DATE: 23/03/2023

Customer Segmentation Model

Mentor:

Dr Arvind Kumar

Presented By:

Zubin Relia & Anant Khemka

Objective

In this project, we will work with **customer datasets** from a company, and the goal is to **create user segments** and optimize **marketing campaigns**.

Motivation

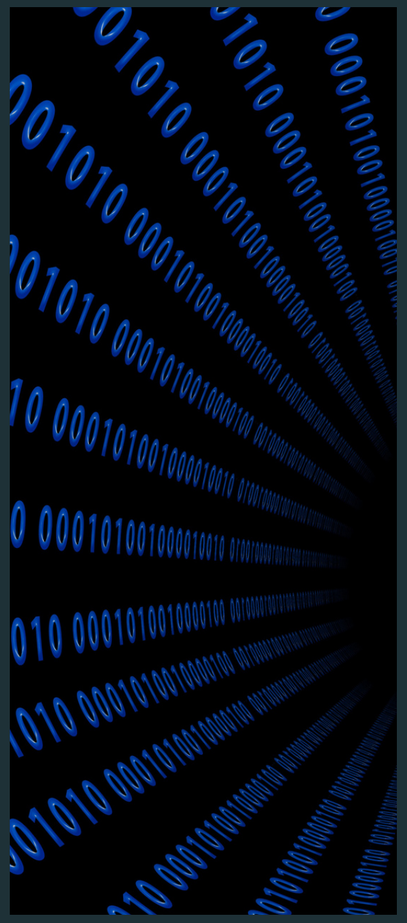
Efficiently acquiring new customers is crucial for business growth.

However, our industry experience highlights that many companies **waste resources** by targeting the **wrong audience**.

To help businesses **achieve maximum ROI**, we will **optimize the marketing strategies** by ensuring they are **targeted towards the right people**.

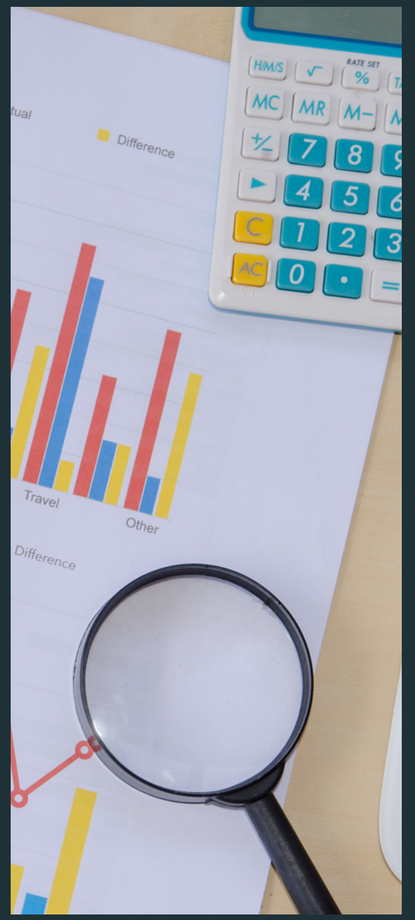
Methodology

The Methodology
involves **3 major parts**.



1. Dataset Handling

Data Cleaning &
Pre-processing



2. Dimensionality Reduction

Reducing dimensions
using PCA and t-SNE

3. Clustering

Select the best clusters
and form Cutomer
Profiles

Implementation

The implementation of the project is divided into **4 steps** :

Step # 1

Import and clean the data

Step # 2

Pre-Process the data by Encoding & Scaling

Step # 3

Reducing dimensions using PCA and t-SNE

Step # 4

Find the best clusters and form customer profiles

Importing and cleaning the data

The data cleaning heavily relied on the pandas functionality and has the following steps involved :

- The first step involved analysing the dimensionality and the **basic information** of the data

- Next **labels were renamed** for a better understanding

- Upon analysing the data, it was noticed that there are **3 major faults** within the data:
  a. Missing values in the income column
  b. Incorrect formatting of the data
  c. Categorical data needs to be encoded

Handling missing values

The handling and imputation of the missing values was done by the following methodology :

- There were 24 missing values in the income section of the dataset.
- Grouping data by education level, calculating average income, and imputing missing values were used to handle missing income data.
- The result of this methodology is that there are no longer any missing values in the income section of the dataset.

As you can see, there are 24 missing values in the Income column. These missing values need to be filled in order to procede.

```
[78] income_missing = df[df['Income'].isna()]
```

```
income_missing.shape
```

```
(24, 29)
```

```
#Calculate the mean values of income grouped by education level
group_means = df.groupby('Education')['Income'].mean()

#Impute missing values in the "Income" column
df['Income'] = df['Income'].fillna(df['Education'].map(group_means))
```

```
#Re-check the missing values of the dataset
print('Missing data: ', df.isna().sum().sum())
```

```
Missing data:  0
```

Applying feature engineering

The next step would be to apply feature engineering to simplify the dataset :

- Feature engineering was applied to reduce complexity in the dataset.
- Features such as "no_of_teenagers" and "no_of_kids" were grouped into a single feature called "family_size".
- Education and Marital Status were also simplified, with Education being classified as undergraduate or postgraduate, and Marital Status being classified as Partner or Alone.

```
#Group marital status into only two status
df['Marital_Status'] = df['Marital_Status'].apply(lambda x: "Partner" if x in {"Married", "Together"} else "Alone")


#Segment education levels in three groups
df['Education'] = df['Education'].replace({'Basic' : 'Undergrade',
                                           '2n Cycle' : 'Undergrade',
                                           'Graduation' : 'Graduate',
                                           'Master' : 'Postgraduate',
                                           'PhD' : 'Postgraduate'})


print('Values of Education levels: ', df['Education'].value_counts())
print('Values of Marital Status: ', df['Marital_Status'].value_counts())
```

```
Values of Education levels:  Graduate       1127
Postgraduate     856
Undergrade       257
Name: Education, dtype: int64
Values of Marital Status:  Partner    1444
Alone       796
Name: Marital_Status, dtype: int64
```

Pre-processing the data

Data preprocessing before clustering includes label encoding for categorical features and scaling using StandardScaler() module.

