

Predict Customer Personality to boost marketing campaign by using Machine Learning

Supported by:
Rakamin Academy
Career Acceleration School
www.rakamin.com



Created by:

Rheza Paleva Uyanto

uyantorheza@gmail.com

<https://www.linkedin.com/in/rheza-uyanto/>

Lulus dari Program Pendidikan Profesi Apoteker Universitas Surabaya pada tahun 2017, memiliki pengalaman praktik kefarmasian di Rumah Sakit selama 4 tahun. Kemampuan komunikasi dan managerial yang baik, dan mampu bekerja sama dalam tim ataupun secara mandiri. Sangat termotivasi dalam bidang mentoring dan pengembangan diri. Mampu dalam penggunaan Microsoft Office. Memiliki ketertarikan dalam bidang Data Analitik, sehingga mengikuti Rakamin Bootcamp Data Science Batch 24. Melalui portofolio ini, saya akan memprediksi karakteristik customer untuk meningkatkan marketing campaign menggunakan Machine Learning.

Overview

“Sebuah perusahaan dapat berkembang dengan pesat saat mengetahui perilaku customer personality nya, sehingga dapat memberikan layanan serta manfaat lebih baik kepada customers yang berpotensi menjadi loyal customers. Dengan mengolah data historical marketing campaign guna menaikkan performa dan menyasar customers yang tepat agar dapat bertransaksi di platform perusahaan, dari insight data tersebut fokus kita adalah membuat sebuah model prediksi kluster sehingga memudahkan perusahaan dalam membuat keputusan ”

Conversion Rate Analysis Based on Income, Spending and Age

- Dataset : marketing_campaign_data.csv
- Dataset terdiri dari 29 kolom, dan 2240 baris.

```
df.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2240 entries, 0 to 2239  
Data columns (total 30 columns):  
 #   Column           Non-Null Count  Dtype     
---  
 0   Unnamed: 0        2240 non-null    int64    
 1   ID               2240 non-null    int64    
 2   Year_Birth       2240 non-null    int64    
 3   Education        2240 non-null    object    
 4   Marital_Status   2240 non-null    object    
 5   Income            2216 non-null    float64  
 6   Kidhome          2240 non-null    int64    
 7   Teenhome         2240 non-null    int64    
 8   Dt_Customer      2240 non-null    object    
 9   Recency           2240 non-null    int64    
 10  MntCoke          2240 non-null    int64    
 11  MntFruits        2240 non-null    int64    
 12  MntMeatProducts  2240 non-null    int64    
 13  MntFishProducts  2240 non-null    int64    
 14  MntSweetProducts 2240 non-null    int64    
 15  MntGoldProd     2240 non-null    int64    
 16  NumDealsPurchases 2240 non-null    int64    
 17  NumWebPurchases  2240 non-null    int64    
 18  NumCatalogPurchases 2240 non-null    int64    
 19  NumStorePurchases 2240 non-null    int64    
 20  NumWebVisitsMonth 2240 non-null    int64    
 21  AcceptedCmp3     2240 non-null    int64    
 22  AcceptedCmp4     2240 non-null    int64    
 23  AcceptedCmp5     2240 non-null    int64    
 24  AcceptedCmp1     2240 non-null    int64    
 25  AcceptedCmp2     2240 non-null    int64    
 26  Complain          2240 non-null    int64    
 27  Z_CostContact    2240 non-null    int64    
 28  Z_Revenue          2240 non-null    int64    
 29  Response          2240 non-null    int64  
dtypes: float64(1), int64(26), object(3)  
memory usage: 525.1+ KB
```

