Customer Segmentation Analysis

Jacob Shamah

Professor Singh

MDA620

12/19/23

Table of Contents

Background:

In the world of marketing and business, customer segmentation is a key component to understanding ones target markets and how to sell to them. Market segmentation is comprised of a number of attributes including demographics (age, profession, gender, etc) and psychographics (likes, dislikes, passions, etc..). Getting a true grasp of the consumer will allow marketers to directly target specific segments through customized methods and targetted advertising.

My dataset, ‘Customer Segmentation’,which I found on Kaggle allows us to learn about market segmentation through practical application of data analysis protocols and methods. The raw dataset contains ten columns and 10,695 rows. The columns are as follows:

ID	Unique customer ID (Int64)
Gender	Male or Female (Object)
Ever Married	Yes or No (Object)
Age	Customer’s age (Int64)
Graduated	Yes or No (Object)
Profession	Customers field of work (Object)
Work Experience	How many years in their field (Float64)
Spending Score	Low, Medium, or High (Object)
Family Size	Family member count (Float64)

Var_1	Customer Segment - Six Categories (Object)

Using the *head* method from Pandas, here is the first five rows printed from the dataset:

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1
0	458989	Female	Yes	36	Yes	Engineer	0.0	Low	1.0	Cat_6
1	458994	Male	Yes	37	Yes	Healthcare	8.0	Average	4.0	Cat_6
2	458996	Female	Yes	69	No	NaN	0.0	Low	1.0	Cat_6
3	459000	Male	Yes	59	No	Executive	11.0	High	2.0	Cat_6
4	459001	Female	No	19	No	Marketing	NaN	Low	4.0	Cat_6

The first five rows reveal that our dataset contains NA values leading us to the next step of dataset exploration, checking for NA values. To do so I used the isna and sum methods from Pandas and got this as a result:

```
ID                    0
Gender                0
Ever_Married        190
Age                   0
Graduated           102
Profession          162
Work_Experience    1098
Spending_Score        0
Family_Size         448
Var_1               108
dtype: int64
```

Dataset Manipulation

Removing NA values and dropping columns:

The first step in manipulating the dataset was to deal with the NA values found in data exploration. Due to the size of the dataset, I dropped the records that contained NA values leaving the dataset with 8819 rows. Additionally, I dropped the ID column as it had no practical use in my analysis.

‘Age Bin’ Column Creation:

Because the age column had 67 unique values and I want to use it in my visualizations, I created a new column ‘Age Bin’ which grouped the customers’ ages by decades. Using binning and cut methods from Pandas, I was able to assign each customer into an Age Bin, and for organization purposes, I inserted the column adjacent to the ‘Age’ column as seen below:

Age	Age Bin
36	30s
37	30s
59	50s
47	40s
61	60s

```
['30s', '50s', '40s', '60s', 'Teens', '20s', '80s', '70s']
Categories (8, object): ['Teens' < '20s' < '30s' < '40s' < '50s' < '60s' < '70s' < '80s']
```

Converting column dtypes:

Float to Integer:

The original datatype of ‘Work Experience’ and ‘Family Size’ is float64, but seeing as they are whole numbers only, I decided to change the datatype to **integer** using the *astype* method. Doing so saves memory and makes the dataset look neater.

