**Customer Segmentation**

Murtuja Afshar

Debug-Tech Task 2

February 23, 2025

**Data Collection:**

Data was collected from Kaggle from a survey conducted to assess the spending habits of population that visited a superstore. The following columns have been created:

Columns:

id: Unique identifier for each customer.

age: Age of the customer.

gender: Gender of the customer (Male, Female, Other).

income: Annual income of the customer (in USD).

spending\_score: Spending score (1-100), indicating the customer's spending behavior and loyalty.

membership\_years: Number of years the customer has been a member.

purchase\_frequency: Number of purchases made by the customer in the last year.

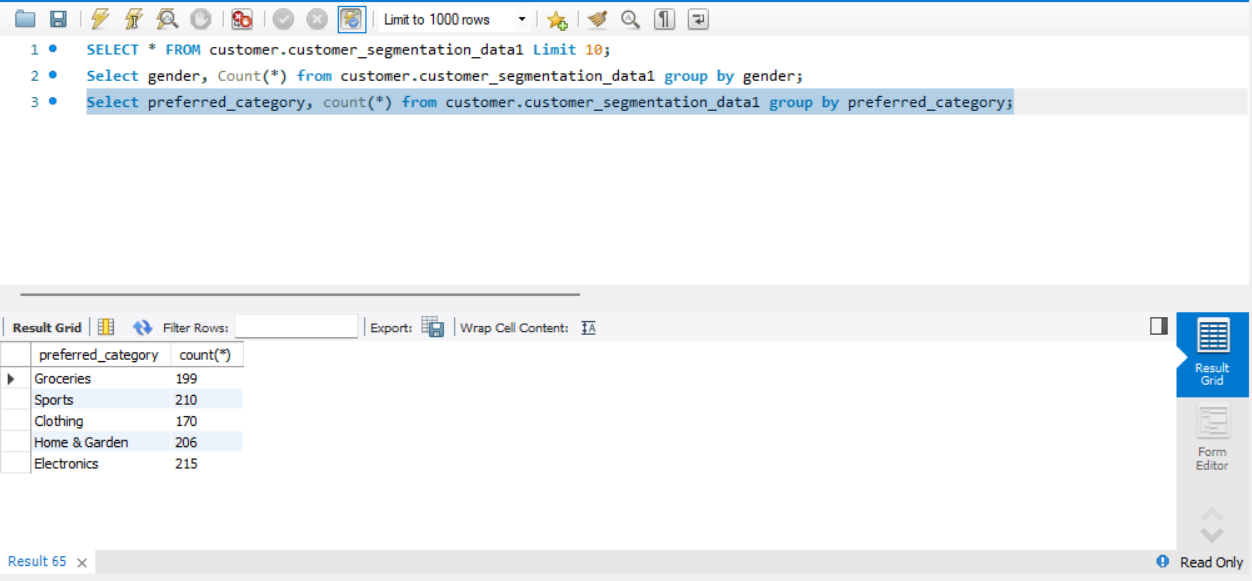
preferred\_category: Preferred shopping category (Electronics, Clothing, Groceries, Home & Garden, Sports).

last\_purchase\_amount: Amount spent by the customer on their last purchase (in USD).

