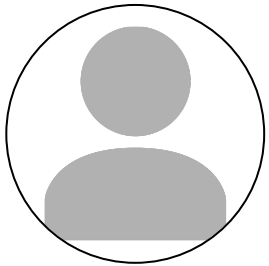
Predict Customer Personality to boost marketing campaign by using Machine Learning



Created by:
Vias Aulia
viasauliaa@gmail.com
linkedIn. com/in/vias-aulia

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

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

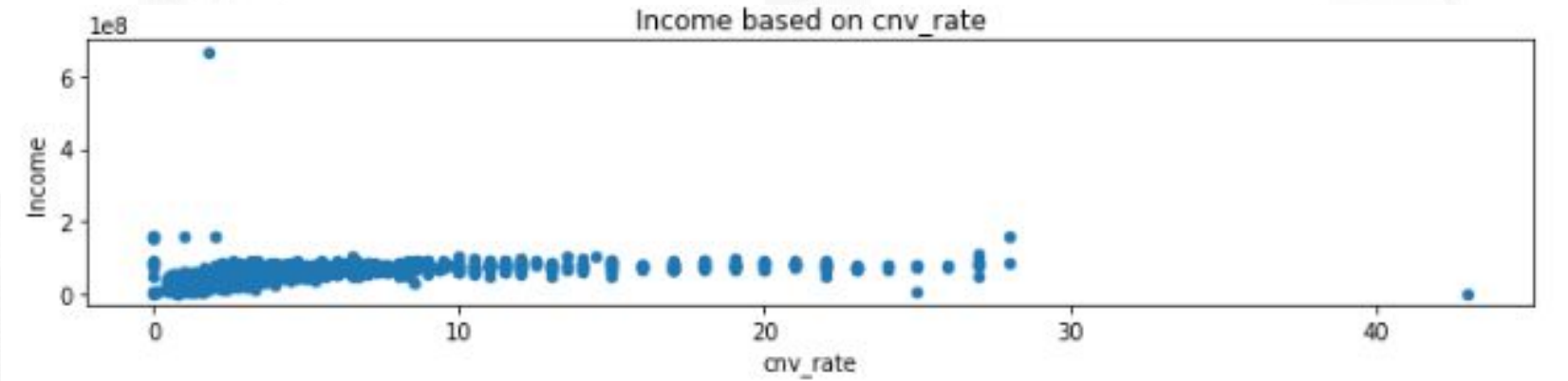
```
def rate(x,y):
  if y == 0:
    return 0
  return x/y
```

```
df1['cnv_rate'] = df1.apply(lambda x : rate(x['total_purchase'], x['NumWebVisitsMonth']),axis=1)
a = df1['cnv_rate']
a.sort_values(ascending=False)
```

- Menghitung cnv_rate dengan melihat total purchase dibagi dengan jumlah kunjungan tiap bulan

Untuk selengkapnya, dapat melihat jupyter notebook disini

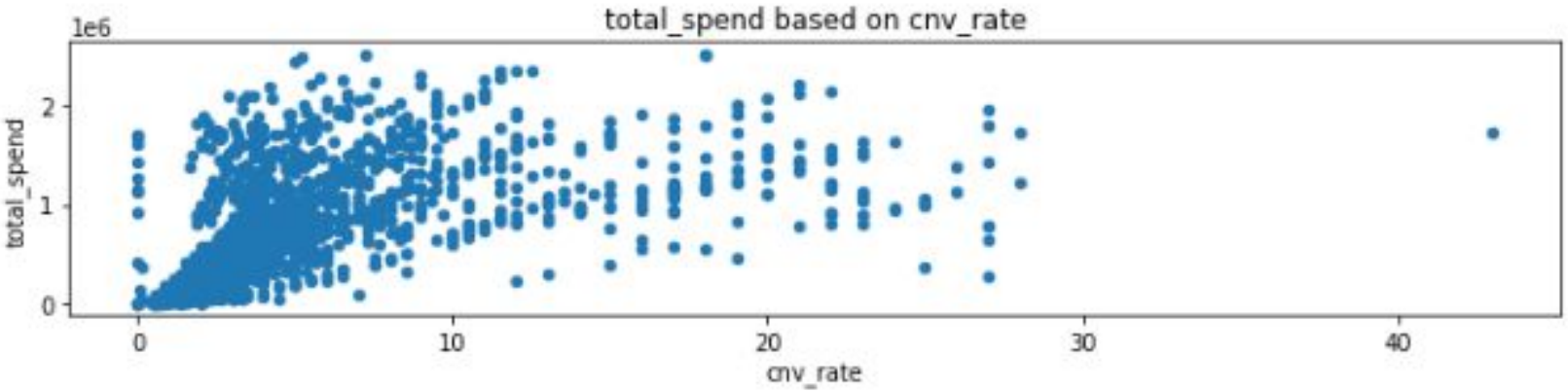
Conversion Rate Analysis Based on Income, Spending and Age