- Label encoding converts categorical features to integers (0 or 1) for machine learning and clustering.
- Scaling using StandardScaler() brings all features to a comparable range for machine learning models.

```
category_columns = ['Education', 'Marital_Status']
```

```
# create an instance of LabelEncoder
le = LabelEncoder()

#copy original dataset
ds = df.copy()
# select categorical columns
categorical_cols = ds.select_dtypes(include=['object']).columns.tolist()

# label encode each column in the list
for col in categorical_cols:
    ds[col] = le.fit_transform(ds[col])

print(ds.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2236 entries, 0 to 2239
Data columns (total 31 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Education            2236 non-null   int64
 1   Marital_Status       2236 non-null   int64
```

```
[96] df_before_scale = ds.drop(bool_columns, axis = 1)
```

```
[97] #Get data for scaling by removing boolean columns
     df_before_scale = ds.drop(bool_columns, axis = 1) #new data frame with numeric

     # create an instance of the StandardScaler class
     scaler = StandardScaler()

     # fit the scaler to the data and transform it
     scaled_data = scaler.fit_transform(df_before_scale)


     # create a new dataframe with the scaled data
     df_scaled = pd.DataFrame(scaled_data, columns=df_before_scale.columns)

     df_scaled.head()
```

	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MWines	MFruits	MMeat	...	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVis
0	-0.89438	-1.347635	0.288195	-0.824939	-0.930615	-1.303716	0.306856	0.983228	1.554170	1.679746	...	1.407639	2.509801	-0.552429	
1	-0.89438	-1.347635	-0.262715	1.032627	0.905974	-0.898592	-0.383971	-0.871064	-0.636431	-0.713455	...	-1.110921	-0.568970	-1.167738	
2	-0.89438	0.742041	0.917627	-0.824939	-0.930615	0.611419	-0.798467	0.362159	0.572177	-0.177201	...	1.407639	-0.226884	1.293496	
3	-0.89438	0.742041	-1.182829	1.032627	-0.930615	-0.677614	-0.798467	-0.871064	-0.560893	-0.651409	...	-0.751127	-0.911056	-0.552429	
4	0.57070	0.742041	0.295435	1.032627	-0.930615	0.327306	1.550344	-0.389661	0.421101	-0.217088	...	0.328256	0.115201	0.062879	

rows × 24 columns

Dimensionsionality Reduction using PCA & t-SNE

- PCA and t-SNE are dimensionality reduction techniques in data science. PCA simplifies data by finding patterns in lower-dimensional space, retaining variation, while t-SNE preserves data relationships while reducing dimensions.

- Since we have high-dimensional data containing 24 features, we decided to reduce it to 3 significant features using PCA.

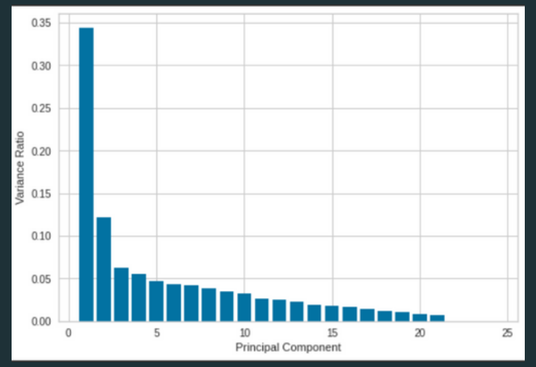
- Reducing the number of features will help with clustering.

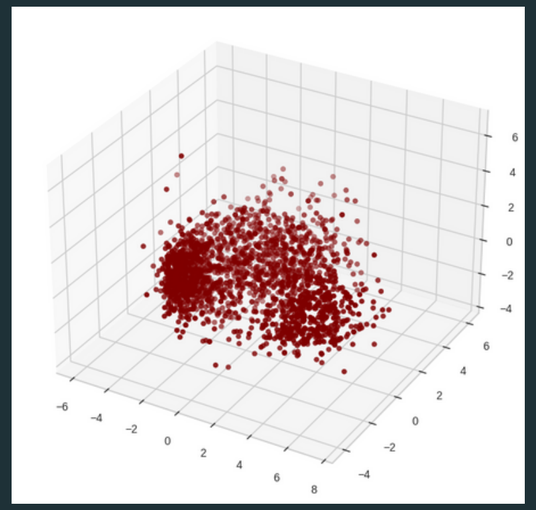
PCA

- PCA is created and **fitt**ed on scaled data, followed by **creat**ing **a bar plot** of all 25 principal components.

- The top 3 components (PC1, PC2 and PC3) are c**h**osen, accoun**t**ing for rou**gh**ly 55% of **the** v**ariance**, and fitted, transformed, and stored in a data frame.

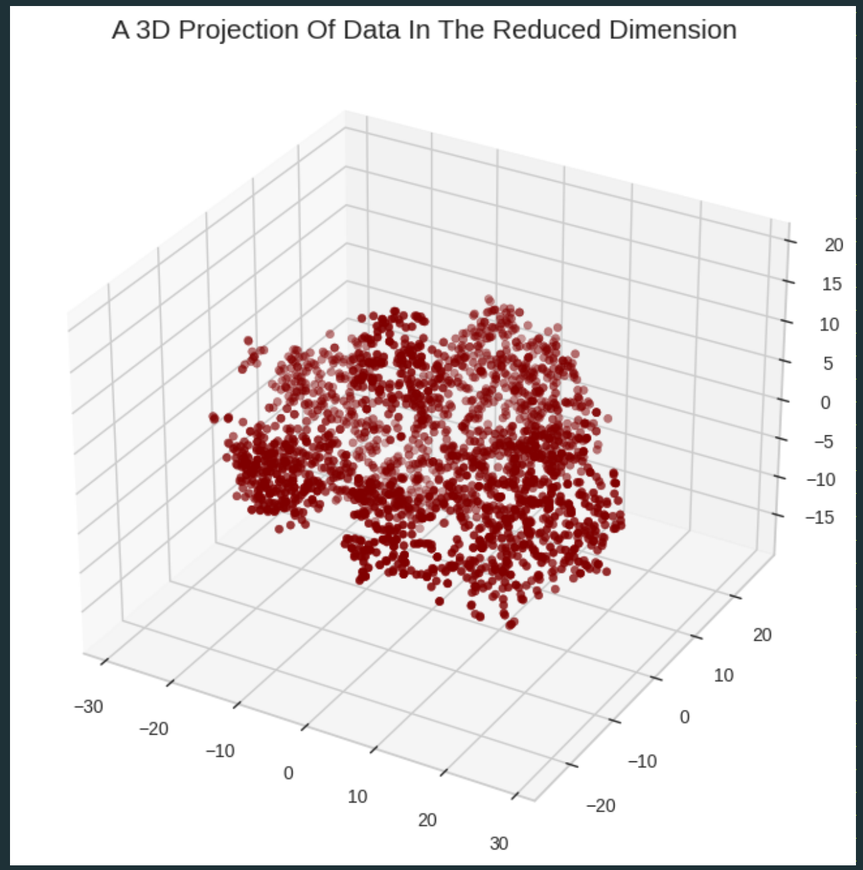
- A 3D figure is plotted using the data frame





