

Group 9

Batch 28

Dokumen Laporan Final Project

(dipresentasikan setiap sesi mentoring)



Stage 0



Latar Belakang Masalah

Sebuah perusahaan diketahui memiliki tingkat acceptance marketing campaign sekitar 14.91%, yang mana hal tersebut masih dianggap kurang oleh manajemen dalam menghadapi persaingan bisnis.

Sehingga pihak manajemen meminta tim marketing untuk meningkatkan lagi tingkat acceptance marketing campaign tersebut agar cost yang dikeluarkan perusahaan dalam melakukan marketing campaign lebih efisien dan jumlah revenue yang mereka raih pada tahun-tahun berikutnya meningkat.

Maka dari itu, tim marketing berencana menerapkan strategi Targeted Marketing dengan bantuan tim data science untuk mengolah data historis penjualan yang telah mereka rekap sebelumnya dan mengelompokkan user ke dalam sebuah kategori tertentu sesuai dengan karakteristiknya masing-masing, sehingga dapat dipilah antara yang layak mendapatkan campaign dengan yang tidak mendapatkan campaign

Peran

Sebagai tim data scientist, kami bertanggungjawab untuk memberikan rekomendasi untuk meningkatkan efektivitas marketing campaign perusahaan berdasarkan pola data yang tersedia.

Goal

Meningkatkan response rate atau tingkat acceptance rate dari marketing campaign yang dilakukan oleh perusahaan, sehingga profit perusahaan serta efisiensi marketing cost dapat lebih optimal.

Objective

Membuat sistem prediksi model klasifikasi/clustering yang dapat menentukan targeting user yang tepat. Dengan ini tentu akan memperbesar nilai business metrics yang telah ditentukan seperti traffic dan sales performance. Sistem sudah menentukan mana user yang memang sedang tertarik atau bagian dari market untuk campaign yang akan dijalankan.

Business Metrics

- Response rate : Rasio jumlah customer yang merespon dibandingkan dengan total impresi campaign
$$\text{Jumlah Customer yang Merespon} / \text{Total Campaign}$$
- Revenue rate : Rasio jumlah keuntungan (profit) yang diperoleh perusahaan berdasarkan total response customer
$$[(\text{Revenue} * \text{Total Response}) - (\text{Cost} * \text{Total Campaign})] / \text{Total Revenue}$$

Stage 1 – EDA

Exploratory Data Analysis



Descriptive Analysis

1. Pada data terdapat tipe data tidak sesuai, yaitu kolom Dt_Customer
2. Terdapat kolom yang memiliki Missing Value yaitu Income sebanyak 24 value

```
# Check Missing Value
```

```
df.info()
```

```
df.isna().sum()
```

```
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)
```

```
ID 0
Year_Birth 0
Education 0
Marital_Status 0
Income 24
Kidhome 0
Teenhome 0
Dt_Customer 0
Recency 0
MntWines 0
MntFruits 0
MntMeatProducts 0
MntFishProducts 0
MntSweetProducts 0
MntGoldProds 0
NumDealsPurchases 0
NumWebPurchases 0
NumCatalogPurchases 0
NumStorePurchases 0
NumWebVisitsMonth 0
AcceptedCmp3 0
AcceptedCmp4 0
AcceptedCmp5 0
AcceptedCmp1 0
AcceptedCmp2 0
Complain 0
Z_CostContact 0
Z_Revenue 0
Response 0
dtypes: int64(1), float64(1), object(3)
```

```
[ ] # Check Duplicate
```

```
df.duplicated().sum()
```

```
0
```


Descriptive Analysis

3. Terdapat kolom dengan value yang perlu diperhatikan, yaitu kolom Marital_Status dan Education karena memiliki kategori terlalu bervariasi, selain itu pada kolom Year_Birth kurang valid untuk data minnya yang terlalu jauh.

Check Value for each Columns -- Check Invalid Data

```
for x in df.columns.to_list():
    print(x + '=')
    print(df[x].sort_values().unique())
    print('')
```

```
ID=
[ 0    1    9 ... 11187 11188 11191]

Year_Birth=
[1893 1899 1900 1940 1941 1943 1944 1945 1946 1947 1948 1949 1950 1951
 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965
 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979
 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993
 1994 1995 1996]

Education=
['2n Cycle' 'Basic' 'Graduation' 'Master' 'PhD']

Marital_Status=
['Absurd' 'Alone' 'Divorced' 'Married' 'Single' 'Together' 'Widow' 'YOLO']

Income=
[ 1730.   2447.   3502. ... 160803. 162397. 666666.]
```

```
[ ] #categorized each column based on num or cat data type, column that has only 1 unique value (Z_CostContact and Z_Revenue) is not categorized
cats = ["Education", "Marital_Status", "Kidhome", "Teenhome", "AcceptedCmp3", "AcceptedCmp4", "AcceptedCmp5",
        "AcceptedCmp1", "AcceptedCmp2", "Complain", "Response"]
num = ['ID', 'Year_Birth', 'Income', 'Recency', 'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts',
        'MntSweetProducts', 'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases',
        'NumStorePurchases', 'NumWebVisitsMonth']
product = ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']
purchase = ['NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases']
```

df[num].describe()

	ID	Year_Birth	Income	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	NumDe
count	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	
mean	5588.353339	1968.820397	52247.251354	49.012635	305.091606	26.356047	166.995939	37.637635	27.028881	43.965253	
std	3249.376275	11.985554	25173.076661	28.948352	337.327920	39.793917	224.283273	54.752082	41.072046	51.815414	
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	2814.750000	1959.000000	35303.000000	24.000000	24.000000	2.000000	16.000000	3.000000	1.000000	9.000000	
50%	5458.500000	1970.000000	51381.500000	49.000000	174.500000	8.000000	68.000000	12.000000	8.000000	24.500000	
75%	8421.750000	1977.000000	68522.000000	74.000000	505.000000	33.000000	232.250000	50.000000	33.000000	56.000000	
max	11191.000000	1996.000000	666666.000000	99.000000	1493.000000	199.000000	1725.000000	259.000000	262.000000	321.000000	

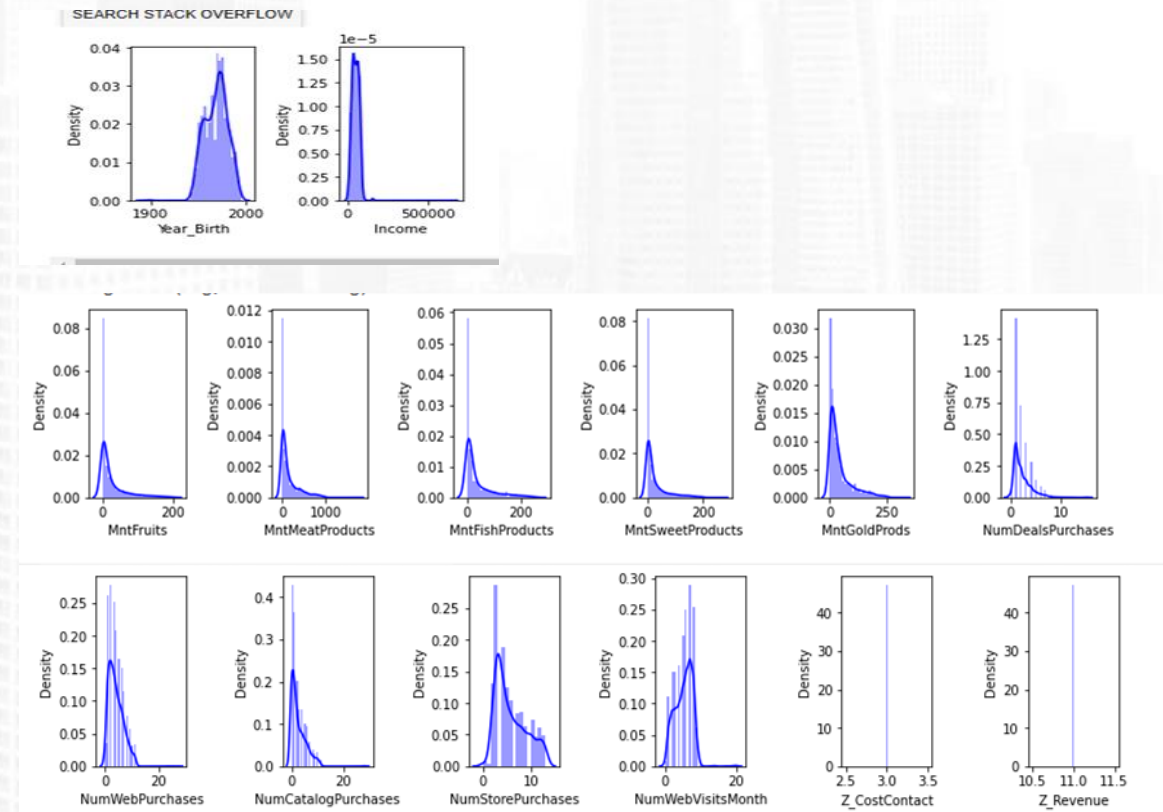
df[cats].describe()

	Education	Marital_Status	Kidhome	Teenhome	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Complain	Response
count	2216	2216	2216	2216	2216	2216	2216	2216	2216	2216	2216
unique	5	8	3	3	2	2	2	2	2	2	2
top	Graduation	Married	no kid	no teen	didn't accept	didn't accept	didn't accept	didn't accept	didn't accept	no complain	no respond
freq	1116	857	1283	1147	2053	2052	2054	2074	2186	2195	1883

Univariate Analysis

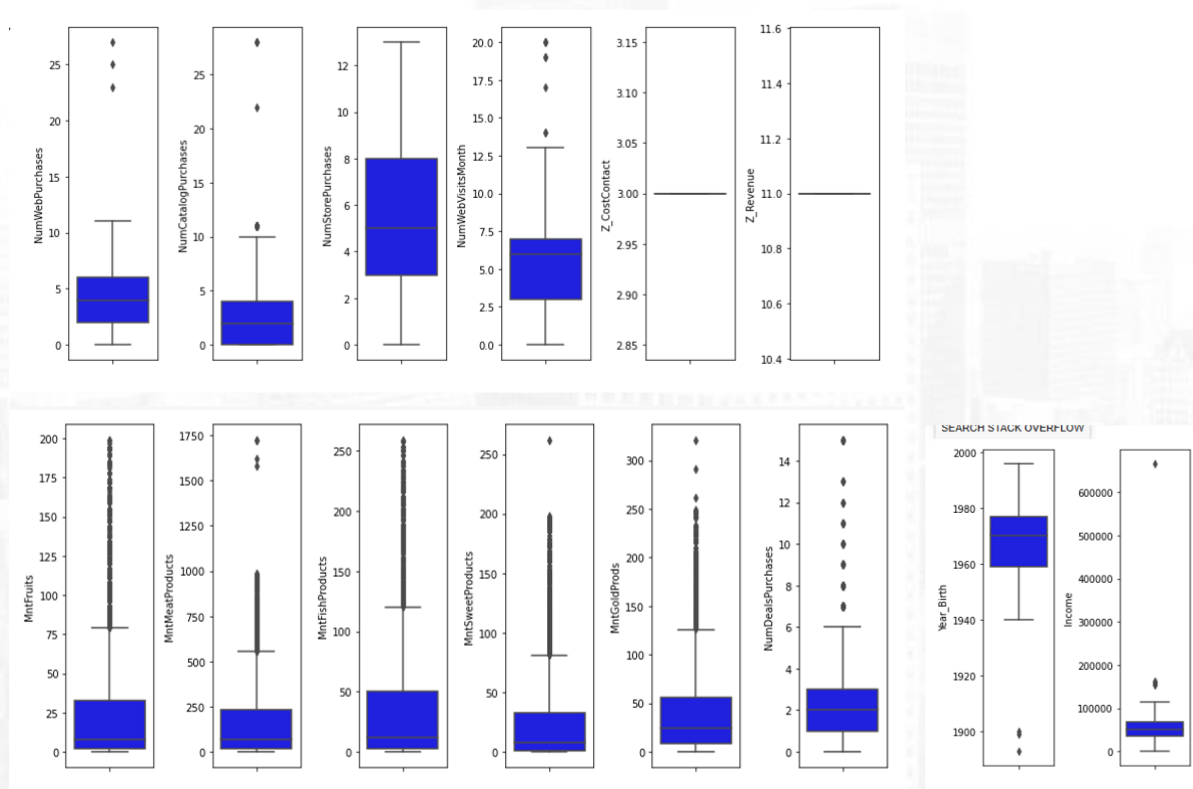
Berikut merupakan beberapa jenis distribusi data yang terdapat pada dataset:

1. Negative Skew : Year_Birth
2. Positive Skew : Income, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth,
3. Bimodal : Teenhome, Kidhome
4. Uniform : Recency



Untuk distribusi positive skew, di proses data pre-processing akan dilakukan log transformation. Untuk distribusi negative skew, di proses data pre-processing akan dilakukan standardization. Untuk distribusi

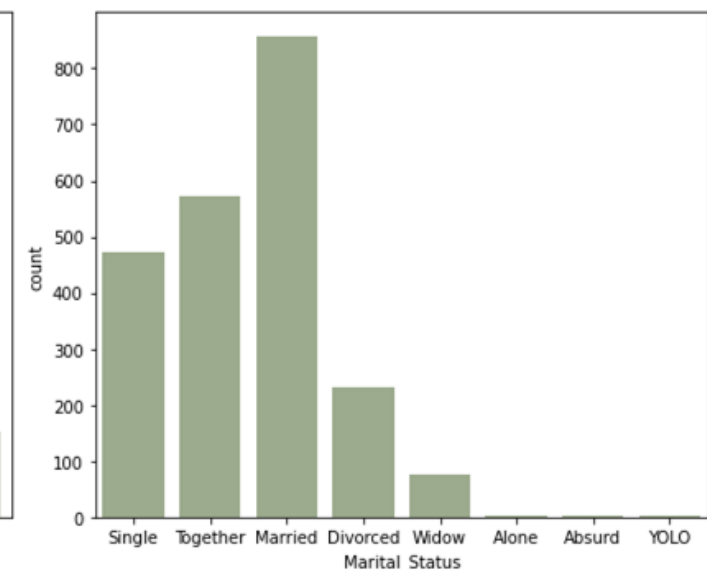
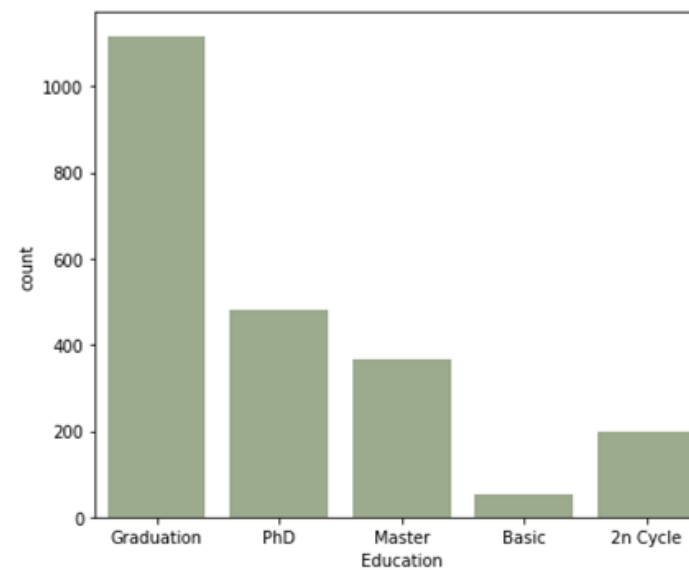
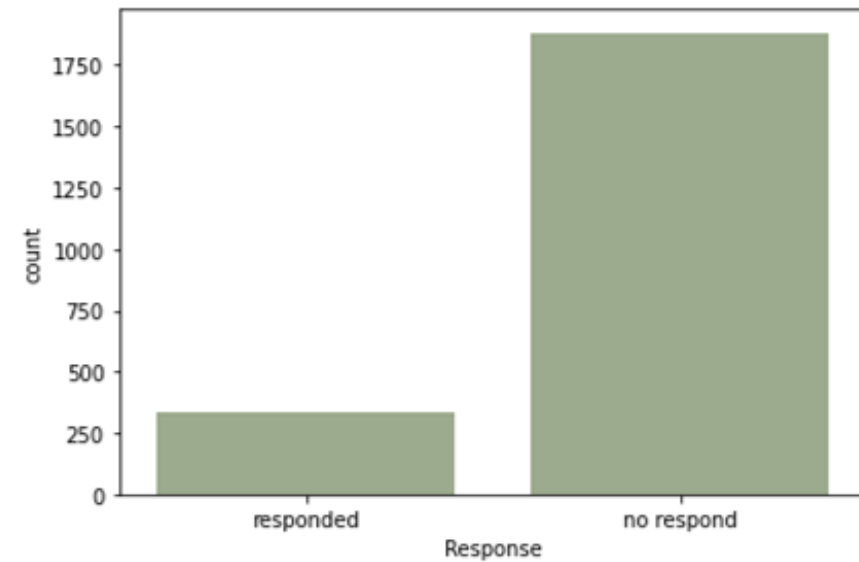
Univariate Analysis

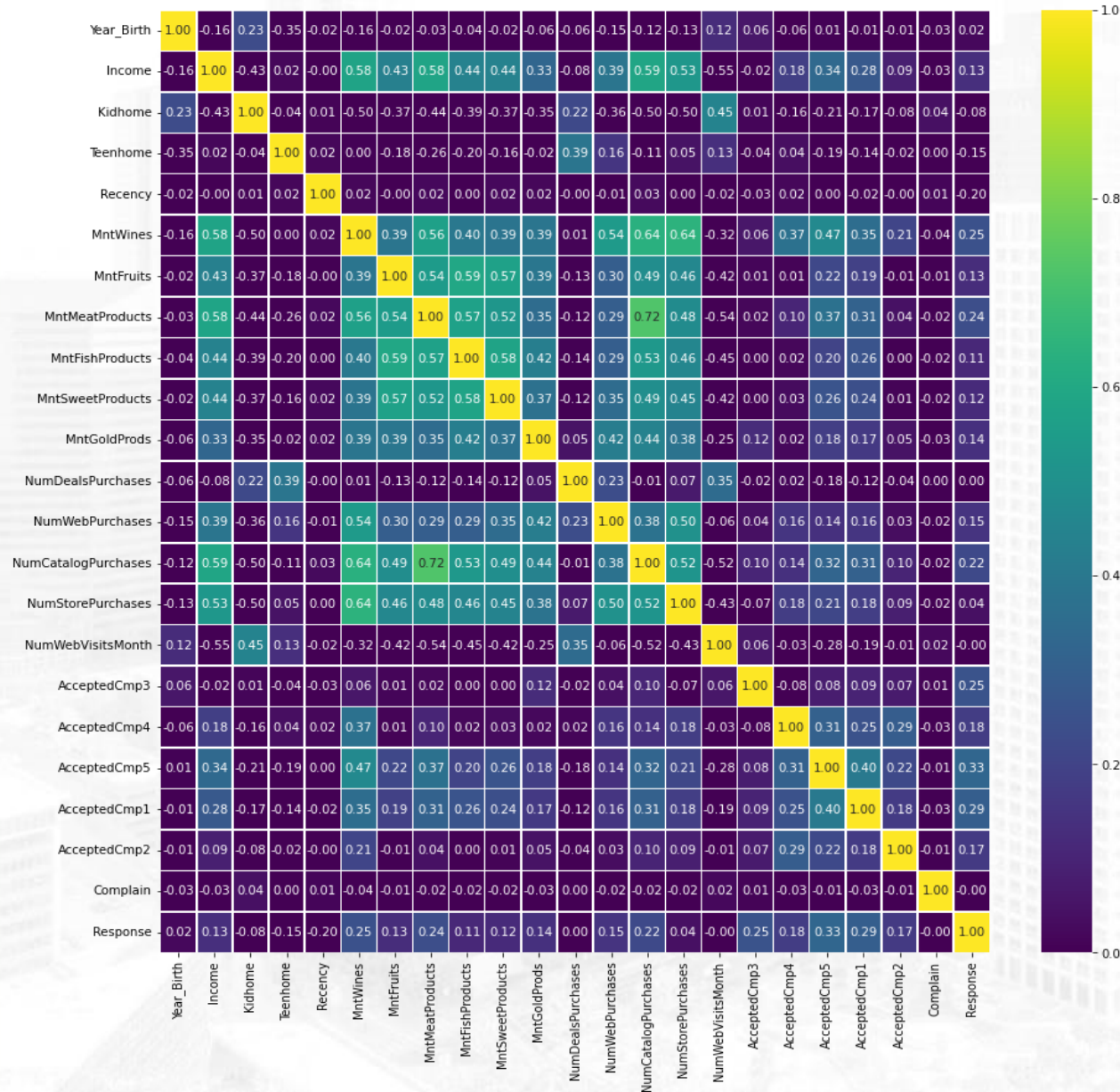


Kemudian, ditemukan outlier pada kolom Year_Birth, Income, Recency, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumWebVisitsMonth. Sehingga pada proses pre-processing, kolom yang memiliki outlier dapat dihapus berdasarkan z-score atau memakai IQR.

Univariate Analysis

Kemudian, untuk data target ditemukan bahwa terdapat ketimpangan sehingga pada pre-processing data dapat dilakukan over/under sampling.





Multivariate Analysis

Dari Heatmap diatas diketahui bahwa fitur yang memiliki korelasi yang cukup tinggi dengan response adalah income, mntwines, mnfruit, mntmeatproducts, mntfishproducts, mntsweetproducts, mntgoldprods, numwebpurchases, numcatalogpurchases dan acceptedcmp1-5

Dari seluruh korelasi antara feature-target, seluruhnya berada di range 0.00 sampai 0.33. Oleh karena itu, kami memutuskan untuk membuat nilai threshold di angka 0.20. Feature-feature di atas yang kami pertahankan adalah feature yang memiliki nilai korelasi >0.20.

