COMPREHENSIVE EXPLORATORY DATA ANALYSIS OF CUSTOMER DEMOGRAPHICS AND SPENDING BEHAVIOR

PREPARED BY SYED HASEEB UL HASSAN

Introduction

This report presents an exploratory data analysis (EDA) of the dataset provided, which contains information related to customer demographics, spending habits, and various customer-specific attributes. The goal of this analysis is to understand the patterns in the data, identify key insights, and provide recommendations for further actions or studies

Data Overview

The dataset consists of multiple columns related to customer information, including:

- ID: Unique identifier for each customer.
- Year_Birth: The birth year of the customer.
- Education: Educational level of the customer.
- Marital_Status: Marital status of the customer.
- Income: The annual income of the customer.
- Kidhome: Number of children at home.
- Various Spending Columns: Including MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, and MntGoldProds.

Data Cleaning and Preprocessing

This report presents an exploratory data analysis (EDA) of the dataset provided, which contains information related to customer demographics, spending habits, and various customer-specific attributes. The goal of this analysis is to understand the patterns in the data, identify key insights, and provide recommendations for further actions or studies

Data Processing

Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the Dataset

```
food_df=pd.read_csv("C:/Users/Hasseb/Downloads/ifood
_df_raw.csv.xls")
food_df.head(5)
```

Checking for Missing Values and Datatypes
food_df.isnull().sum
there is some missing values in income column
food_df.info()
two column has incorrect datatype
income and Dt_Customer

Correcting the datetype
Dt_Customer datatype object to data
food_df['Dt_Customer']=
pd.to_datetime(food_df['Dt_Customer'],format='%m/%d/
%y')
Income Object to float
First Removing space in Column
food_df.columns = food_df.columns.str.strip()
food_df['Income'] = food_df['Income'].replace({r'\$': '', ',': ''}, regex=True).astype(float)

Filling the Null values in Income Column

fill the null values in the Income column based on the Education column

```
education_mean_income = food_df.groupby('Education')['Income'].mean()
food_df['Income'] = food_df.apply(
lambda row: education_mean_income[row['Education']] if pd.isnull(row['Income']) else row['Income'],
    axis=1
)
```

Checking the Outilers

```
numerical_columns = food_df.select_dtypes(include=['int64', 'float64']).columns
plt.figure(figsize=(16, 10))
for i, col in enumerate(numerical_columns):
    plt.subplot(len(numerical_columns)//3 + 1, 3, i + 1)
    sns.boxplot(y=food_df[col])
    plt.title(f'Boxplot of {col}')
    plt.tight_layout()
plt.show()
```

Handling the Outliers in Year_Birth and Income Column

```
columns_outlier = ['Income', 'Year_Birth']
for column in columns_outlier:
    q1 = food_df[column].quantile(0.25)
    q3 = food_df[column].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
food_df[column]=food_df[column].clip(lower=lower_bound, upper=upper_bound)
```

Variable Transformation

Creating Age Column from Year_Birth

```
food_df['Age'] = 2024 - food_df['Year_Birth']
```

Education Column

```
food_df['Education'] = food_df['Education'].replace({
    'Graduation': 'Graduation',
    'PhD': 'PhD',
    'Master': 'Master',
    '2n Cycle': 'Secondary Education',
    'Basic': 'Primary Education'
})
food_df['Education'].value_counts()
```

Feature Engineering

We will create new features that could be useful for analysis, such as "TotalAmount_Spent" by summing all the amount columns

Total Amount Spent

food_df['TotalAmount_Spent'] = food_df[['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']].sum(axis=1)

Total Purhase Column

food_df['Total_Purchases']=
food_df[['NumDealsPurchases','NumWebPurchases',
'NumCatalogPurchases',
'NumStorePurchases']].sum(axis=1)

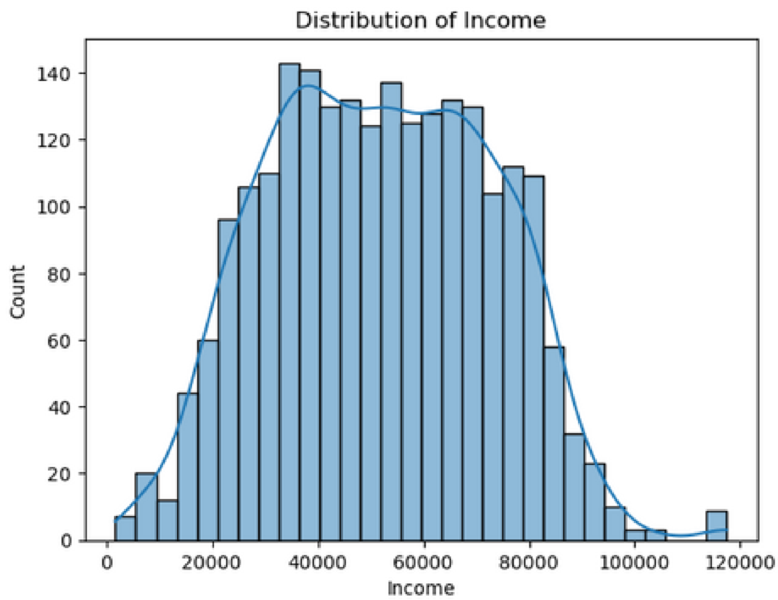
Total Campaigns Accept

food_df['TotalCampaignsAcc']=
food_df[['AcceptedCmp1','AcceptedCmp2','AcceptedCm
p3','AcceptedCmp4', 'AcceptedCmp5']].sum(axis=1)

