4lk4cn3od

February 18, 2024

1 Analyse et Modélisation de la Rétention des Avis Clients dans le Domaine du Commerce Électronique : Une Approche par Régression Logistique et Évaluation Approfondie des Performances

Lien pour telecharger le jeu de données : https://www.kaggle.com/datasets/swathiunnikrishnan/amazon-consumer-behaviour-dataset

2 Partie 1: Analyse descriptive des données

2.1 Importation des bibliothèques nécessaires et chargement de jeu de données "Amazon Customer behavior Survery" :

[612]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import warnings
from itertools import combinations
from scipy.stats import chi2_contingency
import pandas as pd
from sklearn.feature_selection import chi2, SelectKBest
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,␣
↪classification_report
import statsmodels.api as sm
warnings.filterwarnings('ignore')
```

[613]:
```
data  = pd.read_csv("C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
↪Regression\\Amazon Customer Behavior Survey\\dataset\\Amazon Customer␣
↪Behavior Survey.csv")
```

2.2 Analyse éxploratoire de données :

2.2.1 Aperçu des données :

[614]:
```
# Affichage des données :
data
```

[614]:
```
                          Timestamp  age             Gender  \
0    2023/06/04 1:28:19 PM GMT+5:30   23             Female
1    2023/06/04 2:30:44 PM GMT+5:30   23             Female
2    2023/06/04 5:04:56 PM GMT+5:30   24  Prefer not to say
3    2023/06/04 5:13:00 PM GMT+5:30   24             Female
4    2023/06/04 5:28:06 PM GMT+5:30   22             Female
..                                 …  …                …
597  2023/06/12 4:02:02 PM GMT+5:30   23             Female
598  2023/06/12 4:02:53 PM GMT+5:30   23             Female
599  2023/06/12 4:03:59 PM GMT+5:30   23             Female
600  2023/06/12 9:57:20 PM GMT+5:30   23             Female
601  2023/06/16 9:16:05 AM GMT+5:30   23             Female

        Purchase_Frequency  \
0         Few times a month
1              Once a month
2         Few times a month
3              Once a month
4    Less than once a month
..                      …
597             Once a week
598             Once a week
599            Once a month
600       Few times a month
601             Once a week

                                  Purchase_Categories  \
0                            Beauty and Personal Care
1                                Clothing and Fashion
2      Groceries and Gourmet Food;Clothing and Fashion
3    Beauty and Personal Care;Clothing and Fashion;…
4        Beauty and Personal Care;Clothing and Fashion
..                                                 …
597                          Beauty and Personal Care
598                              Clothing and Fashion
599                          Beauty and Personal Care
600  Beauty and Personal Care;Clothing and Fashion;…
601                              Clothing and Fashion

    Personalized_Recommendation_Frequency    Browsing_Frequency  \
0                                     Yes      Few times a week
```

```
1                                   Yes     Few times a month
2                                    No     Few times a month
3                                Sometimes  Few times a month
4                                   Yes     Few times a month
..                                  …                  …
597                              Sometimes   Few times a week
598                              Sometimes   Few times a week
599                              Sometimes   Few times a week
600                                 Yes     Few times a month
601                              Sometimes  Multiple times a day

    Product_Search_Method Search_Result_Exploration  \
0                 Keyword            Multiple pages
1                 Keyword            Multiple pages
2                 Keyword            Multiple pages
3                 Keyword                First page
4                  Filter            Multiple pages
..                    …                       …
597            categories            Multiple pages
598                Filter            Multiple pages
599            categories            Multiple pages
600               Keyword            Multiple pages
601               Keyword            Multiple pages

     Customer_Reviews_Importance  … Saveforlater_Frequency Review_Left  \
0                              1  …              Sometimes         Yes
1                              1  …                 Rarely          No
2                              2  …                 Rarely          No
3                              5  …              Sometimes         Yes
4                              1  …                 Rarely          No
..                           …  …                    …         …
597                            4  …              Sometimes         Yes
598                            3  …              Sometimes         Yes
599                            3  …              Sometimes         Yes
600                            1  …              Sometimes          No
601                            3  …              Sometimes         Yes

    Review_Reliability Review_Helpfulness  \
0         Occasionally               Yes
1              Heavily               Yes
2         Occasionally                No
3              Heavily               Yes
4              Heavily               Yes
..                 …                 …
597          Moderately          Sometimes
598             Heavily          Sometimes
599         Occasionally         Sometimes
```

```
600            Heavily                Yes
601         Moderately          Sometimes

    Personalized_Recommendation_Frequency  Recommendation_Helpfulness  \
0                                       2                         Yes
1                                       2                   Sometimes
2                                       4                          No
3                                       3                   Sometimes
4                                       4                         Yes
..                                    ...                         ...
597                                     3                   Sometimes
598                                     3                   Sometimes
599                                     3                   Sometimes
600                                     2                         Yes
601                                     3                   Sometimes

    Rating_Accuracy   Shopping_Satisfaction     Service_Appreciation  \
0                 1                       1       Competitive prices
1                 3                       2   Wide product selection
2                 3                       3       Competitive prices
3                 3                       4       Competitive prices
4                 2                       2       Competitive prices
..              ...                     ...                      ...
597               3                       4       Competitive prices
598               3                       3  Product recommendations
599               2                       3   Wide product selection
600               2                       2   Wide product selection
601               3                       3  Product recommendations

                  Improvement_Areas
0           Reducing packaging waste
1           Reducing packaging waste
2      Product quality and accuracy
3      Product quality and accuracy
4      Product quality and accuracy
..                              ...
597  Customer service responsiveness
598         Reducing packaging waste
599     Product quality and accuracy
600     Product quality and accuracy
601     Product quality and accuracy

[602 rows x 23 columns]
```

[615]:
```
#Renommer la colonne 'Personalized_Recommendation_Frequency'
#en 'Personalized_Recommendation_Frequency_Nominale' dans les données.
data.rename(columns={'Personalized_Recommendation_Frequency':
```

