Data Analyst Intern at Data Glacier

Week 13: Final Report

Project: Customer Segmentation

Name: Blessed Adjei-Gyan

University: Paderborn University

Email: blessedadjei13@gmail.com

Country: Germany

Specialization: Data Analyst

Internship Batch: LISUM 39

Date:27th January, 2024

Table of Contents

I. Problem Description

XYZ Bank wants to segment its customers into no more than five distinct groups to send personalized Christmas offers. The goal is to automate the process, uncover hidden patterns in customer behavior, and improve campaign efficiency. The segmentation model should help target relevant customer groups with tailored offers, optimizing engagement and conversion rates.

II. Business Understanding

Problem Context: XYZ Bank is looking for a data-driven solution to divide its customer base into groups that have similar behaviors, so that they can create targeted offers.
Objective: Create a machine learning model to perform segmentation, aiming to group customers into 5 or fewer segments.

III. Project Lifecycle

Week 8 (26 Dec 2024): Submit data intake report with initial EDA.
Week 9 (2 Jan 2025): Deliver advanced EDA and feature engineering insights.
Week 10 (9 Jan 2025): Propose model-building plan and clustering approach.
Week 11 (16 Jan 2025): Present EDA findings and modeling technique.
Week 12 (23 Jan 2025): Finalize model, segmentation results, and dashboard.
Week 13 (30 Jan 2025): Submit final report and complete code repository.

1. Data Understanding

1.1. Data Size

The data contains 1000000 row and 48 columns.

In [2]:
```
data
```

Out[2]:

	Unnamed: 0	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	fecha_alta	ind_nuevo	antiguedad	...	ind_hip_fin_ult1	ind_plan_fin_ult1
0	0	2015-01-28	1375586	N	ES	H	35	2015-01-12	0.0	6	...	0	0
1	1	2015-01-28	1050611	N	ES	V	23	2012-08-10	0.0	35	...	0	0
2	2	2015-01-28	1050612	N	ES	V	23	2012-08-10	0.0	35	...	0	0
3	3	2015-01-28	1050613	N	ES	H	22	2012-08-10	0.0	35	...	0	0
4	4	2015-01-28	1050614	N	ES	V	23	2012-08-10	0.0	35	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
999995	999995	2015-02-28	1183296	N	ES	H	27	2013-09-25	0.0	22	...	0	0
999996	999996	2015-02-28	1183295	N	ES	H	56	2013-09-25	0.0	22	...	0	0
999997	999997	2015-02-28	1183294	N	ES	V	39	2013-09-25	0.0	22	...	0	0
999998	999998	2015-02-28	1183293	N	ES	V	36	2013-09-25	0.0	22	...	0	0
999999	999999	2015-02-28	1183289	N	ES	H	38	2013-09-25	0.0	22	...	0	0

1000000 rows × 48 columns

In [3]:
```
data.shape
```

Out[3]:
```
(1000000, 48)
```

```
duplicates = data.duplicated().sum()
print(f"\nNumber of Duplicate Rows: {duplicates}")
```

```
Number of Duplicate Rows: 0
```

1.2. Data Types

Data types include int64, object, float64

In [4]:
```
print("Basic Dataset Information:")
print(data.info())

print("\nBasic Statistics:")
print(data.describe())

print("\nFirst 5 Rows of Dataset:")
print(data.head())
```

```
Basic Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 48 columns):
 #   Column                 Non-Null Count    Dtype
---  ------                 --------------    -----
 0   Unnamed: 0             1000000 non-null  int64
 1   fecha_dato             1000000 non-null  object
 2   ncodpers               1000000 non-null  int64
 3   ind_empleado           989218 non-null   object
 4   pais_residencia        989218 non-null   object
 5   sexo                   989214 non-null   object
 6   age                    1000000 non-null  object
 7   fecha_alta             989218 non-null   object
 8   ind_nuevo              989218 non-null   float64
 9   antiguedad             1000000 non-null  object
 10  indrel                 989218 non-null   float64
 11  ult_fec_cli_1t         1101 non-null     object
 12  indrel_1mes            989218 non-null   float64
```

2. Data Cleaning

2.1. Renaming Unnecessary Columns

The column names were renamed for better understanding. The column description given by the company were used for the renaming.

In [7]:
```
# Define a dictionary with old column names as keys and new descriptive names as values
column_renames = {
    'fecha_dato': 'Partition_Date',
    'ncodpers': 'Customer_Code',
    'ind_empleado': 'Employee_Index',
    'pais_residencia': 'Country_Residence',
    'sexo': 'Gender',
    'age': 'Age',
    'fecha_alta': 'First_Contract_Date',
    'ind_nuevo': 'New_Customer_Index',
    'antiguedad': 'Customer_Seniority',
    'indrel': 'Primary_Customer_Status',
    'ult_fec_cli_1t': 'Last_Primary_Customer_Date',
    'indrel_1mes': 'Customer_Type_Begin_Month',
    'tiprel_1mes': 'Customer_Relation_Begin_Month',
    'indresi': 'Residence_Index',
    'indext': 'Foreigner_Index',
    'conyuemp': 'Spouse_Index',
    'canal_entrada': 'Joining_Channel',
    'indfall': 'Deceased_Index',
    'tipodom': 'Address_Type',
    'cod_prov': 'Province_Code',
    'nomprov': 'Province_Name',
    'ind_actividad_cliente': 'Customer_Activity_Index',
    'renta': 'Gross_Household_Income',
    'ind_ahor_fin_ult1': 'Saving_Account',
    'ind_aval_fin_ult1': 'Guarantees',
    'ind_cco_fin_ult1': 'Current_Accounts',
    'ind_cder_fin_ult1': 'Derivada_Account',
    'ind_cno_fin_ult1': 'Payroll_Account',
    'ind_ctju_fin_ult1': 'Junior_Account',
    'ind_ctma_fin_ult1': 'Más_Particular_Account',
    'ind_ctop_fin_ult1': 'Particular_Account',
    'ind_ctpp_fin_ult1': 'Particular_Plus_Account',
```