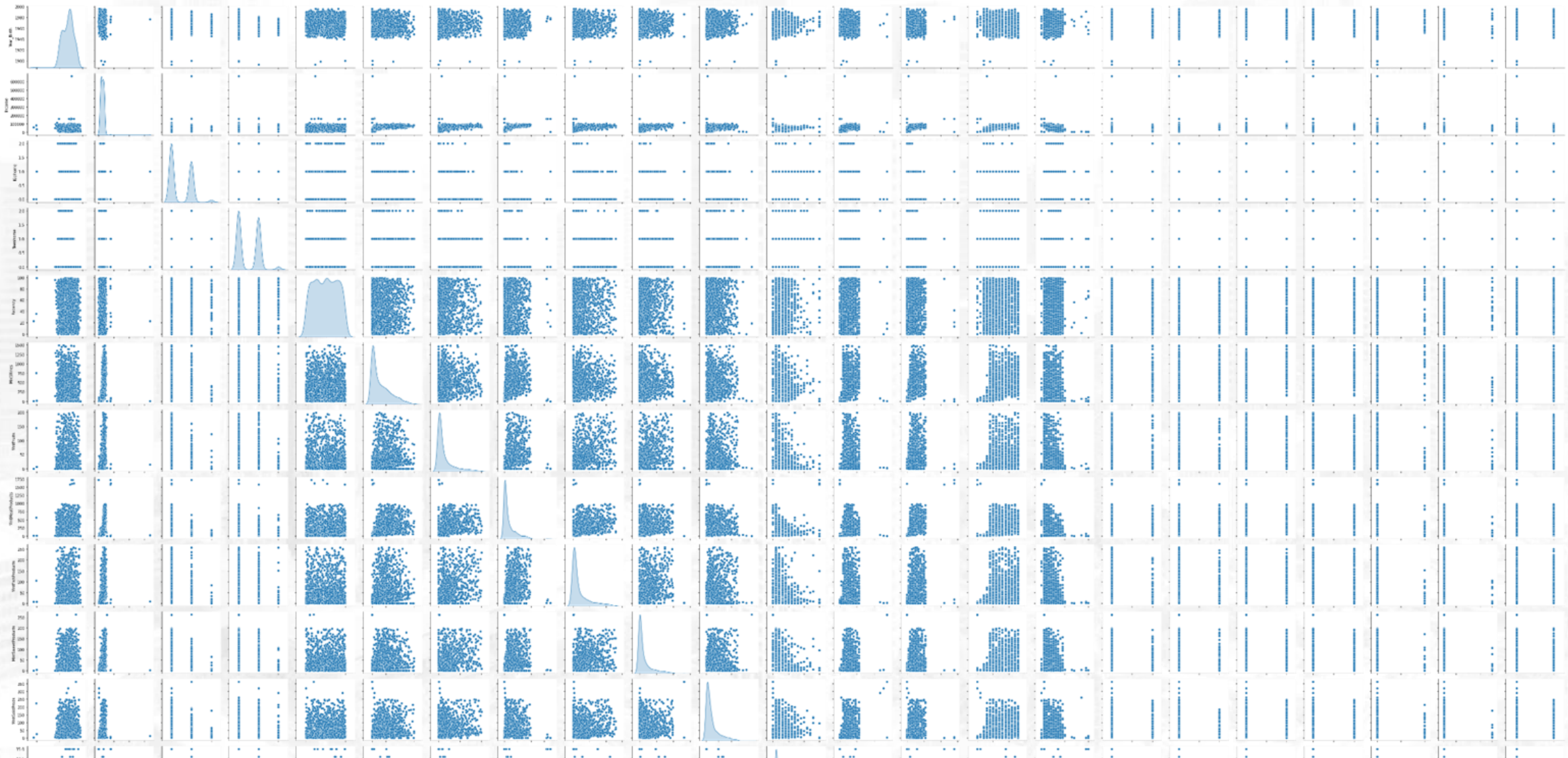
Multivariate Analysis

Berdasarkan analisa awal antar fitur yang kami lakukan terhadap fitur yang memiliki korelasi lebih tinggi dengan target, didapatkan hasil sebagai berikut:

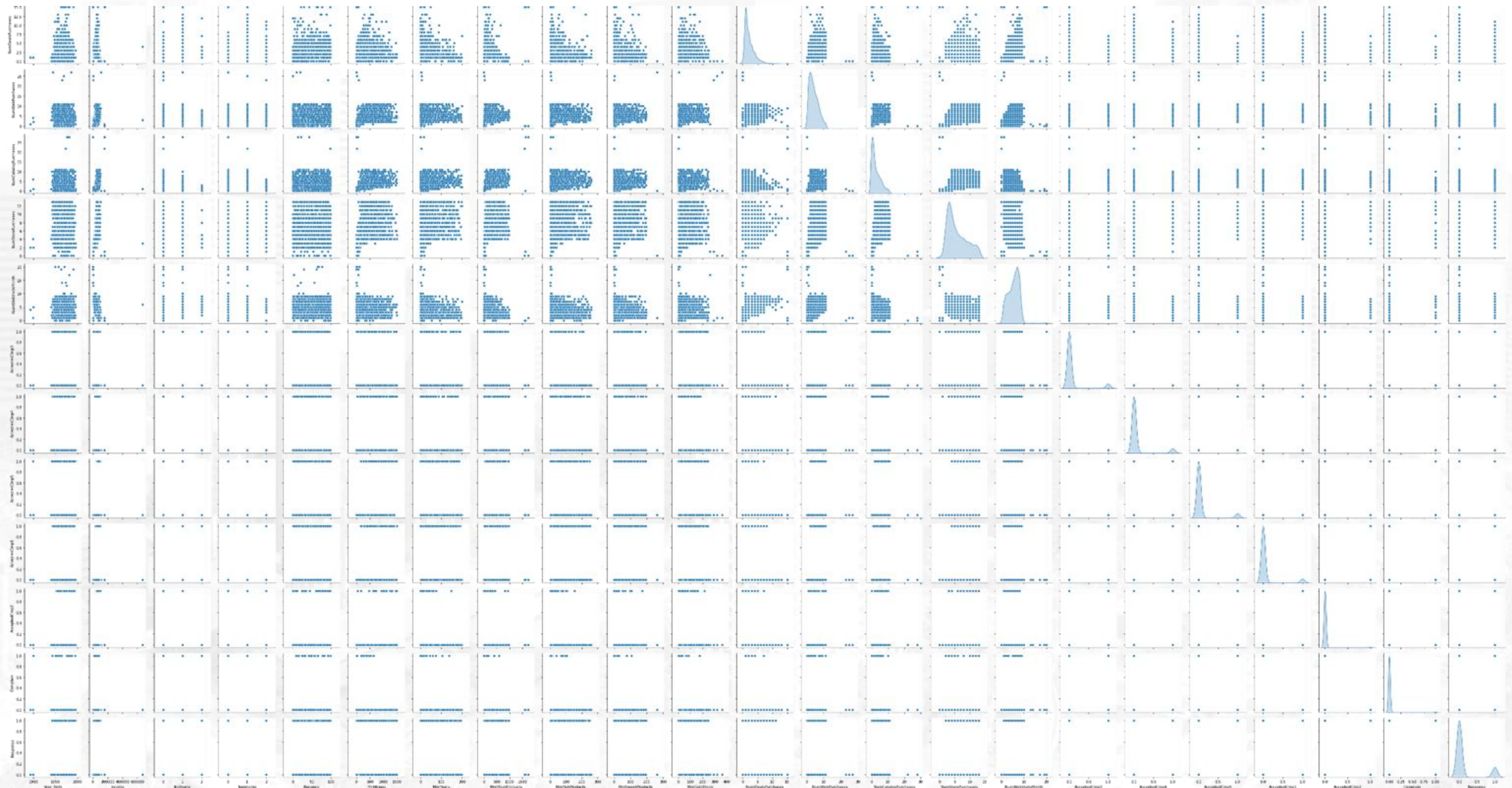
- [Recency]: Nilai korelasi Recency dengan feature lainnya memiliki range 0.00 sampai 0.05
- [MntWines]: Berikut adalah feature yang berkorelasi dengan MntWines: Income(0.58), NumCatalogPurchases(0.64), NumStorePurchases(0.64)
- [MntMeatProducts]: Berikut adalah feature yang berkorelasi dengan MntMeatProducts: NumCatalogPurchases(0.72), Income(0.58), MntWines(0.56)
- [NumCatalogPurchases]: Berikut adalah feature yang berkorelasi dengan NumCatalogPurchases: MntMeatProducts(0.72), MntWines(0.64), Income(0.59)
- [AcceptedCmp3]: Berikut adalah feature yang berkorelasi dengan AcceptedCmp3: MntGoldProducts(0.12)
- [AcceptedCmp5]: Berikut adalah feature yang berkorelasi dengan AcceptedCmp5: MntWines(0.47), MntMeatProducts(0.37), Income(0.34)
- [AcceptedCmp1]: Berikut adalah feature yang berkorelasi dengan AcceptedCmp1: AcceptedCmp5(0.40), MntWines(0.35), MntMeatProducts(0.31), NumCatalogPurchases(0.31)

Dari hasil tersebut, kemungkinan besar nantinya akan kami gunakan sebagai fitur prioritas dalam keputusan penentuan indikator pendukung untuk pengkategorisasian customer mana yang layak diberikan campaign.