```
                    'Personalized_Recommendation_Frequency_Nominale␣
↪'},inplace=True)
```

[616]:
```
# Affichages des informations des varibales :
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 602 entries, 0 to 601
Data columns (total 23 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   Timestamp                                     602 non-null    object
 1   age                                           602 non-null    int64
 2   Gender                                        602 non-null    object
 3   Purchase_Frequency                            602 non-null    object
 4   Purchase_Categories                           602 non-null    object
 5   Personalized_Recommendation_Frequency_Nominale  602 non-null  object
 6   Browsing_Frequency                            602 non-null    object
 7   Product_Search_Method                         600 non-null    object
 8   Search_Result_Exploration                     602 non-null    object
 9   Customer_Reviews_Importance                   602 non-null    int64
 10  Add_to_Cart_Browsing                          602 non-null    object
 11  Cart_Completion_Frequency                     602 non-null    object
 12  Cart_Abandonment_Factors                      602 non-null    object
 13  Saveforlater_Frequency                        602 non-null    object
 14  Review_Left                                   602 non-null    object
 15  Review_Reliability                            602 non-null    object
 16  Review_Helpfulness                            602 non-null    object
 17  Personalized_Recommendation_Frequency         602 non-null    int64
 18  Recommendation_Helpfulness                    602 non-null    object
 19  Rating_Accuracy                               602 non-null    int64
 20  Shopping_Satisfaction                         602 non-null    int64
 21  Service_Appreciation                          602 non-null    object
 22  Improvement_Areas                             602 non-null    object
dtypes: int64(5), object(18)
memory usage: 108.3+ KB
```

[617]:
```
# Affichages des colonnes :
data.columns
```

[617]:
```
Index(['Timestamp', 'age', 'Gender', 'Purchase_Frequency',
       'Purchase_Categories',
       'Personalized_Recommendation_Frequency_Nominale ', 'Browsing_Frequency',
       'Product_Search_Method', 'Search_Result_Exploration',
       'Customer_Reviews_Importance', 'Add_to_Cart_Browsing',
       'Cart_Completion_Frequency', 'Cart_Abandonment_Factors',
       'Saveforlater_Frequency', 'Review_Left', 'Review_Reliability',
```

```
       'Review_Helpfulness', 'Personalized_Recommendation_Frequency ',
       'Recommendation_Helpfulness', 'Rating_Accuracy ',
       'Shopping_Satisfaction', 'Service_Appreciation', 'Improvement_Areas'],
      dtype='object')
```

à propos des données :

1. age = âge
2. gender = genre
3. Purchase_Frequency = How frequently do you make purchases on Amazon? (À quelle fréquence effectuez-vous des achats sur Amazon ?)
4. Purchase_Categories = What product categories do you typically purchase on Amazon? (Dans quelles catégories de produits achetez-vous généralement sur Amazon ?)
5. Personalized_Recommendation_Frequency = Have you ever made a purchase based on personalized product recommendations from Amazon? (Avez-vous déjà effectué un achat basé sur des recommandations de produits personnalisées d'Amazon ?)
6. Browsing_Frequency = How often do you browse Amazon's website or app? (À quelle fréquence parcourez-vous le site web ou l'application d'Amazon ?)
7. Product_Search_Method = How do you search for products on Amazon? (Comment recherchez-vous des produits sur Amazon ?)
8. Search_Result_Exploration = Do you tend to explore multiple pages of search results or focus on the first page? (Avez-vous tendance à explorer plusieurs pages de résultats de recherche ou à vous concentrer sur la première page ?)
9. Customer_Reviews_Importance = How important are customer reviews in your decision-making process? (Dans quelle mesure les avis des clients sont-ils importants dans votre processus de prise de décision ?)
10. Add_to_Cart_Browsing = Do you add products to your cart while browsing on Amazon? (Ajoutez-vous des produits à votre panier tout en naviguant sur Amazon ?)
11. Cart_Completion_Frequency = How often do you complete the purchase after adding products to your cart? (À quelle fréquence finalisez-vous un achat après avoir ajouté des produits à votre panier ?)
12. Cart_Abandonment_Factors = What factors influence your decision to abandon a purchase in your cart? (Quels facteurs influencent votre décision d'abandonner un achat dans votre panier ?)
13. Saveforlater_Frequency = Do you use Amazon's "Save for Later" feature, and if so, how often? (Utilisez-vous la fonction "Enregistrer pour plus tard" d'Amazon, et si oui, à quelle fréquence ?)
14. Review_Left = Have you ever left a product review on Amazon? (Avez-vous déjà laissé un avis sur un produit sur Amazon ?)
15. Review_Reliability = How much do you rely on product reviews when making a purchase? (Dans quelle mesure faites-vous confiance aux avis sur les produits lorsque vous effectuez un achat ?)
16. Review_Helpfulness = Do you find helpful information from other customers' reviews? (Trouvez-vous des informations utiles dans les avis des autres clients ?)
17. Personalized_Recommendation_Frequency = How often do you receive personalized product recommendations from Amazon? (À quelle fréquence recevez-vous des recommandations de produits personnalisées d'Amazon ?)
18. Recommendation_Helpfulness = Do you find the recommendations helpful? (Trouvez-vous

les recommandations utiles ?)

19. Rating_Accuracy = How would you rate the relevance and accuracy of the recommendations you receive (Comment évalueriez-vous la pertinence et la précision des recommandations que vous recevez ?)
20. Shopping_Satisfaction = How satisfied are you with your overall shopping experience on Amazon? (Dans quelle mesure êtes-vous satisfait de votre expérience d'achat globale sur Amazon ?)
21. Service_Appreciation = What aspects of Amazon's services do you appreciate the most? (Quels aspects des services d'Amazon appréciez-vous le plus ?)
22. Improvement_Areas = Are there any areas where you think Amazon can improve? (Y a-t-il des domaines où vous pensez qu'Amazon pourrait s'améliorer ?)
23. Timestamp = Time and date at which the data was entered (Heure et date à laquelle les données ont été saisies)

2.2.2 Statistiques descriptives :

[618]:
```
data.describe()
```

[618]:

	age	Customer_Reviews_Importance	Personalized_Recommendation_Frequency	Rating_Accuracy	Shopping_Satisfaction
count	602.000000	602.000000	602.000000	602.000000	602.000000
mean	30.790698	2.480066	2.699336	2.672757	2.463455
std	10.193276	1.185226	1.042028	0.899744	1.012152
min	3.000000	1.000000	1.000000	1.000000	1.000000
25%	23.000000	1.000000	2.000000	2.000000	2.000000
50%	26.000000	3.000000	3.000000	3.000000	2.000000
75%	36.000000	3.000000	3.000000	3.000000	3.000000
max	67.000000	5.000000	5.000000	5.000000	5.000000

Interprétation des statistiques descriptives pour les variables : `age`, `Customer_Reviews_Importance`, `Personalized_Recommendation_Frequency`, `Rating_Accuracy`, et `Shopping_Satisfaction` :

1. `age` :
   - Comptage (count) : Il y a 602 observations pour l'âge, couvrant 602 clients.
   - Moyenne (mean) : L'âge moyen des clients est d'environ 30.79 ans.
   - Écart-type (std) : L'écart-type de 10.19 indique que l'âge des clients varie considérablement autour de la moyenne.
   - Minimum (min) : L'âge minimum est de 3 ans, ce qui semble inhabituellement bas et nécessite une vérification de la qualité des données.
   - Maximum (max) : L'âge maximum est de 67 ans, ce qui semble raisonnable.
   - Quartiles (25%, 50%, 75%) : Les quartiles révèlent la répartition de l'âge. Par exemple, au 25e percentile (Q1), l'âge est d'environ 23 ans, tandis qu'au 75e percentile (Q3), il est d'environ 36 ans. La médiane (50e percentile) est de 26 ans.
2. `Customer_Reviews_Importance`, `Personalized_Recommendation_Frequency`, `Rating_Accuracy`, et `Shopping_Satisfaction` :
   - Comptage (count) : Il y a 602 observations pour chacune de ces variables.
   - Moyenne (mean) : La moyenne pour chaque variable représente la valeur moyenne donnée par les clients sur une échelle de 1 à 5.
   - Écart-type (std) : L'écart-type mesure la dispersion des réponses des clients par rapport à la moyenne.
   - Minimum (min) : Le minimum est de 1 pour chacune de ces variables, indiquant que 1 est la note minimale possible.
   - Maximum (max) : Le maximum est de 5 pour chacune de ces variables, indiquant que 5 est la note maximale possible.
   - Quartiles (25%, 50%, 75%) : Les quartiles montrent la répartition des réponses des clients pour chaque variable.

Ces statistiques offrent un aperçu des caractéristiques des variables et de leur distribution. Par exemple, pour `age`, nous observons une large dispersion d'âges, de 3 à 67 ans, avec une moyenne d'environ 30.79 ans. Pour les autres variables, les évaluations varient de 1 à 5, avec des moyennes spécifiques et des niveaux de dispersion distincts. Ces informations sont utiles pour mieux comprendre les données et orienter les décisions relatives à la segmentation des clients ou à d'autres analyses ultérieures.

[621]:
```
data.drop(data[data['age'] == 3].index, axis=0, inplace=True)
```

2.2.3 Distributions des variables :

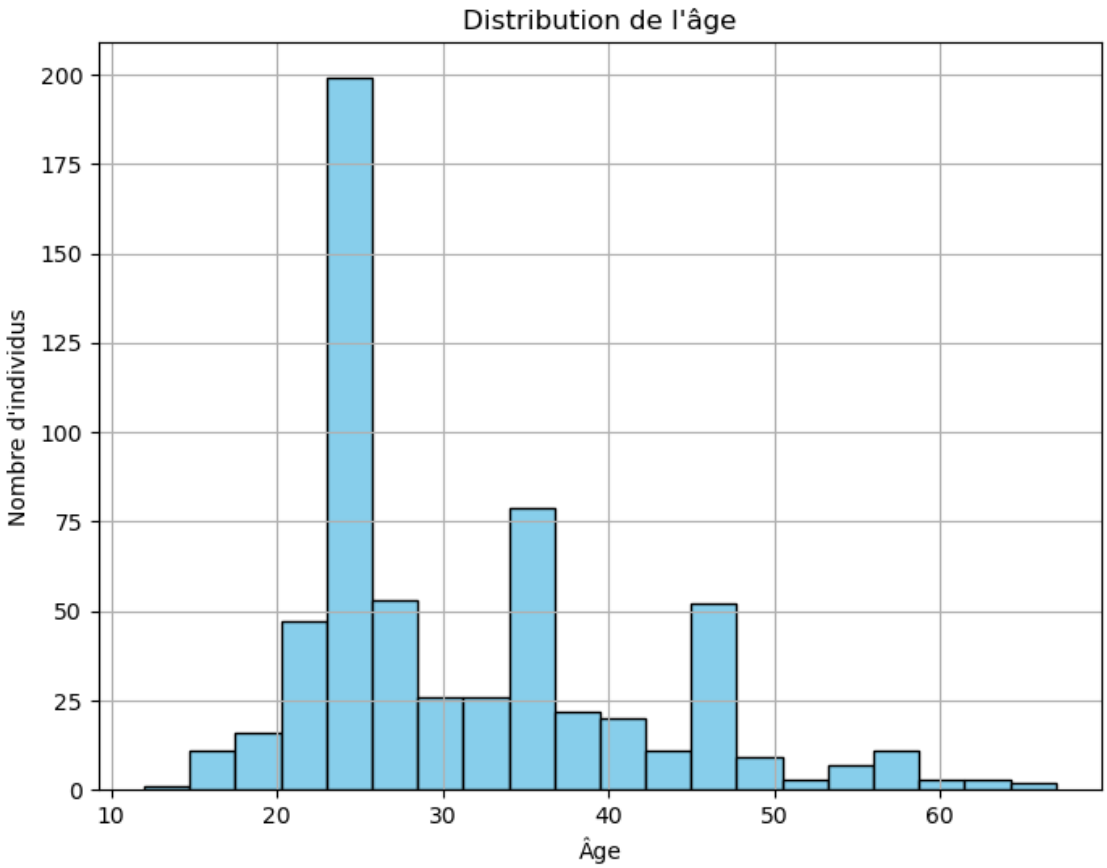
Quelle est la répartition de l'âge des clients et quel est l'âge moyen ?

[622]:
```
# comptes des valeurs unique de variable age :
data['age'].value_counts()
```

[622]:
```
age
23    123
34     48
24     40
```

