**MKT 568 -Assignment 1**

**Names: Arafat Ahmed**

* 1. Variable name: Repurchase\_Intentions

Ans. **Scale**: **Ordinal**

This scale is ordinal because the values represent a ranked order (from "Not at all" to "definitely repurchase"), but the intervals between the numbers may not be equal.

* 1. Variable name: Gender:

Ans. **Scale**: **Nominal**

This is a nominal scale because gender categories are distinct with no inherent order.

* 1. Variable name: income

Ans. **Scale**: **Ratio**

This is a ratio scale because it has a true zero point (no income), and the differences between income levels are meaningful. I can calculate ratios (e.g., "twice as much income").

* 1. Variable name: Satisfaction Product

Ans. **Scale**: **Ordinal**

This is an ordinal scale because it represents levels of satisfaction, which are ranked. However, the intervals between the rankings are not necessarily equal (e.g., the difference between “moderately satisfied” and “completely satisfied” might not be the same as between “moderately dissatisfied” and “neither satisfied nor dissatisfied”).

* 1. Ans. **Scale**: **Nominal**

This is a nominal scale because the locations are distinct categories with no inherent ranking or order.

**B.**

Ans.

**1. Describe the business problem:**

The business problem centers around **reducing delivery times** by predicting and shipping products to customers before they even place an order. The challenge Amazon is facing is that delays between ordering and receiving purchases may discourage customers from buying items online.

Amazon's goal is to taking advantage of its vast amounts of customer data to anticipate what items customers are likely to buy in the near future. By shipping products closer to customers in advance, Amazon aims to cut delivery times, potentially converting more sales and enhancing customer satisfaction. However, the risk is that they may send products customers don't order, resulting in logistical inefficiencies or customer dissatisfaction if items need to be returned.

**2. What type of data and variables would be needed:**

To run this analysis successfully, Amazon would need to gather and analyze several types of data and variables, both as input (independent variables) and output (dependent variables):

**Data (Collected from whom and where):**

* **Customer Data**:
  + Past purchase history (e.g., products bought, categories of items, brands).
  + Browsing behavior (e.g., products viewed, items added to cart but not purchased).
  + Customer demographics (e.g., age, gender, location, income level).
  + Wishlist items or products saved for later purchase.
  + Returns history (e.g., frequency of returns, reasons for returns).
* **Interaction Data**:
  + Engagement with product recommendations (e.g., clicks on recommended products).
  + Time spent on product pages.
  + Search queries (e.g., frequency of searches for certain types of products).
  + Email or marketing campaign responses (e.g., whether the customer clicked through promotional emails or ads).
* **Shipping and Delivery Data**:
  + Previous delivery times and locations.
  + Current stock levels in fulfillment centers close to the customer.
  + Popular products in the customer’s geographic area.
* **Behavioral Data**:
  + Time of the day or week when customers tend to make purchases.
  + Seasonal buying trends (e.g., holidays, Black Friday).
  + Cursor hover time on product pages (e.g., products frequently considered but not purchased).

**Variables for Analysis:**

**Input (Independent) Variables**:

1. **Customer's historical purchase frequency**: How often a customer buys from Amazon (weekly, monthly, etc.).
2. **Previous purchase categories**: Types of products frequently bought by the customer.
3. **Browsing and wish list data**: Products viewed or saved but not yet purchased.
4. **Product popularity**: Items that are popular with similar customers or in the same geographical area.
5. **Customer demographics**: Age, income, location, and other characteristics that may influence buying decisions.
6. **Time of year**: Whether the shopping behavior aligns with holiday seasons or other events (e.g., back-to-school season).
7. **Customer behavior with returns**: Whether the customer frequently returns products.

**Output (Dependent) Variables**:

1. **Likelihood of purchase**: The probability that a customer will buy a specific product.
2. **Estimated delivery location**: Where the package should be pre-shipped (likely based on the customer's address or nearby fulfillment centers).
3. **Optimal product shipment timing**: Predicting when the product should be shipped to arrive just as the customer places the order.

**Conclusion:**

The key to anticipatory shipping is to correctly predict which products a customer is most likely to order and when they will need them. Using advanced data analytics and machine learning models, the data scientist can evaluate the patterns in customer behavior and predict future purchases.

**C.**

|  |  |
| --- | --- |
| Variable | Categorical or Continuous? |
| Neighbourhood | Categorical |
| Room\_type | Categorical |
| Price | Continuous |
| minimum\_nights | Continuous |
| number\_of\_reviews | Continuous |
| reviews\_per\_month | Continuous |
| availability\_365 | Continuous |

2. **Response to: What steps would I take to clean the BostonListings dataset?**

Cleaning the BostonListings dataset is a crucial part of preparing it for analysis. The cleaning process involved several key steps to ensure the dataset was accurate, complete, and free of inconsistencies or irrelevant information. Below is a detailed description of the steps I took to clean the dataset, along with justifications for each decision.