**Data Cleaning:**

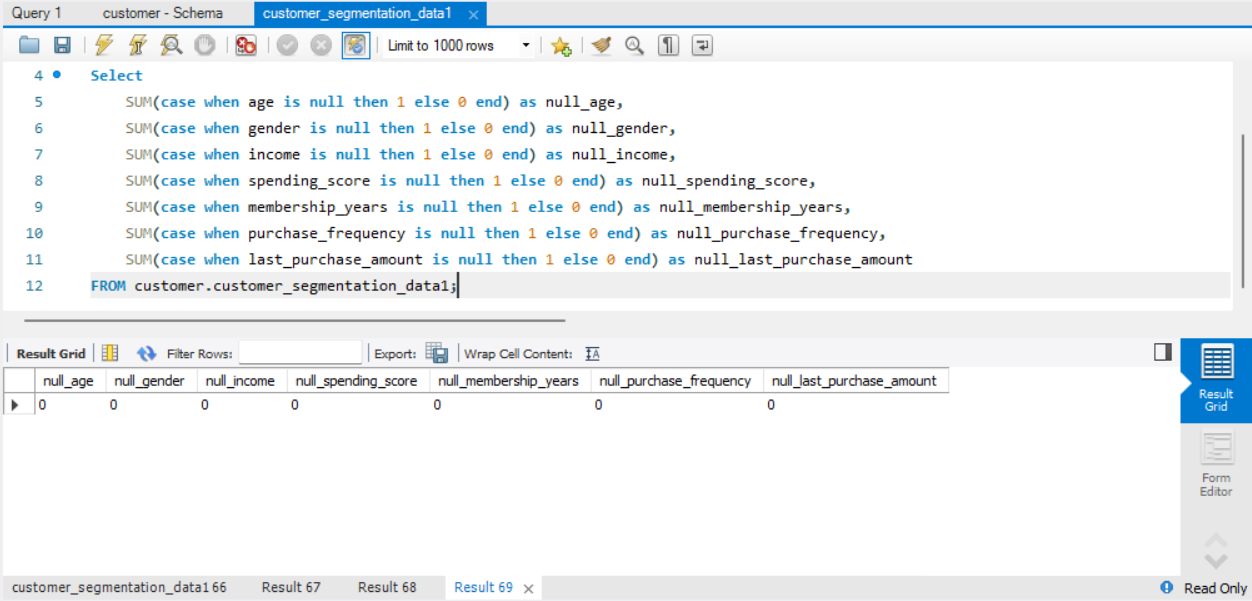
Data was uploaded in MySQL workbench and SQL queries were passed to create a cleaning process:

Identified different categories in gender and preferred category columns. Then treated null values and Duplicates

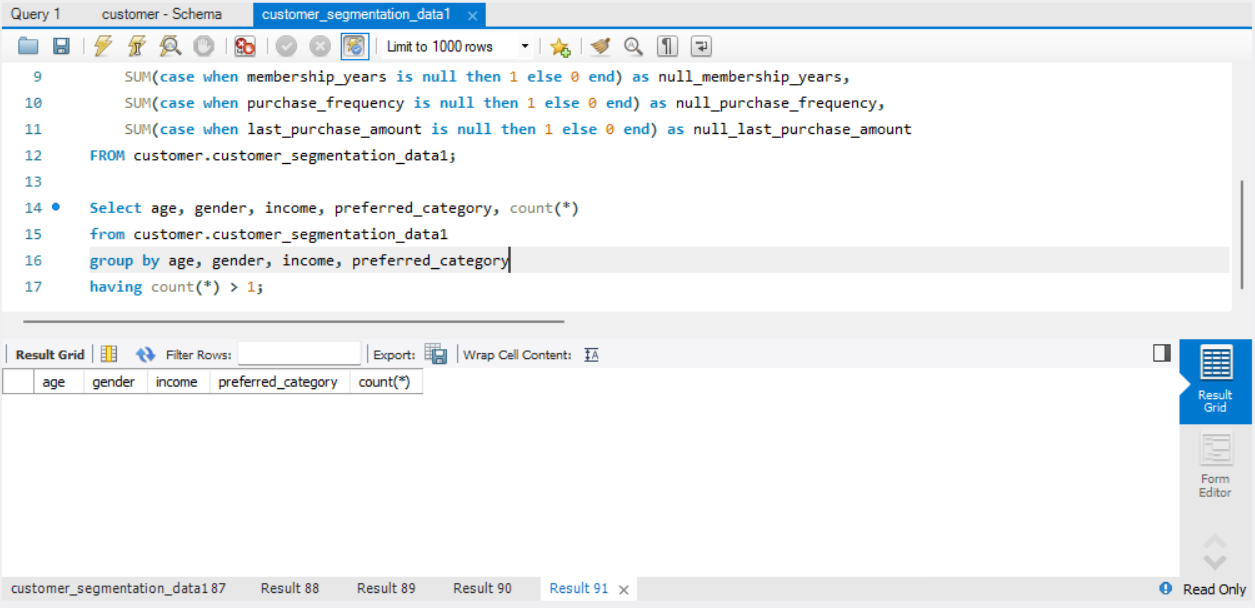


**Null values:**

Null values in a database represent missing, unknown, or inapplicable data. They indicate the absence of a value rather than zero or an empty string. In databases, null values can create challenges in data analysis, as they may distort calculations, affect relationships between tables, and lead to logical inconsistencies. The following SQL queries were used to find Null values of individual columns:

**Duplicate:**

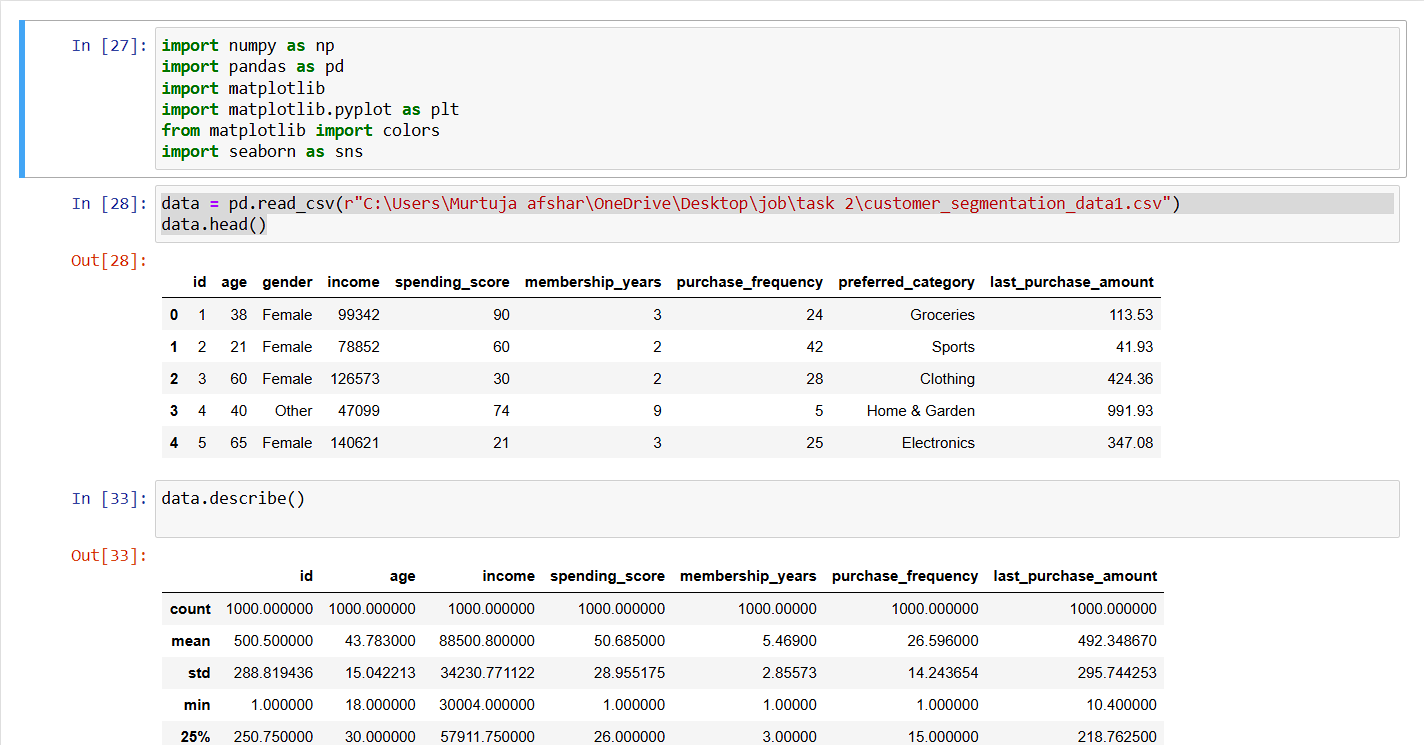
Duplicates in a dataset refer to repeated records that contain identical or highly similar values across one or more fields. They can occur due to data entry errors, system glitches, or multiple data sources merging. Duplicates can distort analysis, inflate statistics, and mislead decision-making by overrepresenting certain values. SQL helps manage duplicates by detecting, filtering, and removing them to ensure data accuracy and consistency. It allows identifying duplicate records based on specific criteria and provides methods to retain only unique entries. By handling duplicates effectively, SQL enhances data quality, ensuring reliable and meaningful analysis. The following SQL queries were used to find Duplicate values of data using age, income and preferred\_category:



Once data was cleaned we imported the data in jupyter notebook.