- Customer dengan income lebih rendah cenderung memiliki cnv_rate yang rendah pula
- Customer dengan income menengah cenderung memiliki cnv_rate lebih tingg

Untuk selengkapnya, dapat melihat jupyter notebook disini

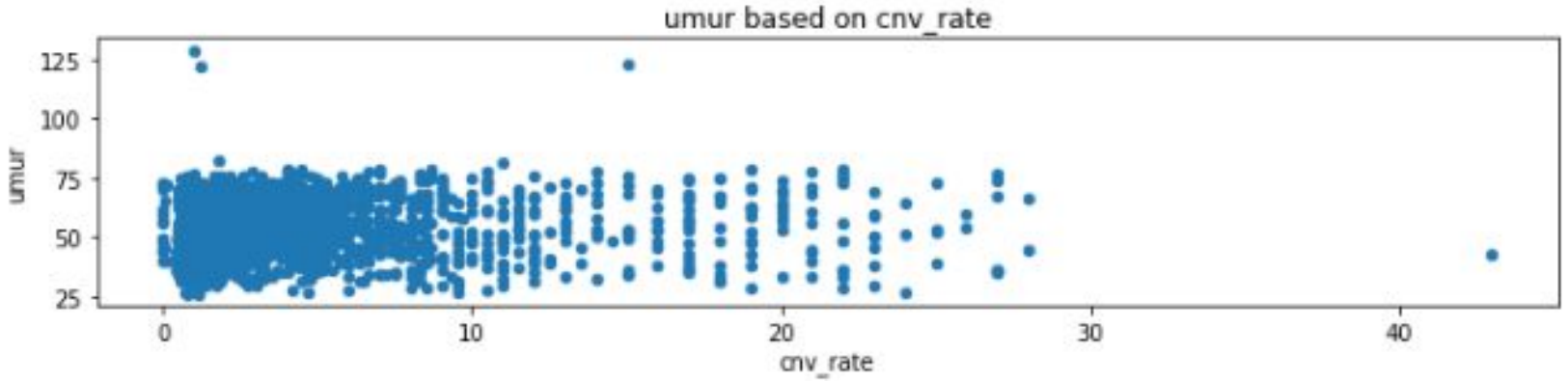
Conversion Rate Analysis Based on Income, Spending and Age



- Terlihat dengan total spending yang meningkat, cnv_rate pun ikut meningkat
- Customer dengan total spend lebih tinggi cenderung memiliki cnv_rate yang tinggi, sedangkat total spend yang lebih rendah memiliki cnv_rate yang rendah seperti yang terlihat sebaran titik pada plot yang berkumpul pada kiri bawah

Untuk selengkapnya, dapat melihat jupyter notebook disini

Conversion Rate Analysis Based on Income, Spending and Age



- Tidak terdapat pola yang jelas pada scatter plot umur pada cnv rate
- Dapat dikatakan cnv_rate tidak dipengaruhi oleh umur

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

```
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  MntGoldProds         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
```

- Total kolom awal sejumlah 30 kolom yang nantinya akan kita proses pada tahap data cleaning dan data prep
- Terdapat null values pada kolom income, selanjutnya null tersebut akan diisi dengan median kolom tersebut karena ada kecendurungan skew