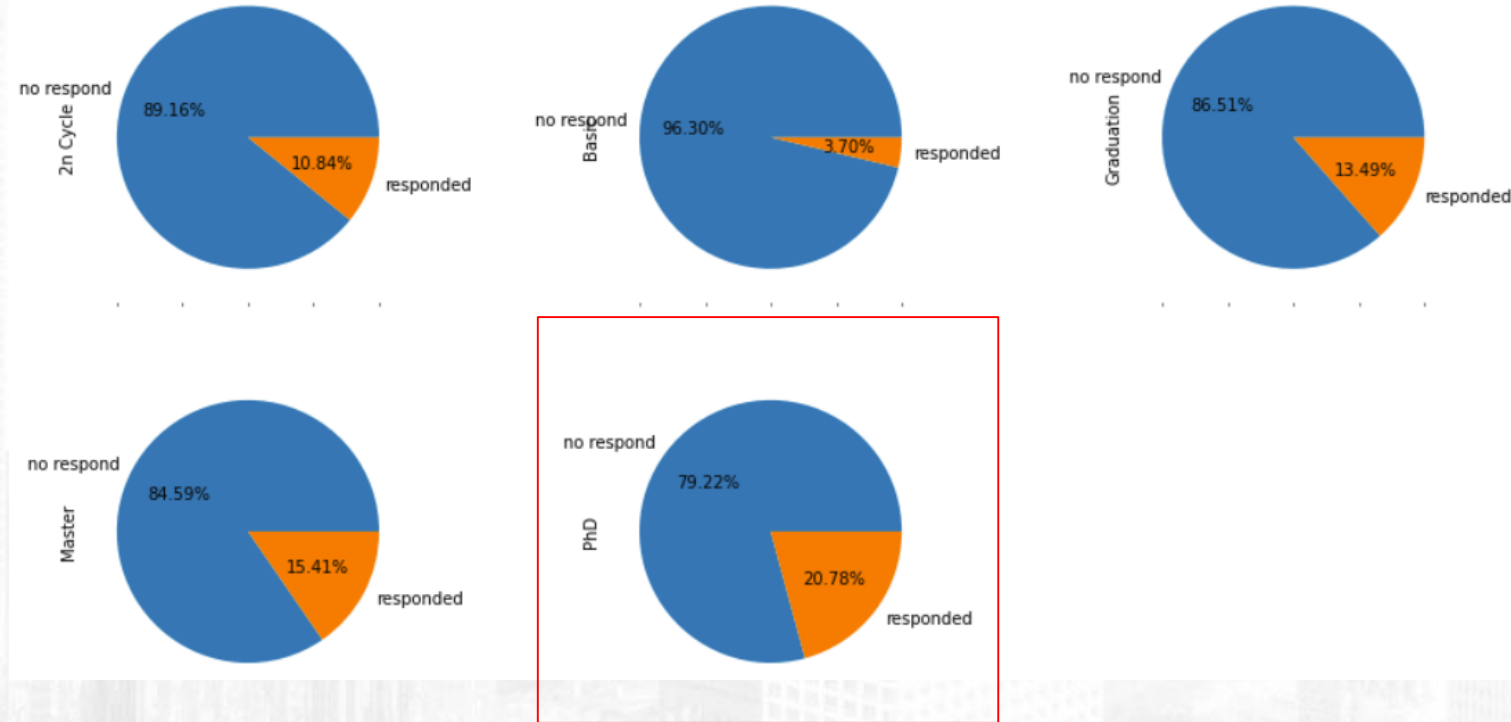
Multivariate Analysis



Multivariate Analysis

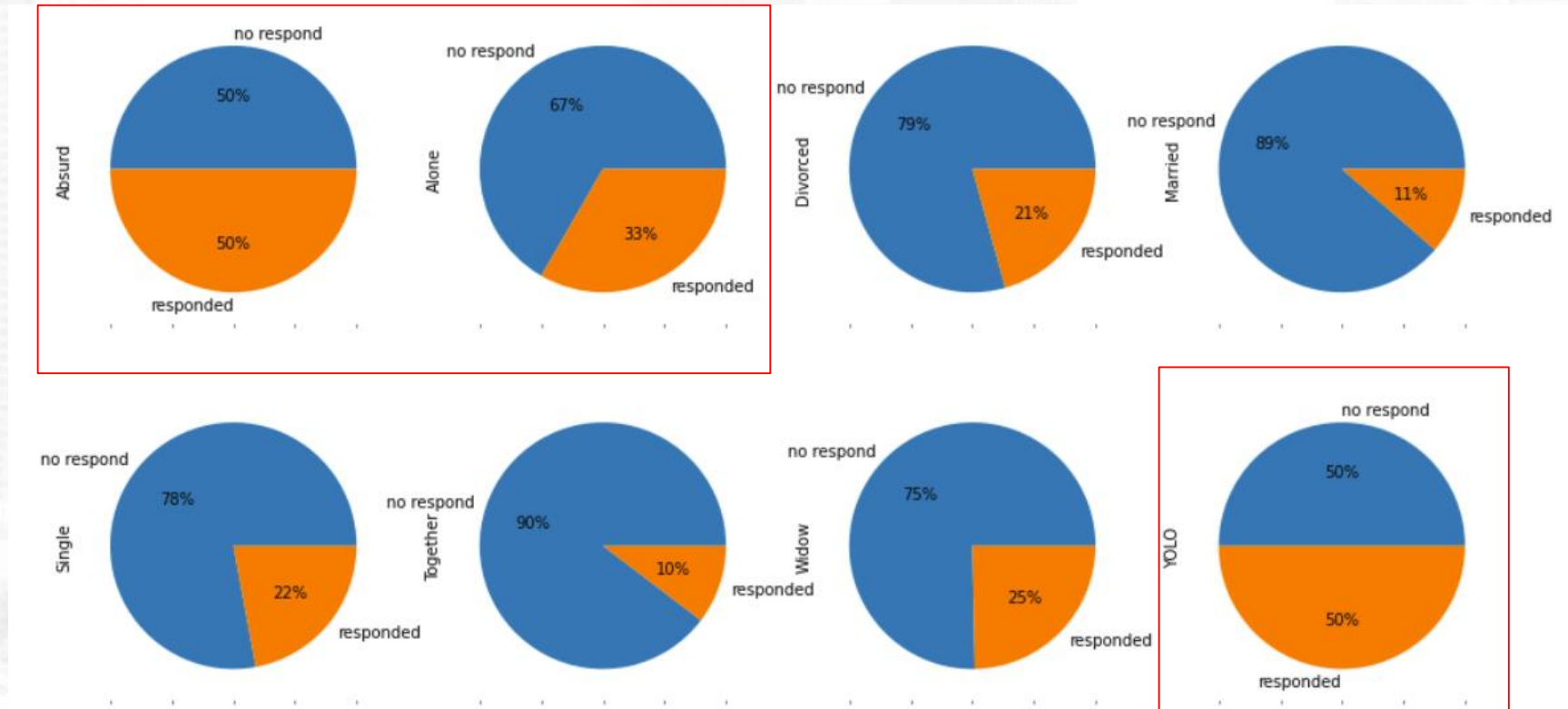


Business Insight



Dari visualisasi Juga dapat dilihat bahwa customer yang merespon terbanyak berasal dari customer yang memiliki edukasi PhD (20.78%), disusul dengan Master

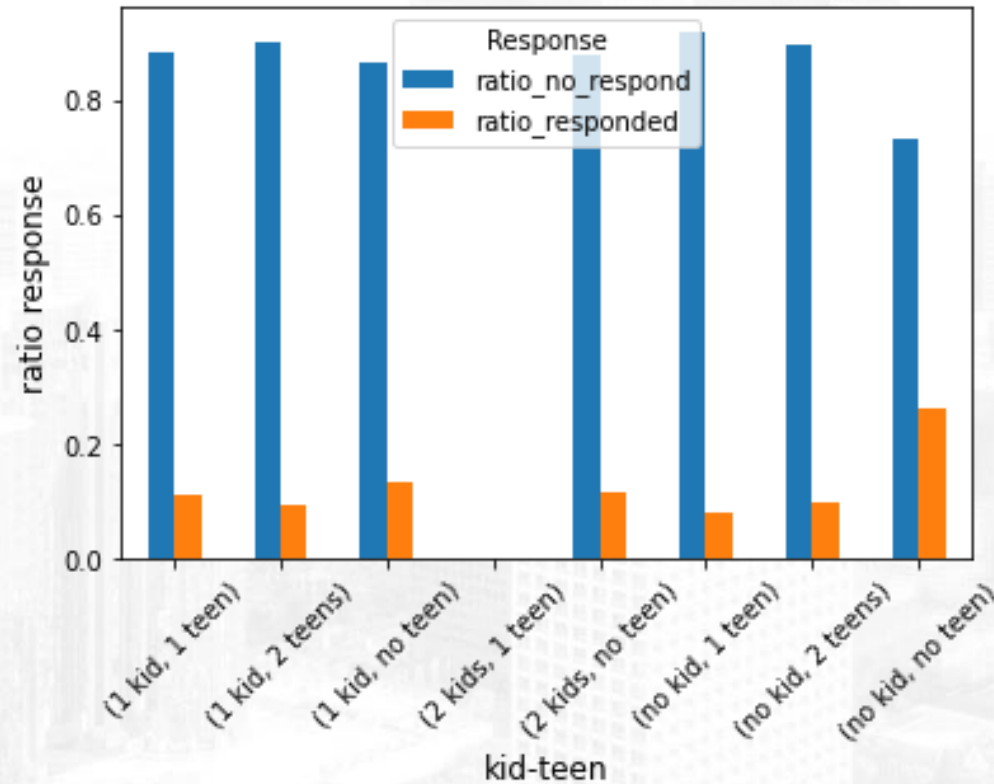
Business Insight



Dari visualisasi piechart Marital Status, dapat dilihat bahwa customer yang merespon terbanyak berasal dari customer yang berstatus Absurd (50%) dan Yolo(50%), disusul dengan Alone(33%) dan Widow (25%), sehingga marketing team dapat memfokuskan campaign ke customer "Absurd" dan "Yolo".

Business Insight

Perbandingan ratio response dengan Customer yang memiliki Kid-Teen



Dari visualisasi kidhome dan teenhome, dapat dilihat bahwa customer yang merespon terbanyak berasal dari customer yang tidak mempunyai anak dan tidak mempunyai remaja (0.265403), sehingga marketing team dapat memfokuskan campaign ke customer yang tidak mempunyai anak dan tidak mempunyai remaja.

Next Improvement

Selain dari tiga business insight tersebut, kami juga memiliki satu buah insight lagi berupa sebuah trend suatu product yang memiliki korelasi kuat (Gold, Meat, dan Wines) terhadap campaign 1 sampai dengan campaign 5. Kemudian, hasil dari visualisasi insight tersebut nantinya dapat digunakan oleh perusahaan untuk memprioritaskan produk mana yang akan dijual atau dipromosikan guna menarik jumlah customer. Sehingga diharapkan dengan adanya kenaikan jumlah customer tersebut, jumlah revenue perusahaan pun dapat bertambah

Namun, dikarenakan keterbatasan waktu dan skill, kami belum sempat membuat visualisasi insight terakhir tersebut dan berencana untuk menjadikannya sebagai salah satu next improvement.

Stage 2

Data Pre-Processing



Data Pre Processing

Data Cleansing - Handle missing values

Melakukan pengecekan data, terdapat missing values pada kolom income sebanyak 24. Dengan adanya missing values tersebut, dilakukan penghapusan (Drop), karena jumlah missing values income kurang dari 10%.

```
# check missing value
```

```
df.isna().sum()
```

ID	0
Year_Birth	0
Education	0
Marital_Status	0
Income	24

```
#deleting rows with missing value  
df.dropna(inplace = True)  
df.info()
```

Data Cleansing - Handle duplicated data

Tidak terdapat data duplikat, setelah melakukan pengecekan.

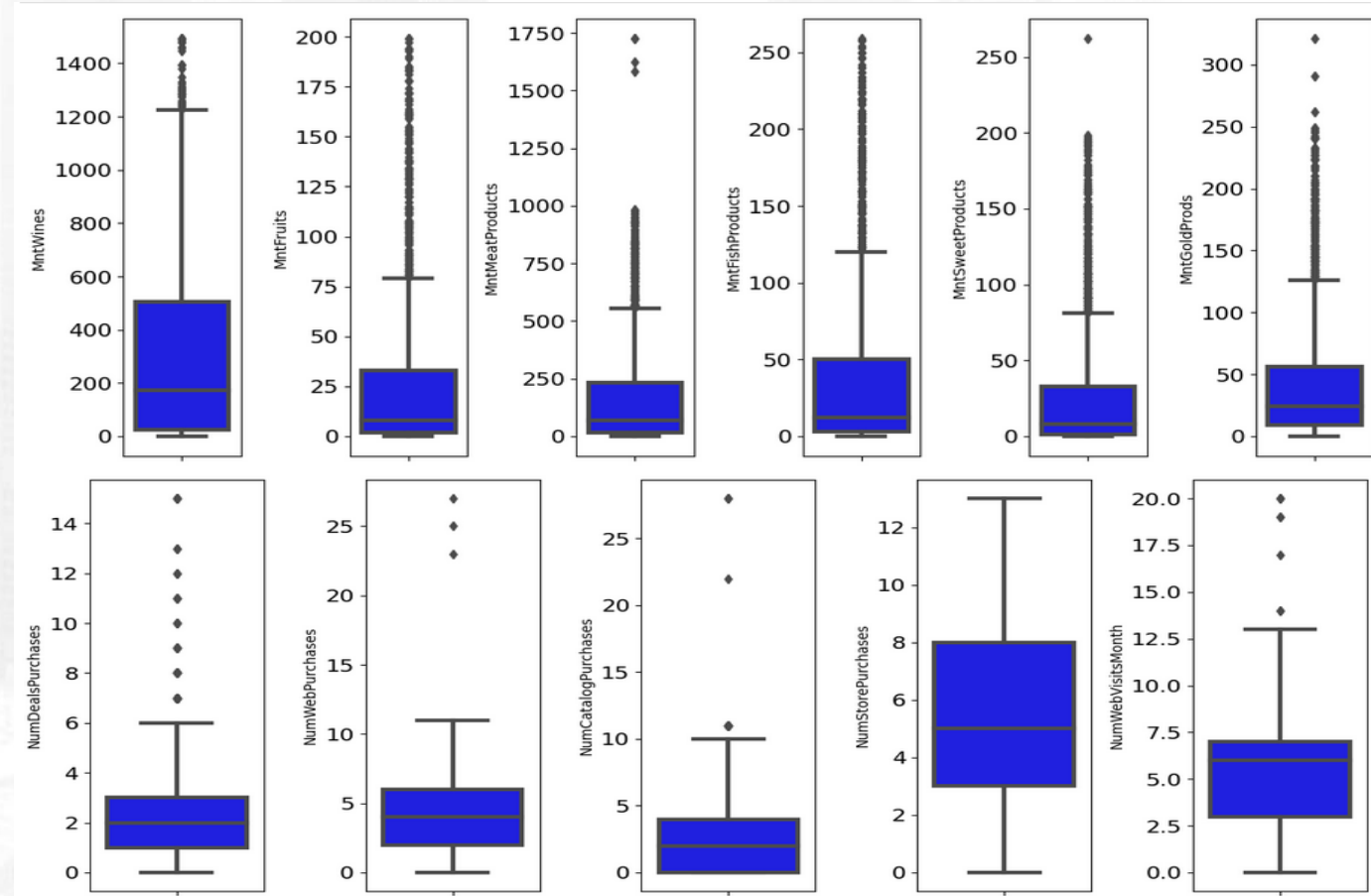
```
#check any duplicated  
df.duplicated().any()
```

```
False
```

Data Pre Processing

Data Cleansing - Handle outliers

Melakukan pengecekan apakah ada outlier pada data kolom Year_Birth, Income, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumWebVisitsMont.