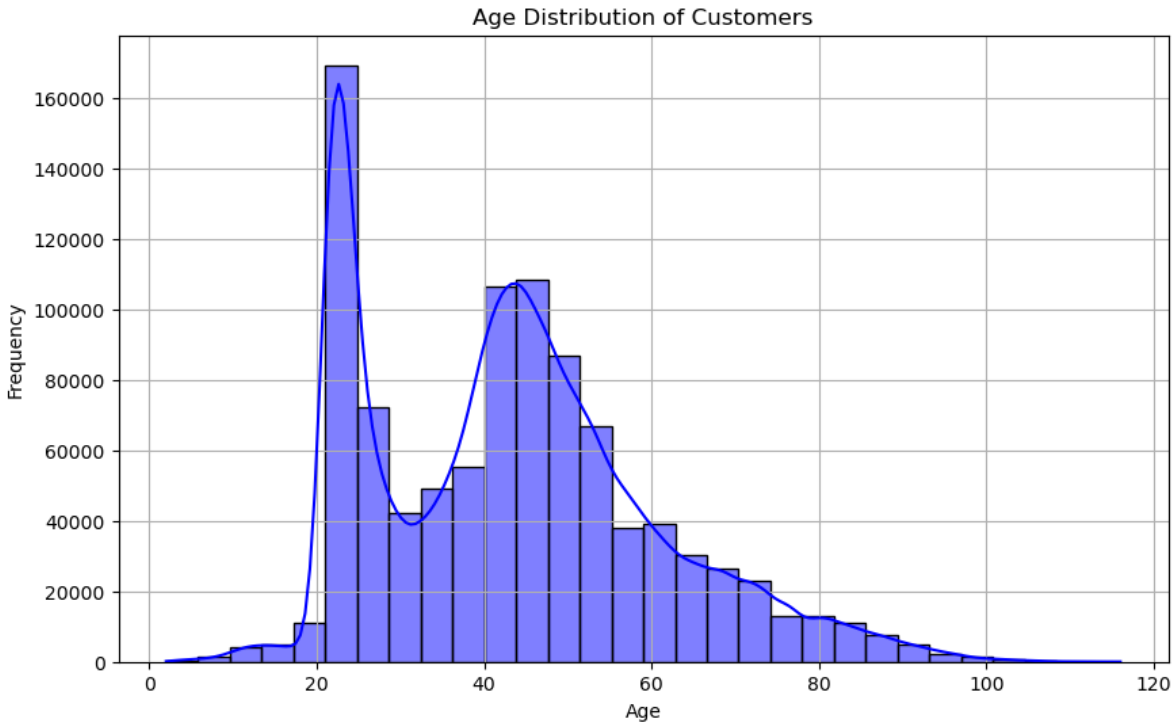
2.2. Data Mapping

Categorical values were mapped in multiple columns to more meaningful labels for better readability and analysis. It converts encoded values for **Employee Index, Gender, Residence Index, Foreigner Index, and Deceased Index** into understandable terms.

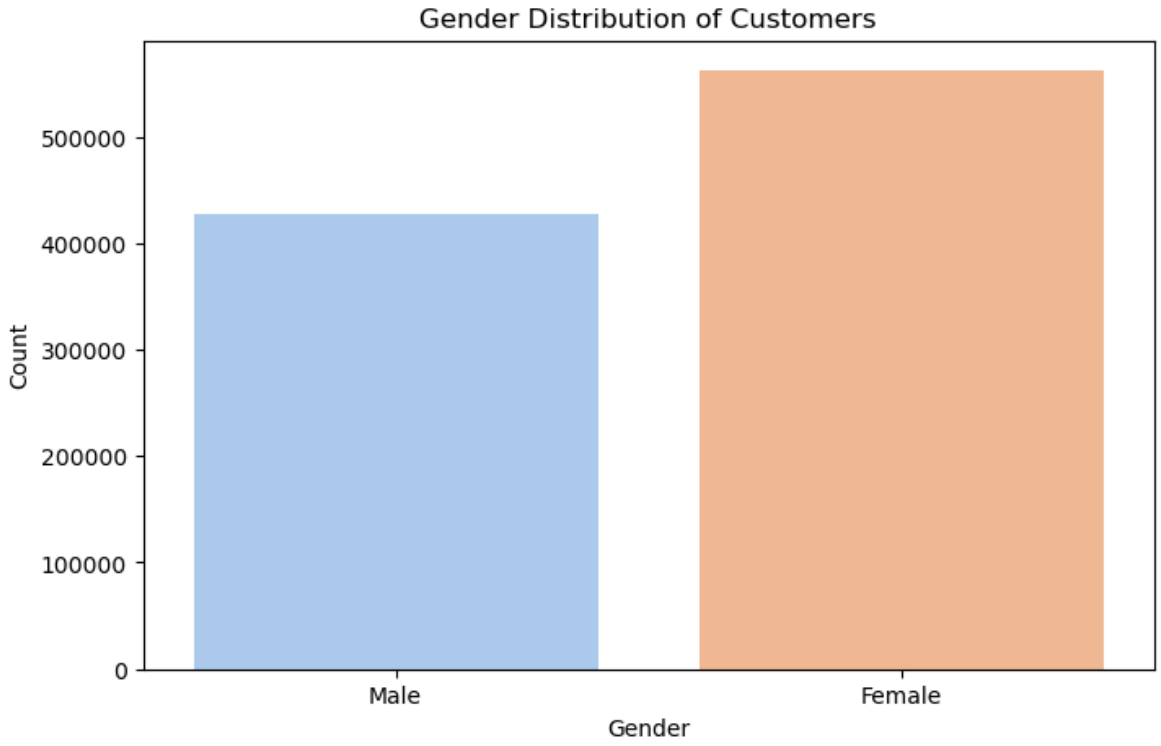
In [8]:
```python
# Map 'Employee_Index' values to meaningful labels
employee_index_map = {
    'A': 'Active',
    'B': 'Ex-Employee',
    'F': 'Filial',
    'N': 'Not_Employee',
    'P': 'Passive'
}

data['Employee_Index'] = data['Employee_Index'].map(employee_index_map)

# Map 'Gender' values to meaningful labels
gender_map = {
    'H': 'Male',
    'V': 'Female'
}

data['Gender'] = data['Gender'].map(gender_map)

# Map 'Residence_Index' values to meaningful labels
residence_index_map = {
    'S': 'Resident',
    'N': 'Non-Resident'
}

data['Residence_Index'] = data['Residence_Index'].map(residence_index_map)

# Map 'Foreigner_Index' values to meaningful labels
foreigner_index_map = {
    'S': 'Foreigner',
    'N': 'Local'
}

data['Foreigner_Index'] = data['Foreigner_Index'].map(foreigner_index_map)

# Map 'Deceased_Index' values to meaningful labels
deceased_index_map = {
    'S': 'Deceased',
```

