Customer Marketing Analysis

We have a dataset containing the data of customers of a retail company in FMCG. I'm going to analyse this dataset and then carry out a clustering exercise to determine a new online customer acquisition strategy for the company.

1) What is Clustering ?

Clustering is a technique of machine learning that involves clustering data points based on similarity or distance. The most used distance is Euclidean distance, although other measures can also be used depending on the context, like correlation.
It is an unsupervised learning method and a popular technique for statistical data analysis. For a given set of points, you can use classification algorithms to sort these individual data points into specific groups. As a result, the data points in a particular group exhibit similar properties. At the same time, data points in different groups have different characteristics.

2) Data importing/Cleaning

To deal with the final assessment I decided to start by importing my various libraries that I would use to deal with the subject.

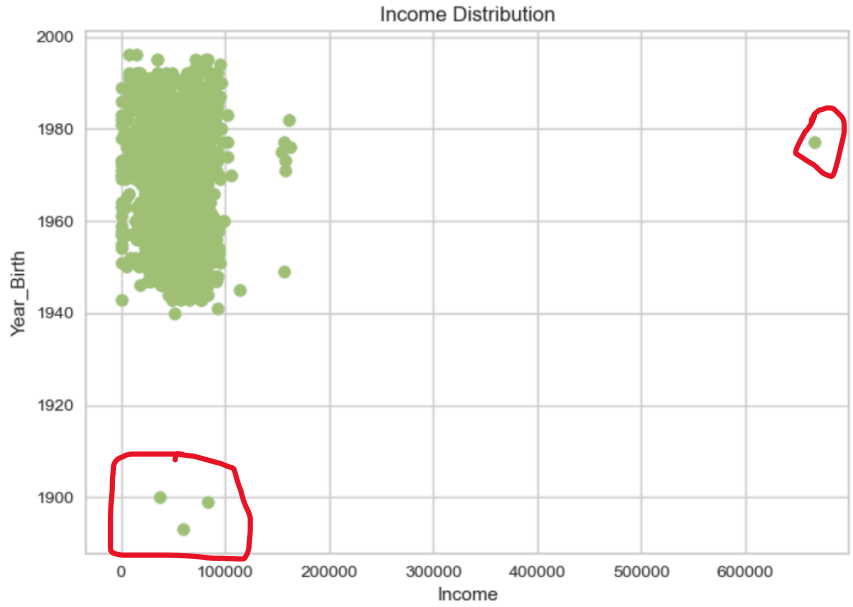
```
]: import pandas as pd
   import numpy as np
   import matplotlib.pyplot as plt
   import seaborn as sns
   from sklearn.preprocessing import LabelEncoder
   from sklearn.preprocessing import StandardScaler
   from sklearn.decomposition import PCA
   from sklearn.cluster import KMeans
   from sklearn import metrics
   from yellowbrick.cluster import KElbowVisualizer
   from sklearn.cluster import AgglomerativeClustering
```

I first imported the file, and ran some codes to get statistics on the dataset. I noticed that the maximum Income amount was abnormally high and the minimum Year_Birth abnormally low. We'll do a scatter plot to check this later.

I've carried out a clean-up and found that 24 items of data are missing from the Income column. This means either that there was a problem retrieving the data, or that the customers didn't get any income in the previous year. When in doubt I preferred to drop the missing data because I thought it would be more logical given the number of rows rather than replacing them with averages.

2) Manipulation and Visualization

First of all I checked the anomaly for Income and Year Birth and there was indeed an error.



So I cleaned up the 2 columns to remove the anomalies.