Data Pre Processing

Data Cleansing - Handle outliers

Dari pengecekan tersebut terdapat outlier pada beberapa kolom tersebut. Sehingga untuk menghilangkan outlier tersebut, digunakan Z-Score, karena dengan menggunakan Z-Score untuk handle outlier data yang dihilangkan tidak lebih dari 30%.

Remove Outliers berdasarkan Z-score

```
print(f'jumlah baris sebelum memfilter outlier: {len(df)}')

filtered_entries = np.array([True] * len(df))

for col in ['Income', 'Year_Birth', 'Recency', 'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
            'MntTobacco']:
    zscore = abs(stats.zscore(df[col]))
    filtered_entries = (zscore < 3) & filtered_entries

df = df[filtered_entries]

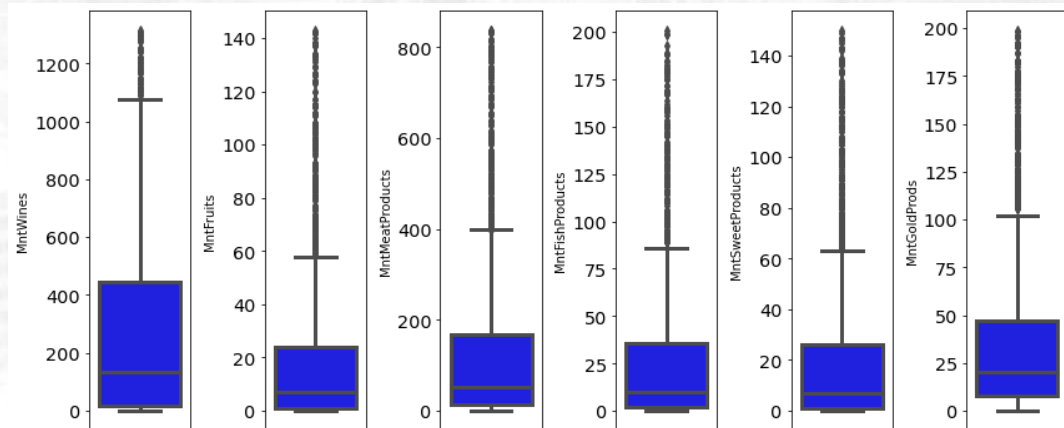
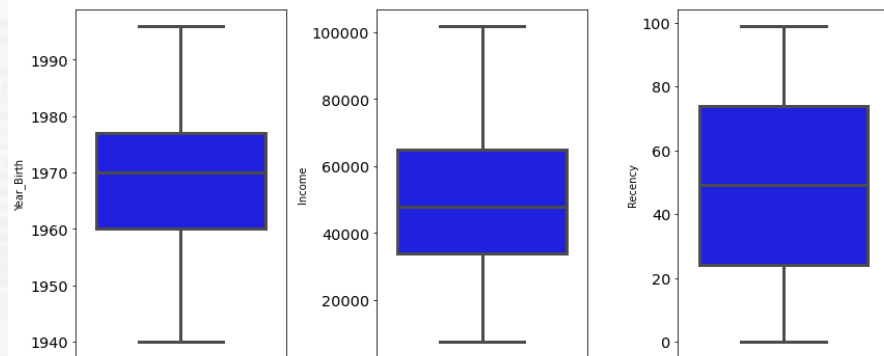
print(f'jumlah baris sesudah memfilter outlier: {len(df)}')
```

< >

jumlah baris sebelum memfilter outlier: 2216
jumlah baris sesudah memfilter outlier: 1953

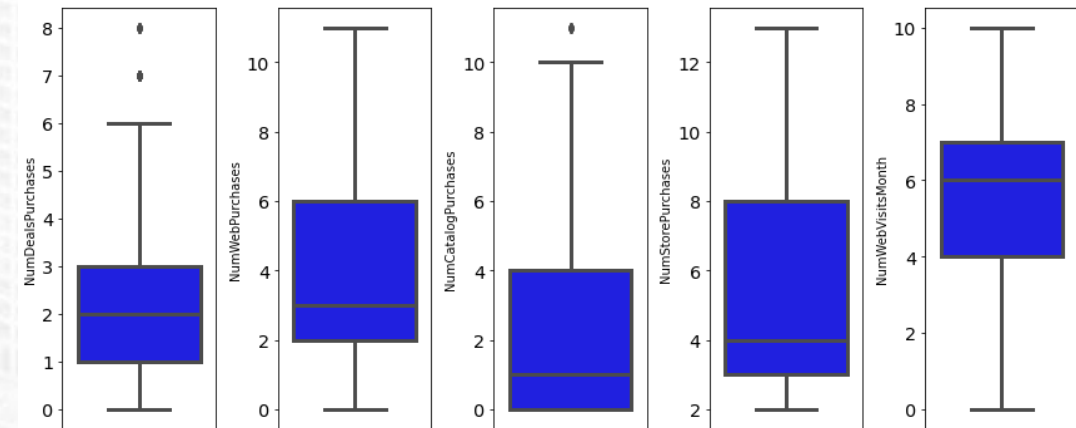
Data Pre Processing

GAMBAR BOXPLOT YG SUDAH TIDAK ADA OUTLIER



Data Cleansing - Handle outliers

Dari gambar disamping diketahui bahwa outlier sudah tidak ada pada setiap data.



Data Pre Processing

Data Cleansing - Feature transformation I

Pada feature transformation, dilakukan log transformation untuk memperkecil range pada beberapa kolom MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases

Menggunakan Code

```
# log transformation for positively skewed features
plt.figure(figsize=(20,20))
for i in range(0, len(positive_skewed)):
    plt.subplot(6, 2, i+1)
    sns.kdeplot(np.log(df[positive_skewed[i]]), color='blue')
plt.tight_layout()
```

Before

```
positive_skewed = ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds',
                  'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases']

# check skewness value
for i in positive_skewed:
    skewness = df[i].skew(axis=0, skipna=True)
    print('skewness feature', i, 'adalah', skewness)
```

```
skewness feature MntWines adalah 1.2704786643137906
skewness feature MntFruits adalah 2.108011346690099
skewness feature MntMeatProducts adalah 1.9170849507890875
skewness feature MntFishProducts adalah 1.965486056831826
skewness feature MntSweetProducts adalah 2.101542394722874
skewness feature MntGoldProds adalah 1.758032451602574
skewness feature NumDealsPurchases adalah 1.31591642250104
skewness feature NumWebPurchases adalah 0.7869758319537213
skewness feature NumCatalogPurchases adalah 1.348368752280952
```

After

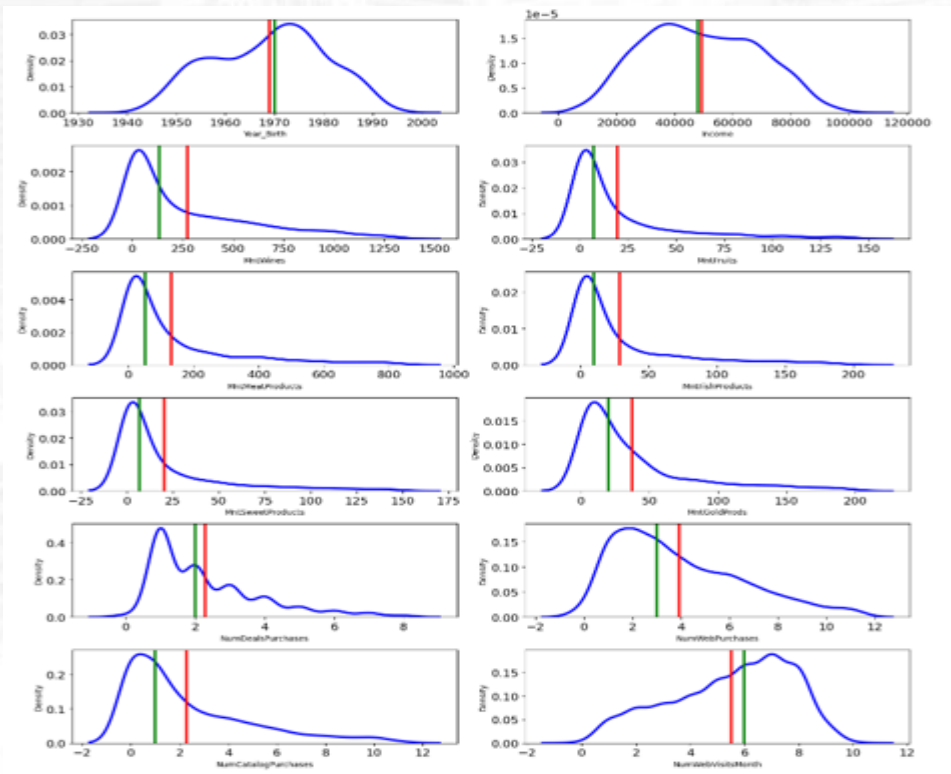
```
[ ] # check skewness value
for i in positive_skewed:
    skewness = np.log(df[i].skew(axis=0, skipna=True))
    print('skewness feature', i, 'adalah', skewness)
```

```
skewness feature MntWines adalah 0.2393937304956276
skewness feature MntFruits adalah 0.745745013344942
skewness feature MntMeatProducts adalah 0.6508057775256199
skewness feature MntFishProducts adalah 0.6757395718998083
skewness feature MntSweetProducts adalah 0.7426715488131432
skewness feature MntGoldProds adalah 0.5641952584797426
skewness feature NumDealsPurchases adalah 0.27453332214854864
skewness feature NumWebPurchases adalah -0.2395577401166722
skewness feature NumCatalogPurchases adalah 0.2988955301853037
```

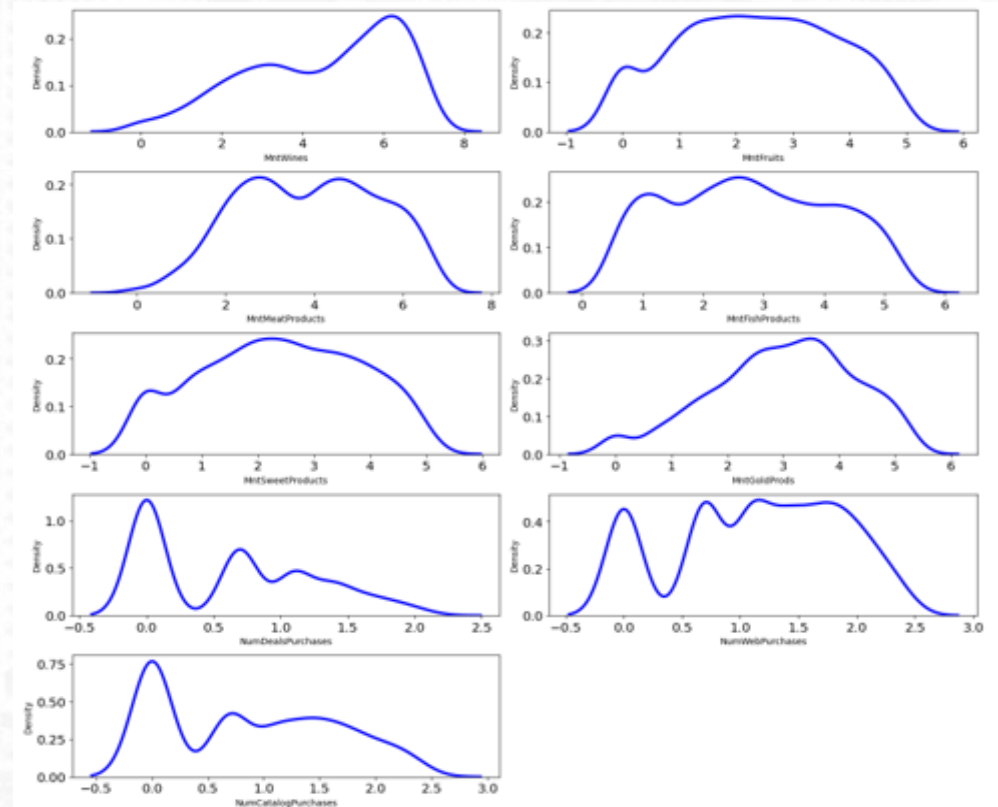
Data Pre Processing

Feature transformation

Sebelum dilakukan log transformation, hampir setiap kolom memiliki range data yg besar, seperti pada gambar dibawah ini.