- Feature Engineering (1):
 - Num_Transaction
 - Dengan menambahkan kolom *NumDealsPurchases*, *NumWebPurchases*, *NumCatalogPurchases*, dan *NumStorePurchases*
 - Conversion_Rate
 - Membagi *Num_Transaction* dengan *NumWebVisitsMonth*
 - Umur
 - Mengurangi 2022 dengan *Year_Birth*
 - Kategori Usia
 - Mengelompokkan Usia ≤ 11 tahun adalah Anak, ≤ 25 adalah Remaja, ≤ 35 adalah Dewasa Awal, ≤ 45 adalah Dewasa Akhir, ≤ 55 tahun adalah Lansia Awal, ≤ 65 tahun adalah Lansia Akhir, > 65 adalah Manula
 - Jumlah_anak
 - Menambahkan *Kidhome* dan *Teenhome*
 - Parent
 - Mengelompokkan berdasarkan Jumlah Anak, 0 : bukan parent, > 0 adalah parent

Untuk selengkapnya, dapat melihat jupyter notebook [disini](#)

Conversion Rate Analysis Based on Income, Spending and Age

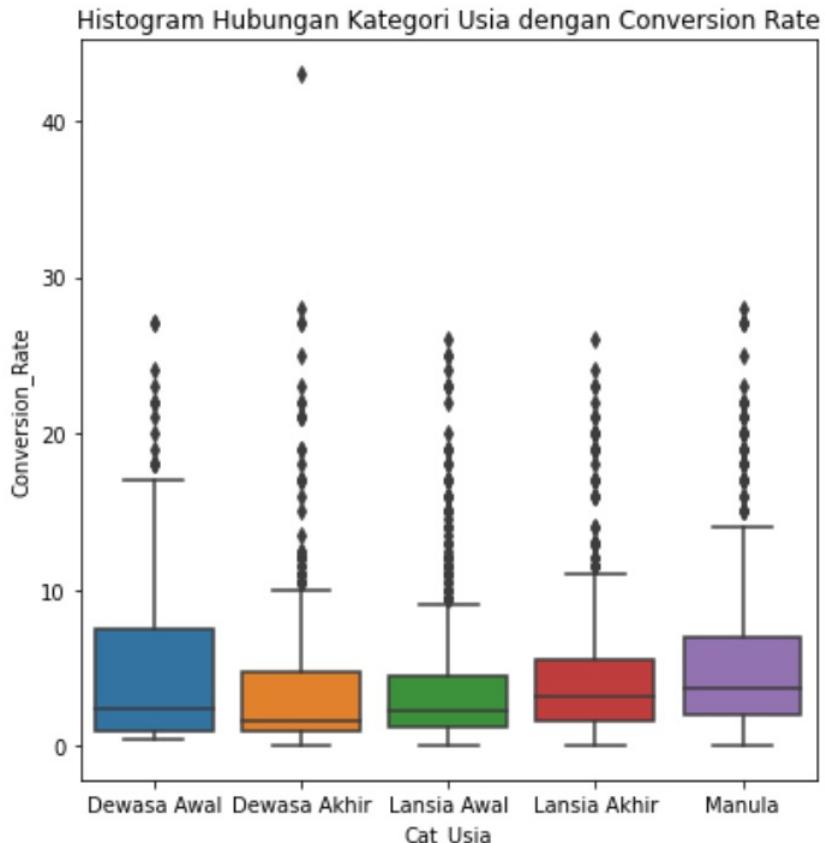
- Feature Engineering (2) :
 - Accepted Campaign
Dengan menambahkan kolom *AcceptedCmp1*, *AcceptedCmp2*, *AcceptedCmp3*, *AcceptedCmp4*, dan *AcceptedCmp5*
 - Income
Mengelompokkan Income, $\leq 1.000.000$ (Low Income), $\leq 3.000.000$ (Low-Middle), $\leq 5.000.000$ (Middle Income)
 $\leq 7.000.000$ (Middle-High), $> 7.000.000$ High
 - Transaction Amount
Menambahkan MntCoke, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, dan MntGoldProds
 - Kategori Transaction
Mengelompokkan Income, ≤ 500.000 (Low Transaction), $\leq 1.000.000$ (Middle Transaction),
 $\leq 1.500.000$ (High Transaction), dan $> 1.500.000$ (Very High Transaction)

