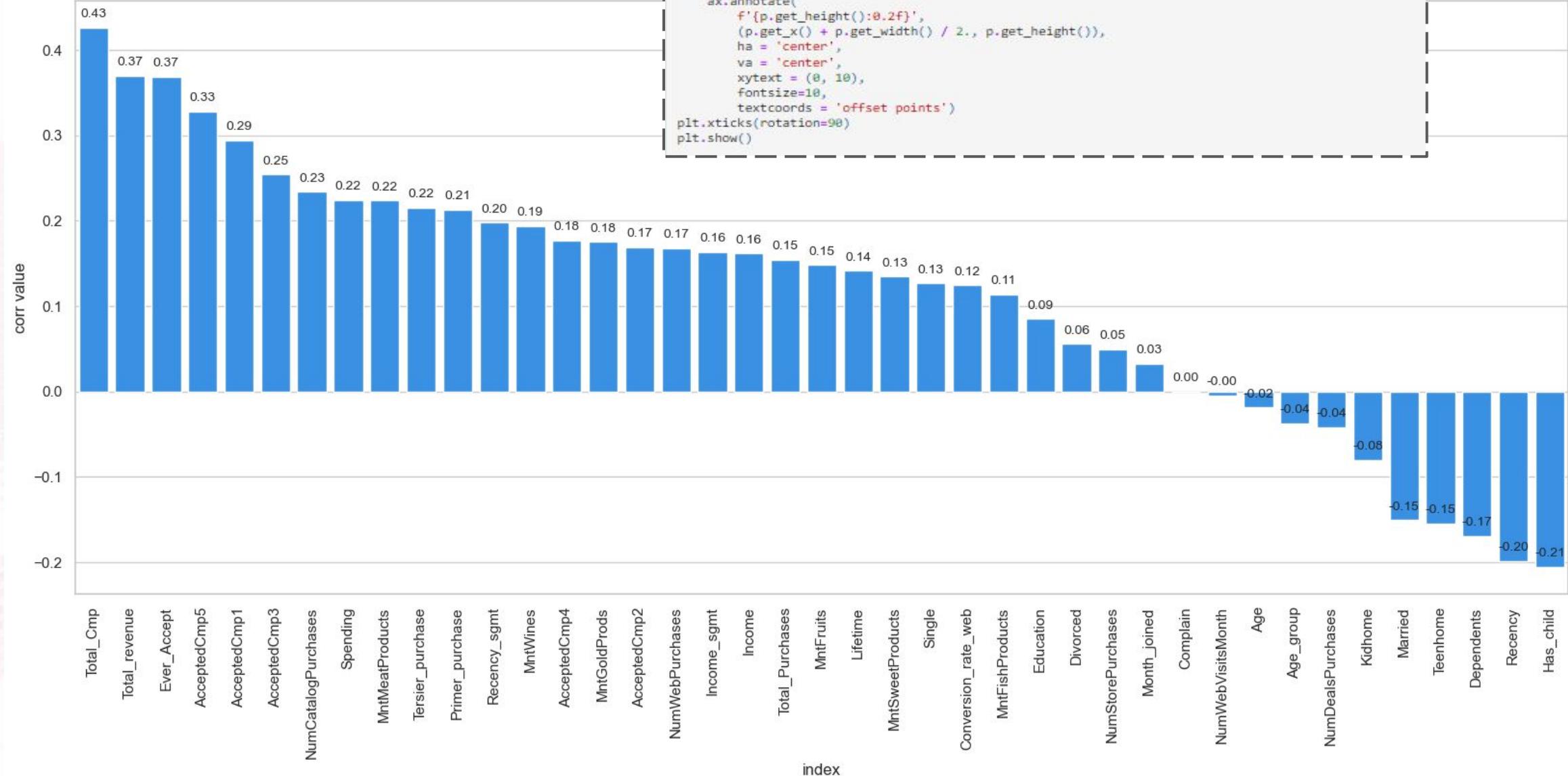


Feature Selection

4. Correlation Matrix with Heatmap



Checking Correlation with Target (Response)



Mengambil 20 Top Features

```
for i in corr["index"].values[:20]:
    if i not in feature_importance:
        feature_importance.append(i)
feature_importance
```

output:

```
['Total_Cmp', 'Total_revenue',
'Ever_Accept', 'AcceptedCmp5',
'AcceptedCmpl', 'AcceptedCmp3',
'NumCatalogPurchases', 'Spending',
'MntMeatProducts', 'Tersier_purchase',
'Primer_purchase', 'Has_child',
'Recency', 'Recency_sgmt',
'MntWines', 'AcceptedCmp4',
'MntGoldProds', 'Dependents',
'AcceptedCmp2', 'NumWebPurchases',
'Conversion_rate_web', 'NumStorePurchases',
'MntFruits', 'MntFishProducts',
'MntSweetProducts', 'NumDealsPurchases',
'Education', 'Income_sgmt',
'Income', 'Total_Purchases',
'Married', 'Lifetime',
'NumWebVisitsMonth', 'Age']
```