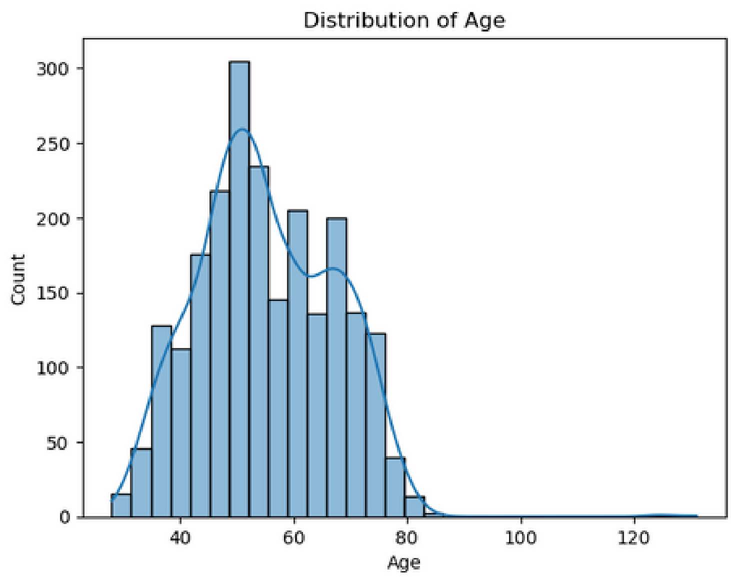
Exploratory Data Analysis (EDA)

Univariate Analysis

Distribution in Income Column

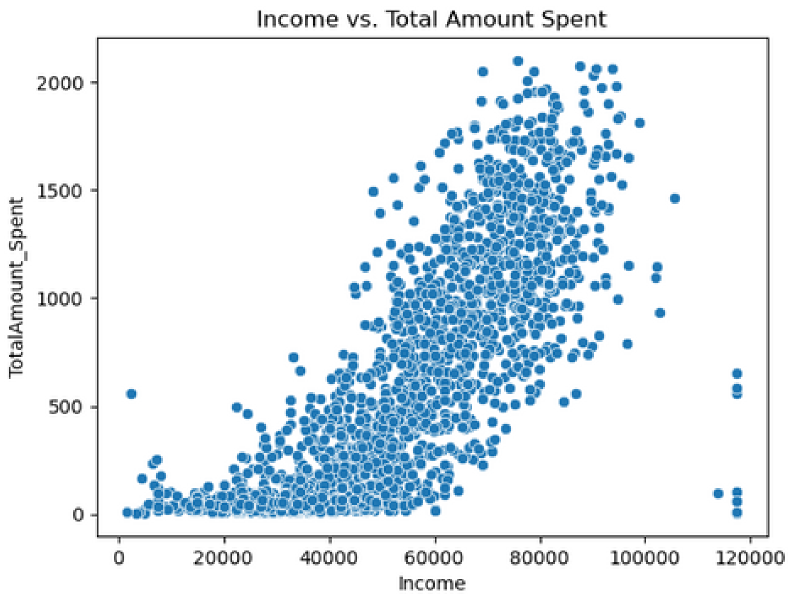


Distribution in Age Column

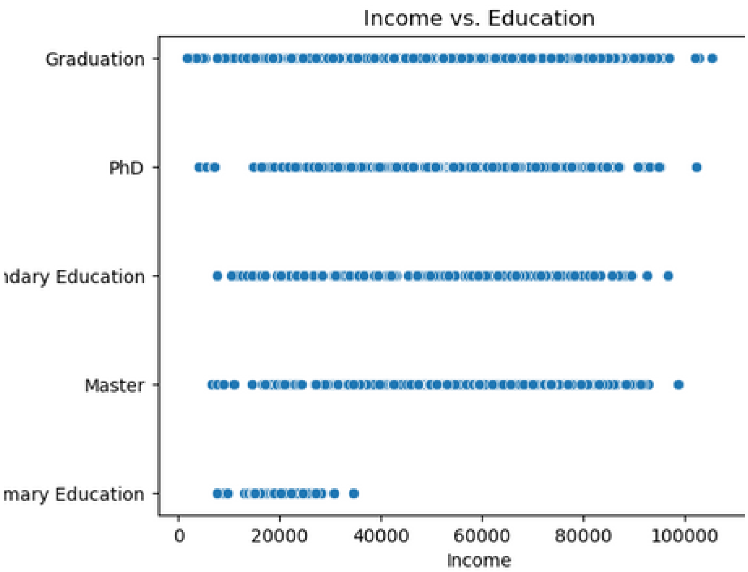


Bivariate Analysis

Relationships between variables, such as "Income" vs. "TotalAmount_Spent."