```
25     36
45     34
21     30
26     27
32     19
27     17
22     17
36     16
35     15
37     14
40     12
46     12
29      9
31      9
28      9
30      8
56      8
33      7
18      7
47      6
54      6
43      6
50      5
44      5
20      5
16      5
38      4
41      4
19      4
39      4
17      4
42      4
48      3
60      3
67      2
15      2
57      2
53      2
64      1
58      1
49      1
63      1
52      1
55      1
62      1
12      1
Name: count, dtype: int64
```

[623]:
```
# vizualisation de la distribution de l'age :
plt.figure(figsize=(8,6))
plt.hist(data['age'],bins=20,color='skyblue',edgecolor='black')
plt.title('Distribution de l\'âge')
plt.xlabel('Âge')
plt.ylabel('Nombre d\'individus')
plt.grid(True)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\1␣
 ↪Distribution de l\'âge.png'  # Remplacez par le chemin et le nom de fichier␣
 ↪souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



La répartition des âges des clients dans notre jeu de données est la suivante :

- L'âge le plus fréquent est de 23 ans, avec 123 individus.
- L'âge le moins fréquent est de 64 ans, avec seulement 1 individu.

- Plusieurs autres âges ont un nombre relativement élevé d'individus, tels que 34 ans (48 individus), 24 ans (40 individus) et 25 ans (36 individus).