Conversion Rate Analysis Based on Income, Spending and Age

Exploration Data Analysis (EDA)

- EDA Hubungan Kategori Usia dan Conversion Rate

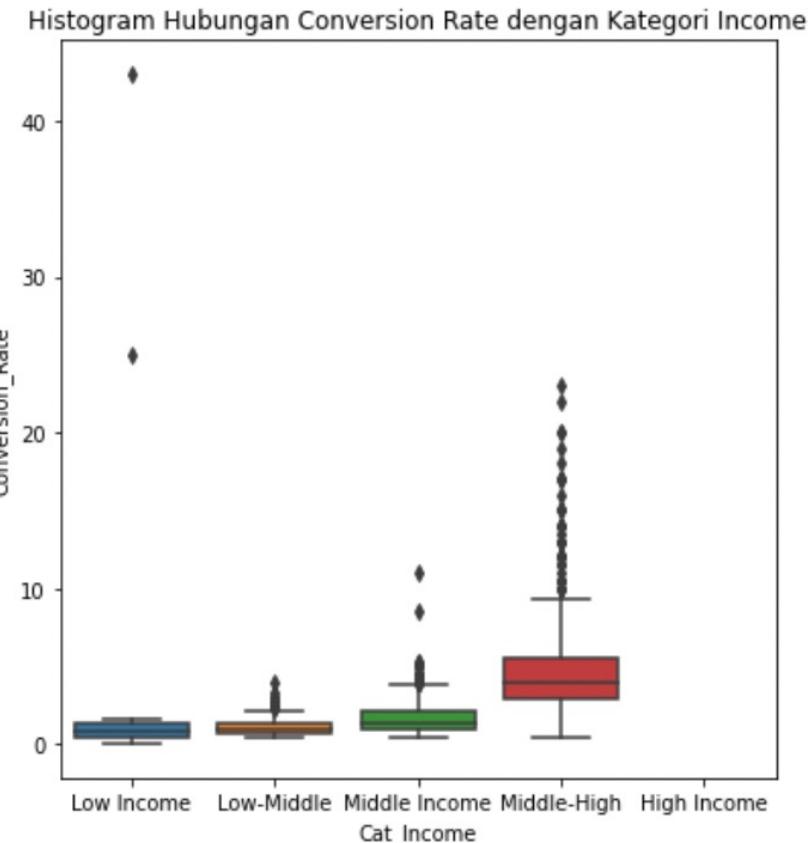
Tidak ditemukan perbedaan Conversion rate yang signifikan antar Kategori Usia.



Conversion Rate Analysis Based on Income, Spending and Age

Exploration Data Analysis (EDA)

- EDA Hubungan Kategori Income dan Conversion Rate
Untuk Kategori Middle High Income (3.000.001-5.000.000), memiliki Conversion Rate yang lebih tinggi diantara ketiga kategori lainnya.