t-SNE

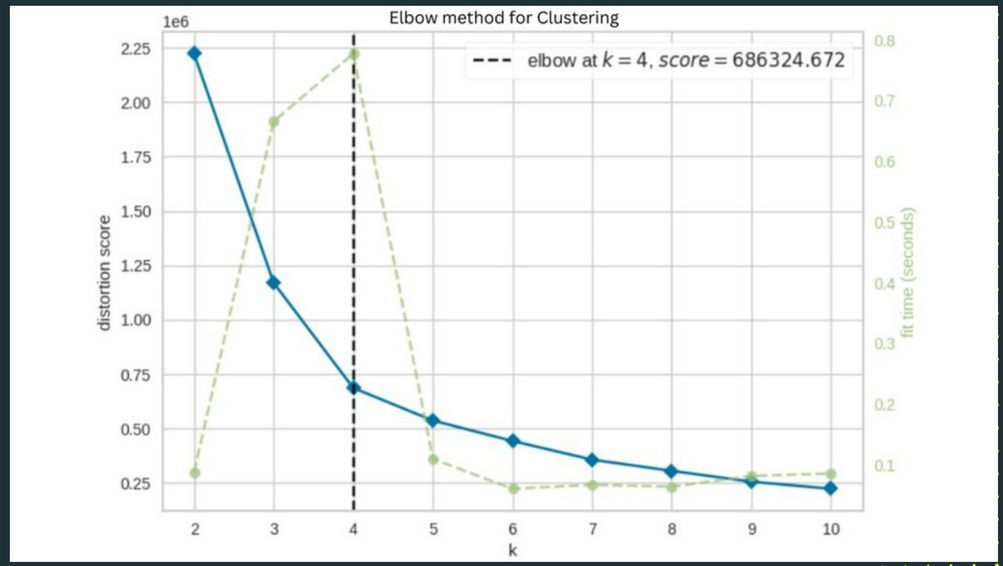
- t-SNE is applied to the df_scaled data using the TSNE function.
- The transformed data is stored in a new data frame called df_tsne, with 3 columns named D1, D2 and D3.
- The resulting t-SNE visualization is plotted as a 3d scatter plot.



Clustering

- Clustering is an unsupervised machine-learning technique that groups similar data points into clusters based on inherent similarities or differences, without prior knowledge of the data labels.

- Using the elbow method we determined the number of clusters to be formed as 4.

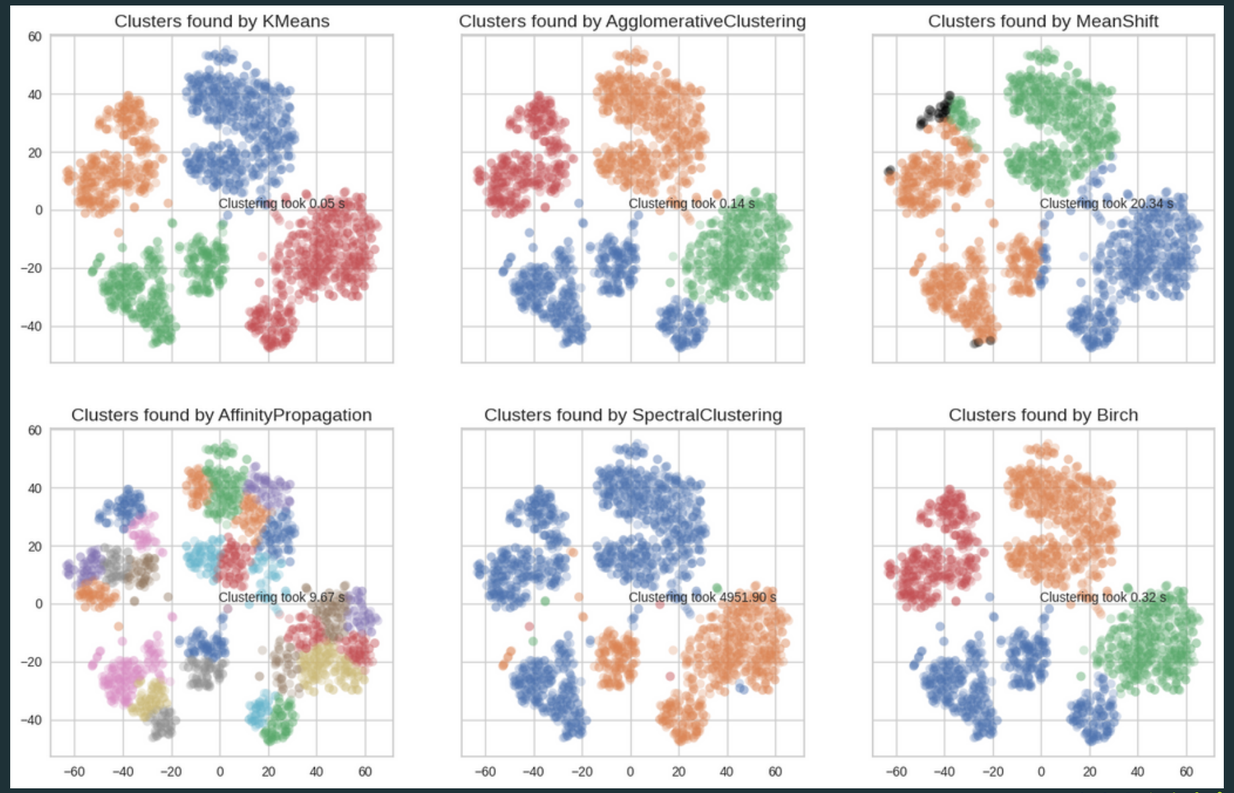


Applying Clustering after PCA

- Applying various clustering methods after PCA.
- Best clusters formed using K-means with the highest silhouette score of 0.374
- As K-means clustering has the highest silhouette score we select the K-means cluster for comparison.