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

```
median = df1['Income'].median()
df1['Income'].fillna(median,inplace=True)
```

```
df1.isnull().sum()
```

```
bin =[0,26,40,60,80,130]
df1['umur_bin'] = pd.cut(df1['umur'],bin)
df1['umur_bin']
```

```
#create total of days joined
import datetime as dt
from datetime import date

df1['Dt_Customer'] = pd.to_datetime(df1['Dt_Customer'])
df1['Dt_Collected'] = date.today()
df1['Dt_Collected'] = df1['Dt_Collected'].astype('datetime64[ns]')
df1['Dt_Days_Customer'] = df1['Dt_Collected'] - df1['Dt_Customer']
df1['Dt_Days_Customer'] = df1['Dt_Days_Customer'].dt.days
```

```
def child(x,y):
  if x == 0 or y == 0:
    return 0
  return 1
```

```
df1['child'] = df1.apply(lambda x : child(x['Kidhome'], x['Teenhome']), axis=1)
```

- Selain cnv rate juga dikelompokkan umur dengan pd.cut sehingga distribusi lebih merata
- Dihitung juga rentang waktu customer join ecommerce hingga saat ini dengan kolom dt_days_customer

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

```
median = df1['Income'].median()
df1['Income'].fillna(median,inplace=True)
```

```
df1.isnull().sum()
```

```
bin =[0,26,40,60,80,130]
df1['umur_bin'] = pd.cut(df1['umur'],bin)
df1['umur_bin']
```

```
#create total of days joined
import datetime as dt
from datetime import date

df1['Dt_Customer'] = pd.to_datetime(df1['Dt_Customer'])
df1['Dt_Collected'] = date.today()
df1['Dt_Collected'] = df1['Dt_Collected'].astype('datetime64[ns]')
df1['Dt_Days_Customer'] = df1['Dt_Collected'] - df1['Dt_Customer']
df1['Dt_Days_Customer'] = df1['Dt_Days_Customer'].dt.days
```

```
def child(x,y):
  if x == 0 or y == 0:
    return 0
  return 1
```

```
df1['child'] = df1.apply(lambda x : child(x['Kidhome'], x['Teenhome']), axis=1)
```

- Selain cnv rate juga dikelompokkan umur dengan pd.cut sehingga distribusi lebih merata
- Dihitung juga rentang waktu customer join ecommerce hingga saat ini dengan kolom dt_days_customer

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

- Kolom-kolom yang sudah dilakukan feature engineering dan tidak informatif selanjutnya akan didrop
- Selain itu untuk duplicate handling tidak dilakukan karena tidak ada duplikat pada data

Drop Unused Column

```
df1.drop(columns=['Unnamed: 0', 'ID','MntCoke','MntFruits', 'MntMeatProducts', 'MntFishProducts',
                  'MntSweetProducts','MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
                  'NumCatalogPurchases', 'NumStorePurchases','AcceptedCmp3', 'AcceptedCmp4',
                  'AcceptedCmp5', 'AcceptedCmp1','AcceptedCmp2','Z_CostContact', 'Z_Revenue'],inplace=True)
```

```
df1.columns
```

```
Index(['Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'NumWebVisitsMonth', 'Complain',
       'Response', 'total_purchase', 'cnv_rate', 'umur', 'total_spend',
       'campaign', 'umur_bin', 'Dt_Collected', 'Dt_Days_Customer', 'child'],
      dtype='object')
```

Drop Duplicated Values

```
[149] print('total baris sebelum handling duplikat:\n',df1.shape)
```

```
total baris sebelum handling duplikat:
 (2057, 20)
```

```
[151] print('data yang duplikat sebanyak:\n',df1.duplicated().sum())
```

```
data yang duplikat sebanyak:
 0
```

Untuk selengkapnya, dapat melihat jupyter notebook disini

```
encode = pd.get_dummies(df1,columns=['Education', 'Marital_Status'],drop_first= True)
print(encode)
```