**Data Loading:**

First import the libraries that are important and load the data using read.csv function. Also describe the data to get an idea of the data:

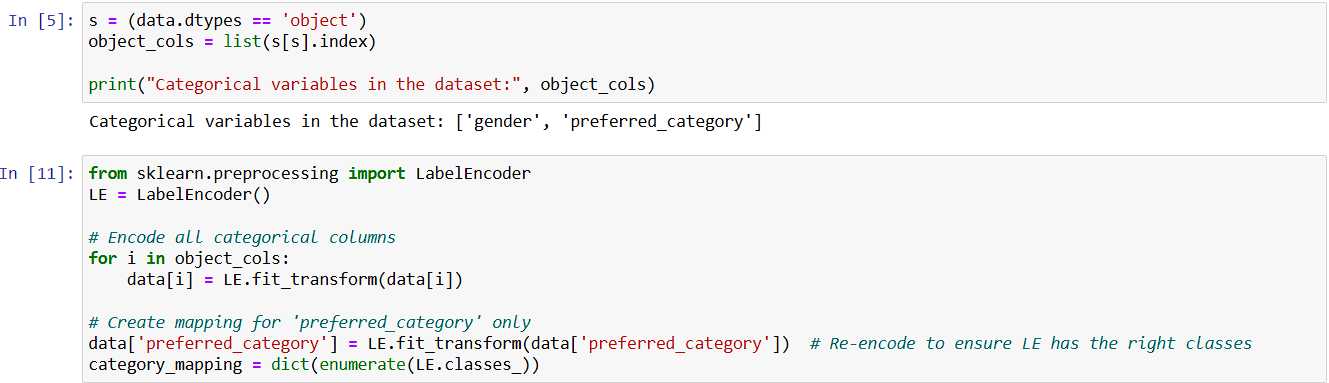


**Preprocessing Data:**

Since we are using unsupervised machine learning algorithms and for that we will be using K-Means classification. It is important to preprocess the data to carry out the analysis.

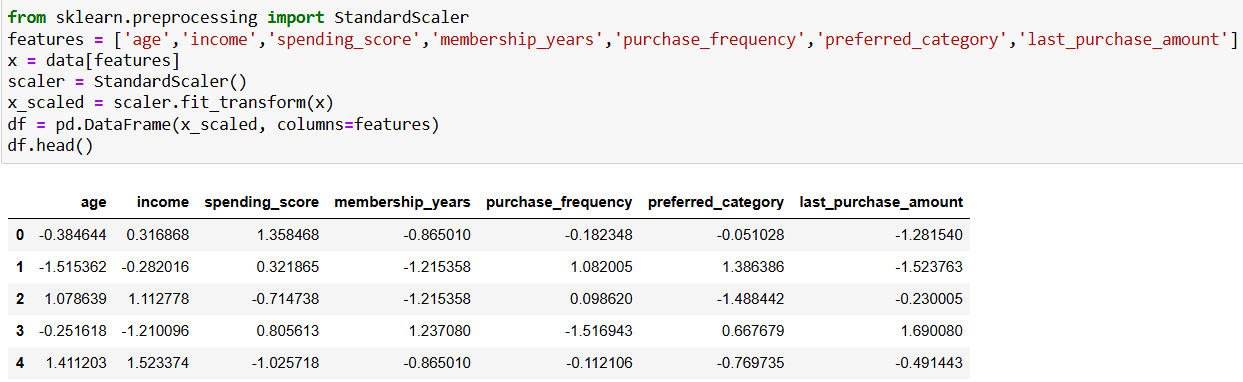
**Label Encoder**

The data contains columns that have string variables which cannot be used for K-means, thus we use label encoder to turn them to float data:



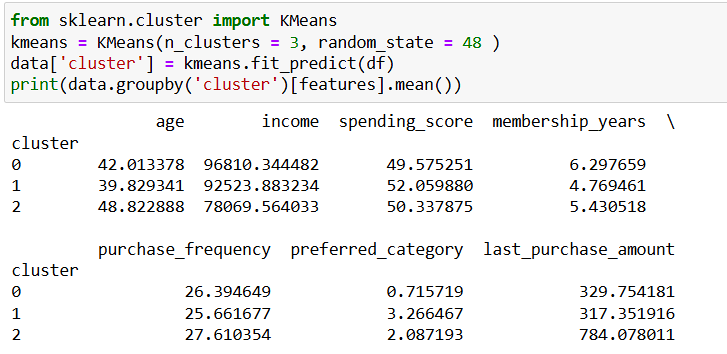
**Standard Scaler**

The data also has values that range over a wide variety. For example the age data varies between 18 to 69 whereas the income ranges between 30004 to 149973. The variation is huge which will cause the classification features with larger numeric ranges to dominate the clustering process. Since K-Means uses Euclidean distance to assign data points to clusters, the algorithm may give more importance to features with larger values, leading to biased cluster assignments.