Then some codes for manipulation and visualisation. In particular with Marital Status and Education Level:

```
df["Marital_Status"].value_counts()
```

```
Marital_Status
Married     864
Together    578
Single      479
Divorced    231
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: count, dtype: int64
```

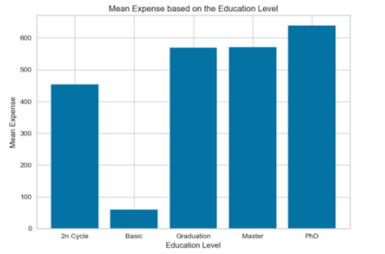
```
df["Education"].value_counts()
```

```
Education
Graduation    1126
PhD            485
Master         370
2n Cycle       201
Basic           54
Name: count, dtype: int64
```

```
df["Dt_Customer"] = pd.to_datetime(df["Dt_Customer"], format='%d-%m-%Y')
dates = []
for i in df["Dt_Customer"]:
    i = i.date()
    dates.append(i)
#Dates of the newest and oldest recorded customer
print("The newest customer's arrive the :",max(dates))
print("The oldest customer's arrive the :",min(dates))
```

```
The newest customer's arrive the : 2014-06-29
The oldest customer's arrive the : 2012-07-30
```

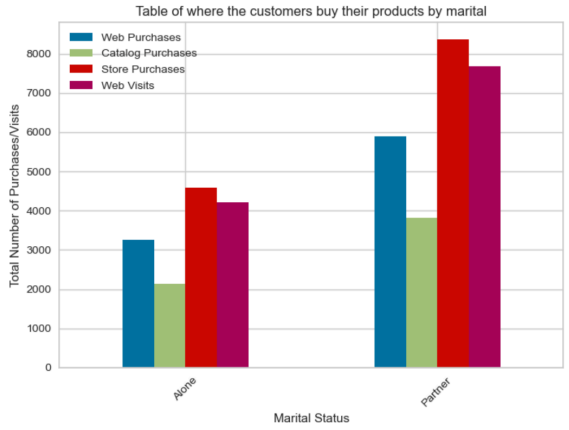
```
mean_expense_by_education = df.groupby('Education')['MntTotal'].mean()

plt.bar(mean_expense_by_education.index, mean_expense_by_education.values)
plt.title('Mean Expense based on the Education Level')
plt.xlabel('Education Level')
plt.ylabel('Mean Expense')
plt.show()
```

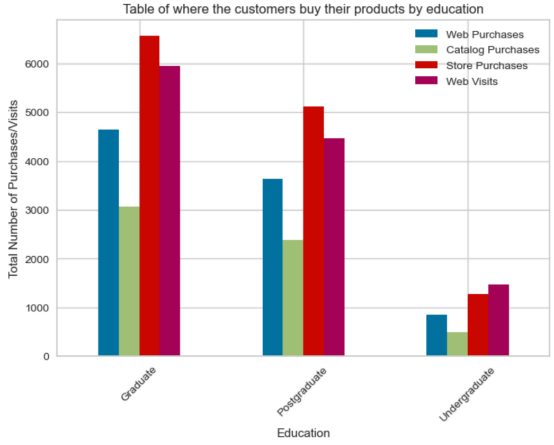


I then manipulated the data to create new columns and remove columns to simplify the dataset.

```
Entrée [22]: df["Education"]=df["Education"].replace({"Basic":"Undergraduate","2n Cycle":"Undergraduate", "Graduation":"Graduate"
Entrée [23]: df["Marital_Status"]=df["Marital_Status"].replace({"Married":"Partner", "Together":"Partner", "Absurd":"Alone", "Wid
Entrée [24]: df["Age"] = 2024-df["Year_Birth"]
Entrée [25]: df["Children"]=df["Kidhome"]+df["Teenhome"]
```

Then I did some coding according to the marketing strategy that was going to be used for the clusters, on online purchases, in-store purchases, catalogue purchases and website visits. I used pivot tables to drop my tables