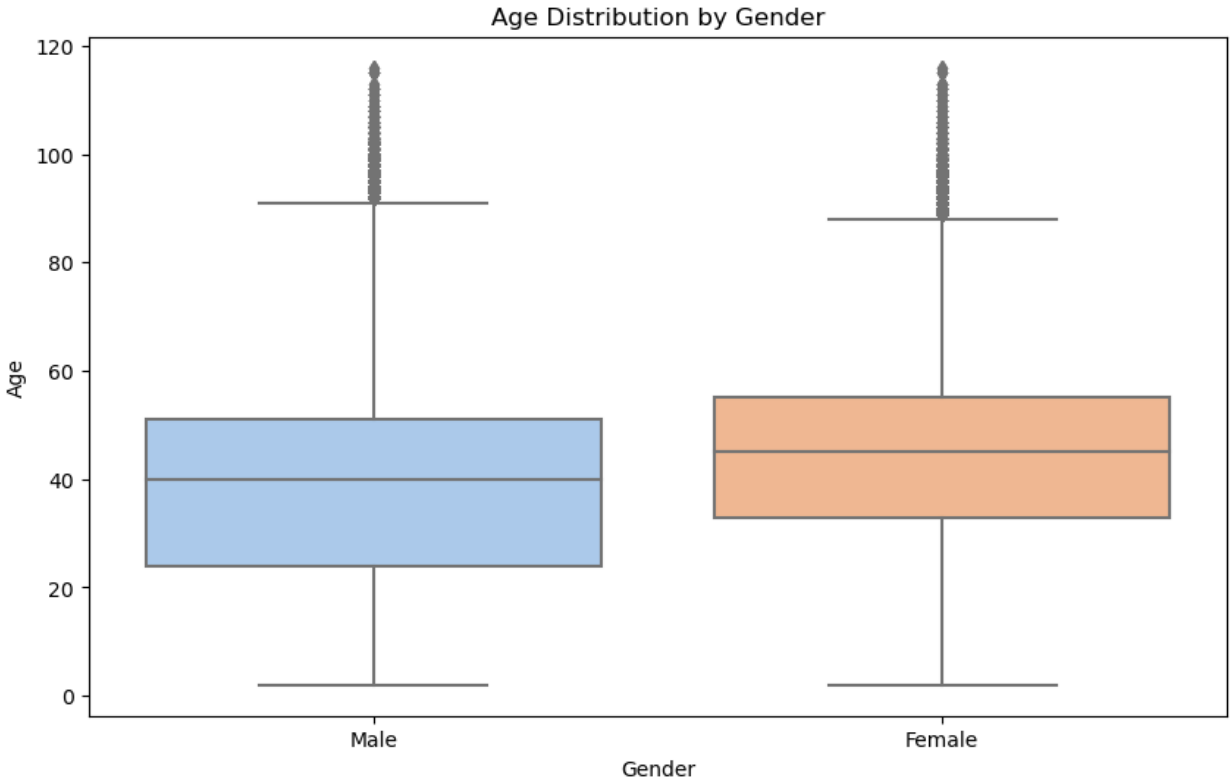
3. Exploratory Data Analysis

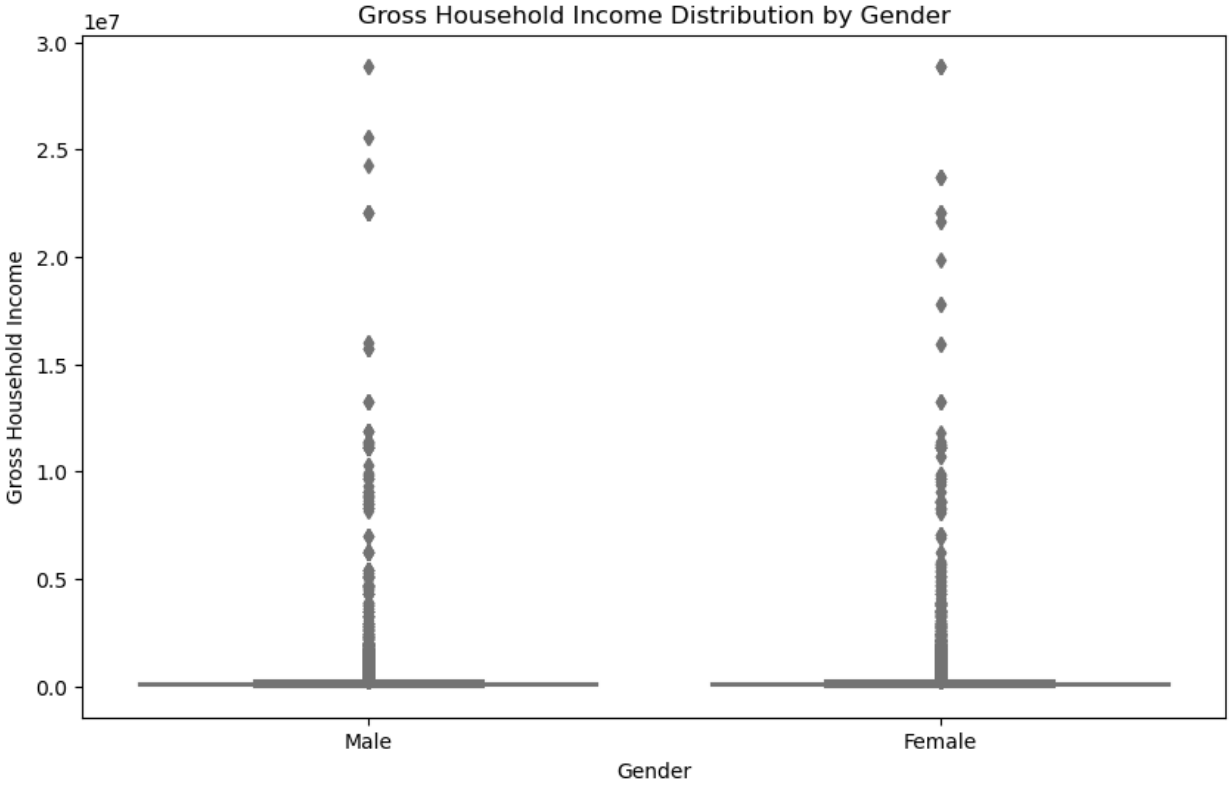
From the visualization below, we can see that most customers belong in the age range of 20-60. The number of adults is 42.8% higher than the rest of the population.