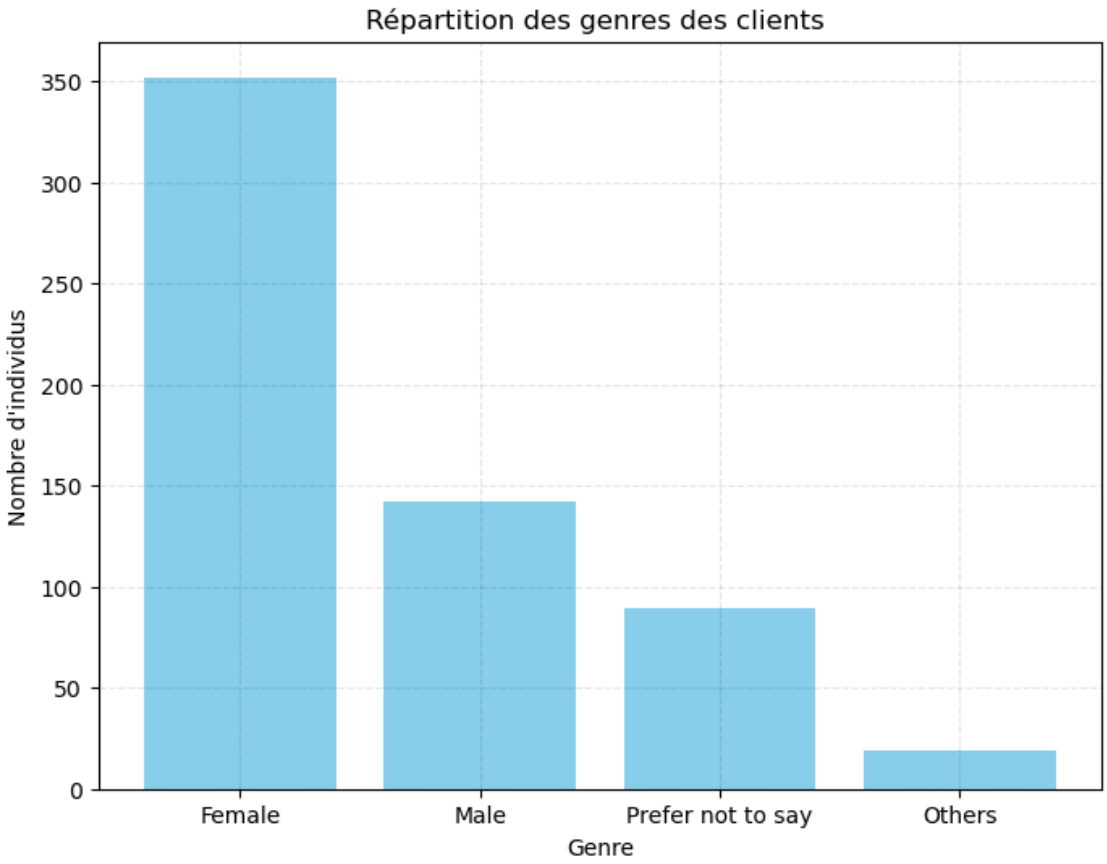
D'après le tableau de la statistique descriptive, l'âge moyen est de 30 ans.

Quelle est la répartition des genres des clients (homme, femme, autre) ?

[625]:
```
data['Gender'].value_counts()
```

[625]:
```
Gender
Female               352
Male                 142
Prefer not to say     88
Others                19
Name: count, dtype: int64
```

[626]:
```
Gender_Frequency = [352,142,89,19]
Genre = ['Female','Male','Prefer not to say','Others']
plt.figure(figsize=(8,6))
plt.bar(Genre,Gender_Frequency,color='skyblue')
plt.title('Répartition des genres des clients ')
plt.xlabel("Genre")
plt.ylabel("Nombre d\'individus")
plt.grid(True,linestyle='--',color='black',alpha=0.1)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\2␣
 ↪Répartition des genres des clients.png'  # Remplacez par le chemin et le nom␣
 ↪de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres, voici l'interprétation des données :

- **Femmes** : Il y a plus de clientes féminines que toute autre catégorie. Cela pourrait indiquer que les produits ou services offerts sont plus populaires ou plus ciblés vers les femmes.
- **Hommes** : Le nombre de clients masculins est inférieur à celui des femmes, mais reste une part importante de la clientèle.
- **Préfère ne pas dire** : Un certain nombre de clients ont choisi de ne pas divulguer leur genre. Cela pourrait indiquer une préférence pour la confidentialité ou l'anonymat.
- **Autres** : Il y a moins de clients qui s'identifient comme "autres". Cela pourrait inclure une variété de genres non binaires ou non conformes.

la répartition des genres des clients est donc dominée par les femmes, suivies des hommes, avec un nombre plus petit de clients qui préfèrent ne pas divulguer leur genre ou qui s'identifient comme "autres". Cette information pourrait être utile pour comprendre le profil démographique de notre clientèle et pour élaborer des stratégies de marketing ciblées.

À quelle fréquence les clients effectuent-ils des achats sur Amazon ?

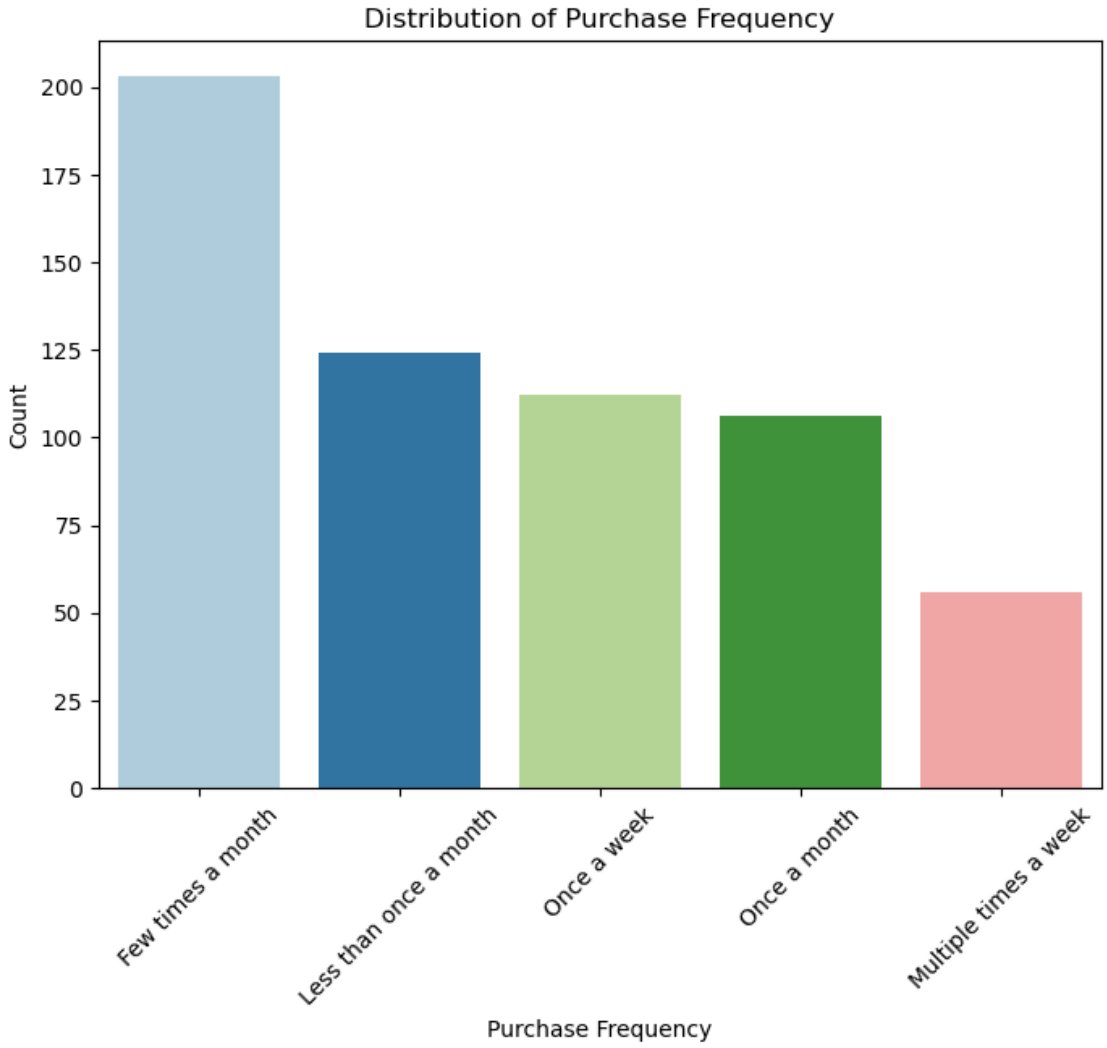
[627]:
```
data['Purchase_Frequency'].value_counts()
```

[627]:
```
Purchase_Frequency
Few times a month          203
Less than once a month     124
Once a week                112
Once a month               106
Multiple times a week       56
Name: count, dtype: int64
```

Certainly! Here are 10 other color palettes that you can consider using in Seaborn:

'deep' 'muted' 'colorblind' 'husl' 'pastel' 'dark' 'RdBu_r' 'cubehelix' 'Set2' 'Paired'

[628]:
```
# Count the occurrences of each category
purchase_frequency_counts = data['Purchase_Frequency'].value_counts()