Now, before using the scaler function, I've removed the unnecessary columns from my df dataframe in order to get the most optimal clustering for my code. I've also changed my object columns to INT64, so I've replaced Alone by 1 and Partner by 2. And for Education I replaced Undergraduate by 1, Graduate by 2, and Postgraduate by 3.

```
e_column = ["Dt_Customer", "AcceptedCmp1", "AcceptedCmp2", "AcceptedCmp3", "Year_Birth", "ID","Complain","Response"]
drop(drop_some_column, axis=1)
```

```
df["Marital_Status"]=df["Marital_Status"].replace({"Partner":"2", "Alone":"1",})
df['Marital_Status'] = df['Marital_Status'].astype('int64')
```

```
df["Education"]=df["Education"].replace({"Graduate":"1","Undergraduate":"2","Postgraduate":"3",})
df['Education'] = df['Education'].astype('int64')
```

3) Scaler and Reduction

Now I've created a new dataframe called dfnew which I'll use to reduce my data using the standard scaler function, and then I've decided to use the PCA method to create a

PCA_dfnew dataframe which I'll use to visualise the reduced dimensions.

```
dfnew = df.copy()
```

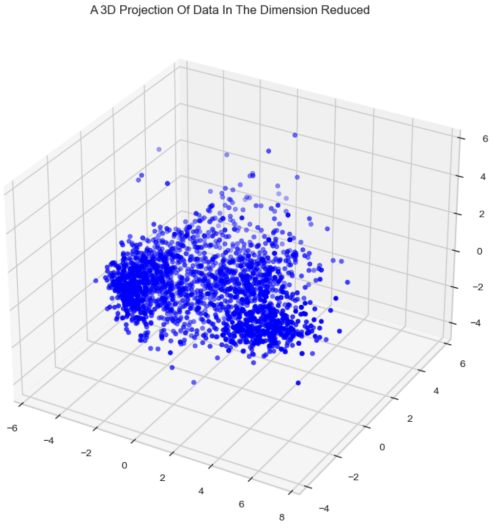
```
scaler = StandardScaler()
scaler.fit(dfnew)
scaled_dfnew = pd.DataFrame(scaler.transform(dfnew),columns= dfnew.columns )
```

```
scaled_dfnew.head()
```

Education	Marital_Status	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts
-0.411891	-1.347635	0.305306	-0.824939	-0.930615	0.306856	0.983228	1.554170	1.679746	2.461068
-0.411891	-1.347635	-0.229181	1.032627	0.905974	-0.383971	-0.871064	-0.636431	-0.713455	-0.650414
-0.411891	0.742041	0.915974	-0.824939	-0.930615	-0.798467	0.362159	0.572177	-0.177201	1.344595
-0.411891	0.742041	-1.121867	1.032627	-0.930615	-0.798467	-0.871064	-0.560893	-0.651409	-0.503991

```
pca = PCA(n_components=3)
pca.fit(scaled_dfnew)
PCA_dfnew = pd.DataFrame(pca.transform(scaled_dfnew), columns=(["col1","col2", "col3"]))
PCA_dfnew.describe().T
```

	count	mean	std	min	25%	50%	75%	max
col1	2236.0	-2.224418e-17	2.675033	-5.468214	-2.424303	-0.775813	2.310474	7.458665
col2	2236.0	3.177740e-17	1.523000	-3.803879	-1.284723	-0.116904	1.090483	5.645733
col3	2236.0	3.972175e-17	1.143294	-4.200302	-0.809319	-0.046281	0.699803	5.682971

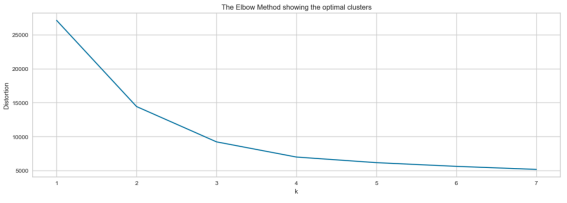


4) Clustering with Distortion Score

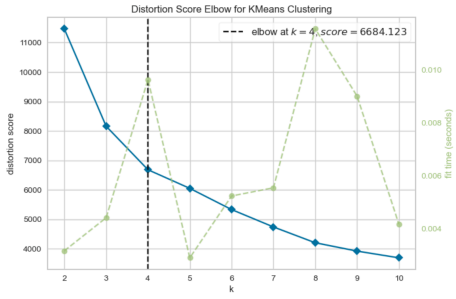
So for my clustering I used all my reduced dataframe, having removed the columns I wasn't interested in at the end of the manipulation.