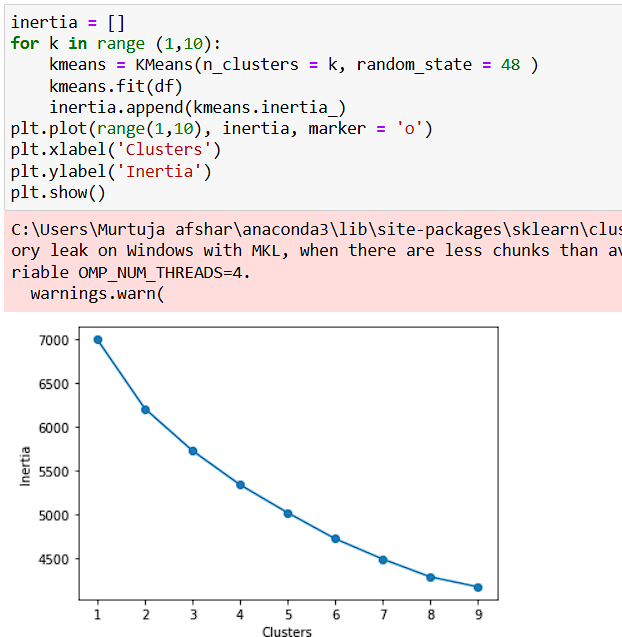


**Data Clustering:**

Initially a cluster of 3 is formed and a random state of 48 is used. Cluster of 3 is for initial process if there is improper clustering or need more detailing we can increase the number of clusters. Random state ensures that this initial selection is reproducible when you run the algorithm multiple times.. The initial centroids can affect the final clusters, as K-Means converges to local minima.

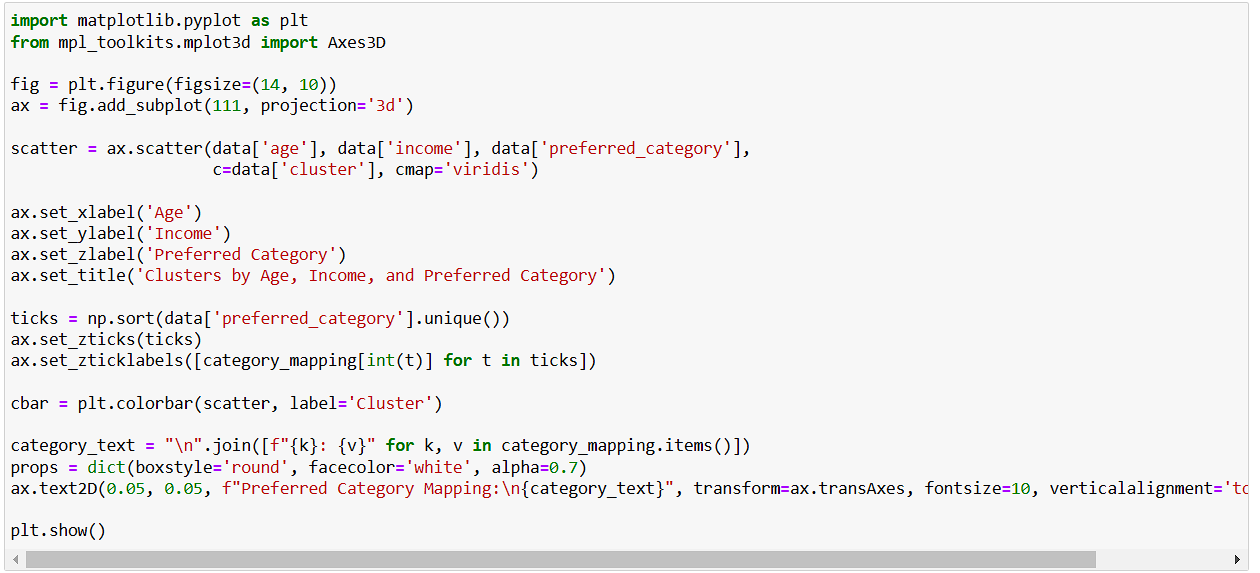


The elbow plot is a graphical method used to determine the optimal number of clusters for K-means clustering. It plots the inertia (within-cluster sum of squares) against the number of clusters. Inertia measures how compact the clusters are; lower inertia indicates more compact clusters. As the number of clusters increases, inertia decreases, but after a certain point, the reduction in inertia becomes very less. We can increase the number of clusters to 7, 8 or 9 but I wanted to create lesser confusion and broader category I have gone with 3 clusters.