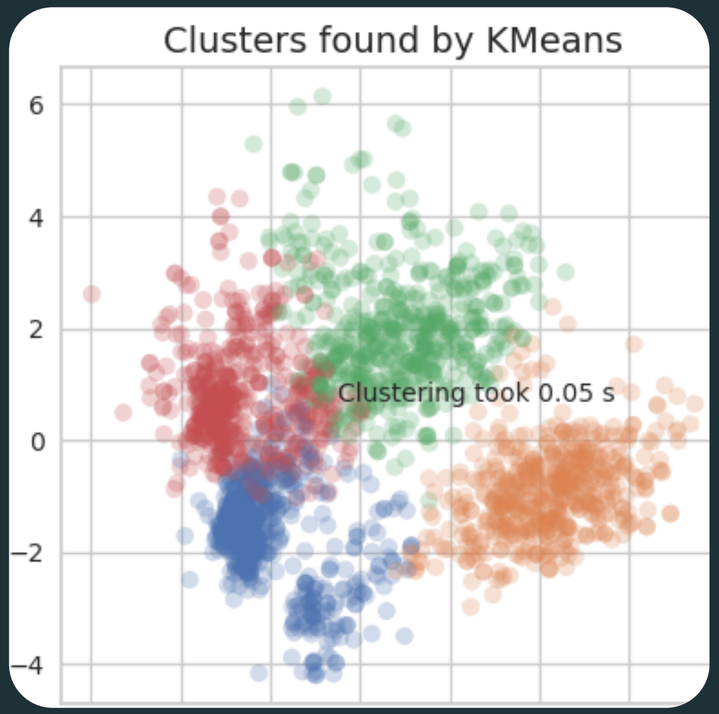
Applying Clustering after t-SNE

- Applying various clustering methods after t-SNE.
- Best clusters formed using K-means with the highest silhouette score of 0.408
- As K-means clustering has the highest silhouette score we select the K-means cluster for comparison.