Now to carry out the clustering I first used the classic Elbow method seen in class. However, the result gave me 2 clusters. So I decided to use the Distortion Score method with the Yellowbrick.cluster package. This method enabled me to obtain 4 clusters, which is satisfactory for the rest of my process.

```
distortions = []
K = range(1,8)
for k in K:
 kmeanModel = KMeans(n_clusters=k)
 kmeanModel.fit(PCA_dfnew)
 distortions.append(kmeanModel.inertia_)
# Plotting the distortions
plt.figure(figsize=(16,5))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal clusters')
plt.show()
```



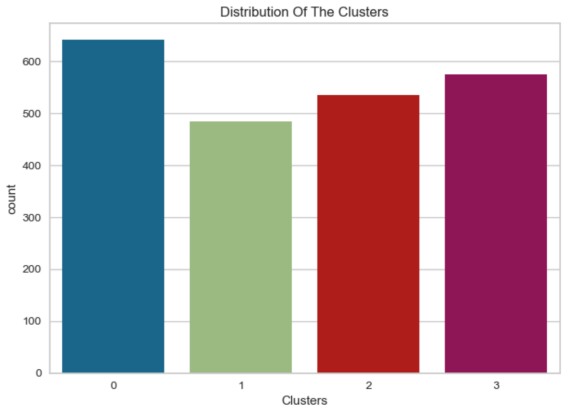

```
Elbow_M = KElbowVisualizer(KMeans(), k=10)
Elbow_M.fit(PCA_dfnew)
Elbow_M.show()
```

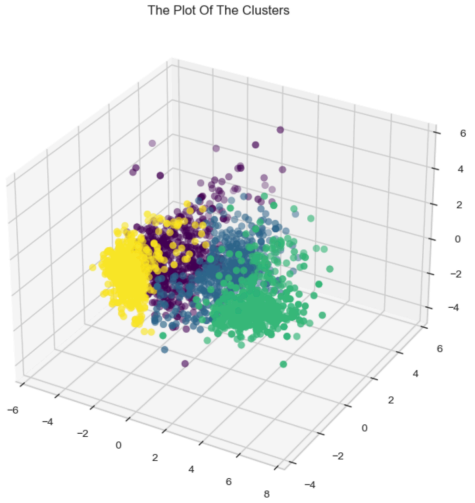


The next step is to fit the model to create 4 clusters and add the clusters to my df dataframe so that I can visualise and perform profiling.