5. Check Data Redundancy

Manampilkan Korelasi Feature > Threshold 0.70

```
def corrtarget(x):
    target = "Response"
    return df[x].corr(df[target])

def corrresp(x):
    target = "Response"
    col1 = x["A"]
    col2 = x["B"]

    cor1 = df[col1].corr(df[target])
    cor2 = df[col2].corr(df[target])

    if cor1 < cor2:
        return col1
    else:
        return col2
    return col1
```

```
corr_matrix = df[feature_importance].corr()
target = "Response"

# Flatten correlation matrix.
flat_cm = corr_matrix.stack().reset_index()
flat_cm.columns = ['A', 'B', 'correlation']
flat_cm = flat_cm.loc[flat_cm.correlation < 1, :]
flat_cm = flat_cm.sort_values("correlation", ascending=False)
redundan = flat_cm[flat_cm["correlation"] >= 0.7].reset_index(drop=True)
redundan['A vs Target'] = redundan['A'].apply(lambda x: corrtarget(x))
redundan['B vs Target'] = redundan['B'].apply(lambda x: corrtarget(x))
redundan = redundan.drop_duplicates(subset=["correlation"])
redundan["drop"] = redundan.apply(corrresp, axis=1)
redundan
```

5. Check Data Redundancy (.....continue)

Feature Selection

	A	B	correlation	A vs Target	B vs Target	drop
0	Ever_Accept	Total_revenue	0.999982	0.368726	0.369906	Ever_Accept
2	MntMeatProducts	Primer_purchase	0.977667	0.223704	0.212989	Primer_purchase
4	MntWines	Tersier_purchase	0.976146	0.193849	0.215490	MntWines
6	Tersier_purchase	Spending	0.970442	0.215490	0.223961	Tersier_purchase
8	MntMeatProducts	Spending	0.937875	0.223704	0.223961	MntMeatProducts
10	MntWines	Spending	0.937196	0.193849	0.223961	MntWines
12	Spending	Primer_purchase	0.929771	0.223961	0.212989	Primer_purchase
14	Spending	NumCatalogPurchases	0.894305	0.223961	0.233715	Spending
16	Income	Income_sgmt	0.892218	0.161643	0.163239	Income
18	Spending	Total_Purchases	0.873134	0.223961	0.154715	Total_Purchases
20	Total_revenue	Total_Cmp	0.862955	0.369906	0.426206	Total_revenue
22	Tersier_purchase	Total_Purchases	0.861373	0.215490	0.154715	Total_Purchases
24	Ever_Accept	Total_Cmp	0.860109	0.368726	0.426206	Ever_Accept
26	Tersier_purchase	NumCatalogPurchases	0.856828	0.215490	0.233715	Tersier_purchase
28	Tersier_purchase	MntMeatProducts	0.852559	0.215490	0.223704	Tersier_purchase
30	NumCatalogPurchases	MntMeatProducts	0.845912	0.233715	0.223704	MntMeatProducts

110	Income_sgmt	Tersier_purchase	0.738756	0.163239	0.215490	Income_sgmt
112	NumWebPurchases	MntWines	0.737381	0.168142	0.193849	NumWebPurchases
114	NumStorePurchases	Primer_purchase	0.736144	0.049484	0.212989	NumStorePurchases
116	Conversion_rate_web	Income_sgmt	0.728599	0.124409	0.163239	Conversion_rate_web
118	Total_Cmp	AcceptedCmp5	0.716298	0.426206	0.328148	AcceptedCmp5
120	Primer_purchase	Income_sgmt	0.714907	0.212989	0.163239	Income_sgmt
122	MntMeatProducts	MntFishProducts	0.708560	0.223704	0.113616	MntFishProducts
124	MntFruits	MntMeatProducts	0.702425	0.148318	0.223704	MntFruits

Drop Redundancy & List of feature Importances

```
for i in list(redundan["drop"].unique()):
    feature_importance.remove(i)

feature_importance = sorted(feature_importance)
feature_importance
```

output:

```
['AcceptedCmp1',
 'AcceptedCmp2',
 'AcceptedCmp3',
 'AcceptedCmp4',
 'Age',
 'Dependents',
 'Education',
 'Lifetime',
 'Married',
 'MntGoldProds',
 'NumCatalogPurchases',
 'NumDealsPurchases',
 'NumWebVisitsMonth',
 'Recency',
 'Recency_sgmt',
 'Total_Cmp']
```

Data Splitting

split training set and testing set

75:25

```
# define X and y
X = df.drop(['Response'], axis=1)[feature_importance] #features
y = df['Response'] #target
```

```
from sklearn.model_selection import train_test_split

# splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, stratify=y, random_state=42)
print(X_train.shape, X_test.shape)
```

output:

(1677, 16) (559, 16)

Handling Imbalanced Data

Status risiko → **highly imbalanced** (15% Response & 85% No Response)