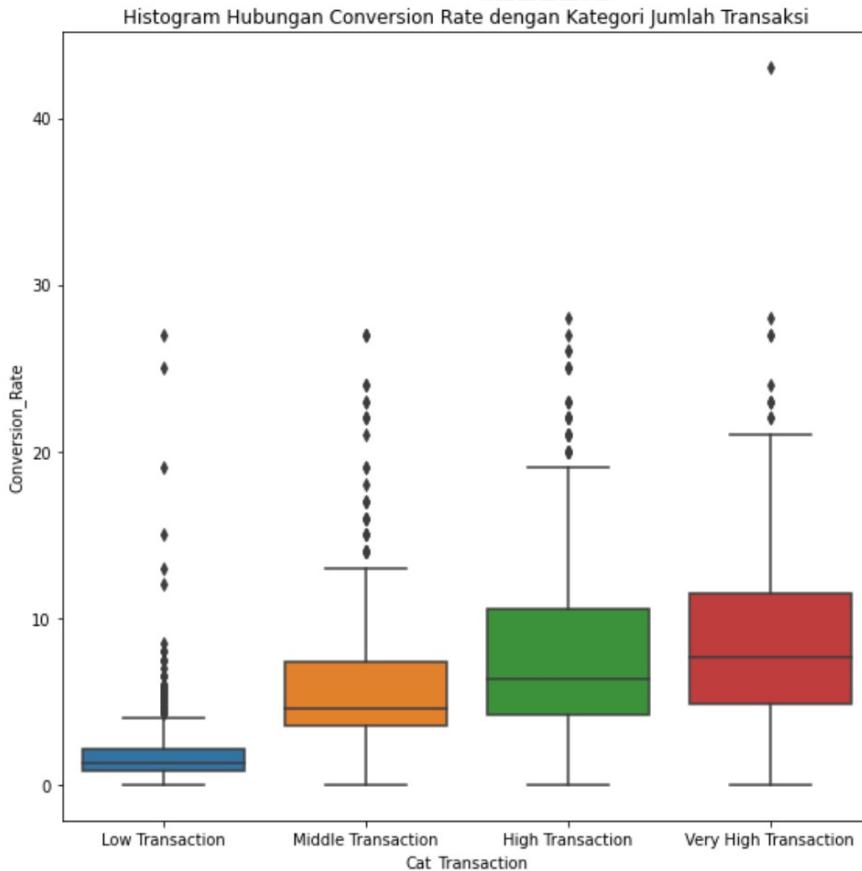


Untuk selengkapnya, dapat melihat jupyter notebook [disini](#)

Conversion Rate Analysis Based on Income, Spending and Age

Exploration Data Analysis (EDA)

- EDA Hubungan Kategori Jumlah Transaksi dan Conversion Rate
Untuk Kategori Middle Transaction, High Transaction, dan Very High Transaction ($>1.000.000$), memiliki Conversion Rate yang tidak berbeda signifikan.



Untuk selengkapnya, dapat melihat jupyter notebook [disini](#)

Data Cleaning & Preprocessing

- Null-Value

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

```

↳ Unnamed: 0      0
ID          0
Year_Birth   0
Education    0
Marital_Status 0
Income       24
Kidhome     0
Teenhome    0
Dt_Customer 0
Recency     0
MntCoke     0
MntFruits   0
MntMeatProducts 0
MntFishProducts 0
MntSweetProducts 0
MntGoldProducts 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
Num_Transaction 0
Conversion_Rate 2
Umur        0
Cat_Usia     0
Jumlah_anak 0
is_parent    0
AcceptedCmp_Tot 0
Cat_Income   0
Transaction_tot 0
Cat_Transaction 0
dtype: int64

```

- Handling Null –Value (Income)

- Null Value diisi dengan nilai Modus

```

| df['Conversion_Rate'].fillna(df['Conversion_Rate'].mode()[0], inplace=True)
| df.Conversion_Rate.value_counts()
| df.Conversion_Rate.isna().sum()

0

```

- Handling Null –Value (Conversion_Rate)

- Null Value diisi dengan nilai Modus
- Nilai Inf akibat pembagian dengan nol
(NumWebVisit) menyebabkan hasil Conversion rate menjadi Inf, nilai Inf diganti nol

```

df['Income'].fillna(df['Income'].mode()[0], inplace=True)
df.Income.value_counts()
df.Income.isna().sum()

0

# Terdapat Conversion_Rate yang bernilai Inf, diganti dengan 0
df['Conversion_Rate'] = df['Conversion_Rate'].replace(np.Inf, 0)

```

- Null-Value (Checking)