# Create a bar plot
plt.figure(figsize=(8, 6))
sns.barplot(x=purchase_frequency_counts.index, y=purchase_frequency_counts.
 ↪values, palette='Paired')

# Add labels and title
plt.xlabel('Purchase Frequency')
plt.ylabel('Count')
plt.title('Distribution of Purchase Frequency')
plt.xticks(rotation=45)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\3␣
 ↪Distribution of Purchase Frequency.png'  # Remplacez par le chemin et le nom␣
 ↪de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
# Show the plot
plt.show()
```



D'après le graphique à barres :

- **Une fois par mois** : C'est la fréquence d'achat la plus courante parmi les clients, avec environ 175 personnes effectuant des achats mensuels. Cela pourrait indiquer que la majorité des clients utilisent Amazon pour des achats réguliers ou planifiés.
- **2-3 fois par mois** : Un nombre significatif de clients achète 2 à 3 fois par mois. Ces clients pourraient être ceux qui effectuent des achats plus fréquents ou qui utilisent Amazon pour des besoins variés.
- **Une fois par semaine** : Un nombre plus restreint de clients effectue des achats une fois par semaine. Ces clients pourraient être ceux qui dépendent fortement d'Amazon pour leurs achats hebdomadaires.
- **2-3 fois par semaine** : Très peu de clients achètent 2 à 3 fois par semaine. Ces clients pourraient être ceux qui font des achats fréquents ou qui dépendent fortement d'Amazon pour leurs besoins quotidiens.

- **4+ fois par semaine** : Le nombre de clients qui achètent 4 fois ou plus par semaine est le plus bas, avec environ 25 personnes. Ces clients pourraient être ceux qui utilisent Amazon pour une grande variété de besoins ou qui effectuent des achats fréquents.

La majorité des clients effectuent des achats sur Amazon une fois par mois, suivis de ceux qui achètent 2 à 3 fois par mois. Un nombre plus restreint de clients effectue des achats une fois par semaine ou plus. Cette information pourrait être utile pour comprendre les habitudes d'achat de notre clientèle et élaborer des stratégies de marketing ciblées.

Quelles catégories de produits sont les plus populaires parmi les clients d'Amazon ?

[231]:
```
data['Purchase_Categories'].unique()
```

[231]:
```
array(['Beauty and Personal Care', 'Clothing and Fashion',
       'Groceries and Gourmet Food;Clothing and Fashion',
       'Beauty and Personal Care;Clothing and Fashion;others',
       'Beauty and Personal Care;Clothing and Fashion',
       'Beauty and Personal Care;Clothing and Fashion;Home and Kitchen',
       'Clothing and Fashion;Home and Kitchen', 'others',
       'Clothing and Fashion;others',
       'Beauty and Personal Care;Home and Kitchen',
       'Groceries and Gourmet Food',
       'Groceries and Gourmet Food;Clothing and Fashion;others',
       'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and
Fashion;Home and Kitchen',
       'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and
Fashion;Home and Kitchen;others',
       'Home and Kitchen', 'Beauty and Personal Care;others',
       'Beauty and Personal Care;Home and Kitchen;others',
       'Home and Kitchen;others',
       'Groceries and Gourmet Food;Home and Kitchen',
       'Beauty and Personal Care;Clothing and Fashion;Home and Kitchen;others',
       'Groceries and Gourmet Food;Beauty and Personal Care;Home and Kitchen',
       'Groceries and Gourmet Food;Home and Kitchen;others',
       'Groceries and Gourmet Food;Clothing and Fashion;Home and
Kitchen;others',
       'Groceries and Gourmet Food;Beauty and Personal Care',
       'Clothing and Fashion;Home and Kitchen;others',
       'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and
Fashion',
       'Groceries and Gourmet Food;Clothing and Fashion;Home and Kitchen',
       'Groceries and Gourmet Food;Beauty and Personal Care;others',
       'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and
Fashion;others'],
      dtype=object)
```

Remarque : Il est à noter que les catégories semblent être correctement répertoriées dans les données, cependant, certaines d'entre elles sont des combinaisons de plusieurs catégories. Cette observation suggère la présence de catégories d'achat multiples ou de regroupements de produits

au sein de l'échantillon, ce qui peut fournir des informations précieuses sur les préférences d'achat des clients.

On va faire des étapes pour résoudre ce problème :

[301]:
```
# étape 1 (Netoyage des categories ) :
#Pour garantir que toutes les catégories sont uniformes, sans espaces␣
 ↪supplémentaires et en lettres minuscules.
data['Purchase_Categories'] = data['Purchase_Categories'].str.strip().str.
 ↪lower()
```

[302]:
```
# étape 2 création d'une dataFrame de categorie et counts
category_counts = data['Purchase_Categories'].value_counts()
df = category_counts.reset_index()
df.columns = ['Category','Counts']
print(df)
```