- Karena tujuannya adalah clustering maka kolom dengan object value akan diencode dengan one hot encoding dan diberlakukan drop_first sehingga diharapkan meminimalisir multicollinearity

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Modeling

Rakamin Academy

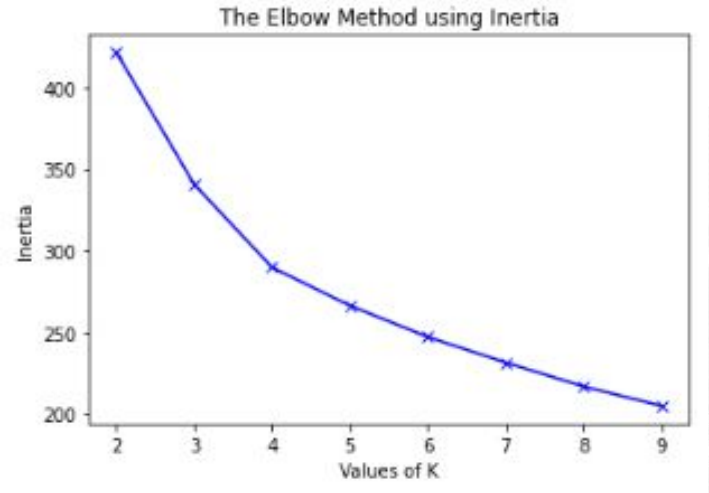
- Reduce dimensionality dengan RFM analysis sehingga menghasilkan kolom recency, total purchase, total spend dan dt_days_customer serta umur sebagai L dan C

```
encode1 =encode[['Recency','total_purchase','total_spend','Dt_Days_Customer','umur']]
encode1.columns = ['R', 'F', 'M','L','C']
encode1.describe()
```

- Data tersebut selanjutnya dilakukan handling outlier menggunakan Z score dan distandarisasi dengan MinMaxScaler

```
from scipy import stats
import numpy as np
z = np.abs(stats.zscore(encode_z))
filter = (z<3).all(axis=1)
encode_z = encode_z[filter]
encode_z
```

- Untuk melihat jumlah cluster yang optimal dilakukan dengan elbow method dengan Inertia



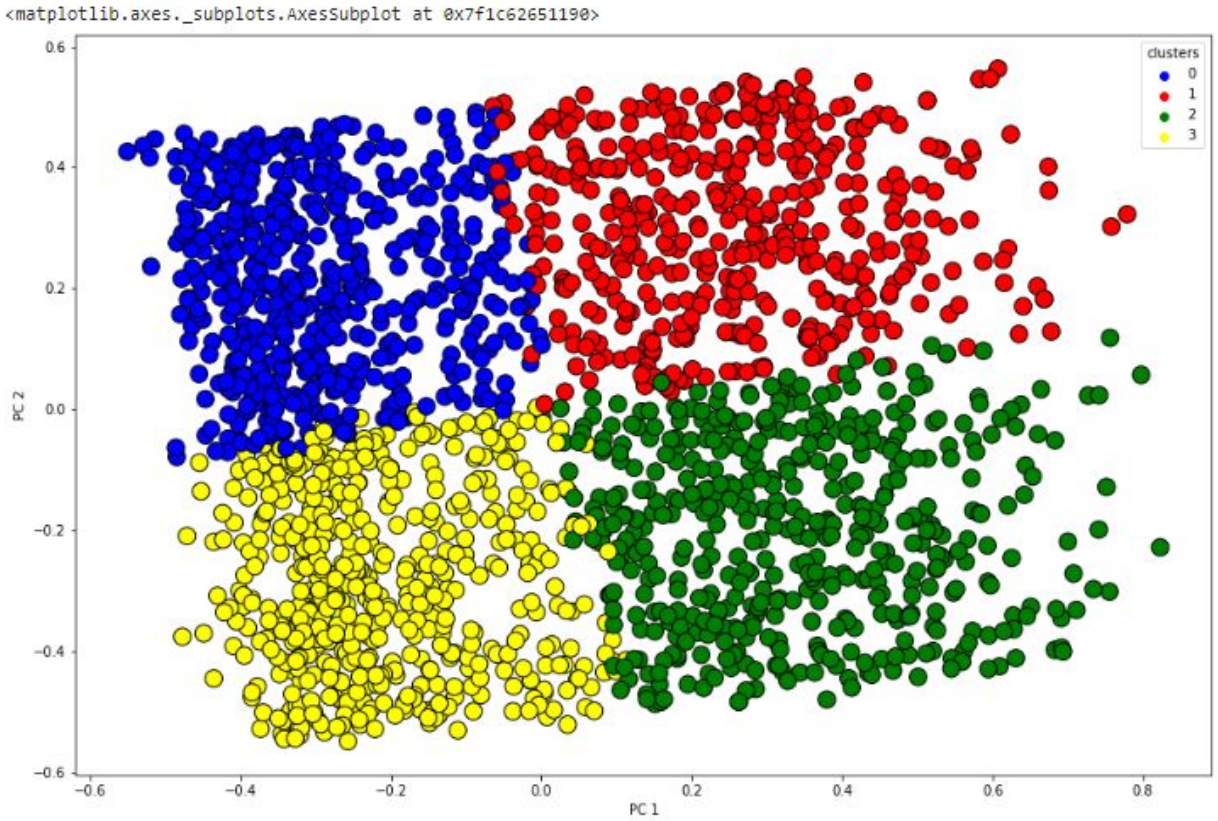
- Untuk memudahkan memilih nilai K pada elbow method, gunakan package kneed dan KneeLocator dan didapat optimal K adalah 4