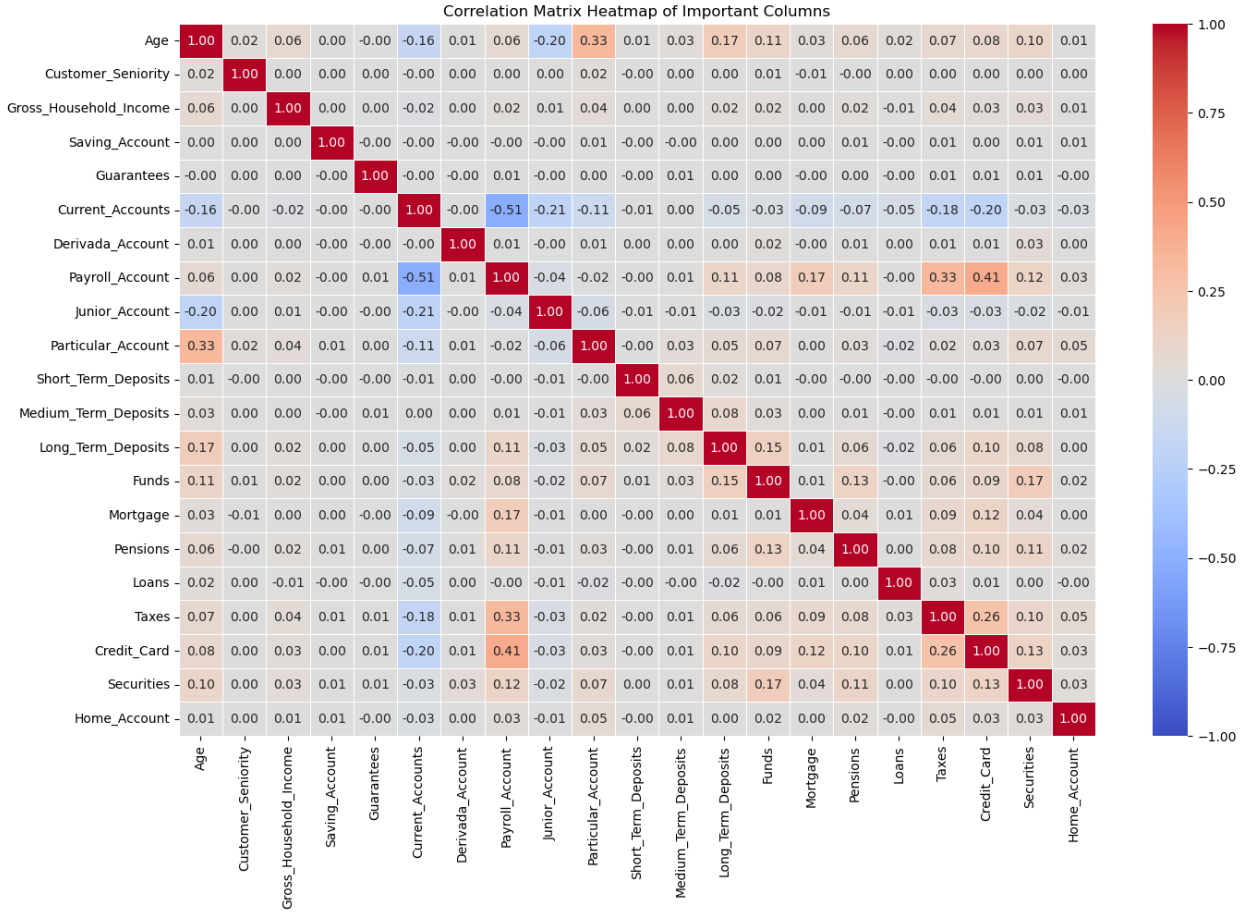


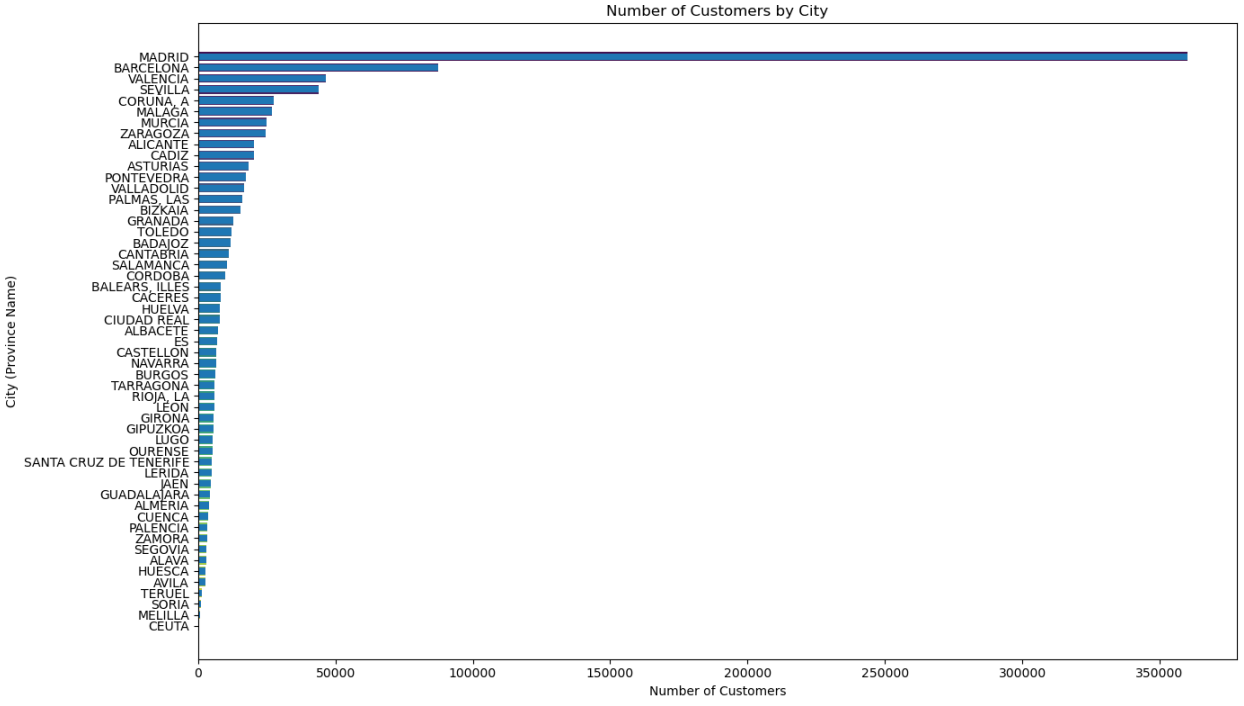
43.2% of customers are women and 56.8% are men.




Age Distribution by Gender
120
100
80
60
40
20
0
Age
Male
Female
Gender

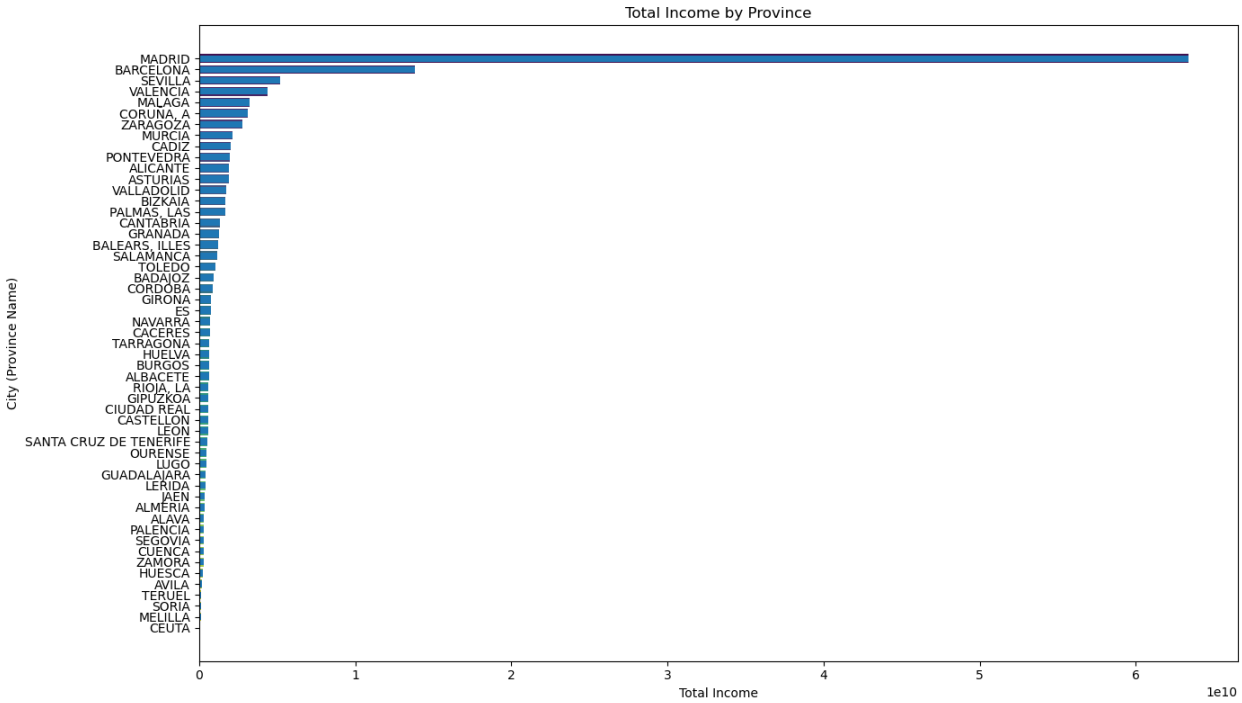

Gross Household Income Distribution by Gender
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Gross Household Income
Male
Female
Gender