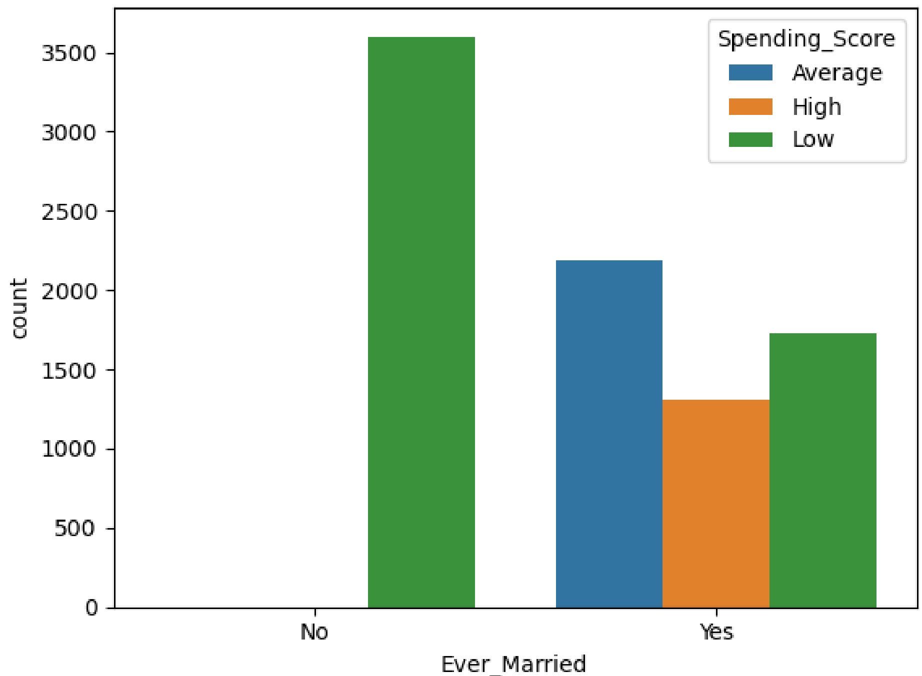
Object to Numerical:

Customer columns 'Gender', 'Ever_Married', 'Graduated', 'Spending_Score' are of object dtype originally, but they all lean towards the categorical datatype, so I changed it to **category** dtype. I did this to make plotting more efficient when working with these specific columns.

Data Visualizations:

Marriage & Spending Score:

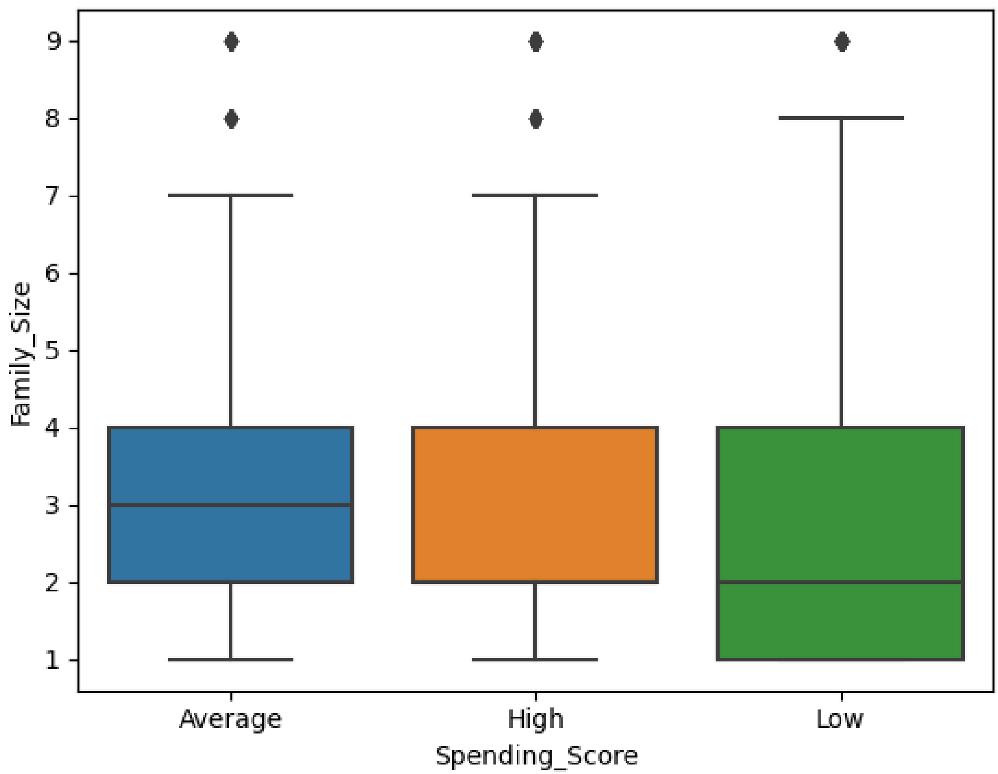
Using seaborn, a visualization library, I plotted a bar graph to explore the effect marriage had on the customers' spending habits. The plot revealed something interesting regarding those that aren't married.