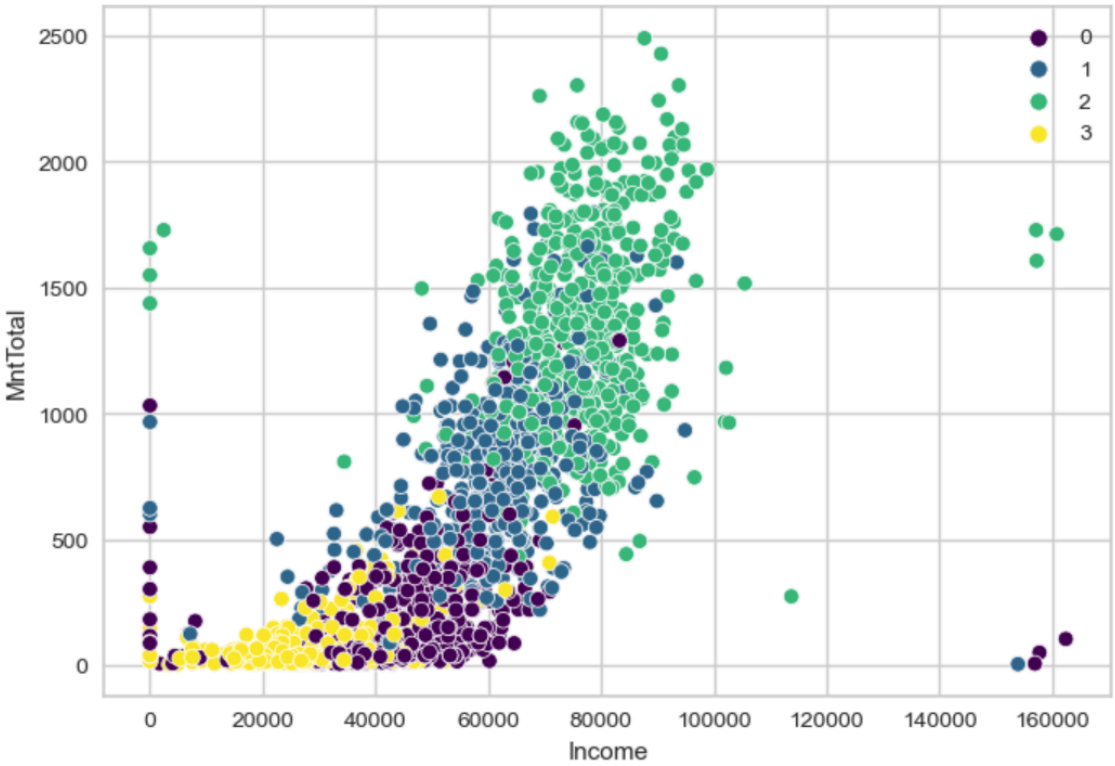
```
AC = AgglomerativeClustering(n_clusters=4)
yhat_AC = AC.fit_predict(PCA_dfnew)
PCA_dfnew["Clusters"] = yhat_AC

df["Clusters"]= yhat_AC
```

Now let's look at the 4 clusters. The distribution is fairly even and balanced, despite cluster 0 having more customers.





We can also see from the 4 clusters that they are fairly well distributed in terms of Income and amount spent



5) Profiling

Now that we've done the clustering to obtain 4 different types of customer, as well as the visualisation of these customers, here's the profiling to understand the 4 new types of customer.

Looking at the different codes produced using Pivot Table and the groupby function, we can say that :

Cluster 0 is the oldest cluster, it has the most children on average, 2/3 of the cluster's customers are in a couple, almost all of the cluster has studied to be at least a graduate, it is the 3rd cluster in terms of Income and MntTotal.

Cluster 1 is the 2nd oldest, more than half of the cluster has no children, like cluster 1 2/3 of the customers are in couples, and most of them have also studied to at least graduate level. This is the 2nd largest cluster in terms of Income and MntTotal.

Cluster 3 is the 2nd youngest. It has the fewest children (average 0.18), more than half the cluster is in a couple, and almost all have studied to be at least a graduate. It is the leading cluster in terms of Income, and therefore MntTotal.

Cluster 4 is the youngest, and also the 2nd in terms of Children (on average 0.9 per Customer). Like clusters 1 and 2, there are many more users in couples, and there are also more users who are at least Graduate, although it is the only cluster where there are more undergraduates than postgraduates. It is the last in terms of Income and therefore MntTotal.