Correlation Matrix Heatmap of Important Columns
Age 1.00 0.02 0.06 0.00 -0.00 -0.16 0.01 0.06 -0.20 0.33 0.01 0.03 0.17 0.11 0.03 0.06 0.02 0.07 0.08 0.10 0.01
Customer_Seniority 0.02 1.00 0.00 0.00 0.00 -0.00 0.00 0.00 0.00 0.02 -0.00 0.00 0.00 0.01 -0.01 -0.00 0.00 0.00 0.00 0.00 0.00
Gross_Household_Income 0.06 0.00 1.00 0.00 0.00 -0.02 0.00 0.02 0.01 0.04 0.00 0.00 0.02 0.02 0.00 0.02 -0.01 0.04 0.03 0.03 0.01
Saving_Account 0.00 0.00 0.00 1.00 -0.00 -0.00 -0.00 -0.00 -0.00 0.01 -0.00 -0.00 0.00 0.00 0.00 0.01 -0.00 0.01 0.00 0.01 0.01
Guarantees -0.00 0.00 0.00 -0.00 1.00 -0.00 -0.00 0.01 -0.00 0.00 -0.00 0.01 0.00 0.00 -0.00 0.00 -0.00 0.01 0.01 0.01 -0.00
Current_Accounts -0.16 -0.00 -0.02 -0.00 -0.00 1.00 -0.00 -0.51 -0.21 -0.11 -0.01 0.00 -0.05 -0.03 -0.09 -0.07 -0.05 -0.18 -0.20 -0.03 -0.03
Derivada_Account 0.01 0.00 0.00 -0.00 -0.00 -0.00 1.00 0.01 -0.00 0.01 0.00 0.00 0.00 0.02 -0.00 0.01 0.00 0.01 0.01 0.03 0.00
Payroll_Account 0.06 0.00 0.02 -0.00 0.01 -0.51 0.01 1.00 -0.04 -0.02 -0.00 0.01 0.11 0.08 0.17 0.11 -0.00 0.33 0.41 0.12 0.03
Junior_Account -0.20 0.00 0.01 -0.00 -0.00 -0.21 -0.00 -0.04 1.00 -0.06 -0.01 -0.01 -0.03 -0.02 -0.01 -0.01 -0.01 -0.03 -0.03 -0.02 -0.01
Particular_Account 0.33 0.02 0.04 0.01 0.00 -0.11 0.01 -0.02 -0.06 1.00 -0.00 0.03 0.05 0.07 0.00 0.03 -0.02 0.02 0.03 0.07 0.05
Short_Term_Deposits 0.01 -0.00 0.00 -0.00 -0.00 -0.01 0.00 -0.00 -0.01 -0.00 1.00 0.06 0.02 0.01 -0.00 -0.00 -0.00 -0.00 -0.00 0.00 -0.00
Medium_Term_Deposits 0.03 0.00 0.00 -0.00 0.01 0.00 0.00 0.01 -0.01 0.03 0.06 1.00 0.08 0.03 0.00 0.01 -0.00 0.01 0.01 0.01 0.01
Long_Term_Deposits 0.17 0.00 0.02 0.00 0.00 -0.05 0.00 0.11 -0.03 0.05 0.02 0.08 1.00 0.15 0.01 0.06 -0.02 0.06 0.10 0.08 0.00
Funds 0.11 0.01 0.02 0.00 0.00 -0.03 0.02 0.08 -0.02 0.07 0.01 0.03 0.15 1.00 0.01 0.13 -0.00 0.06 0.09 0.17 0.02
Mortgage 0.03 -0.01 0.00 0.00 -0.00 -0.09 -0.00 0.17 -0.01 0.00 -0.00 0.00 0.01 0.01 1.00 0.04 0.01 0.09 0.12 0.04 0.00
Pensions 0.06 -0.00 0.02 0.01 0.00 -0.07 0.01 0.11 -0.01 0.03 -0.00 0.01 0.06 0.13 0.04 1.00 0.00 0.08 0.10 0.11 0.02
Loans 0.02 0.00 -0.01 -0.00 -0.00 -0.05 0.00 -0.00 -0.01 -0.02 -0.00 -0.00 -0.02 -0.00 0.01 0.00 1.00 0.03 0.01 0.00 -0.00
Taxes 0.07 0.00 0.04 0.01 0.01 -0.18 0.01 0.33 -0.03 0.02 -0.00 0.01 0.06 0.06 0.09 0.08 0.03 1.00 0.26 0.10 0.05
Credit_Card 0.08 0.00 0.03 0.00 0.01 -0.20 0.01 0.41 -0.03 0.03 -0.00 0.01 0.10 0.09 0.12 0.10 0.01 0.26 1.00 0.13 0.03
Securities 0.10 0.00 0.03 0.01 0.01 -0.03 0.03 0.12 -0.02 0.07 0.00 0.01 0.08 0.17 0.04 0.11 0.00 0.10 0.13 1.00 0.03
Home_Account 0.01 0.00 0.01 0.01 -0.00 -0.03 0.00 0.03 -0.01 0.05 -0.00 0.01 0.00 0.02 0.00 0.02 -0.00 0.05 0.03 0.03 1.00
Age
Customer_Seniority
Gross_Household_Income
Saving_Account
Guarantees
Current_Accounts
Derivada_Account
Payroll_Account
Junior_Account
Particular_Account
Short_Term_Deposits
Medium_Term_Deposits
Long_Term_Deposits
Funds
Mortgage
Pensions
Loans
Taxes
Credit_Card
Securities
Home_Account
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Number of Customers by City
City (Province Name)
MADRID
BARCELONA
VALENCIA
SEVILLA
CORUNA, A
MALAGA
MURCIA
ZARAGOZA
ALICANTE
CADIZ
ASTURIAS
PONTEVEDRA
VALLADOLID
PALMAS, LAS
BIZKAIA
GRANADA
TOLEDO
BADAJOZ
CANTABRIA
SALAMANCA
CORDOBA
BALEARS, ILLES
CACERES
HUELVA
CIUDAD REAL
ALBACETE
ES
CASTELLON
NAVARRA
BURGOS
TARRAGONA
RIOJA, LA
LEON
GIRONA
GIPUZKOA
LUGO
OURENSE
SANTA CRUZ DE TENERIFE
LERIDA
JAEN
GUADALAJARA
ALMERIA
CUENCA
PALENCIA
ZAMORA
SEGOVIA
ALAVA
HUESCA
AVILA
TERUEL
SORIA
MELILLA
CEUTA
0
50000
100000
150000
200000
250000
300000
350000
Number of Customers

From the above visualization, Madrid, Barcelona and Valencia have the highest number of customers by City.

We can also see Madrid, Barcelona and Sevilla as the cities with the most income.



4. Final Recommendation

A model can be built using K-means and other techniques for customer segmentation. The Elbow method is employed to determine the optimal number of clusters. Additionally, a detailed analysis is conducted to evaluate clustering effectiveness, ensuring that customer groups are well-defined. Further, dimensionality reduction techniques such as PCA are applied to visualize the clusters, making the segmentation process more interpretable. This approach allows for meaningful insights into customer behavior, facilitating data-driven decision-making for personalized marketing strategies.

5. K-means clustering Method

5.1. **StandardScaler** is used to normalize numerical features by transforming them to have zero mean and unit variance, ensuring all variables contribute equally to the clustering process. The code selects Age, Customer_Seniority, and Gross_Household_Income, then applies StandardScaler() to scale these features. This prevents features with larger magnitudes from dominating the analysis. Finally, the transformed data is converted into a DataFrame for further processing, improving model performance and interpretability.