**Evaluating Models:**

After classification I have done a 3-D scatter plot and 2 2-D scatter plots using age, income and Preferred category as main references:



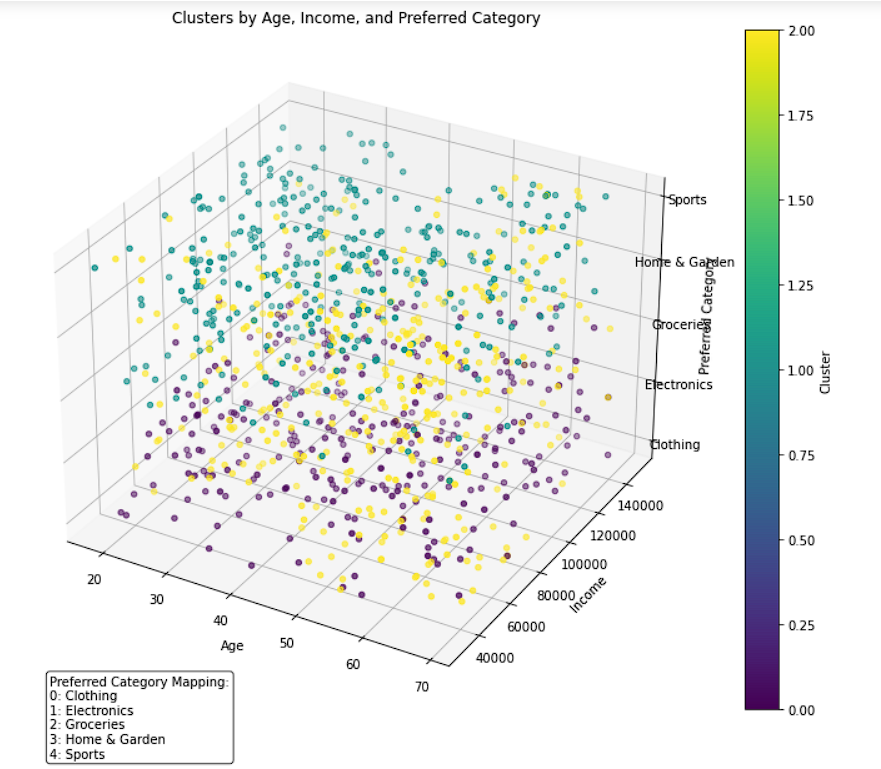
Here in ax variable subplot(111) refers to 1 row, 1 column, and the first subplot. In the code cmap='viridis', cmap refers to the colormap used for mapping the values (in this case, the cluster labels in data['cluster']) to colors in the scatter plot. 'viridis' is a specific colormap that is perceptually uniform. Also since the label encoder has changed the name of the z axis labels to numbers we will use the below process to create the 3d plot with preferred category labels.

1. ticks = np.sort(data['preferred\_category'].unique()): It gets the unique values from the preferred\_category column, sorts them, and stores them in ticks.
2. ax.set\_zticks(ticks): It sets the tick positions on the Z-axis to the sorted unique values (ticks).
3. ax.set\_zticklabels([category\_mapping[int(t)] for t in ticks]): It maps the numeric preferred\_category values to their corresponding labels from category\_mapping and sets them as the tick labels on the Z-axis.