	Clusters	Age
0	0	59.759750
1	1	58.637113
2	2	55.885981
3	3	46.198261

	Clusters	Children
0	0	1.675507
1	1	0.896907
2	2	0.183178
3	3	0.902609

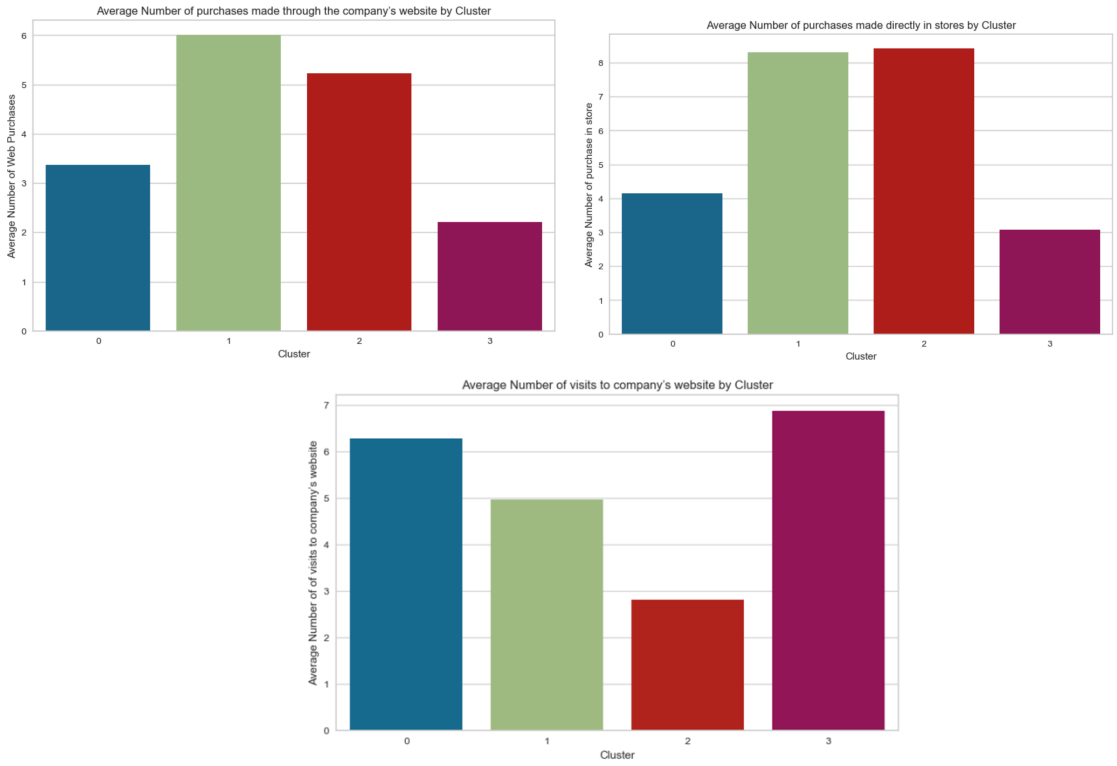
	Clusters	Marital_Status	Count
0	0	2	429
1	0	1	212
2	1	2	313
3	1	1	172
4	2	2	332
5	2	1	203
6	3	2	368
7	3	1	207

	Clusters	Education	Count
0	0	3	313
1	0	2	296
2	0	1	32
3	1	2	240
4	1	3	217
5	1	1	28
6	2	2	290
7	2	3	192
8	2	1	53
9	3	2	300
10	3	1	142
11	3	3	133

	Clusters	MntTotal
0	0	194.716069
1	1	748.779381
2	2	1353.949533
3	3	77.053913

	Clusters	Income
0	0	43944.507020
1	1	60454.412371
2	2	76395.629907
3	3	28821.579130

6) Marketing Strategy



Now the marketing strategy. Clusters 1 and 4 are the 2 clusters with the most visits to the company's website. However, they make fewer online purchases than in-store purchases and fewer online purchases than clusters 2 and 3. However, these are the clusters with the most children and the least Income. We therefore need to carry out an SEA acquisition campaign via search, display and YouTube, as well as Social Media ADS campaigns targeting these 2 clusters, with the aim of converting them into purchases. These campaigns will offer discounts for placing online orders on the brand's website for home delivery of groceries.

Clusters 2 and 3 are the clusters with the most Income but the fewest children. They buy their products on the shop's website, but also in-store. The objective of the campaign will therefore be divided into 2 parts: there will be campaigns to drive traffic to the company's website, which will also help to generate a larger audience on the site, and above all campaigns to acquire customers. The campaigns will also be broadcast in SEA

search, display and YouTube, as well as in social ads. The aim will be to convert these 2 clusters into purchases.

If the 2 acquisition campaigns are satisfactory, we can also go a step further by carrying out a lookalike to generate an audience similar to the clusters in order to expand the shop's online customer base.