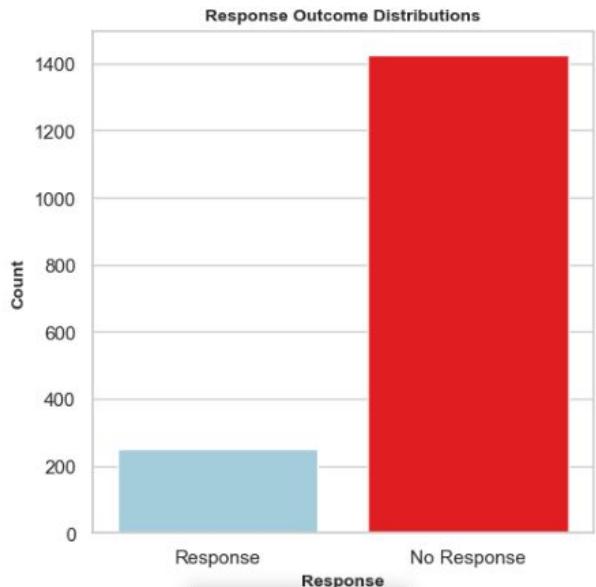
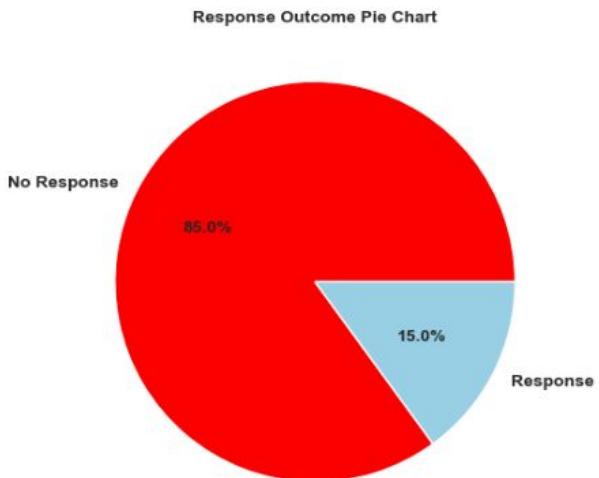
→ **resampling**

```
fig = plt.figure(figsize = (10, 5))

plt.subplot(121)
plt.pie(y_train.value_counts(),
         labels = ['No Response', 'Response'],
         autopct = '%.1f%%',
         radius = 1,
         colors=["#e31a1c", "#a6cee3"],
         textprops={'fontsize': 10, 'fontweight': 'bold'})
plt.title('Response Outcome Pie Chart', fontsize = 10, fontweight = 'bold')

plt.subplot(122)
resp = y_train.apply(lambda x: "No Response" if x == 0 else "Response")
t = sns.countplot(x=resp, palette=["#a6cee3", "#e31a1c"])
t.set_xlabel('Response', fontweight = 'bold', fontsize = 10)
t.set_ylabel('Count', fontweight = 'bold', fontsize = 10)

plt.title('Response Outcome Distributions', fontsize = 10, fontweight = 'bold')
plt.tight_layout()
```



Handling Imbalanced Data

SAMPLING

```
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler, SMOTE

print('Before OverSampling, the shape of X_train: {}'.format(X_train.shape))
print('Before OverSampling, the shape of y_train: {} \n'.format(y_train.shape))

print("Before OverSampling, counts of label '1': {}".format(sum(y_train == 1))) # Response
print("Before OverSampling, counts of label '0': {} \n".format(sum(y_train == 0))) # No Response
```

```
# Oversampling SMOTE
sm = SMOTE(sampling_strategy=0.5, random_state = 2)
X_balanced_res, y_balanced_res = sm.fit_resample(X_train,y_train)

print('After OverSampling, the shape of X_train: {}'.format(X_balanced_res.shape))
print('After OverSampling, the shape of y_train: {} \n'.format(y_balanced_res.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_balanced_res == 1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_balanced_res == 0)))

X_train = X_balanced_res
y_train = y_balanced_res
```

output:

SMOTE

```
Before OverSampling, counts of label '1': 334
Before OverSampling, counts of label '0': 1895

After OverSampling, the shape of train_X: (2274, 44)
After OverSampling, the shape of train_y: (2274,)

After OverSampling, counts of label '1': 758
After OverSampling, counts of label '0': 1516
```

Stage 3

*Machine Learning Modelling &
Evaluation*



STAGE2 - Modelling Experiments

(1-3 slide)









STAGE3 - Executive Summary & Recommendation (1 slide)

Jelaskan kesimpulan yang didapatkan dari semua bagian, dan rekomendasi keputusan yang dapat diimplementasikan pada masalah yang diangkat

Pembagian Tugas (1 slide)

Penjelasan kontribusi dari masing-masing anggota team (siapa mengerjakan apa saja). Tuliskan secara detail, termasuk pembagian dalam penulisan laporan, pembuatan slide, dan presentasi.