Comparing the Clusters

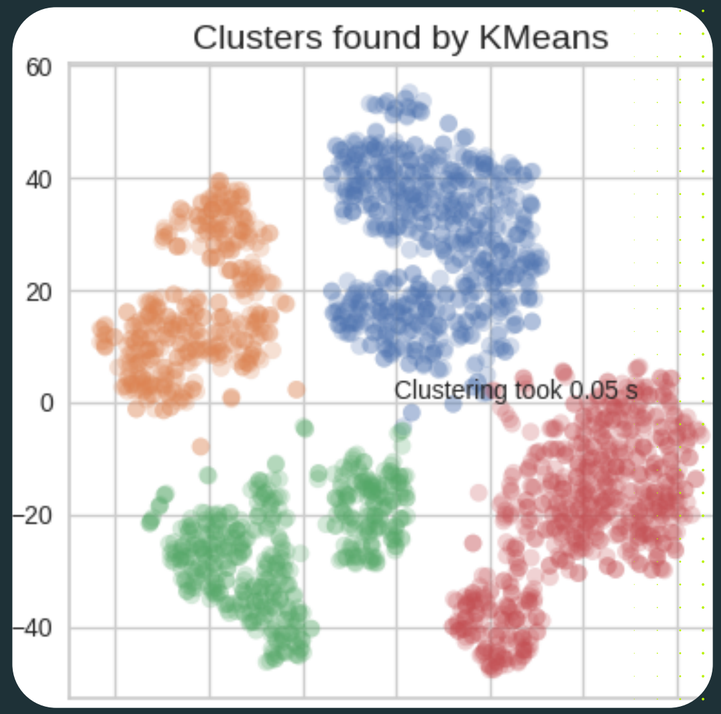
PCA



Silhouette score:
0.374

<

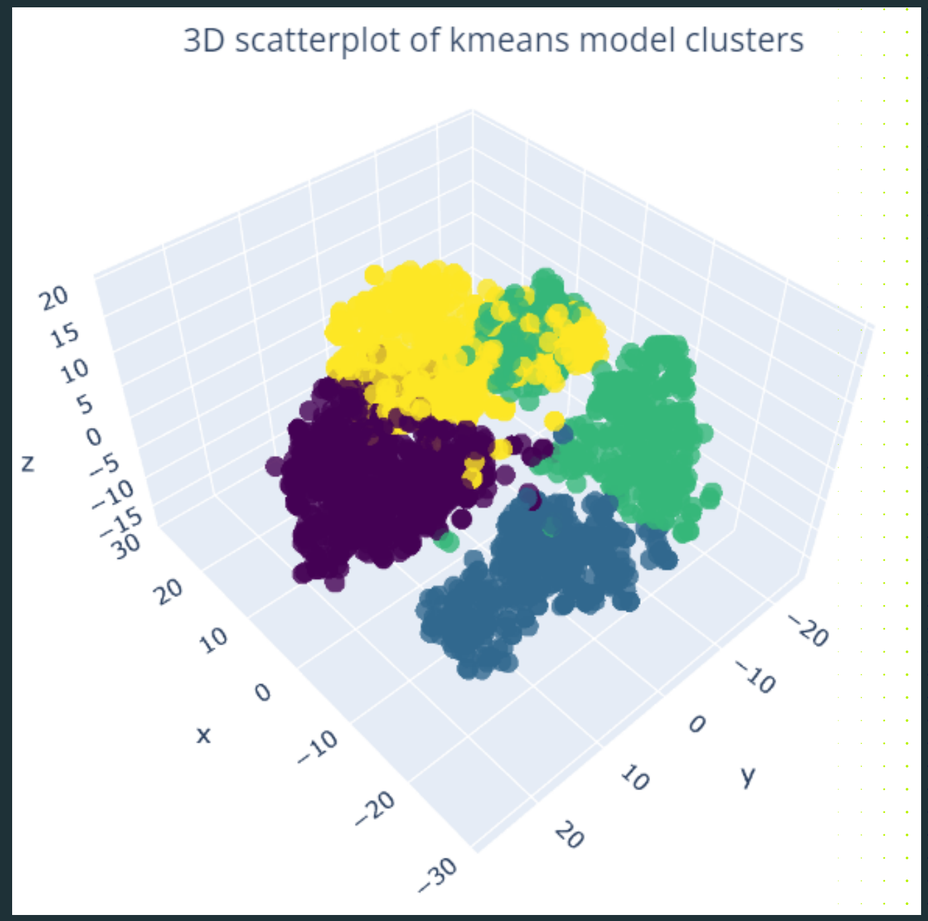
t-SNE



Silhouette score:
0.408

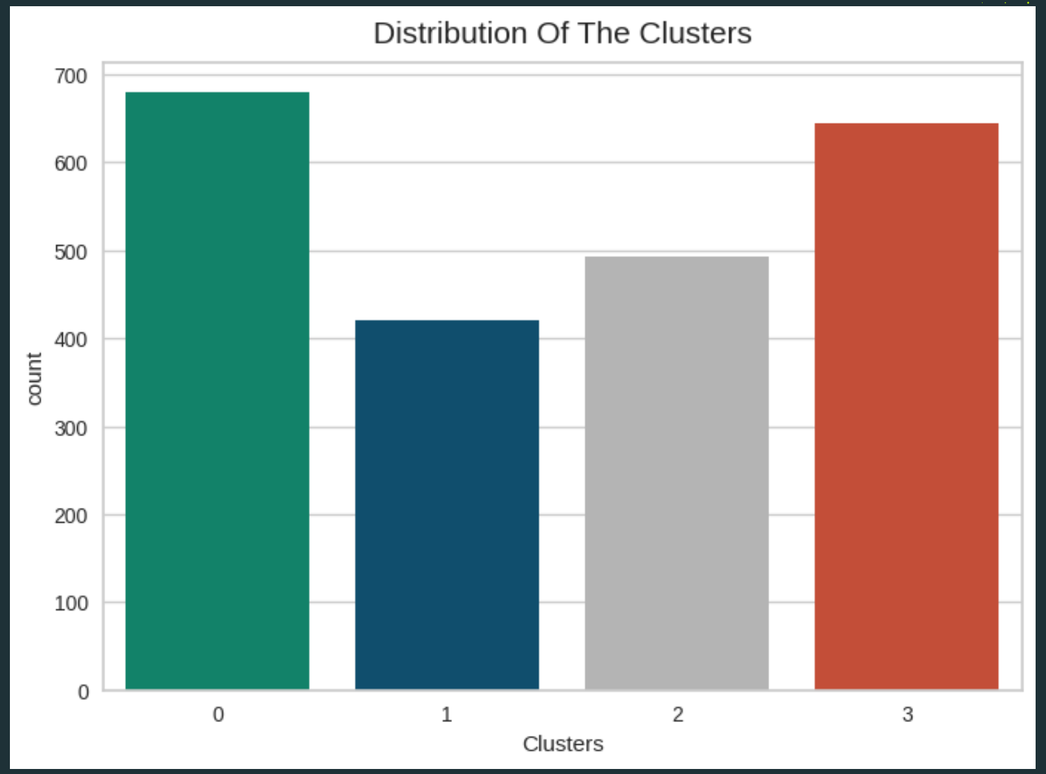
Final Cluster

- The algorithm used for Dimensionality Reduction: t-SNE
- The algorithm used for Clustering: K-means
- Silhouette Score: 0.408



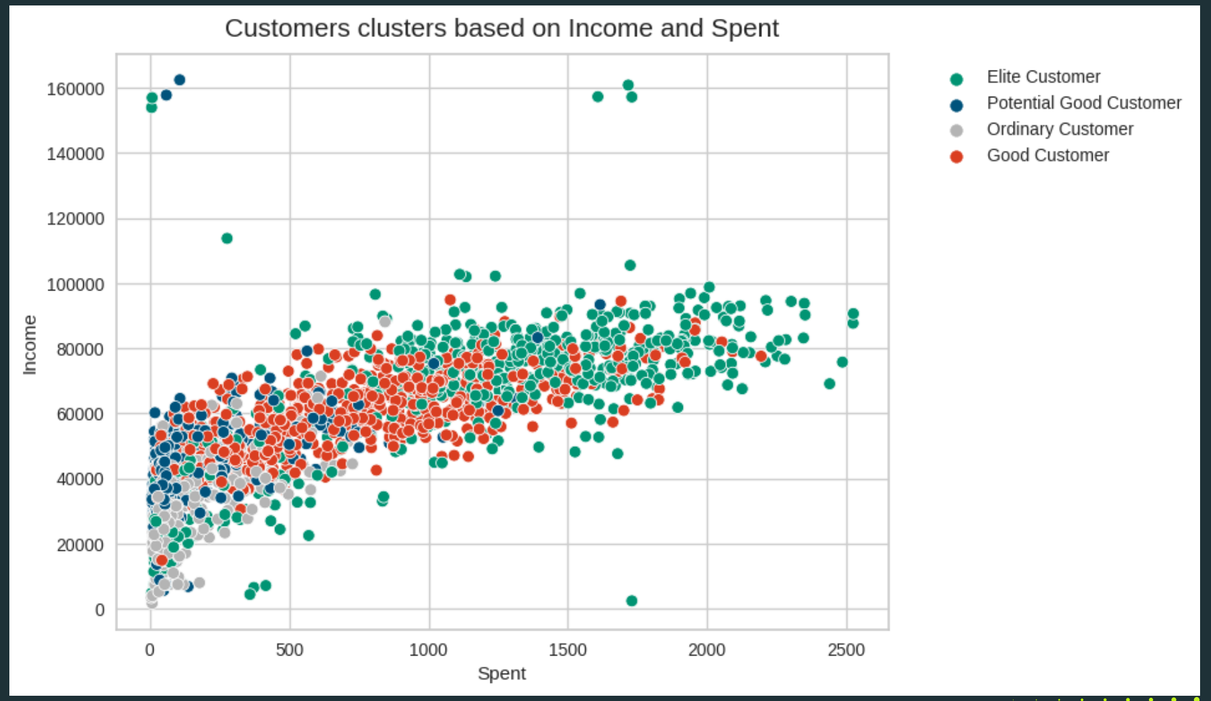
Distribution of the Clusters

- The distribution is relatively uniform.
- Clusters 0 and 1 have the highest and lowest number of customers respectively.