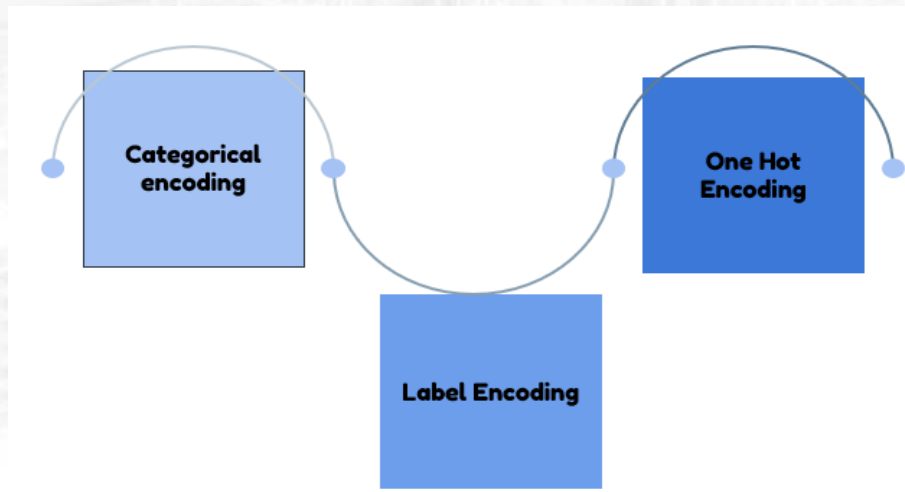
Setelah dilakukan log transformation, setiap kolom memiliki range data yg kecil, seperti pada gambar dibawah ini.



Data Pre Processing

Data Cleansing - Feature encoding

Feature encoding dilakukan mulai dari categorical encoding, label encoding dan one hot encoding. Kolom marital status dan education akan dilakukan feture encoding dengan mengubah data categorical menjadi numerical.



Code feture encoding Marital_Status dan Education

```

▶ mapping_marital = {
    'Absurd' : 0,
    'Alone' : 0,
    'Divorced' : 0,
    'Single' : 0,
    'Widow' : 0,
    'YOLO' : 0,
    'Together' : 1,
    'Married' : 1
}
df['Marital_Status'] = df['Marital_Status'].map(mapping_marital)
df.head()
  
```

```

▶ mapping_education = {
    'Basic' : 0,
    'Graduation' : 1,
    'Master' : 2,
    '2n Cycle' : 2,
    'PhD' : 3
}
df['Education'] = df['Education'].map(mapping_education)
df
  
```

```

▶ #menggabungkan value pada kolom education
df.Education = df.Education.apply(lambda x : "Master" if (x=="2n Cycle") else x)

#One Hot Encoding
prefix_educ = pd.get_dummies(df['Education'], prefix='is')

df = df.join(prefix_educ)
df
  
```

Data Pre Processing Feature Extraction

Pada feature extraction, terdapat 9 feature yaitu:

1. **primer_purchase & tersier_purchase**, yaitu sebuah fitur yang menggabungkan kolom product purchases ke dalam 2 golongan, yaitu primer dan tersier.
2. **total_accepted_campaign**, yaitu sebuah fitur yang menggabungkan acceptedcmp 1 - 5. Fitur ini dibuat untuk melihat intensitas customer dalam accepting campaign
3. **total_revenue**, yaitu sebuah fitur yang dibuat dengan menjumlahkan total acceptance customer pada keseluruhan campaign dengan jumlah revenue per accepted campaign
4. **total_spent**, yaitu sebuah fitur yang menggabungkan total pembelian pada keseluruhan produk untuk merekap total pengeluaran yang telah dilakukan
5. **total_order**, yaitu sebuah fitur yang berisikan summary dari total purchases atau order yang telah dilakukan oleh pelanggan dari berbagai metode purchases
6. **month_customer**, yaitu fitur bulan dimana customer mulai enroll/ register ke marketing campaign
7. **age**, yaitu sebuah fitur yang mengkategorisasikan customer ke dalam 3 kelompok umur, yaitu: Elderly (2), Middle Age (1), dan Young (0)
8. **income_category**, yaitu sebuah fitur yang mengkategorisasikan customer berdasarkan pendapatannya ke dalam 3 kategori, yaitu High-Income (2), Mid-Income (1), dan Low-Income (0)
9. **total_dependents**, yaitu sebuah fitur yang menggabungkan kolom marital status, kidhome, dan teenhome untuk melihat jumlah orang dalam 1 rumah yang dianggap sebagai tanggungan rumah tangga.

- Code untuk Feature Extraction

```
[ ] # total revenue
df['total_revenue'] = (df['AcceptedCmp1'] + df['AcceptedCmp2'] + df['AcceptedCmp3'] +
                      df['AcceptedCmp4'] + df['AcceptedCmp5']) * df['Z_Revenue']
df[['Z_Revenue', 'total_revenue']].sample(5)
```

```
▶ # jumlah tanggungan
df['total_dependents'] = df['Marital_Status'] + df['Kidhome'] + df['Teenhome']
df.sample(5)
```

```
▶ # primer and tertier product
df['primer_purchase'] = df['MntFruits'] + df['MntMeatProducts'] + df['MntFishProducts']
df['tersier_purchase'] = df['MntWines'] + df['MntSweetProducts'] + df['MntGoldProds']
df.sample(5)
```

```
▶ # convert the date of enrolment to datetime
df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'])

# creating features from date of enrolment
df['month_customer'] = df['Dt_Customer'].apply(lambda x: x.month)

# Check the result
df.sample(5)
```

```
▶ # total spent
df["total_spent"] = df["MntWines"] + df["MntFruits"] + df["MntMeatProducts"] + df["MntFishProducts"] + df["M"]
df.sample(5)
```

```
[ ] # total accepted campaign
df['total_accepted_campaign'] = df['AcceptedCmp1'] + df['AcceptedCmp2'] + df['AcceptedCmp3'] + df['AcceptedCmp4'] +
df.sample(5)
```

```
▶ # Age_category customer menurut WHO
df['age'] = 2023 - df['Year_Birth']

age_category=[]
for i in df['age']:
    if i <= 25 :
        age_category.append(0) #Young
    elif i <=45 :
        age_category.append(1) #Middle-Age
    else :
        age_category.append(2) #Elderly
df['age_category'] = age_category
df.head()
```

```
▶ # Income
Income_category=[]
for i in df['Income']:
    if i >= df['Income'].quantile(0.75) :
        Income_category.append(2) # High-Income
    elif i >= df['Income'].quantile(0.50) :
        Income_category.append(1) # Mid-Income
    else :
        Income_category.append(0) # Low-Income
df['Income_category'] = Income_category
df.head()
```

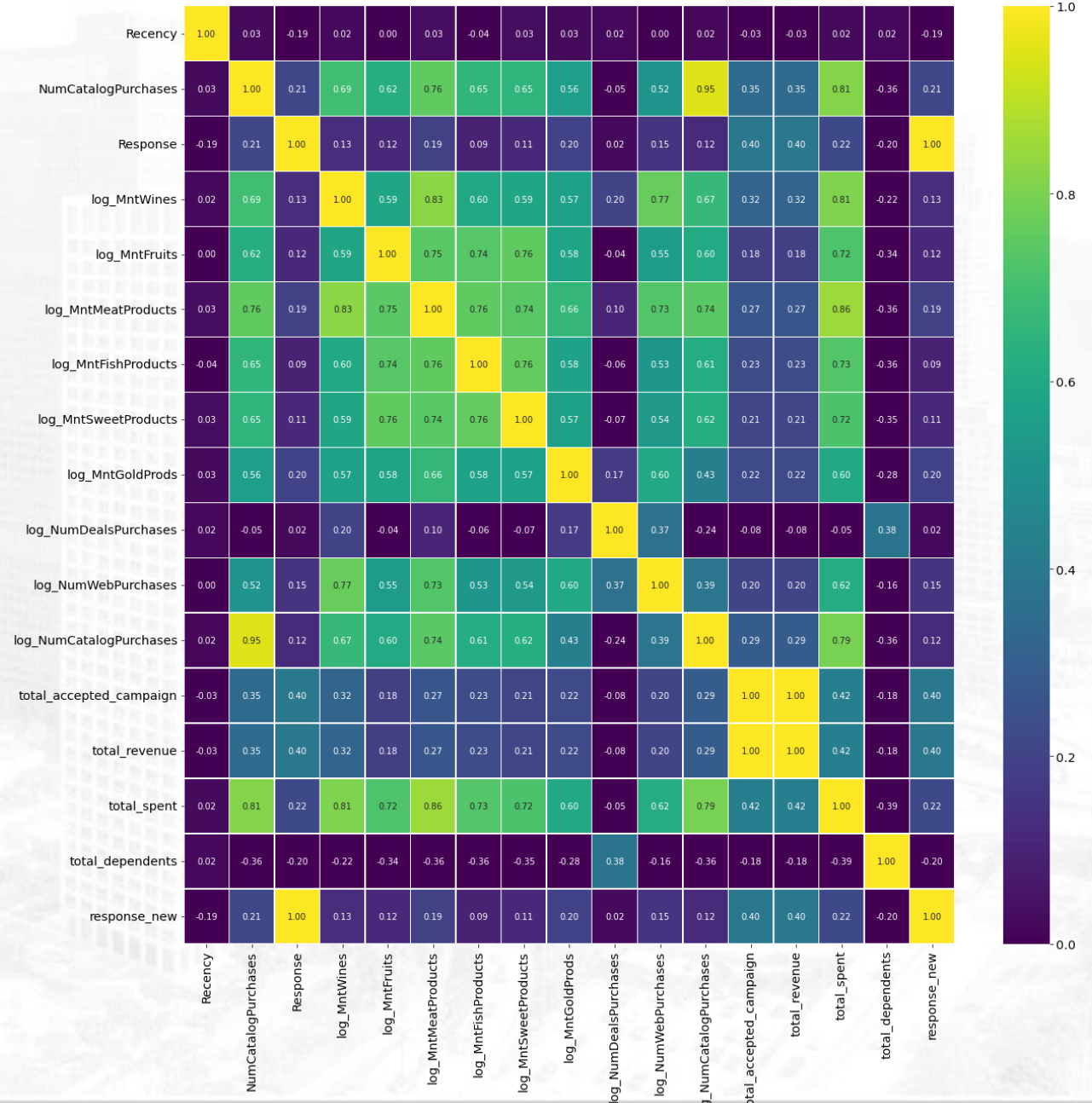

Data Pre Processing

Feature Selection

Pada feature selection semakin banyak feature akan semakin memberatkan Machine Learning. Feature yang akan dipertahankan adalah feature yang korelasinya $> 0,19$ dengan respons

Terdapat beberapa feature yg akan dipertahankan.