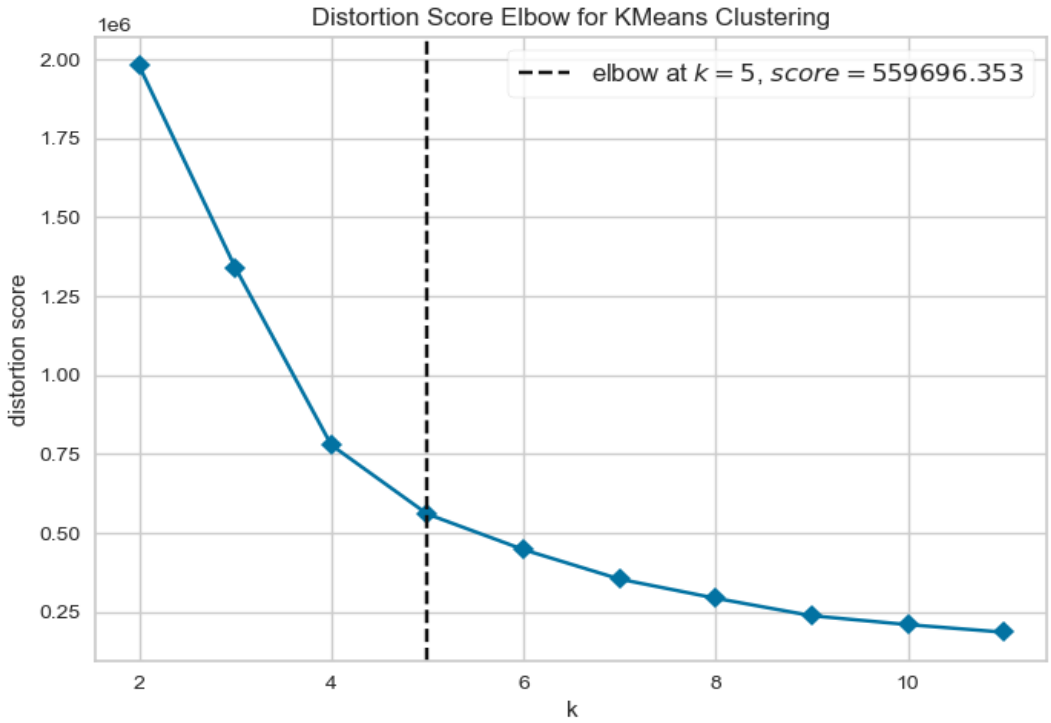
```
In [26]: from sklearn.preprocessing import StandardScaler

features = ['Age', 'Customer_Seniority', 'Gross_Household_Income']
data_selected = data[features]

scaler = StandardScaler()
data_scaled = scaler.fit_transform(data_selected)

data_scaled_df = pd.DataFrame(data_scaled, columns=features)

print(data_scaled_df.head())
```

```
        Age  Customer_Seniority  Gross_Household_Income
0 -0.481959           -0.043284               -0.214780
1 -1.181327           -0.028872               -0.451177
2 -1.181327           -0.028872               -0.054827
3 -1.239608           -0.028872               -0.065824
4 -1.181327           -0.028872               -0.125867
```

5.2. Elbow Method

KElbowVisualizer to determine the optimal number of clusters for KMeans based on the distortion metric, and then applied KMeans with the chosen cluster count. To ensure accuracy, the result was verified by plotting the Elbow Method with inertia values to

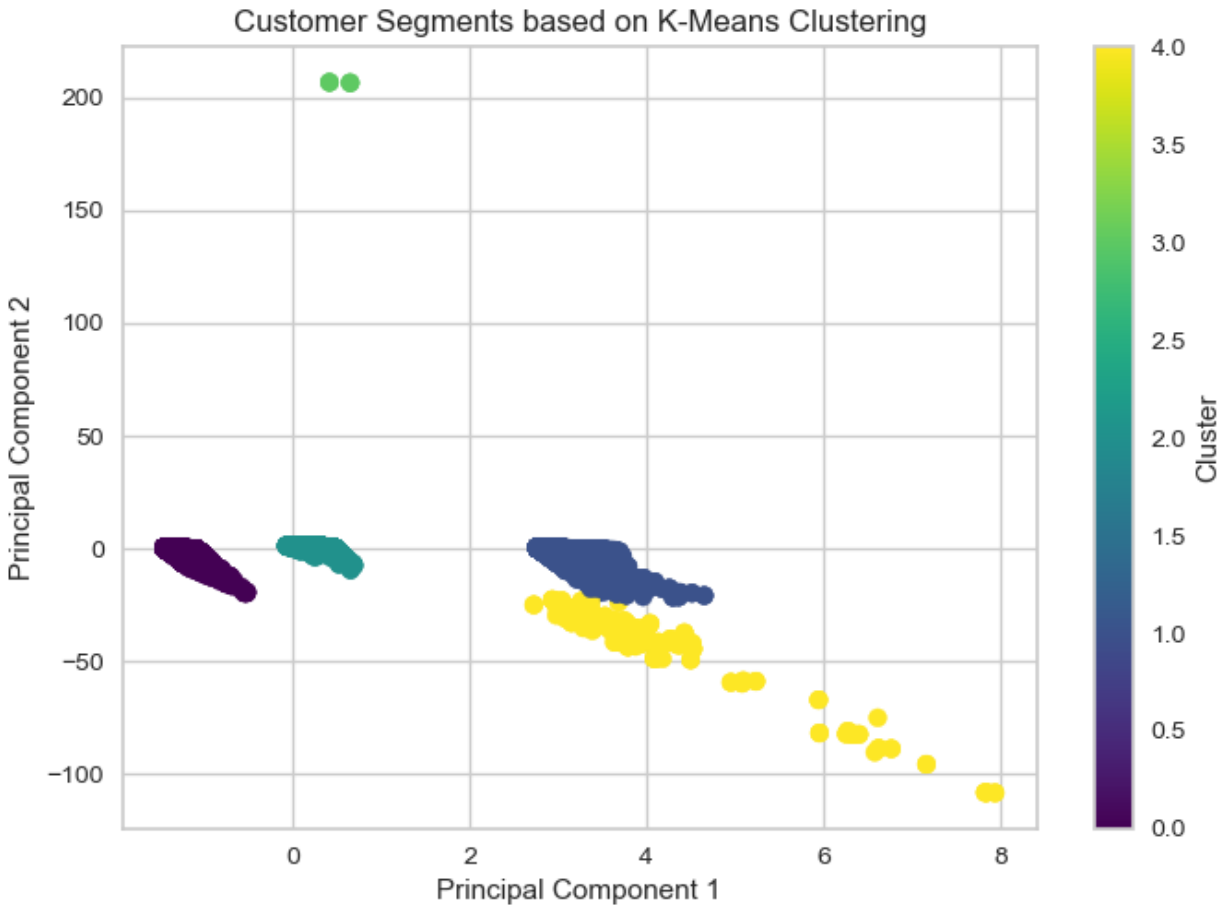
confirm the optimal number of clusters.