Renaming Clusters

- As income and spending have the most significant contributions to the distribution, clusters are plotted based on these variables.
- Based on the distribution of the data points, the clusters are given labels.
- Elite customers being the people with high income & high spending and Ordinary customer being the people with low income and low spending.

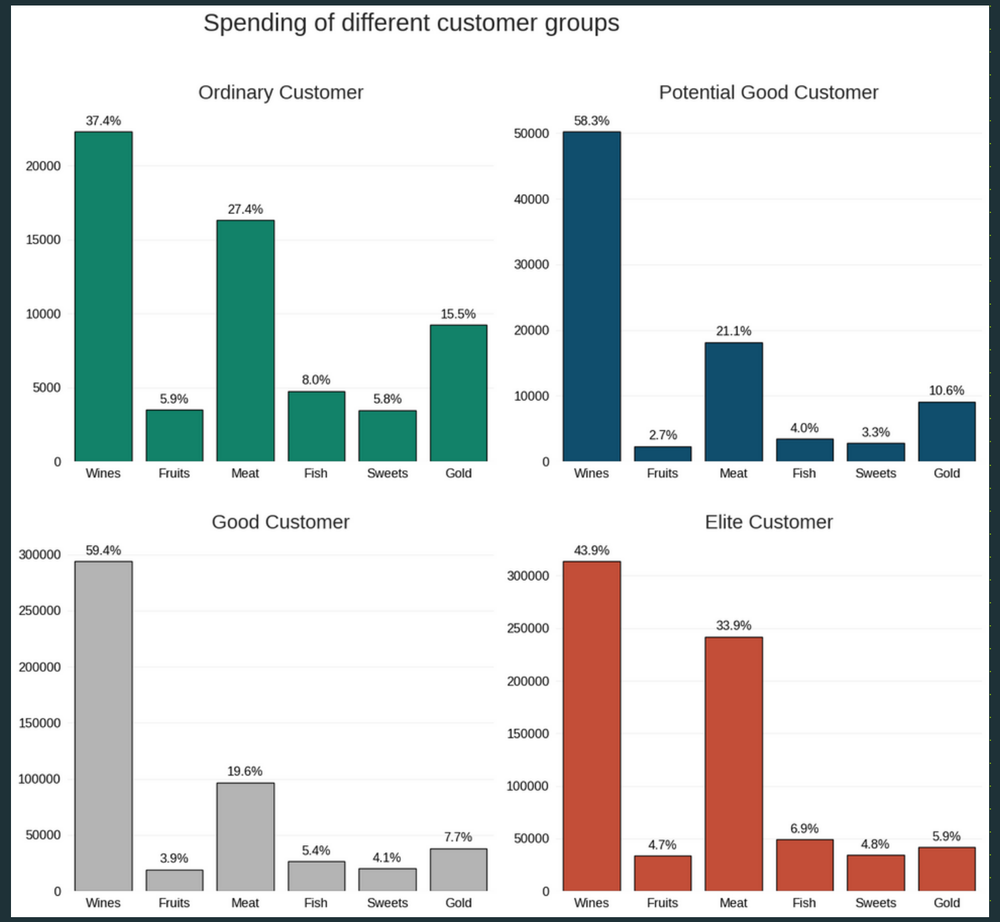


Spending of the Customer Groups

- All four groups spent the highest amount on **Wines**, followed by **Meat.**

- Ordinary and potentially good customers have higher percentages (17.7% and 10.5%) of purchasing gold compared to the other two groups (8% and 5.4%).

- Elite customers spend around £240,000 (33.9%) on Meat, which is more than the other groups' spending (less than 26%).

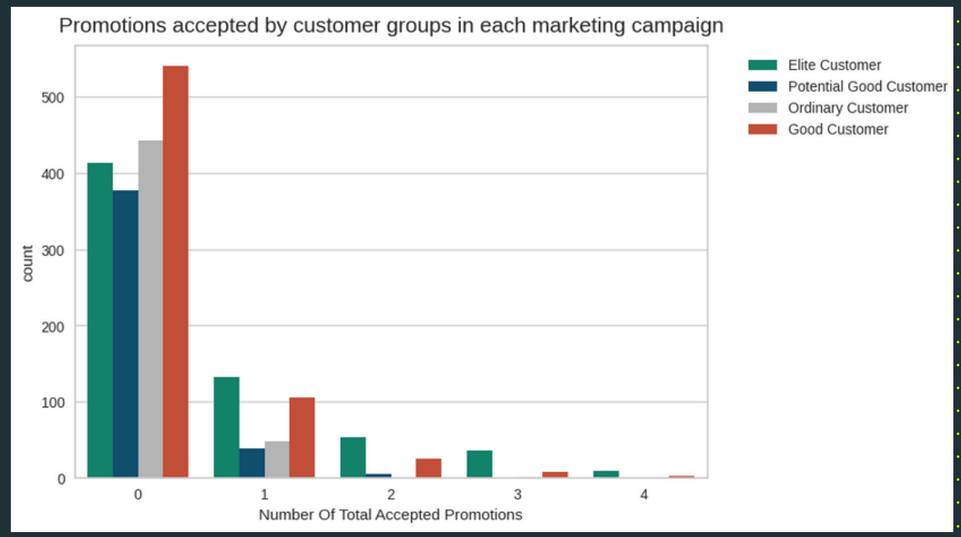


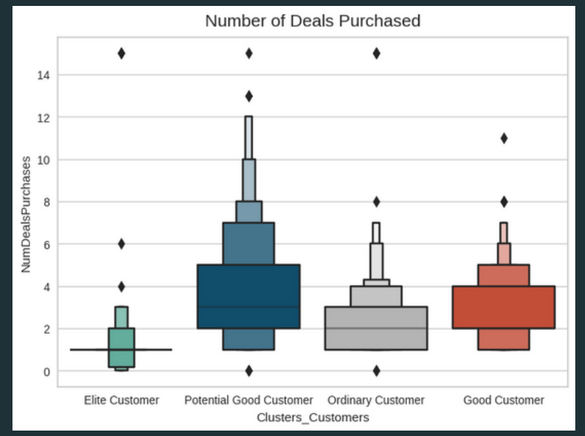
Promotions and Deals

- Initial campaign: 500 acceptan**ces** each for Ordinary/Potentially good customers, 400 and 300 for Good/Elite customers.