1. Nilai korelasi Recency dengan respons adalah 0,19
2. Nilai korelasi Num Catalog Purchases dengan respons adalah 0,21
3. Nilai korelasi Total accepted Campaign dengan respons adalah 0,4
4. Nilai korelasi Total Revenue dengan respons adalah 0,4
5. Nilai korelasi Total Spent dengan respons adalah 0,22
6. Nilai korelasi Family Size dengan respons adalah 0,2



Data Pre Processing - Class Imbalance

Degree Imbalance tergolong Moderate. Sehingga kami Handling dengan Oversampling, karena jumlah sample yang dipelajari oleh Model akan lebih banyak.

```
[ ] #Ratio Check for target
df_response = df.groupby('Response').agg({'ID':'count'}).reset_index().rename(columns={'ID':'Jumlah'})
df_response['Ratio'] = df_response['Jumlah']*100/df_response['Jumlah'].sum()
df_response

#degree of imbalance = moderate
```

	Response	Jumlah	Ratio
0	0	1692	86.635945
1	1	261	13.364055

```
[ ] df['response_new'] = df['Response'] > 0.8 #split dataset
print(df['response_new'].value_counts())
```

```
False    1692
True      261
Name: response_new, dtype: int64
```

```
[ ] x = df[[col for col in df.columns if col not in ['response_new', 'Response']]].values
y = df['response_new'].values
print(x.shape)
print(y.shape)
```

```
(1953, 52)
(1953,)
```

```
[ ] #use oversampling
from imblearn import under_sampling, over_sampling
#x_under, y_under = under_sampling.RandomUnderSampler(sampling_strategy=1).fit_resample(x,y)
x_over, y_over = over_sampling.RandomOverSampler().fit_resample(x,y)
#x_over_SMOTE, y_over_SMOTE = over_sampling.SMOTE().fit_resample(x,y)
```

Output

```
[ ] print(pd.Series(y).value_counts())
print(pd.Series(y_over).value_counts())
#print(pd.Series(y_under).value_counts())
```

```
False    1692
True      261
dtype: int64
True      1692
False    1692
dtype: int64
```

Feature Tambahan

1. Area/ Region

Lokasi tempat tinggal customer dapat mempengaruhi tingkat respon customer terhadap pembelian barang. Semakin dekat tempat tinggal mereka dengan pusat kota, kemungkinan semakin sedikit yang merespon dikarenakan banyaknya kompetisi campaign dari market lainnya di sekitar kota.

2. Time call

Waktu ketika ditelepon: pada saat jam kerja atau jam istirahat.

3. Day call

Hari ketika ditelepon: weekend/ weekday.

4. Payment method

Metode pembayaran yang digunakan untuk membeli barang: credit card/ COD/ Bank transfer/ e-money. Customer yang memakai metode credit card, kemungkinan tingkat respon dapat lebih tinggi dari pada metode pembayaran lainnya.

5. Job position

Jenis pekerjaan customer dapat mempengaruhi tingkat respon campaign: student/ professional/ unemployed.

Stage 3

Modeling



Modelling Experiments

Sebelum dilakukan modelling, kami melakukan split data terlebih dahulu untuk memisahkan antara data train (70%) dan data test (30%).

```
[ ] X = df_for_corr.drop(['Response'], axis=1)
    y = df_for_corr['Response']
```

```
df_for_corr['Response'].value_counts(normalize = True)
0    0.866359
1    0.133641
Name: Response, dtype: float64
```

```
[ ] # split train and test

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
```

Kemudian dikarenakan terdapat class imbalance yang tergolong intermediate pada data target, maka kami memutuskan untuk melakukan oversampling agar algoritma atau model ML yang kami ciptakan dapat mempelajari data dengan lebih seimbang.

```
[ ] #use oversampling
from imblearn import under_sampling, over_sampling
#x_under, y_under = under_sampling.RandomUnderSampler(sampling_strategy=1).fit_resample(x,y)
#X_train, y_train = over_sampling.RandomOverSampler(sampling_strategy=0.5).fit_resample(X_train,y_train)
X_train, y_train = over_sampling.SMOTE(sampling_strategy=0.5).fit_resample(X_train,y_train)
```

Modelling Experiments

Pemilihan dan Perhitungan Metrics

1. Precision

Metrics model ini kami pilih sebagai primary metrics. Hal ini dikarenakan keinginan kami untuk memprioritaskan peningkatan dari primary business metrics, yaitu response rate demi meningkatkan efisiensi cost advertising. Sehingga, kami memerlukan sebuah primary metrics yang dapat mereduksi False Positive atau customer yang diprediksi akan mengikuti campaign namun pada kenyataannya tidak mengikuti. Berikut merupakan rumus perhitungan dari precision

$$\textit{Precision} = \frac{TP}{TP + FP}$$

2. Recall

Selain memprioritaskan peningkatan response rate, kami juga mempertimbangkan pengoptimalan revenue rate perusahaan. Sehingga kami memerlukan metrics model yang dapat mereduksi False Negative atau customer yang diprediksi tidak akan mengikuti campaign namun pada kenyataannya berkeinginan mengikuti. Maka dari itu, kami memilih recall sebagai secondary metrics model. Berikut merupakan rumus perhitungannya

$$\textit{Recall} = \frac{TP}{TP + FN}$$

Modeling Experiments

Pada tahap ini kami melakukan eksplorasi menggunakan seluruh algoritma yang telah diajarkan sebelumnya dengan hasil sebagai berikut :

Algoritma	Precision Train	Precision Test	Recall Train	Recall Test
Logistic Regression	0.62	0.37	0.31	0.37
KNN	0.77	0.29	0.73	0.38
Decision Tree	1.00	0.44	0.98	0.62
Random Forest	0.99	0.55	0.99	0.62
AdaBoost	0.80	0.43	0.73	0.54
XGBoost	0.90	0.51	0.85	0.55

Berdasarkan hasil tersebut, maka kami memutuskan untuk melakukan eksplorasi lebih dalam terhadap 3 algoritma dengan hasil *precision* dan *recall* terbaik, yaitu Decision Tree, Random Forest dan XGBoost.

Modeling Experiments

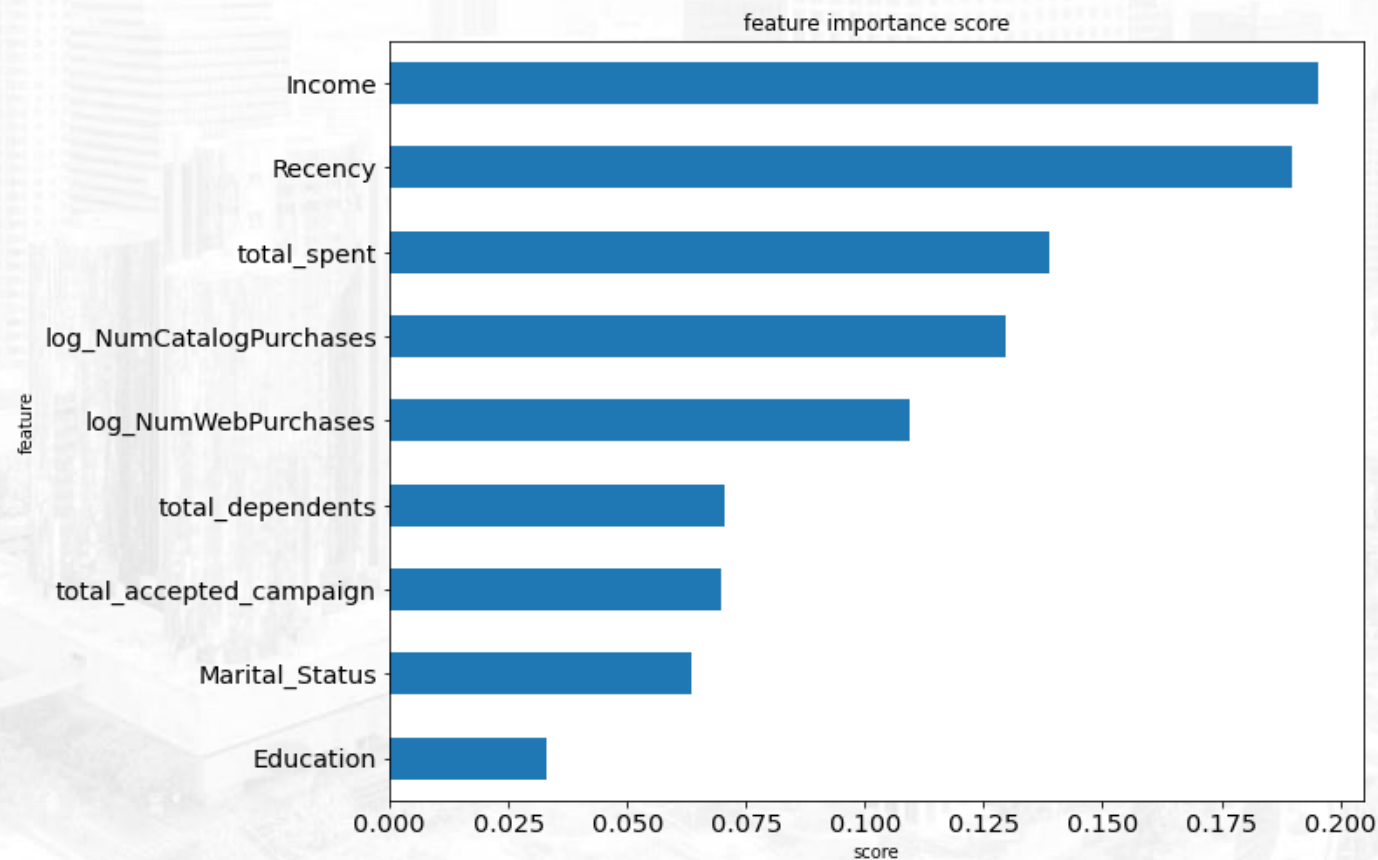
Kemudian, setelah Decision Tree, Random Forest dan XGBoost telah dilakukan eksplorasi lebih lanjut dengan menggunakan Hyperparameter Tuning, berikut merupakan rekap performa akhir yang didapatkan:

Algoritma	Precision Train	Precision Test	Recall Train	Recall Test	ROC Crossval Train	ROC Crossval Test	Performa Model
Decision Tree	0.80	0.51	0.29	0.31	0.76	0.74	Best Fit
Random Forest	0.84	0.58	0.34	0.33	0.82	0.80	Best Fit
XGBoost	0.83	0.58	0.27	0.27	0.71	0.70	Best Fit

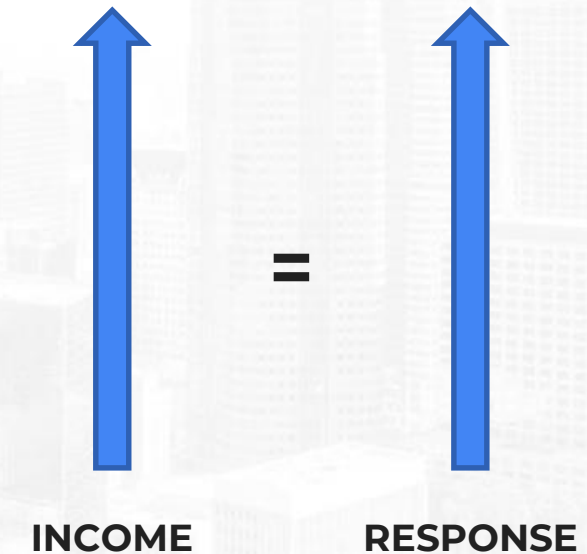
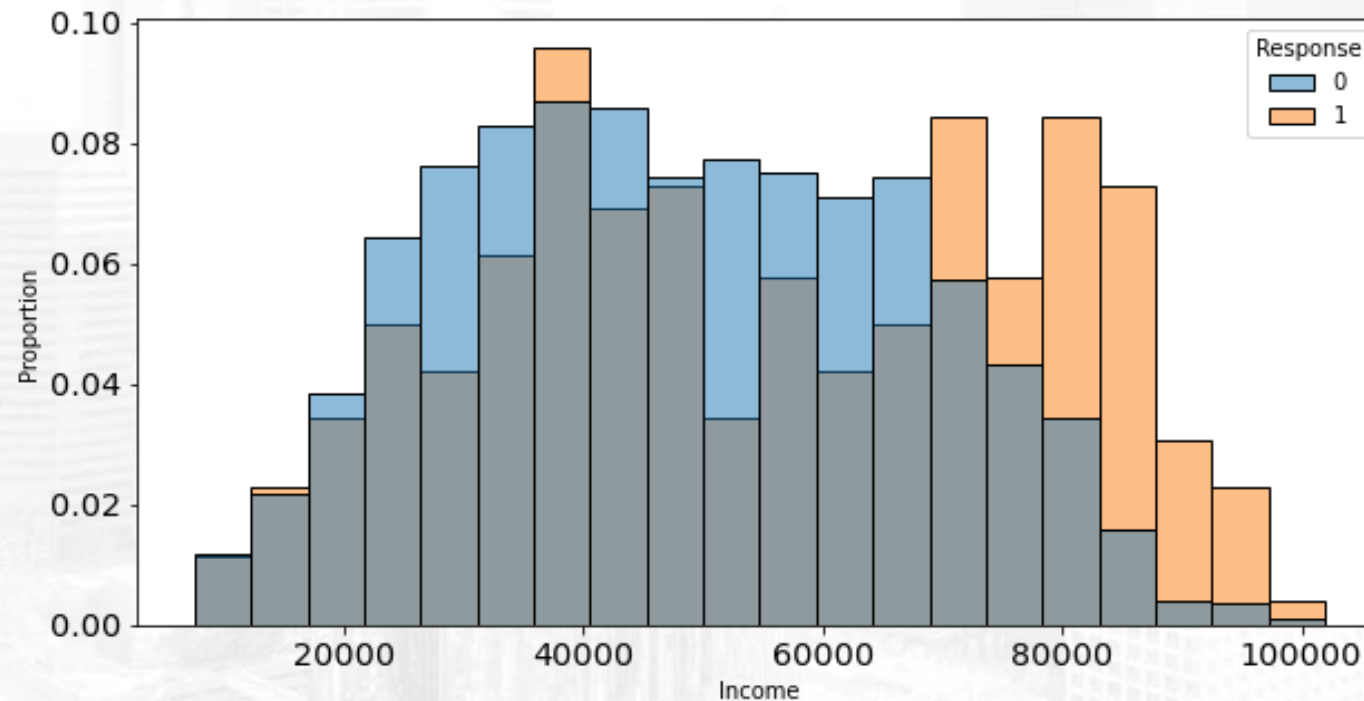
Berdasarkan hasil pertimbangan nilai precision train, recall serta gap train-test yang didapatkan pada ketiga model tersebut, maka dapat diambil kesimpulan bahwa model paling optimal yang kami dapatkan adalah **Random Forest**. Hal ini dikarenakan model tersebut menghasilkan output dengan nilai metrics precision dan juga recall tertinggi. Selain itu, model ini juga cenderung **best fit** sehingga layak untuk dipilih.