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

```

Unnamed: 0      0
ID          0
Year_Birth   0
Education    0
Marital_Status 0
Income       0
Kidhome     0
Teenhome    0
Dt_Customer 0
Recency     0
MntCoke     0
MntFruits   0
MntMeatProducts 0
MntFishProducts 0
MntSweetProducts 0
MntGoldProducts 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
Num_Transaction 0
Conversion_Rate 0
Umur        0
Cat_Usia     0
Jumlah_anak 0
is_parent    0
AcceptedCmp_Tot 0
Cat_Income   0
Transaction_tot 0
Cat_Transaction 0
dtype: int64

```

Data Cleaning & Preprocessing

- Duplicated Data

- Tidak ada data nilai duplikat

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

```
0
```

- Feature Encoding

- Label Encoding : Education
- One Hot Encoding :
 - Cat_Usia (Kategori Usia)
 - Marital_Status
 - Cat_Income (Kategori Income)
 - Cat_Transaction (Kategori Transaksi)

- Feature Dropping :

```
df2 = df.drop(columns = ['Unnamed: 0','ID','Education','Marital_Status','Kidhome','Teenhome',
.....,'Dt_Customer','Recency','MntCoke','MntFruits','MntMeatProducts','MntFishProducts',
.....,'MntSweetProducts','MntGoldProds','NumDealsPurchases','NumWebPurchases','NumCatalogPurchases',
.....,'NumStorePurchases','NumWebVisitsMonth','AcceptedCmp3','AcceptedCmp4','AcceptedCmp5',
.....,'AcceptedCmp1','AcceptedCmp2','Complain','Z_CostContact','Z_Revenue','Response','Cat_Usia',
.....,'Cat_Transaction','Cat_Income'])
```

```
df2.info()
```

- Standarisasi :

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
sc_data = StandardScaler()
df3 = sc_data.fit_transform(df2.astype(float))
df3
```

```
array([[ -0.98534473,   0.25027631,   1.32082612, ..., -1.11960758,
       -0.46056619,   2.84704954],
       [-1.23573295,  -0.21309477,  -1.15459595, ...,  0.89317009,
       -0.46056619,  -0.35124082],
       [-0.3176428 ,   0.77969173,   0.79968463, ..., -1.11960758,
       2.17124059,  -0.35124082],
       ...,
       [ 1.01776106,   0.20481927,   0.53911389, ..., -1.11960758,
       -0.46056619,  -0.35124082],
       [-1.06880747,   0.68665606,   1.06025538, ..., -1.11960758,
       2.17124059,  -0.35124082],
       [-1.23573295,   0.04326408,  -0.50316909, ...,  0.89317009,
       -0.46056619,  -0.35124082]])
```

Untuk selengkapnya, dapat melihat jupyter notebook [disini](#)

Data Modeling

- Visualisasi **Elbow Method** menggunakan **K-Means Clustering**

```
| from sklearn.cluster import KMeans
| from scipy.spatial.distance import cdist
| distortions = []
| inertias = []
| mapping1 = {}
| mapping2 = {}
| K = range(1, 10)

| for k in K:
|     # Building and fitting the model
|     kmeanModel = KMeans(n_clusters=k).fit(df3)
|     kmeanModel.fit(df3)

|     distortions.append(sum(np.min(cdist(df3, kmeanModel.cluster_centers_,
|         'euclidean'), axis=1)) / df3.shape[0])
|     inertias.append(kmeanModel.inertia_)