**Step-by-Step Data Cleaning Process for the BostonListings Dataset**

**Step 1: Import Required Libraries:**  Before any data cleaning can begin, it's essential to load the necessary libraries. I imported the panda’s library, which is critical for working with dataframes and managing data cleaning tasks.

**Step 2: Load the Dataset** The dataset was loaded using the pd.read\_excel() function. I ensured that the BostonListings dataset is read into a pandas dataframe, allowing me to perform all the required cleaning steps.

**Justification**: Loading the dataset into a dataframe is the first step for any data analysis, and pandas make it easy to manipulate and clean data for further analysis.

**Step 3: Examine the Data** After importing the data, the first task was to understand the structure of the dataset. I used listings\_data.shape() and listings\_data.head() to check the number of rows and columns, as well as viewing the first few rows of the dataset. I used functions like .shape() and .head() to examine the number of rows and columns, and to look at the first few rows of data to get a sense of the variables and their contents.

**Justification:** This step ensures that I am aware of how many entries and variables are in the dataset and allows I to visually inspect the data for any irregularities or missing values early on.

**Step 4: Examine Missing Values** I examined missing values using the isnull () function and sum () to get the total number of missing values in each column.

In this dataset, missing values were detected in the price and reviews\_per\_month columns, which were addressed in the following steps.

**Justification:** Handling missing data is a crucial step in any data-cleaning process. It helps I decide how to deal with incomplete observations, which can bias the results of Ir analysis if left unchecked.

**Step 5: Handle Missing Values**

* **For the price column**: I imputed missing values with the median of the column. The median was chosen because it is less sensitive to outliers compared to the mean. This ensures that the imputed values do not skew the data, particularly if there are listings with extremely high or low prices.
* **For the reviews\_per\_month column**: I imputed missing values with 0, under the assumption that a listing with no reviews can be considered to have zero reviews per month.

**Justification:** Imputing missing values with the median ensures that the imputed data is not overly influenced by extreme values, while filling reviews\_per\_month with 0 is logical as properties without reviews should not be ignored or misrepresented.

**Step 6: Check Data Types** To ensure that the data types of the columns are appropriate, I used dtypes. This confirmed whether each column has the correct data type for further analysis. After handling the missing values, I checked the data types of all relevant columns to ensure that they were in the correct format for further analysis. For instance, I ensured that numerical columns like price, minimum\_nights, number\_of\_reviews, reviews\_per\_month, and availability\_365 were stored as integers or floats.

**Justification:** Proper data types are essential to ensure that future analysis, calculations, and manipulations (such as mathematical operations) are valid. For instance, ensuring price is a numeric data type allows for mathematical operations like finding the mean, median, etc.

**Step 7: Convert Object Columns to Categorical** For object-type columns like Host\_name, Neighborhood, etc., I converted them to the category data type. This optimizes memory usage and speeds up the analysis.

Justification: Categorical data types are more memory-efficient and make it easier to work with non-numerical data in subsequent analysis.

**Step 8: Check for Duplicates** I checked for any duplicate rows to ensure that no repeated data entries were present, which could skew the analysis.

**Justification:** Duplicate rows, if present, can create biases in the analysis by giving more weight to certain data points. Hence, identifying and removing duplicates is a necessary step.

**Step 9: Examine Correlations Between Numerical Variables (Heatmap)** I used a correlation heatmap to understand the relationships between key variables such as price, minimum\_nights, and number\_of\_reviews. The heatmap helps to visualize whether outliers in these columns might be connected.

The heatmap showed weak correlations between these variables, which indicated that the outliers in price and minimum\_nights did not necessarily influence each other.

**Justification:** This step was important because if a strong correlation had been found, the outliers in one variable might have justified the extreme values in another. Since the correlation was weak, I proceeded with treating the outliers independently in the final step.

**Step 10: Detect Outliers Using Z-Scores** I calculated Z-scores for the numerical columns (e.g., price, minimum\_nights, etc.) to detect outliers, assuming values outside a threshold (e.g., |Z| > 2.53) as potential outliers.

**Justification:** Z-scores standardize the data and help to identify outliers effectively by checking how far a particular data point deviates from the mean. Values beyond a certain threshold (e.g., 2.53) can be classified as outliers.

**Step 11: Visualize Outliers Using Boxplots** I created boxplots for key numerical columns to visualize the distribution of values and detect outliers graphically.

**Justification:** Boxplots provide a visual representation of the spread and skewness of the data. Outliers, if present, are indicated as points outside the whiskers of the plot, making it easier to detect extreme values.

**Conclusion**: Through these steps, I have effectively cleaned the BostonListings dataset, ensuring it is ready for future analysis. I handled missing values, detected and dealt with outliers, and ensured that the data types were correctly formatted. Each step was essential to ensure that the dataset is in a state where further analysis can be conducted without bias or inaccuracies caused by dirty data.