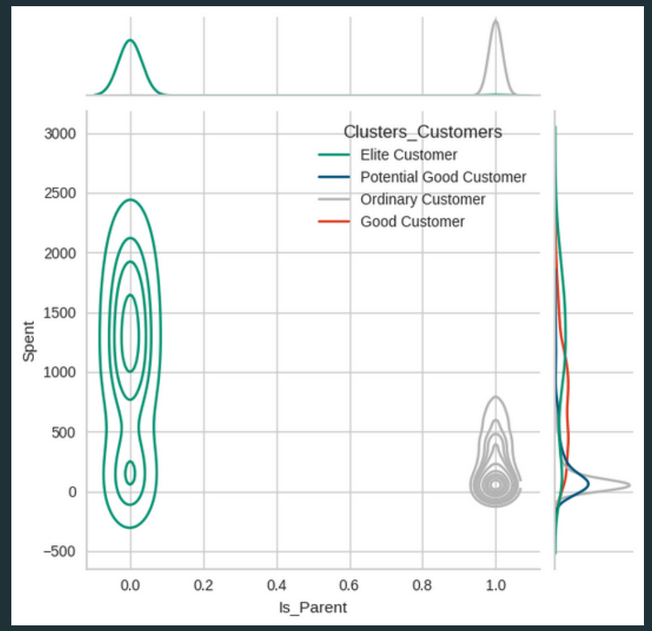
- Subsequent campaigns: significant decline in accepted promotions, esp. for Ordinary/Potentially good customers, small portion for Good/Elite.

- The deals offered were more successful for the groups of Good and Potential good customers, but not for the group of Elite customers.

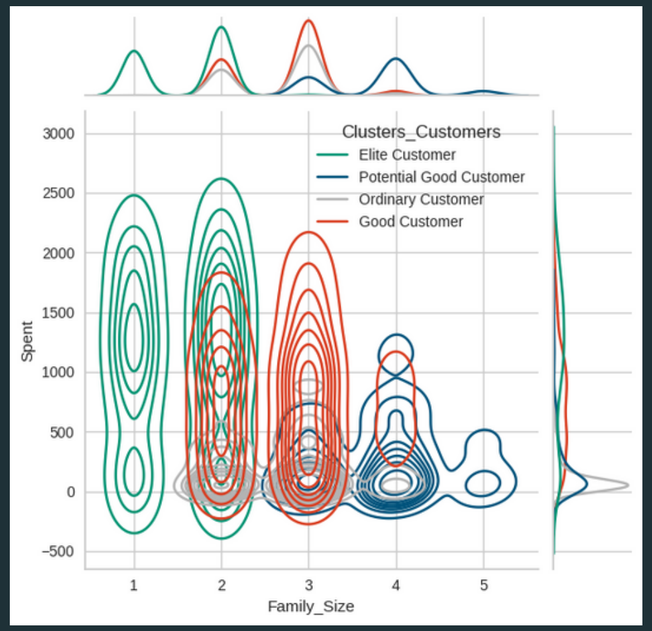




Analyzing Customer Data

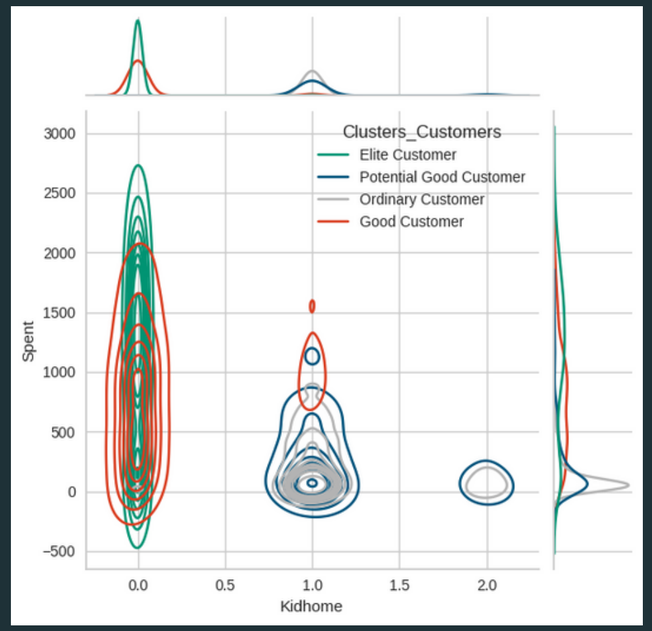


Which Customers are parents?