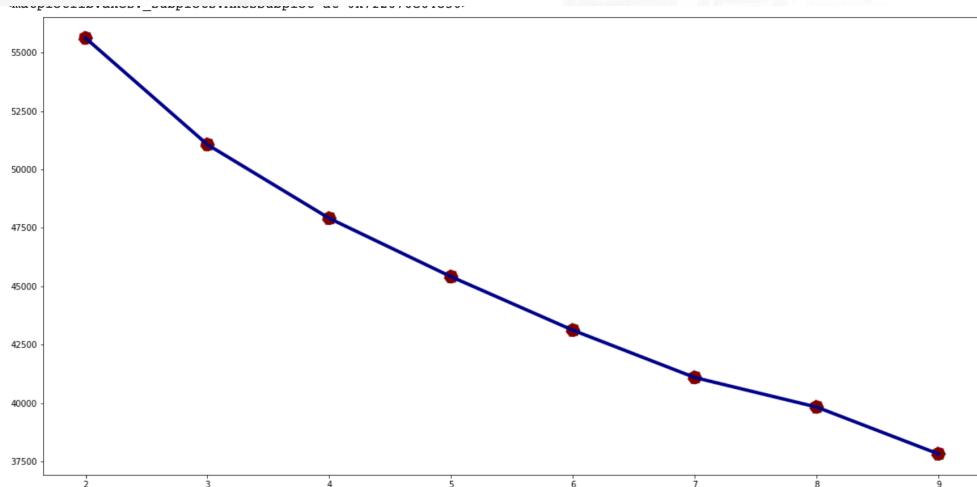
|     mapping1[k] = sum(np.min(cdist(df3, kmeanModel.cluster_centers_,
|         'euclidean'), axis=1)) / df3.shape[0]
|     mapping2[k] = kmeanModel.inertia_

from sklearn.cluster import KMeans
inertia = []

for i in range(2, 10):
    kmeans = KMeans(n_clusters=i, random_state=0)
    kmeans.fit(df3)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(20, 10))
# plt.plot(inertia)

sns.lineplot(x=range(2, 10), y=inertia, color="#000087", linewidth = 4)
sns.scatterplot(x=range(2, 10), y=inertia, s=300, color='#800000', linestyle='--')
```



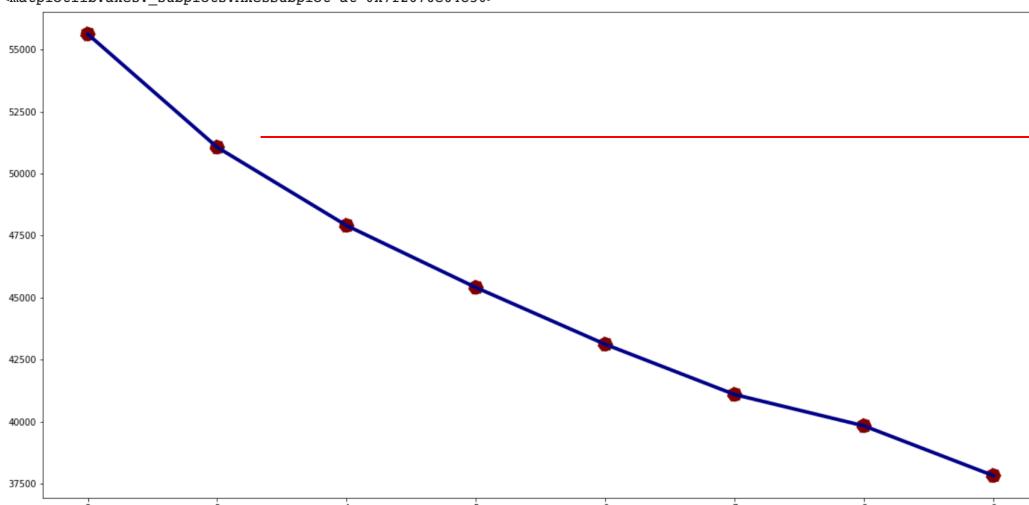
```
for key, val in mapping2.items():
    print(f'{key} : {val}')
```

```
1 : 67199.99999999994
2 : 55622.08433928477
3 : 51061.83383170926
4 : 47905.22628032777
5 : 45614.81085772801
6 : 43025.92385971268
7 : 41263.52243648496
8 : 39659.00047195942
9 : 38270.660110920304
```

Untuk selengkapnya, dapat melihat jupyter notebook [disini](#)

Data Modeling

- Dari Visualisasi Elbow Method, cluster maksimal adalah 3 buah.



Dilihat dari Elbow Method, patahan paling tajam pada angka 3, sehingga jumlah cluster paling optimal adalah 3.