Within the married category, we see a fairly even distribution in the spending score categories, but those that aren't married all fall into the 'Low' spending category. This can be due to the lower expenses one incurs when they live alone.

Family Size and Spending Score:

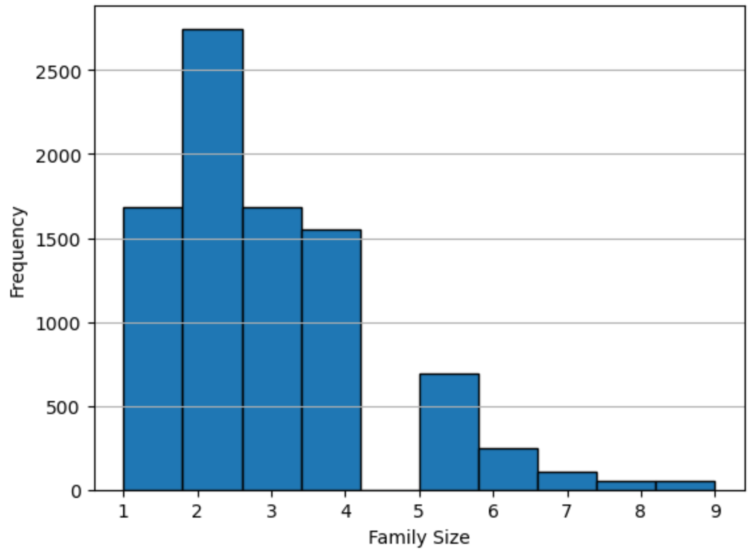
I was curious as to whether or not larger families spend more on average, so to test this theory, I ran a box plot with the X axis being the spending score categories, and the Y axis being number of family members.



The boxplot indicates that family size of two or more is likely to be in the average or high spending category. Additionally, it also shows that there are a couple outliers in the average and high spending categories and one in the low spending category. I am leaving in the outliers because the dataset is so large that it shouldn't have any drastic effects on the prediction model's accuracy.

Distribution of Family Size:

To visualize the distribution of family size, I ran a simple histogram using matplotlib's .hist method. (See visualization on next page)



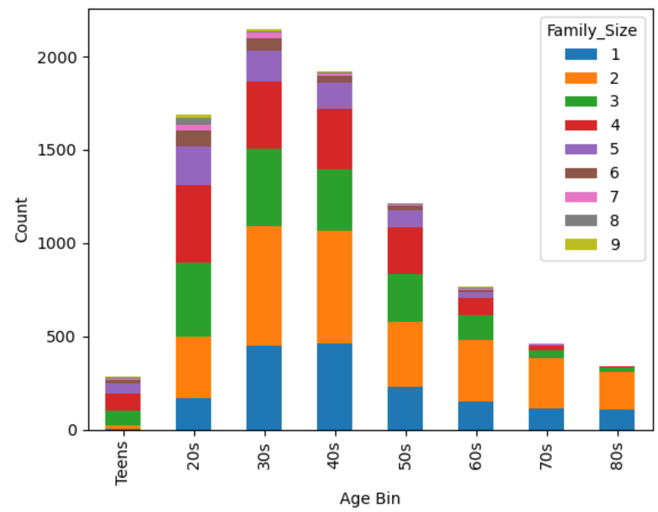
The histogram is right-skewed showing that most values lie onthe lower end of the plot. Most consumers had a family size between 1 and 4, while few had five or more. This insight can help marketers tailor their advertising to family sizes in the common range.