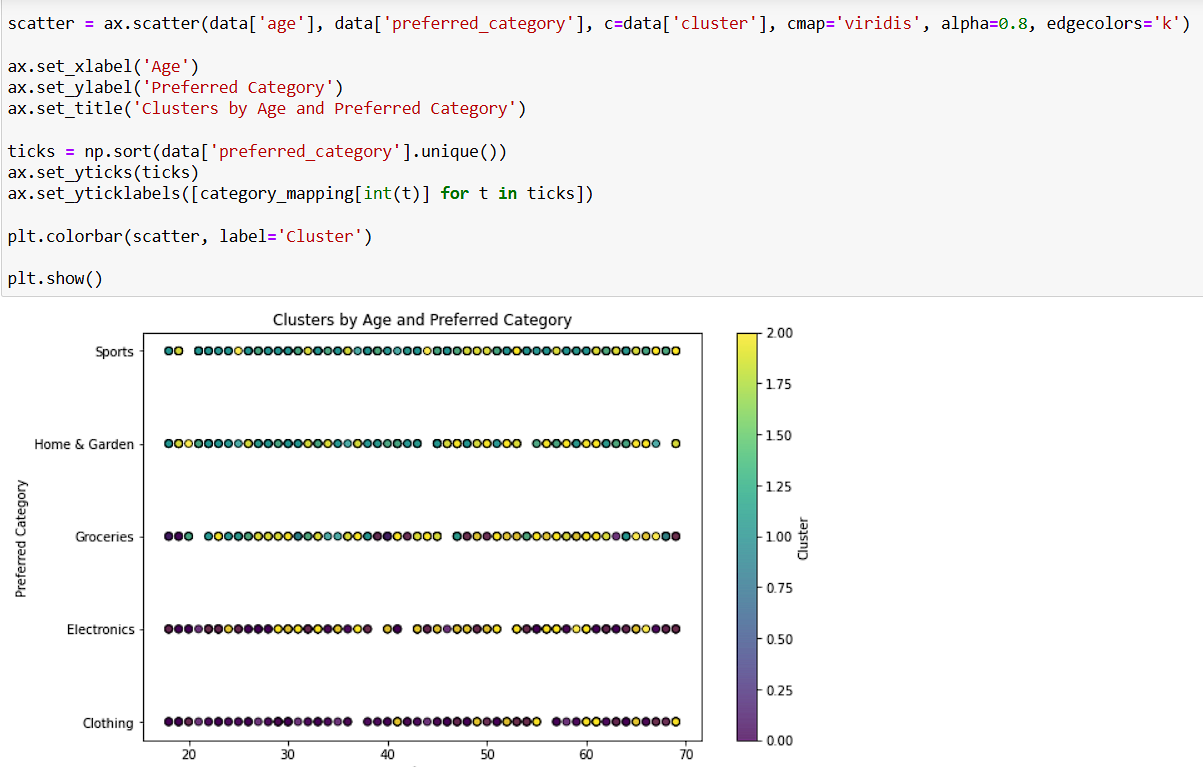
This code displays the category\_mapping dictionary as a text box on the plot:

1. category\_text = "\n".join([f"{k}: {v}" for k, v in category\_mapping.items()]): It creates a formatted string by joining key-value pairs from the category\_mapping dictionary, with each pair on a new line.
2. props = dict(boxstyle='round', facecolor='white', alpha=0.7): It defines the properties of the text box, setting a rounded style, white background color, and 70% transparency (alpha).
3. ax.text2D(...): It adds the formatted category\_text as a text box to the plot at the coordinates (0.05, 0.05) within the axes. The transform=ax.transAxes ensures the position is relative to the axes, and the text box is styled using the props dictionary.

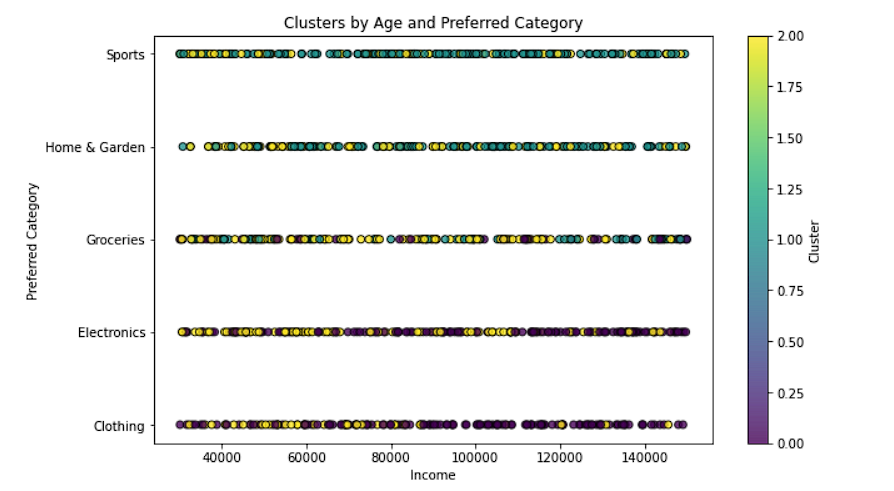
The final 3-D plot:



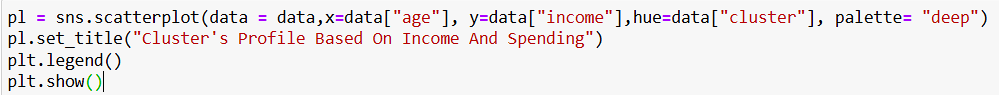
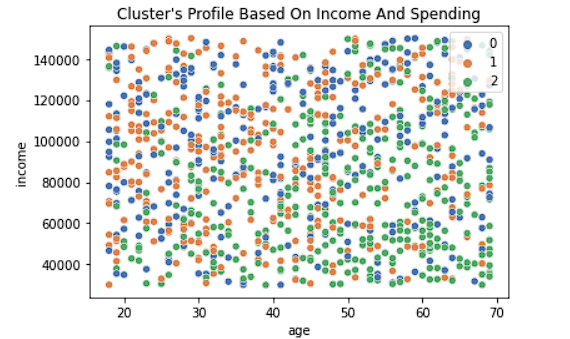
I also created two 2d plots between x and z axis and also y and z axis:

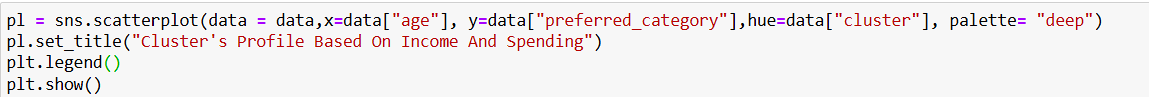






I have also created two scatter plots between age and income and also age and preferred category:

**Final Analysis:**

 **Cluster 0 (Blue)**:

* **Profile**: Likely lower-income customers ($40,000–$80,000) spread across all ages, with a concentration on younger (20–40) and older (60–70) customers preferring Clothing (purple dots in the plots).
* **Fashion Preference**: These customers may prioritize affordable, practical, or essential casual wear due to budget constraints. They’re less likely to invest in high-end, fashion-forward items.
* **Product Recommendations**:
  + Develop budget-friendly casual wear lines, such as basic t-shirts, jeans, and everyday outfits in neutral or classic colors.
  + Offer seasonal sales or loyalty discounts to appeal to price-sensitive shoppers.
  + Focus on durable, versatile clothing that can be mixed and matched for everyday use.

 **Cluster 1 (Orange)**:

* **Profile**: Mid to higher-income customers ($80,000–$140,000) across all ages, with a notable presence among middle-aged (30–50) customers who show interest in Electronics and Sports but also Clothing.
* **Fashion Preference**: This segment is more likely to be fashion-forward or willing to invest in trendy, stylish clothing. They may seek premium or branded fashion items due to their higher disposable income.
* **Product Recommendations**:
  + Create fashion-forward collections with bold patterns, trendy designs, and premium materials (e.g., designer jeans, statement jackets, or seasonal fashion pieces).
  + Introduce limited-edition or exclusive lines to attract these customers, leveraging their interest in unique, high-value products.
  + Offer complementary accessories (e.g., belts, scarves) or upscale casual wear like athleisure for those also interested in Sports.

 **Cluster 2 (Green)**:

* **Profile**: A broad, mixed segment with incomes ranging from $40,000–$120,000 and all ages, showing a balanced interest in Groceries, Home & Garden, and Clothing.
* **Fashion Preference**: These customers likely prefer versatile, functional casual wear that fits into their daily lives (e.g., family-oriented or practical outfits). They may not prioritize high-fashion but value comfort and affordability.
* **Product Recommendations**:
  + Design casual wear lines that balance style and comfort, such as comfortable leggings, hoodies, or work-appropriate outfits.
  + Focus on family-sized packs or multi-purpose clothing (e.g., items suitable for both work and leisure).
  + Use eco-friendly or sustainable materials to appeal to environmentally conscious customers within this segment.