```
                                             Category  Counts
0                             beauty and personal care     106
1                                 clothing and fashion     106
2                                               others      48
3          beauty and personal care;clothing and fashion      46
4   beauty and personal care;clothing and fashion;…      42
5   groceries and gourmet food;beauty and personal…      32
6                clothing and fashion;home and kitchen      27
7                                     home and kitchen      24
8             beauty and personal care;home and kitchen      21
9          clothing and fashion;home and kitchen;others      16
10                        clothing and fashion;others      14
11                         groceries and gourmet food      14
12  groceries and gourmet food;beauty and personal…      14
13  beauty and personal care;clothing and fashion;…      12
14  groceries and gourmet food;beauty and personal…      10
15                            home and kitchen;others       9
16  beauty and personal care;clothing and fashion;…       8
17                     beauty and personal care;others       7
18  groceries and gourmet food;beauty and personal…       7
19  groceries and gourmet food;home and kitchen;ot…       6
20    groceries and gourmet food;clothing and fashion       6
21        groceries and gourmet food;home and kitchen       5
22   beauty and personal care;home and kitchen;others       5
23  groceries and gourmet food;beauty and personal…       4
24  groceries and gourmet food;clothing and fashio…       4
25  groceries and gourmet food;clothing and fashio…       3
26  groceries and gourmet food;beauty and personal…       3
27  groceries and gourmet food;clothing and fashio…       2
28  groceries and gourmet food;beauty and personal…       1
```

[303]:
```
# étape 3 determination de categorie unique (sans combinaison )
unique_categories = []
categories =df['Category'].unique()
# Parcourir chaque catégorie dans la liste initiale
for category in categories:
    # Diviser la catégorie en sous-catégories en utilisant ';' comme séparateur
    sub_categories = category.split(';')

    # Parcourir chaque sous-catégorie
    for sub_category in sub_categories:
        # Si la sous-catégorie n'est pas déjà dans la liste des catégories␣
 ↪uniques, l'ajouter
        if sub_category not in unique_categories:
            unique_categories.append(sub_category)

# Afficher la liste des catégories uniques
print(unique_categories)
```

```
['beauty and personal care', 'clothing and fashion', 'others', 'home and
kitchen', 'groceries and gourmet food']
```

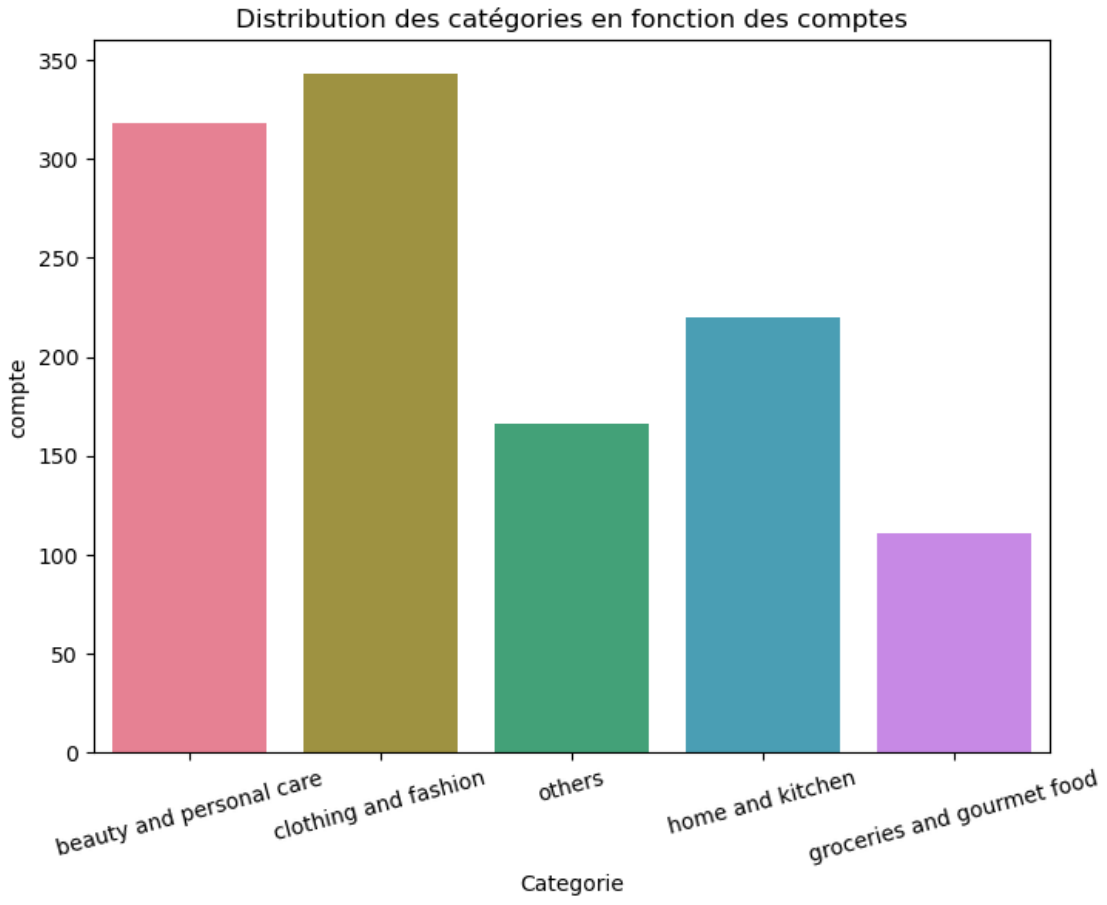
[305]:
```
# étape 4 calcule de comptage de chaque categorie unique :
unique_categories = ['beauty and personal care', 'clothing and fashion',␣
 ↪'others', 'home and kitchen', 'groceries and gourmet food']

# Initialiser un dictionnaire pour stocker les sommes des Counts
sum_counts_dict = {}

# Parcourir chaque catégorie dans la liste unique_categories
for category in unique_categories:
    # Filtrer les lignes où la catégorie contient la catégorie actuelle
    category_df = df[df['Category'].str.contains(category)]

    # Calculer la somme des Counts pour ces lignes
    sum_counts = category_df['Counts'].sum()

    # Ajouter la somme des Counts à la catégorie correspondante dans le␣
 ↪dictionnaire
    sum_counts_dict[category] = sum_counts
```

[306]:
```
# finalement, on trouve le résultats :
# Convertir le dictionnaire en DataFrame
df_counts = pd.DataFrame(list(sum_counts_dict.items()), columns=['Category',␣
 ↪'Counts'])