<u>Age and Family Size:</u>

Before I was able plot the relationship between age and family size, I had to first subset my data. Using groupby, value counts, and unstack, I derived the resulting dataframe:

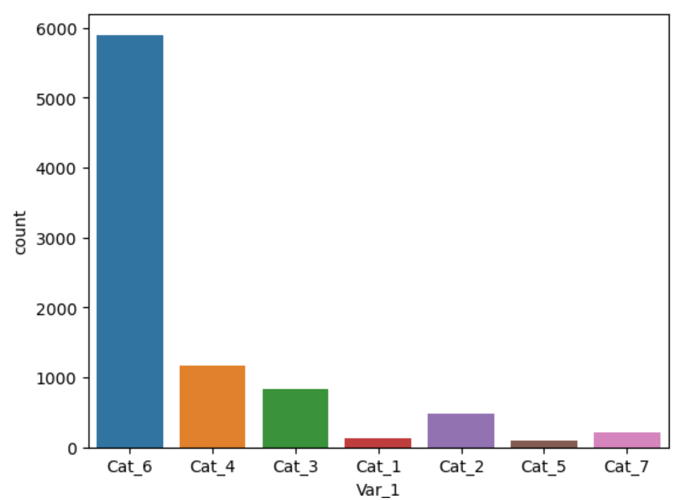
Family_Size	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Age Bin									
Teens	3	21	78	93	51	21	7	5	3
20s	167	330	401	411	207	85	35	32	20
30s	448	642	414	360	167	68	28	8	14
40s	461	604	332	321	139	37	16	3	7
50s	231	349	256	251	86	25	8	4	3
60s	148	332	136	91	31	12	7	2	6
70s	112	270	45	21	8	2	3	0	1
80s	110	199	24	5	1	0	1	0	0

I then plotted a stacked bar chart to see the distribution of family size within each age group:



This plot reinforces the inferences made from the family size distribution chart, and shows that most of their consumers lie between the ages of 20-60.

<u>Countplot for Customer Segments:</u>



The countplot for 'Var_1' revealed that Category 6 has the highest amount of observations, which will come into account later when evaluating the accuracy of the prediction models.

Prediction Models:

Data Preperation for Prediction Models:

For the two model types, I chose to run logistic regression and decision tree models, and to my surprise, both faced many of the same problems. When I tried to run the models with my train & test split, I got an error that it couldn't handle text data, so I had to do some additional data manipulation.

Label Encoding, a method of preparing data for machine learning, assigns a number for each variabel in a column. Given that 'Gender', 'Graduated', 'Ever_Married' are all categorical columns with 2 possible variables I encoded 0 and 1 to their respective options (EX: male = 0, female = 1). Label Encoding is a good method for columns with few categories, but unlike the columns listed above 'Profession' had nine possible values, leading me to create dummy variables (using get_dummies method) for each profession:

Profession_Artist	Profession_Doctor	Profession_Engineer	Profession_Entertainment	Profession_Executive	Profession_Healthcare	Profession_Homemaker	Profession_Law
False	False	True	False	False	False	False	F
False	False	False	False	False	True	False	F
False	False	False	False	True	False	False	F
False	True	False	False	False	False	False	F
False	True	False	False	False	False	False	F
...	...	...	...	...	...	...	
True	False	False	False	False	False	False	F
False	False	False	False	True	False	False	F
False	False	False	False	False	True	False	F
False	False	False	False	False	True	False	F
False	False	False	False	True	False	False	F

Lastly using the .replace method, I altered the spending score column as follows:

0 = 'Low', 1 = 'Average', 2 = 'High'

Now I was able to run my models properly. It is important to note that my correlation matrix didn't indicate any columns that were highly correlated, so I was able to use all the columns without worrying about colinearity.

Model 1 - Decision Tree:

I decided to use a decision tree model because they are one of the ideal choices when working with classification models.