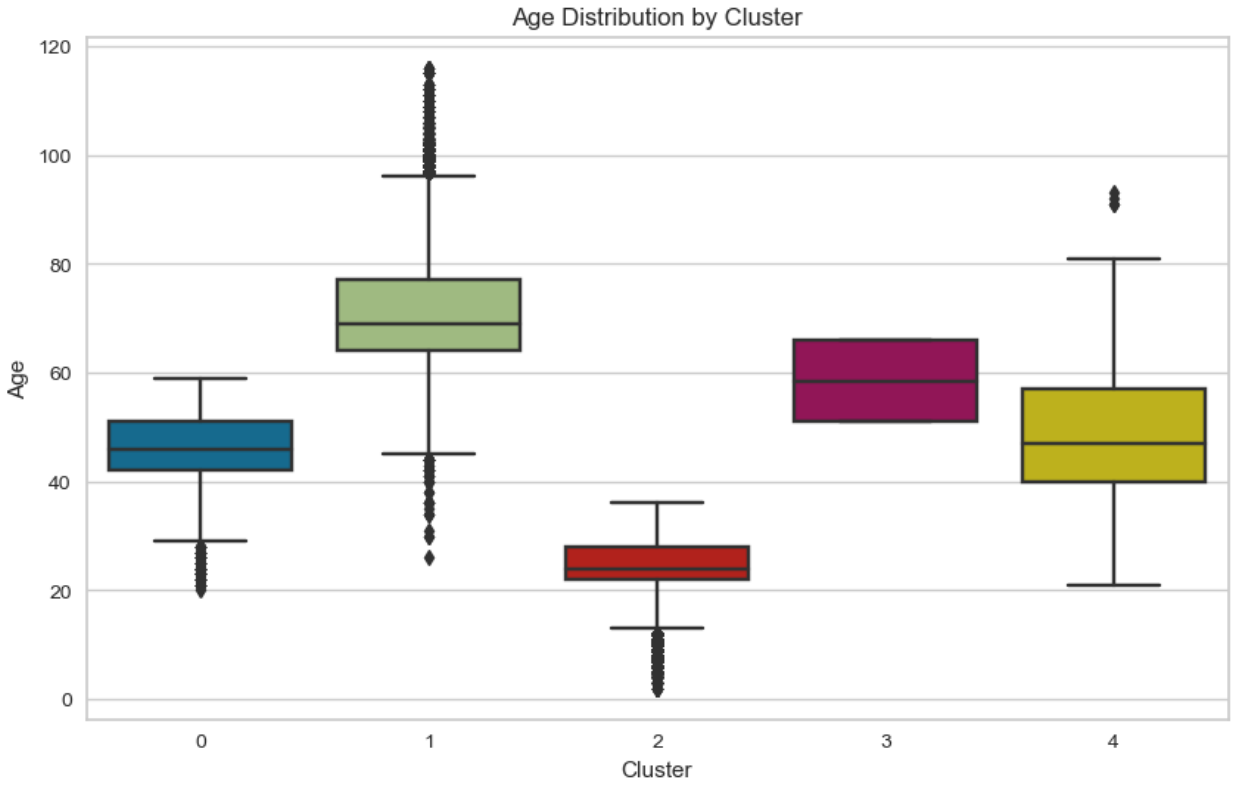


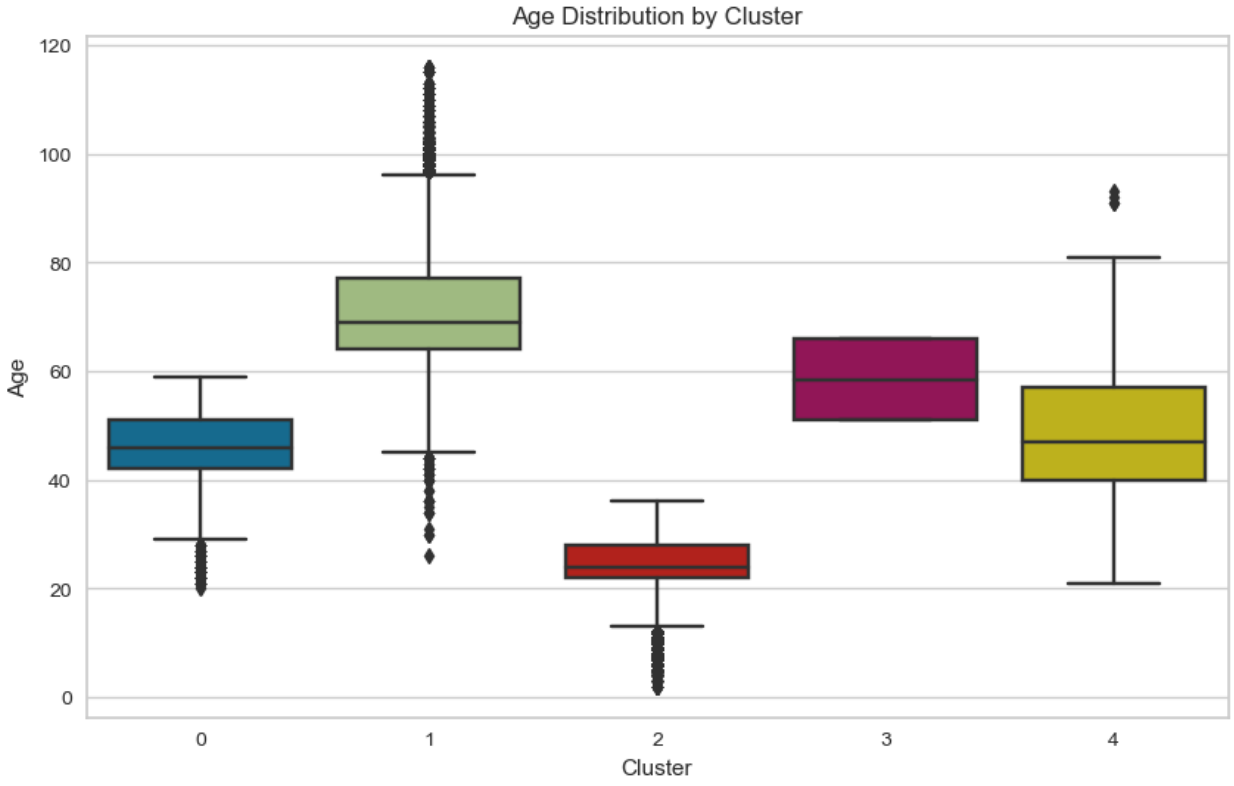
5.3. Visualization of Customer Segments Using PCA

Principal Component Analysis (PCA) was applied to reduce the data to two principal components for easier visualization of customer segments. The scaled data was then transformed, and a 2D scatter plot was created to display the clusters, with each point colored according to its assigned cluster, providing a

clear representation of the customer segments based on KMeans clustering.