Data Modeling

- Untuk reduce dimensionality digunakan PCA dengan n_components sebanyak 2

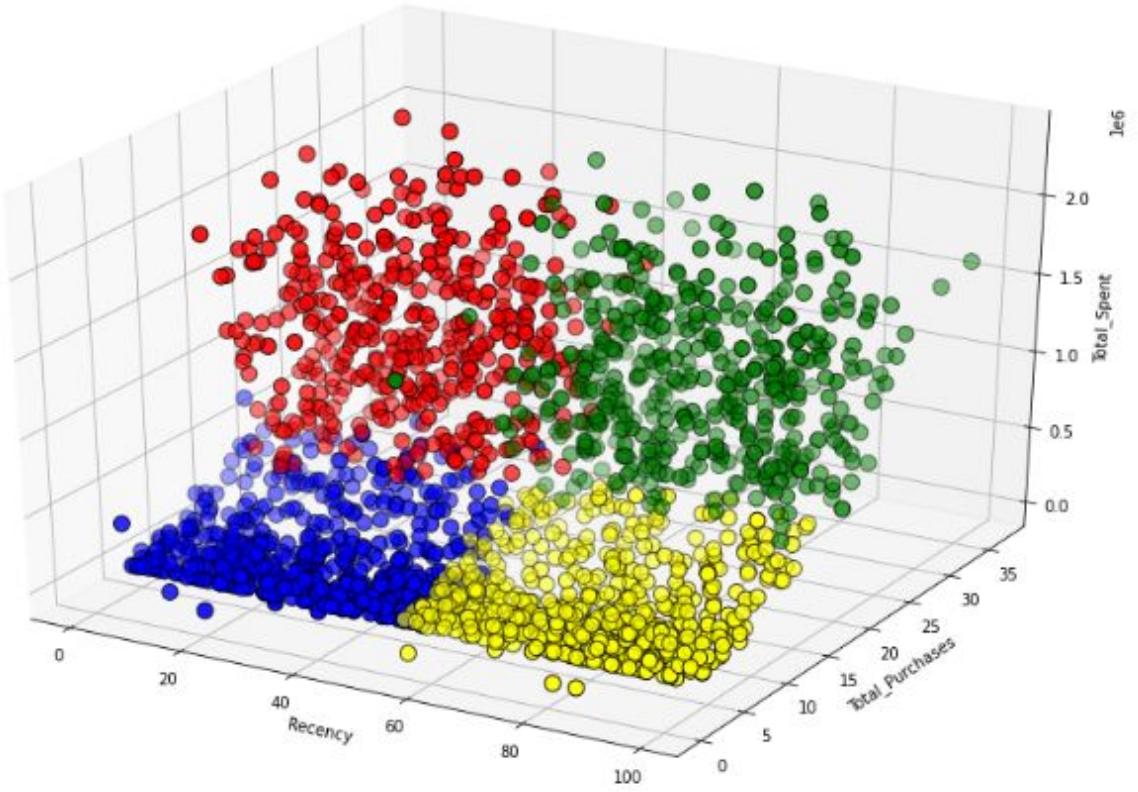
```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(new_encode)
pcs = pca.transform(new_encode)

df_pca = pd.DataFrame(data = pcs, columns = ['PC 1', 'PC 2'])
df_pca['clusters'] = df_std_cluster['clusters']
df_pca
```