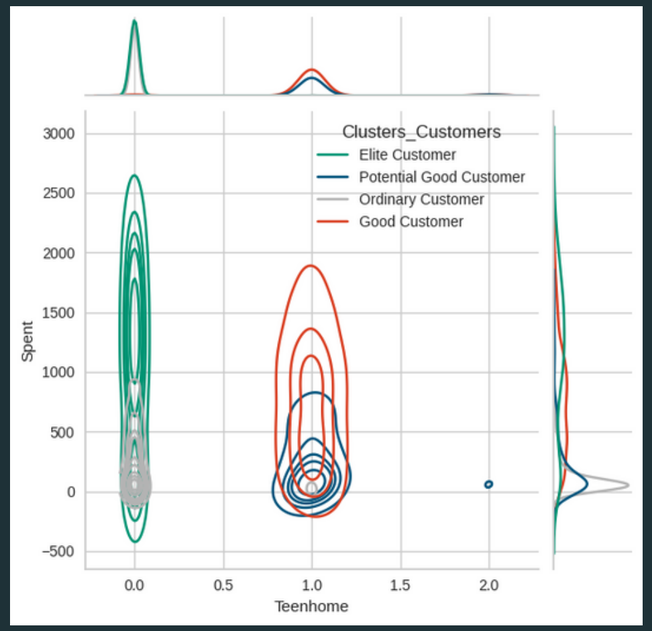


What is the family size of our customers

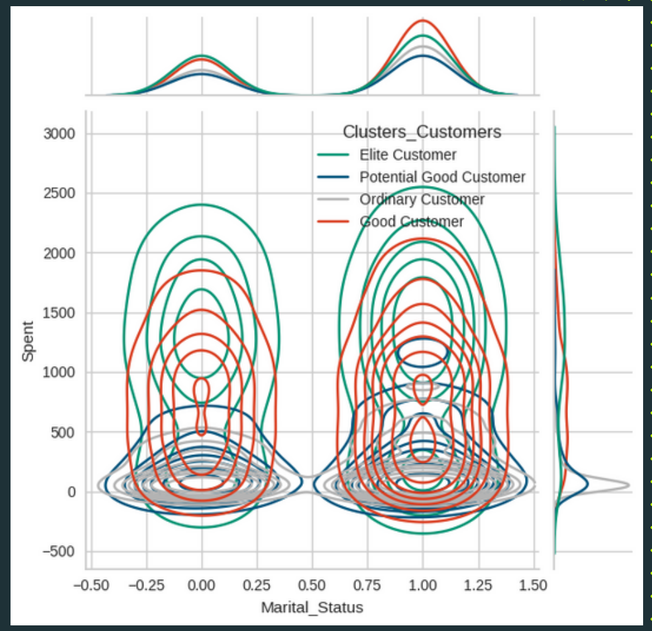
Analyzing Customer Data



How many little children do our customers have?

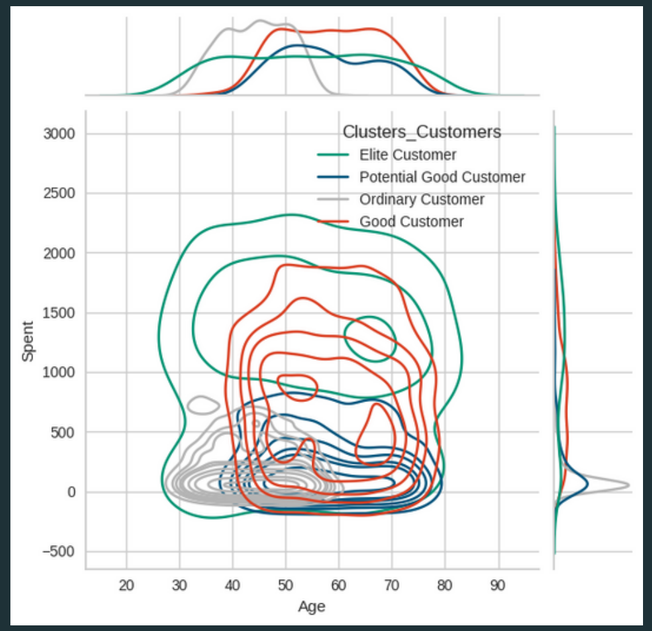


How many teenagers do our customers have?

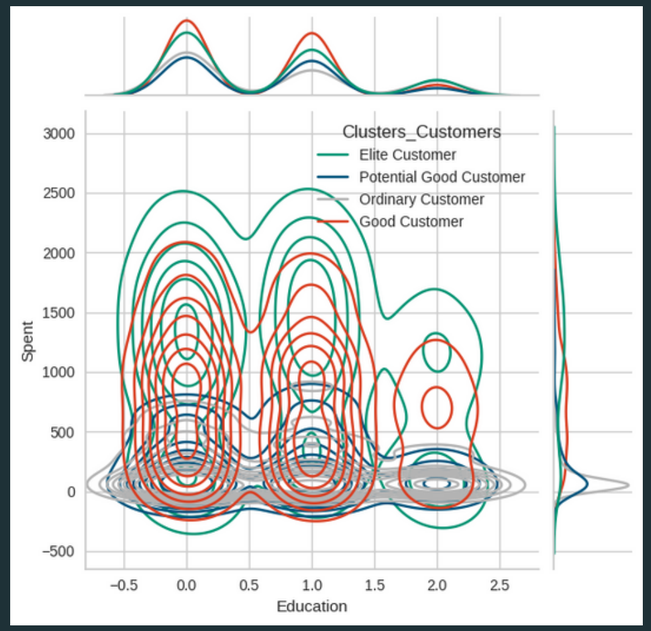


Customer marital status

Analyzing Customer Data



Customers age



Customers education

Analyzing Customer Profiles

Elite Customer

- High spending and high income
- Not parents
- Maximum family size of 2, no children
- There are greater number of couples than single ones.

Good Customer

- High spending and average income
- Have children.
- Most of them have teenagers.
- Family size ranges from 2-4
- Middle aged

Potentially Good Customer

- Low spending and average income
- Family size varies from 3-6
- Have more than 2 children
- Have both teens and kids.

Ordinary Customer

- Low spending and low income
- Most of them are parents.
- Maximum family size is 4.
- Typically have 1 child (usually a kid)

A Wine Ad **before** analysing Customer Profile

A Wine Ad **after** analysing Customer Profile

Conclusion

Customer segmentation using dimensional reduction and clustering is a powerful technique for businesses to identify groups of customers with similar characteristics and tailor their marketing strategies accordingly, leading to increased customer satisfaction and loyalty. As data analytics and machine learning techniques continue to advance, we can expect even more sophisticated segmentation methods in the future. Our future plans include further data analysis and exploration of new techniques, such as deep learning, to enhance our customer segmentation strategies and maintain a competitive advantage in the marketplace.

Thank You for listening!

Zubin Relia

Btech CSE - 6B

Reg no: 209301005

Anant Khemka

Btech CSE-6A

Reg no: 209301508