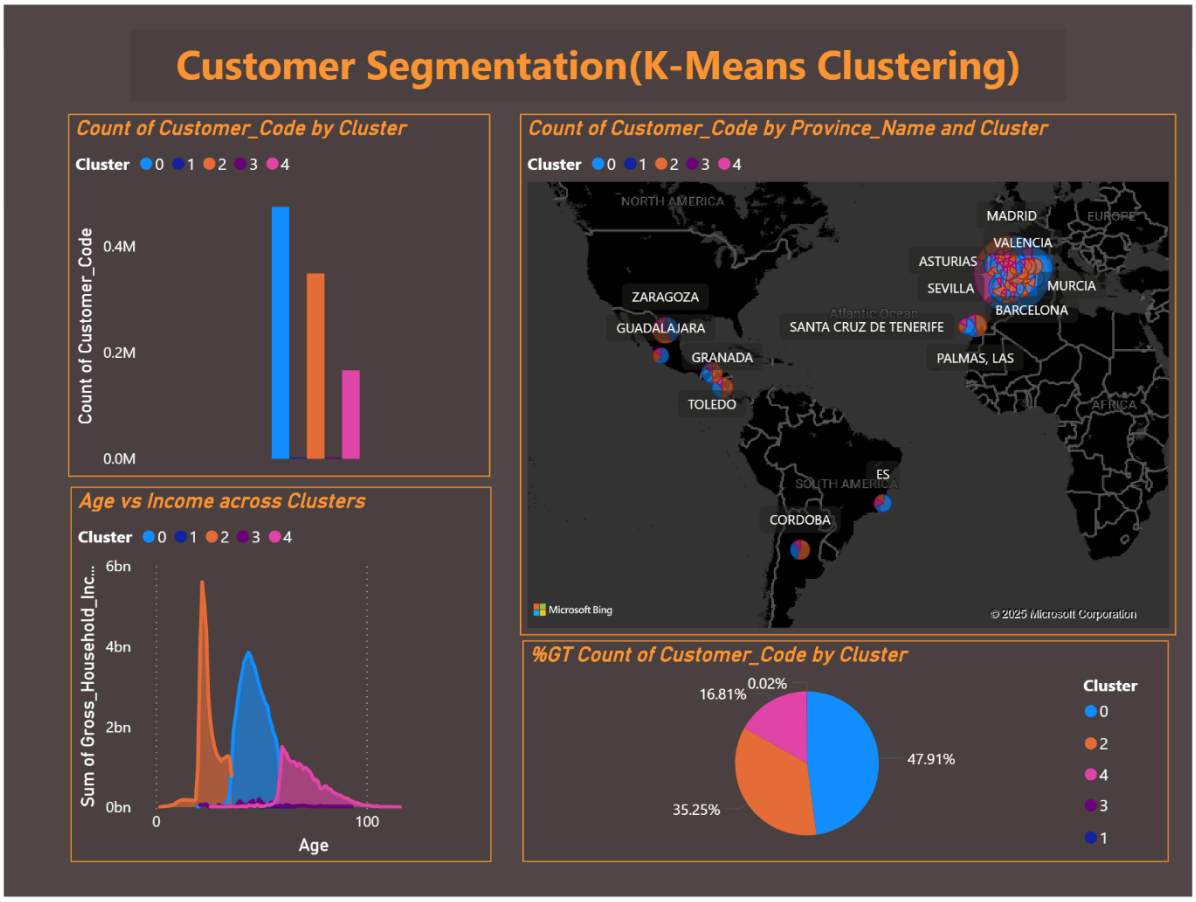




According to KMeans Clustering analysis, the 5 customer groups are as follows:

- *Cluster 0: Customers who age rate 30 to 60*
- *Cluster 1: Customers who age rate 45 to 90*
- *Cluster 2: Customers who age rate 15 to 35*
- *Cluster 3: Customers who age rate 50 to 65*
- *Cluster 4: Customers who age rate 20 to 80*

5.4. Dashboard



Clusters 0 and 1 are 83% of all customers, any improvements achieved in this customer group will dramatically benefit the campaign.

5.5. Summary of K-mean Clustering

The dataset was well-suited for an unsupervised machine learning task. To address this project, the K-Means clustering method was applied alongside PCA, given the dataset's numerous features. The Elbow method was used to determine the optimal number of clusters, which was found to be five. By implementing K-Means clustering, patterns within the data were identified, enabling the creation of distinct groups. This segmentation can facilitate the development of targeted strategies for each group. In the future, specific features from the dataset can be leveraged to form customer groups, allowing for the delivery of personalized offers.

6. Model Performance Evaluation with Supervised Learning

6.1. Training a Random Forest Classifier

A Random Forest Classifier was implemented to predict customer groups using supervised learning. The model was trained on labeled data, and its performance was evaluated using accuracy scores, precision, recall, and F1-score. The Random Forest classifier achieved near-perfect accuracy, with a classification report showing strong results across all categories.

6.2. Training an XGBoost Classifier

XGBoost, another powerful ensemble learning algorithm, was also trained for comparison. After fitting the model to the dataset, evaluation metrics revealed high accuracy but a slightly lower performance compared to Random Forest, particularly in handling class imbalances.

6.3. Hyperparameter Optimization

To enhance model performance, hyperparameter tuning was conducted using RandomizedSearchCV. This was applied to both Random Forest and XGBoost to identify the best parameters for improving accuracy. The optimized Random Forest

model achieved a final accuracy of **99.99%**, while the optimized XGBoost model reached **99.94%**.

6.4. Model Comparison and Selection

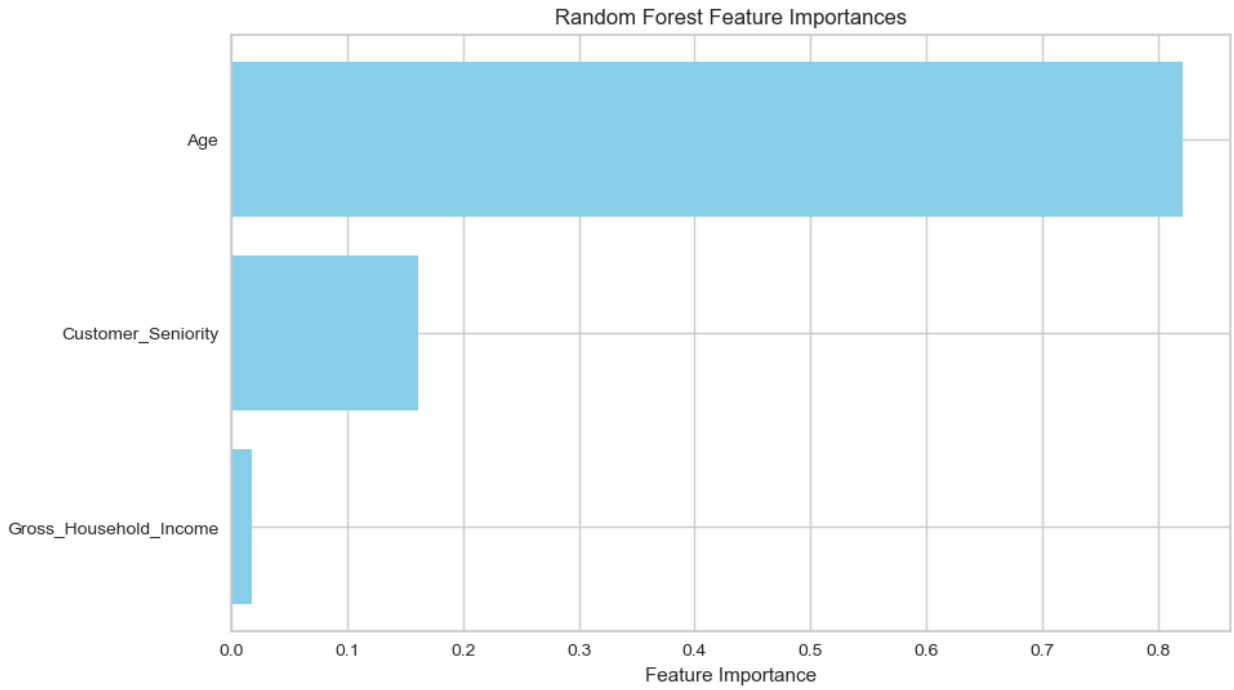
A direct comparison of both models showed that Random Forest performed better than XGBoost in terms of accuracy and classification metrics. The confusion matrix revealed that XGBoost had some misclassifications in specific clusters, while Random Forest demonstrated near-perfect classification across all groups.

6.5. Saving, Loading and Making Predictions

To ensure efficient deployment, the best-performing model was saved using joblib. The saved model was later reloaded and tested on new data. Predictions were generated on a subset of unseen data (first five samples of X_test), confirming that the model remained functional after being saved and reloaded.

6.6. Feature Importance Analysis

To understand which features contributed the most to the classification, feature importance scores were extracted from the trained Random Forest model. A horizontal bar plot was generated to visualize the most influential features, helping to interpret the model's decision-making process.



7. Final Conclusion

This project successfully applied machine learning techniques to analyze and classify data using both unsupervised and supervised learning approaches. K-Means clustering, combined with Principal Component Analysis (PCA), was used for segmentation, allowing patterns within the dataset to be uncovered. The optimal number of clusters was determined using the Elbow Method, providing valuable insights into group structures.

For classification, two powerful machine learning models—Random Forest and XGBoost—were implemented and optimized using hyperparameter tuning. After evaluating both models, Random Forest achieved the highest accuracy and was selected as the final model. The best-performing model was saved and deployed for future predictions, ensuring scalability and usability. Additionally, feature importance analysis provided interpretability, highlighting key factors influencing predictions.

Overall, the project demonstrated the effectiveness of machine learning in data-driven decision-making. The insights derived from clustering and classification can be

leveraged for personalized strategies, improved customer segmentation, and better business outcomes. Future work may involve further fine-tuning, integrating additional features, or exploring deep learning approaches for enhanced performance.