```
for key, val in mapping2.items():
    print(f'{key} : {val}')
```

```
1 : 67199.9999999994
2 : 55622.08433928477
3 : 51061.83383170926
4 : 47905.22628032777
5 : 45614.81085772801
6 : 43025.92385971268
7 : 41263.52243648496
8 : 39659.00047195942
9 : 38270.660110920304
```

Customer Personality Analysis for Marketing Retargeting

- Evaluasi KMeans

```
kmeans = KMeans(n_clusters=3, random_state=0).fit(df3)

df4 = pd.DataFrame(data=df3, columns=list(df2))

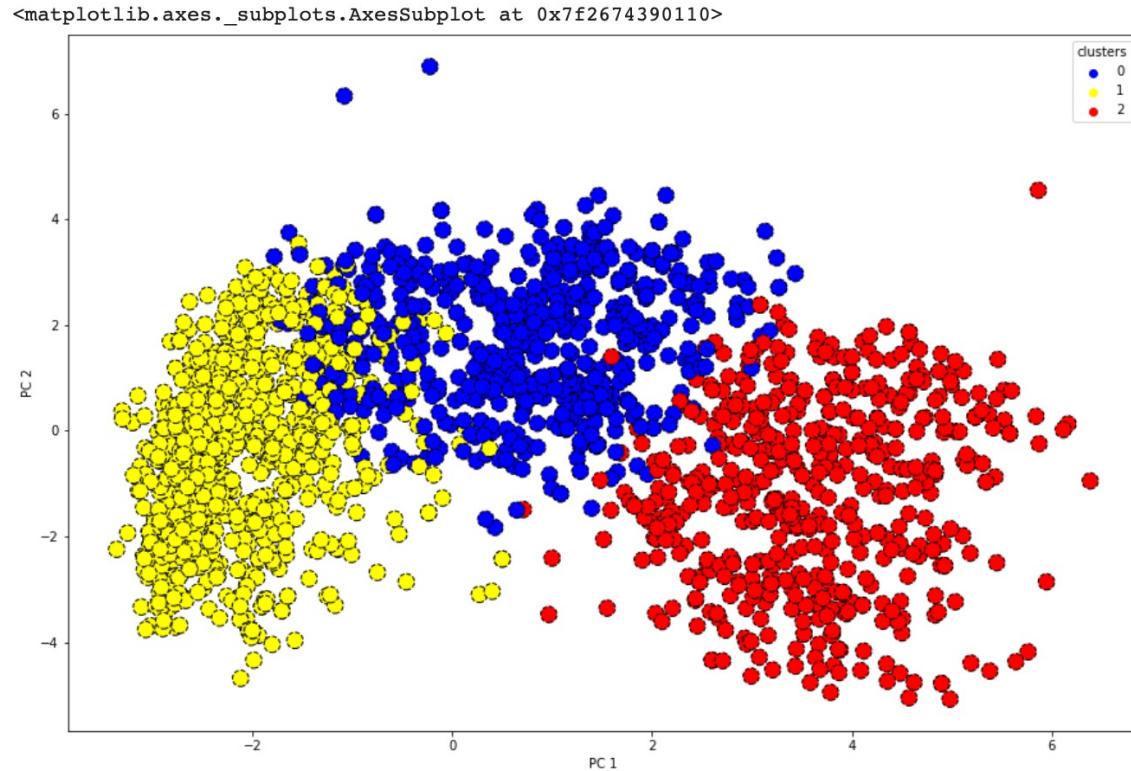
df4['clusters'] = kmeans.labels_

data_pca = pd.DataFrame(data = pcs, columns = ['PC 1', 'PC 2'])
data_pca['clusters'] = df4['clusters']
data_pca.head(10)
```

	PC 1	PC 2	clusters
0	3.089506	0.367726	2
1	-2.043303	2.010721	1
2	2.626746	0.104379	2
3	-2.740333	-2.126766	1
4	-0.908067	-0.485849	0
5	0.616739	1.619470	0
6	0.298263	1.267990	0
7	-2.437924	-1.528546	1
8	-2.510989	-0.020629	1
9	-2.192548	2.227822	1