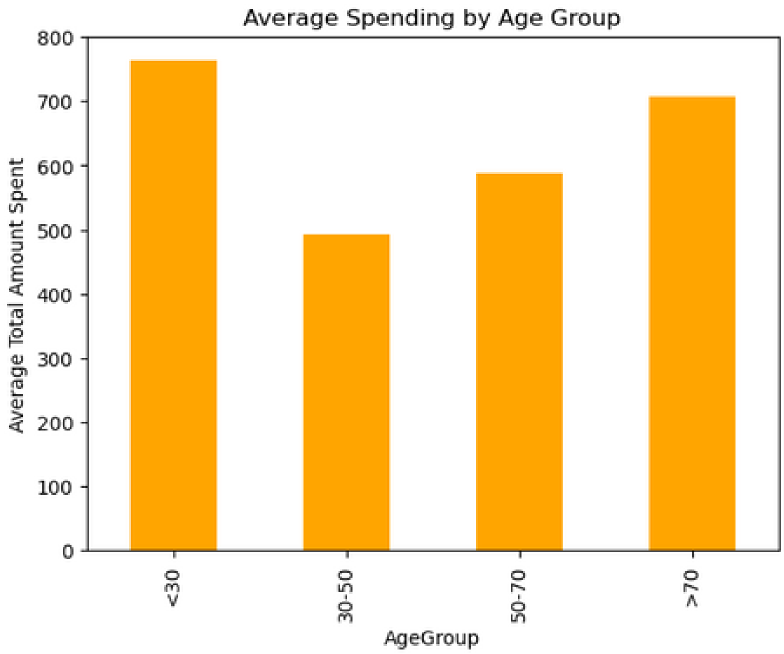


Relationships between variables, such as "Income" vs. "Education."



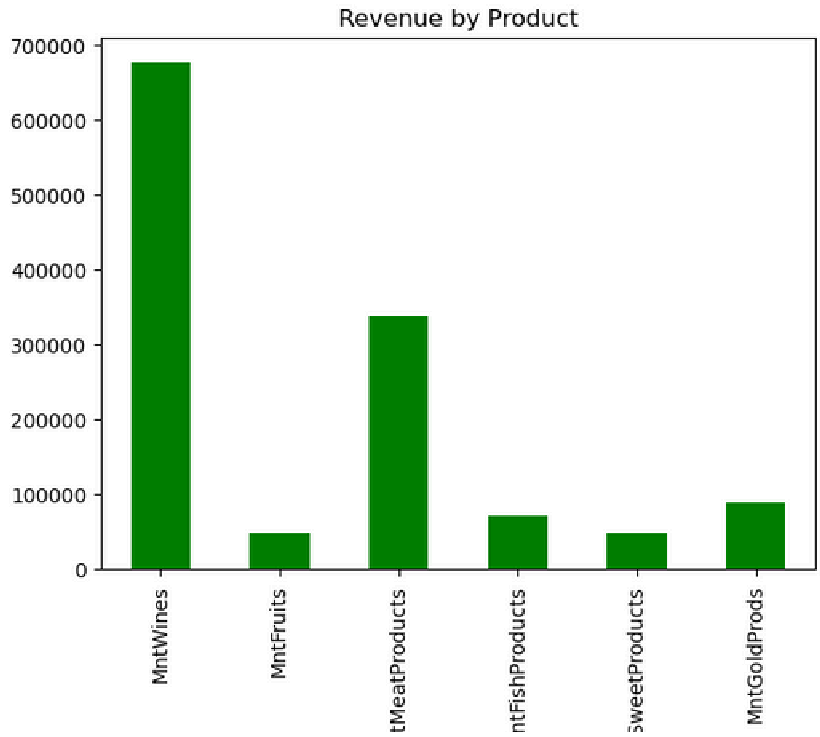
Data Visualization and Insights

Analyzing Spending by Product Categories and Age Groups



Assessing Marketing Campaign Success

Product Performance Analysis



Key Findings

- Age Distribution: A majority of customers fall within a certain age range, suggesting targeting specific age groups might be beneficial.
- Spending Patterns: Higher income is positively correlated with higher spending across all product categories.
- Outlier Management: Effective capping of outliers ensured more robust and reliable statistical analysis.

Recommendations

- Based on the EDA, here are a few recommendations:
- Campaign Targeting: Focus future campaigns on segments with higher acceptance rates. Campaign 5 seems the most successful; future campaigns could model its approach.
- Product Marketing: Allocate more resources to promoting high-revenue products like wines and meats while considering strategies to boost sales of lower-performing items like sweets.
- Customer Segmentation: Segment customers by age groups and tailor marketing messages accordingly. Older customers (50-70) spend more, so offering premium products or loyalty programs may be effective.

Conclusion

The EDA provided valuable insights into customer demographics and spending patterns. The analysis highlighted several key areas for potential business strategies, including targeted marketing and product bundling. Further research could involve predictive modeling to anticipate customer needs and optimize marketing efforts.