# Afficher le DataFrame
df_counts
```

[306]:

	Category	Counts
0	beauty and personal care	318
1	clothing and fashion	343
2	others	166
3	home and kitchen	220
4	groceries and gourmet food	111

[400]:
```
plt.figure(figsize=(8,6))
sns.barplot(x=df_counts['Category'],y=df_counts['Counts'],palette='husl')
plt.title('Distribution des catégories en fonction des comptes')
plt.xlabel('Categorie')
plt.ylabel('compte')
plt.xticks(rotation=15)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\4␣
 ↪Distribution des catégories en fonction des comptes.png'  # Remplacez par le␣
 ↪chemin et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres:

- **Beauté et soins personnels** : Cette catégorie a le plus grand nombre de comptes associés, ce qui indique qu'elle est la plus populaire parmi les clients d'Amazon. Les produits de beauté et de soins personnels sont souvent achetés en ligne pour leur commodité et leur variété.
- **Vêtements et mode** : Cette catégorie est la deuxième plus populaire parmi les clients d'Amazon. Cela pourrait indiquer que les clients apprécient la large sélection de vêtements et d'accessoires de mode disponibles sur Amazon.
- **Maison et cuisine** : Cette catégorie est également populaire parmi les clients d'Amazon, avec un nombre significatif de comptes associés. Cela pourrait indiquer que les clients apprécient la commodité d'acheter des articles ménagers et de cuisine en ligne.
- **Épicerie et gastronomie** : Cette catégorie a le moins de comptes associés, ce qui pourrait indiquer que moins de clients achètent des produits d'épicerie et gastronomiques sur Amazon par rapport aux autres catégories.

Les catégories de produits les plus populaires parmi les clients d'Amazon sont la beauté et les soins personnels, les vêtements et la mode, et la maison et la cuisine.

conclusion sur les 3 images précidentes :

1. **Genre des clients** : La première image montre que la majorité des clients sont des femmes. Cela pourrait influencer les types de produits qui sont les plus populaires, comme le montre la troisième image.

2. **Fréquence d'achat** : La deuxième image montre que la plupart des clients achètent une fois par mois. Cela pourrait indiquer que les clients préfèrent acheter en gros ou planifier leurs achats, ce qui pourrait également influencer les types de produits qu'ils achètent.

3. **Catégories de produits populaires** : La troisième image montre que les catégories de produits les plus populaires sont la beauté et les soins personnels, les vêtements et la mode, et la maison et la cuisine. Cela pourrait être lié au genre des clients (plus de femmes) et à leur fréquence d'achat (une fois par mois).

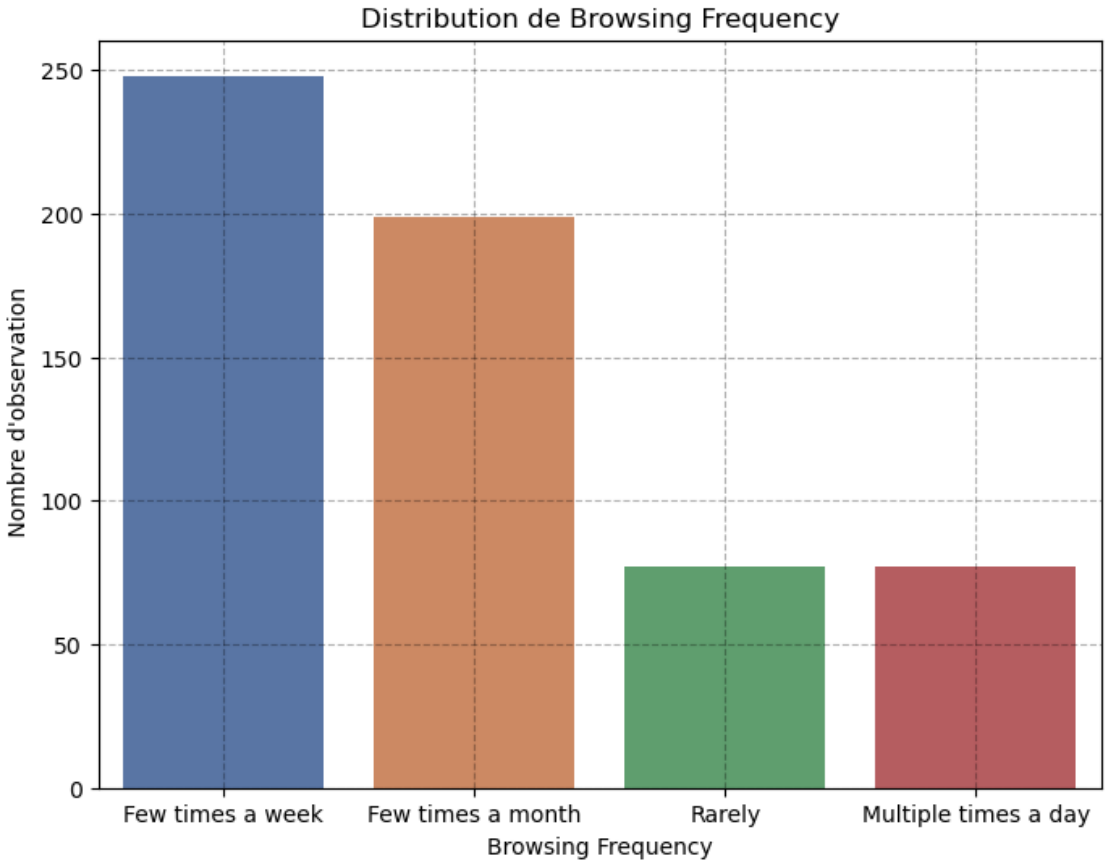
En combinant ces informations, on pourrait dire que les femmes qui achètent une fois par mois sont susceptibles d'acheter des produits de beauté et de soins personnels, des vêtements et de la mode, et des articles pour la maison et la cuisine. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire, comme une analyse de corrélation ou une analyse de régression.

À quelle fréquence les clients parcourent-ils le site web ou l'application d'Amazon ?