Split: 80- Train, 20 - Test

Predictor columns: ['Gender', 'Ever_Married', 'Age', 'Graduated', 'Work_Experience', 'Spending_Score', 'Family_Size', 'Profession_Artist', 'Profession_Doctor', 'Profession_Engineer', 'Profession_Entertainment', 'Profession_Executive', 'Profession_Healthcare', 'Profession_Homemaker', 'Profession_Lawyer', 'Profession_Marketing']

Predicted column: 'Var_1'

To test the accuracy of the model, I used both the accuracy_score method and classification_report methods from ***sklearn***.

```
Accuracy: 0.67
              precision    recall  f1-score   support

       Cat_1       0.00      0.00      0.00        21
       Cat_2       0.00      0.00      0.00       102
       Cat_3       0.00      0.00      0.00       162
       Cat_4       0.50      0.07      0.12       246
       Cat_5       0.00      0.00      0.00        19
       Cat_6       0.67      0.99      0.80      1173
       Cat_7       0.00      0.00      0.00        41

    accuracy                           0.67      1764
   macro avg       0.17      0.15      0.13      1764
weighted avg       0.52      0.67      0.55      1764
```

The models overall accuracy rate is 67%, but it is mostly due to its prediction performance regarding Cat_6. Cat_6 has the highest precision, recall, f1, and support. Furthermore, categories 1, 2, 5 and 7 have no metrics which means that the model predicted them all wrong or that the model didn't even attempt to predict them. All this inaccuracy is likely because of the uneven distribution among categories we saw in the last visualization.

Model 2 - Logistic Regression:

While logistic regression models are more commonly used with binary outcomes, I ran the model with 'multinomial' in the multi_class method to see if it would produce better results.

Split: 80- Train, 20 - Test

Predictor columns: ['Gender', 'Ever_Married', 'Age', 'Graduated', 'Work_Experience', 'Spending_Score', 'Family_Size', 'Profession_Artist', 'Profession_Doctor', 'Profession_Engineer', 'Profession_Entertainment', 'Profession_Executive', 'Profession_Healthcare', 'Profession_Homemaker', 'Profession_Lawyer', 'Profession_Marketing']

Predicted column: 'Var_1'

```
Accuracy: 0.67
              precision    recall  f1-score   support

       Cat_1       0.00      0.00      0.00        21
       Cat_2       0.00      0.00      0.00       102
       Cat_3       0.00      0.00      0.00       162
       Cat_4       0.48      0.11      0.18       246
       Cat_5       0.00      0.00      0.00        19
       Cat_6       0.68      0.99      0.81      1173
       Cat_7       0.00      0.00      0.00        41

    accuracy                           0.67      1764
   macro avg       0.17      0.16      0.14      1764
weighted avg       0.52      0.67      0.56      1764
```

To my surprise, the evaluation metrics for the logistic regression model were practically identical to those of the Decision tree model, reaffirming the theory that the imbalance of 'Var_1' data is likely the root cause of the model performance. Cat_6, again, has extremely high accuracy, giving the model its 67% accuracy rate.

Model 3 - Logistic Regression:

To be sure it is the dataset that is flawed, I changed the predictor columns to see if that would effect model performance.

Split: 80- Train, 20 - Test

Predictor columns: ['Gender', 'Ever_Married', 'Graduated', 'Work_Experience', 'Spending_Score', 'Family_Size']

Predicted column: 'Var_1'

Accuracy and Classification Report:

Accuracy: 0.67

	precision	recall	f1-score	support
Cat_1	0.00	0.00	0.00	21
Cat_2	0.00	0.00	0.00	102
Cat_3	0.00	0.00	0.00	162
Cat_4	0.48	0.10	0.16	246
Cat_5	0.00	0.00	0.00	19
Cat_6	0.68	0.99	0.80	1173
Cat_7	0.00	0.00	0.00	41
accuracy			0.67	1764
macro avg	0.17	0.16	0.14	1764
weighted avg	0.52	0.67	0.56	1764

This model produced the same results as the other two.

Recommendations:

- Collect more balanced data regarding the consumer segments to improve the performance of prediction models. Category 6 is over saturated within the dataset to build an effective model
- Tailor marketing efforts to 'Low' spending customers with families to promote sales and deals, enticing them to come and purchase goods
- Individuals with two or more family members can be shown pricier products as they are more likely to spend more
- Unmarried individuals can potentially be presented with more expensive products as they aren't spending much money on other products, leaving them with more money to spend

Reference Page

https://www.kaggle.com/datasets/abisheksudarshan/customer-segmentation/?select=train.csv