Executive Summary and Recommendation

Dari hasil pemodelan yang telah dipilih sebelumnya, diperoleh feature importance sebagai berikut yang mempengaruhi apakah pelanggan akan meresponse campaign atau tidak.

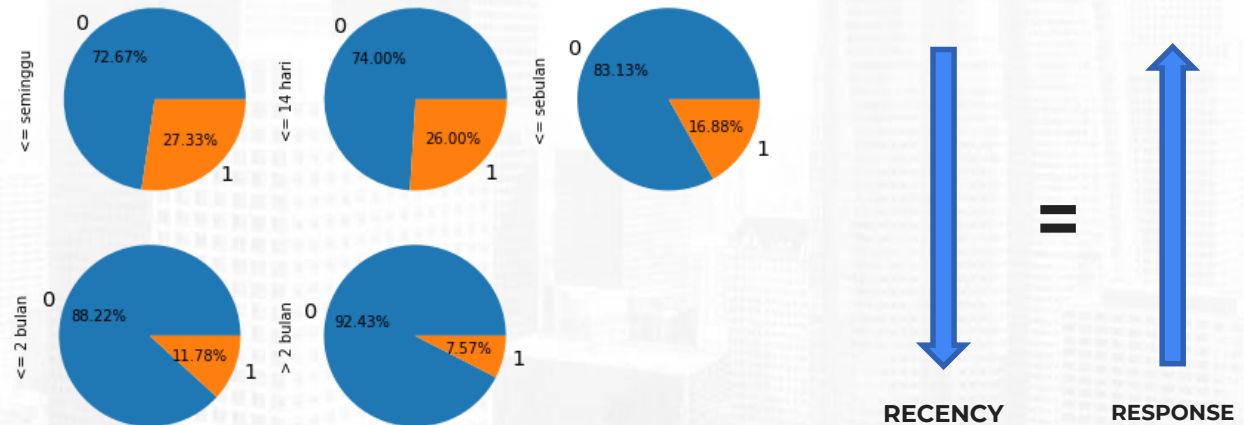
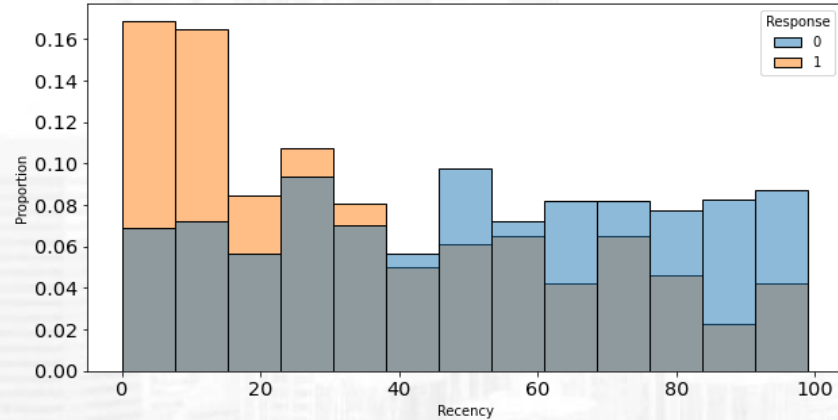


Executive Summary and Recommendation



Berdasarkan hasil visualisasi yang telah ditampilkan di atas, maka kami dapat merekomendasikan perusahaan untuk lebih memfokuskan campaign kepada pelanggan yg memiliki income diatas 70000 untuk meningkatkan campaign response. Kemudian untuk meningkatkan traffic jumlah customer dengan income diatas 70000 tersebut, perusahaan dapat menerapkan sistem loyalty berupa poin khusus bagi customer pengguna kartu debit / credit gold & platinum yang nantinya dapat ditukarkan dengan beberapa reward menarik berupa diskon ataupun pemberian produk secara gratis.

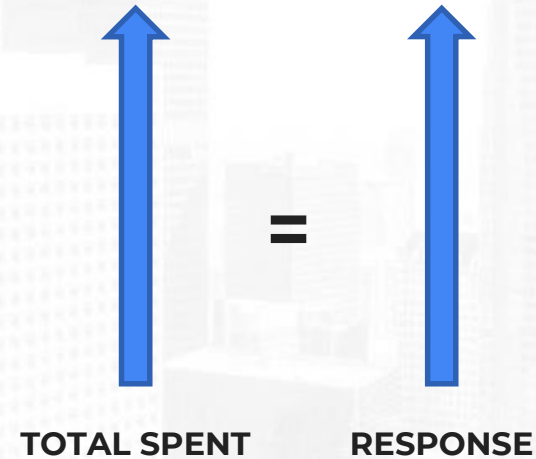
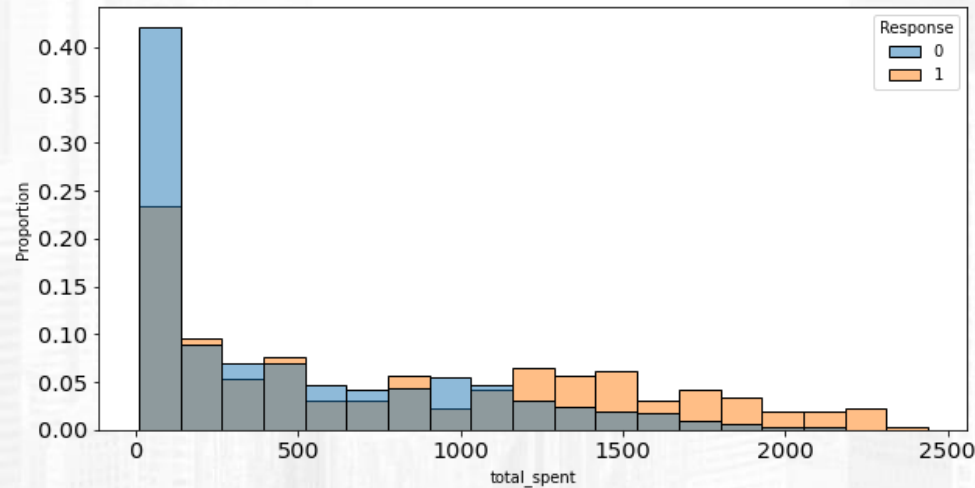
Executive Summary and Recommendation



Berdasarkan hasil visualisasi di atas, maka kedepannya perusahaan bisa memberikan lebih memfokuskan campaign kepada pelanggan yg memiliki kriteria melakukan pembelian terakhir pada waktu terdekat dengan campaign yang akan dilaksanakan.

Kemudian, perusahaan kedepannya juga harus memiliki concern lebih untuk menjangkau customer-customer yang sudah lama tidak belanja ataupun yang belum pernah belanja dengan memberikan promo khusus bisa berupa voucher diskon pengguna baru, voucher diskon pengguna lama, dsb. Selain itu perusahaan juga dapat memberikan promosi langsung yang dilaksanakan secara bulanan (seperti 2.2, 3.3, dst.) untuk menjaga traffic pelanggan dengan membangun habit "belanja bulanan". Dan terakhir, perusahaan juga dapat melakukan pemasangan e-billboard, ataupun ads lain yang secara umum bertujuan untuk menarik atensi pengguna baru.

Executive Summary and Recommendation



Berdasarkan hasil visualisasi di atas, maka kedepannya perusahaan dapat memberikan campaign khusus kepada pelanggan yg memiliki kriteria total spent terbanyak sebelum periode campaign terbaru yang akan dilaksanakan.

Kemudian, perusahaan juga perlu meningkatkan total spent pelanggan dengan cara meningkatkan dan menjaga kualitas dari produk yg dijual, misalnya dengan berusaha untuk selalu menawarkan produk fresh pada jenis buah-buahan, ikan, dan daging. Selain itu, perusahaan juga dapat memberikan promo khusus dengan menggunakan strategi "up-selling / cross-selling" yang menawarkan diskon tambahan pada produk penunjang sesuai dengan barang yang dibeli oleh pelanggan. Dan terakhir, perusahaan juga dapat memberikan voucher khusus bagi pelanggan dengan syarat klaim berupa total min. belanja sekian dollar.

Pembagian Tugas

Nama	Kontribusi
Jesslyn Jane	Ketua <ul style="list-style-type: none"> - Bersinergi dengan Wakil untuk mengkoordinasikan tim dalam menyelesaikan weekly homework dan juga Final Project - Melakukan QC terhadap source code dan juga file Presentasi
Aditya Ridwan	Wakil Ketua <ul style="list-style-type: none"> - Bersinergi dengan Ketua untuk mengkoordinasikan tim dalam menyelesaikan weekly homework dan juga Final Project - Melakukan QC terhadap isi laporan Final Project dan juga file weekly homework
Ann Sinaga	Notulen <ul style="list-style-type: none"> - Bertanggungjawab untuk membuat notulensi baik dalam meeting internal ataupun mentoring bersama Mentor - Bertanggungjawab untuk membuat PPT Final Presentasi
Lhutfia Ichsan	<ul style="list-style-type: none"> - Bertanggungjawab untuk membuat Laporan Final Project - Bertanggungjawab untuk mencari inspirasi/materi tambahan untuk anggota tim
Mustiadi Zakki	<ul style="list-style-type: none"> - Bertanggungjawab untuk membuat Laporan Final Project - Bertanggungjawab untuk mencari inspirasi/materi tambahan untuk anggota tim
M Triargi	<ul style="list-style-type: none"> - Bertanggungjawab untuk mengelola arsip file yang digunakan dalam Final Project oleh Tim - Bertanggungjawab untuk mengelola GitHub Kelompok dan File collab Final
Dian Rahayu	((RESIGN))
Irvandri	((RESIGN))

[Link GDrive - Halcyon](#)

[Link Github - Halcyon](#)