```
fig, ax = plt.subplots(figsize=(15,10))

sns.scatterplot(
    x="PC 1", y="PC 2",
    hue="clusters",
    edgecolor='black',
    linestyle='--',
    data=data_pca,
    palette=['blue', 'yellow', 'red'],
    s=160,
    ax=ax
)
```



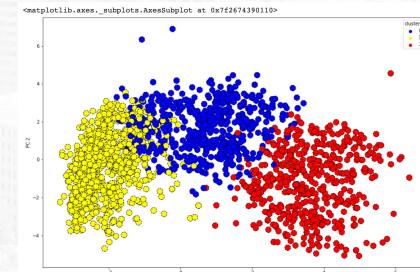
Untuk selengkapnya, dapat melihat jupyter notebook [disini](#)

Customer Personality Analysis for Marketing Retargeting

```
df2['clusters'] = kmeans.labels_
display(df2.groupby('clusters')[['Year_Birth', 'Income', 'Num_Transaction', 'Umur', 'Jumlah_anak', 'is_parent',
'AcceptedCmp_Tot', 'Transaction_tot']].agg(['mean', 'median']))
```

clusters	Year_Birth		Income		Num_Transaction		Umur		Jumlah_anak		is_parent		AcceptedCmp_Tot		Transaction_tot	
	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median
0	1963.372392	1964.0	5.949148e+07	60000000.0	20.346709	21.0	58.627608	58.0	1.094703	1.0	0.871589	1.0	0.218299	0.0	7.350433e+05	730000.0
1	1971.908004	1973.0	3.404444e+07	34633000.0	8.771849	8.0	50.091996	49.0	1.221711	1.0	0.877645	1.0	0.090156	0.0	1.250589e+05	68000.0
2	1968.830189	1970.0	7.903851e+07	77866500.0	20.905660	21.0	53.169811	52.0	0.224528	0.0	0.198113	0.0	0.816981	0.0	1.439843e+06	1435000.0

- Dari hasil clustering :
- Cluster 1 : usia rata-rata 58 tahun, tingkat income rata-rata 6 juta per bulan, total transaksi 20 kali, dengan jumlah nilai transaksi rata-rata 700.000
- Cluster 2 : usia rata-rata 49 tahun, tingkat income rata-rata 3.5 juta per bulan, total transaksi 8 kali, dengan jumlah nilai transaksi rata-rata 120.000
- Cluster 3 : usia rata-rata 54 tahun, tingkat income rata-rata 8 juta per bulan, total transaksi 20 kali, dengan jumlah transaksi rata-rata 1.400.000



- Usia : dari data diatas, customer kebanyakan diusia dewasa akhir sampai lansia, sehingga tidak ada perbedaan yang bermakna.
- Dari clustering, perbedaan yang jelas adalah pembagian segmentasi jumlah nominal transaksi.
- Rekomendasi Bisnis :
 - Untuk Cluster 2 : Jumlah transaksi minimal dibandingkan dua jenis cluster lainnya, dapat dipertimbangkan untuk menambah jenis promo, sehingga meningkatkan jumlah transaksi, perlu adanya peningkatan kunjungan ke Website. Promo kunjungan perlu dipertimbangkan ada. Perlu adanya Program Loyalty dengan indikator jumlah kunjungan, semakin sering customer datang, akan mendapatkan voucher diskon (contoh)
 - Untuk Cluster 1 dan 3, Jumlah kunjungan sudah cukup tinggi, dan nominal transaksi juga tinggi, perlu adanya maintenance agar, tidak terjadi churn. Bisa menggunakan voucher tambahan diskon untuk minimal pembelanjaan tertentu.