Data Modeling



- Grafik dibuat berdasarkan PC1 dan PC2 dan membentuk 4 cluster
- Tiap cluster merepresentasikan segementasi customer berdasarkan feature pada RFM analysis

Data Modeling



Berdasarkan gambar 3D disamping, terdapat 4 cluster pada segementasi customer berdasarkan 3 feature utama yaitu

Recency,

Total_Purchase

Total_Spent

Untuk selengkapnya, dapat melihat jupyter notebook disini

Customer Personality Analysis for Marketing Retargeting

Rakamin Academy

	R		F		M		L		C	
	mean	median	mean	median	mean	median	mean	median	mean	median
clusters										
0	24.526	25.000	8.770	8.000	134926.391	69000.000	3273.115	3258.000	50.039	49.000
1	22.348	23.000	21.780	21.000	1115991.131	1071000.000	3357.477	3379.000	54.625	54.000
2	72.842	72.000	21.508	21.500	1166907.787	1113000.000	3366.178	3376.000	56.453	57.000
3	74.378	75.000	9.293	8.000	146060.000	84000.000	3292.527	3283.500	52.131	51.000

- Pada cluster 0 terlihat R yaitu recency terdapat recent activity oleh customer di e_commerce dan F yaitu total purchase yang masih sedikit dan M yaitu total spent dengan median 69000 namun mean hampir 135000 sedangkan L dan C tidak berbeda signifikan pada seluruh cluster
- Pada cluster 1 terlihat R terdapat recent activity dengan F dan M tinggi
- Pada cluster 2 dan 3 memiliki R dengan no recent activity namun pada cluster 2 terdapat total purchase dan total spent yang tinggi dibanding cluster 3

Customer Personality Analysis for Marketing Retargeting

	clusters	Total Customers
0	0	557
1	1	451
2	2	488
3	3	550

- Berdasarkan segmentasi yang telah dilakukan, customer dibagi menjadi 4 cluster dan total customer tiap cluster dapat dilihat pada gambar
- Selanjutnya tiap cluster tersebut dilakukan strategi yang berfokus pada keadaan atau tren yang ada pada cluster

Customer Personality Analysis for Marketing Retargeting

Rakamin Academy

Priority Customers (Cluster 1)

Customer with high value in all three factor on RFM analysis. Customer with recent activities on marketplace, high frequencies of purchase and high spending on marketplace should be targeted with special promotions like triggered campaigns or free shipping fee to keep them active

New Customers (Cluster 0)

Customers with recent activities but low frequencies of purchase maybe a new customers. A targeted follow-up like events, promotion or new product may convert them into repeat customer

Customer Personality Analysis for Marketing Retargeting

Old Customers (Cluster 2 and 3)

Customers with no recent activities but high spending were once valueable customers but have stopped either because competitor or any other reasons such as poor experience or dissatisfied with product or service. Identify the problem and work on it. A targeted message like 'we fix it' campaign that explain the marketplace listened to customer feedback to make a better experience on our marketplace may reactivate them

Untuk selengkapnya, dapat melihat jupyter notebook disini