[318]:
```
data['Browsing_Frequency'].value_counts()
```

[318]:
```
Browsing_Frequency
Few times a week        249
Few times a month       199
Rarely                   77
Multiple times a day     77
Name: count, dtype: int64
```

[402]:
```
data_B_F = data['Browsing_Frequency'].value_counts()
plt.figure(figsize=(8,6))
sns.barplot(x=data_B_F.index,y=data_B_F,palette='deep')
plt.title('Distribution de Browsing Frequency')
plt.xlabel('Browsing Frequency')
plt.ylabel(' Nombre d\'observation')
plt.grid(True,color='black',linestyle='--',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\6␣
 ↪Distribution de Browsing Frequency.png'  # Remplacez par le chemin et le nom␣
 ↪de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres:

- **Quelques fois par semaine** : La majorité des clients, représentés par la barre bleue la plus haute, parcourent le site web ou l'application d'Amazon quelques fois par semaine. Cela

pourrait indiquer que ces clients utilisent régulièrement Amazon pour parcourir les produits, lire les critiques, comparer les prix, etc.

- **Quelques fois par mois** : Un nombre significatif de clients, représentés par la barre orange, parcourent le site web ou l'application d'Amazon quelques fois par mois. Cela pourrait indiquer que ces clients utilisent Amazon moins fréquemment, peut-être pour des achats planifiés ou occasionnels.
- **Plusieurs fois par jour** : Un nombre plus petit de clients, représentés par la barre verte, parcourent le site web ou l'application d'Amazon plusieurs fois par jour. Ces clients pourraient être ceux qui dépendent fortement d'Amazon pour leurs besoins quotidiens ou qui font des achats fréquents.
- **Rarement** : Le nombre de clients qui parcourent rarement le site web ou l'application d'Amazon, représentés par la barre rouge, est le plus bas. Ces clients pourraient être ceux qui utilisent Amazon très occasionnellement ou qui préfèrent d'autres méthodes d'achat.

La majorité des clients parcourent le site web ou l'application d'Amazon quelques fois par semaine, suivis de ceux qui le font quelques fois par mois. Un nombre plus petit de clients le font plusieurs fois par jour ou rarement.

La fréquence de navigation pourrait être liée à la fréquence d'achat (deuxième image) et à l'utilisation des recommandations de produits personnalisées (quatrième image). Par exemple, les clients qui parcourent le site plusieurs fois par jour pourraient être plus susceptibles d'acheter fréquemment et d'utiliser des recommandations de produits personnalisées. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Comment les clients recherchent-ils des produits sur Amazon, et quel est le moyen le plus courant de recherche ?

[328]:
```
data['Product_Search_Method'].value_counts()
```

[328]:
```
Product_Search_Method
categories    223
Keyword       214
Filter        127
others         36
Name: count, dtype: int64
```

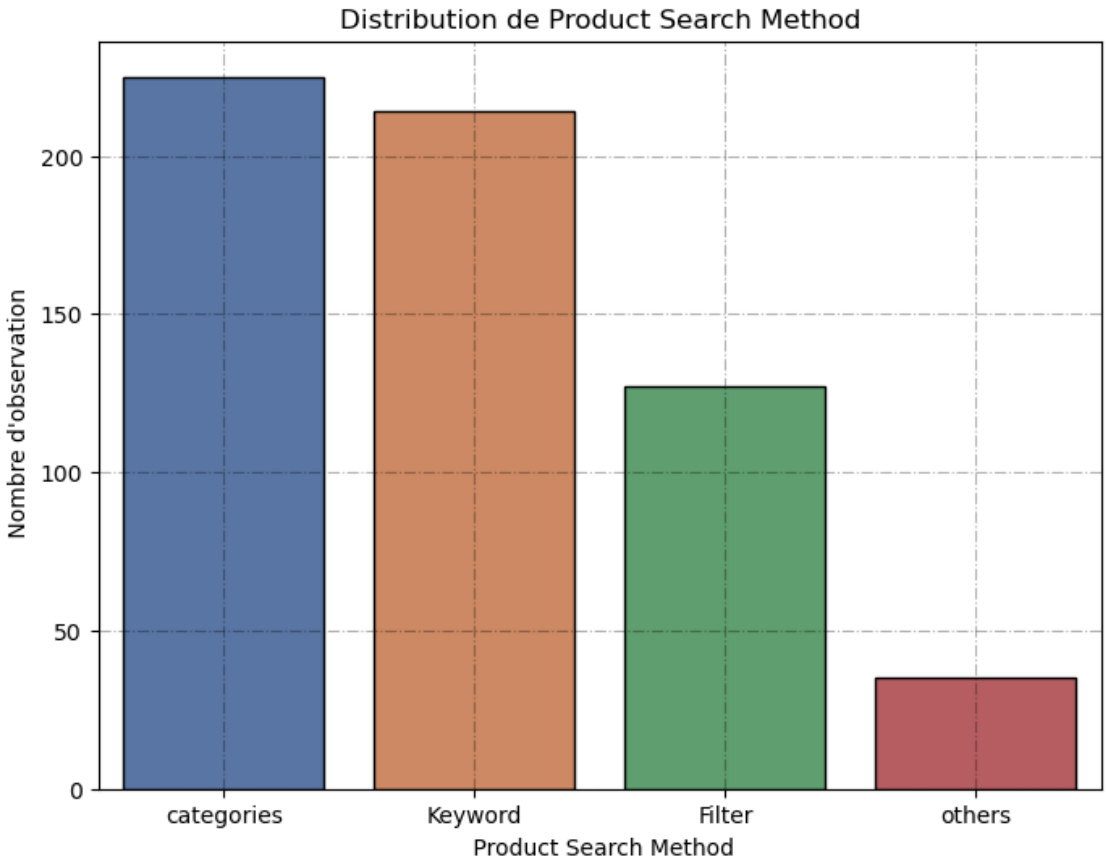
[329]:
```
# vérifié s'il y a des valeurs manquantes dans cette colonnes :
data['Product_Search_Method'].isna().sum()
```

[329]:
```
2
```

[330]:
```
mode_value = data['Product_Search_Method'].mode()[0]
data['Product_Search_Method'] = data['Product_Search_Method'].fillna(mode_value)
```

[403]:
```
data_S_M =data['Product_Search_Method'].value_counts()
plt.figure(figsize=(8,6))
sns.barplot(x=data_S_M.index,y=data_S_M,palette='deep',edgecolor='black')
plt.title('Distribution de Product Search Method')
plt.xlabel(' Product Search Method')
```

```
plt.ylabel(' Nombre d\'observation')
plt.grid(True,linestyle='dashdot',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\7␣
 ↪Distribution de Product Search Method.png'  # Remplacez par le chemin et le␣
 ↪nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **Mots-clés** : La majorité des clients, représentés par la barre bleue la plus haute, recherchent des produits en utilisant des mots-clés. Cela pourrait indiquer que ces clients ont une idée précise de ce qu'ils cherchent et utilisent des mots-clés pour trouver rapidement et efficacement le produit souhaité.
- **Catégories** : Un nombre significatif de clients, représentés par la barre orange, recherchent des produits en naviguant dans les catégories. Cela pourrait indiquer que ces clients préfèrent explorer une variété de produits dans une certaine catégorie avant de faire un choix.

- **Filtres** : Un nombre plus petit de clients, représentés par la barre verte, utilisent des filtres pour rechercher des produits. Ces clients pourraient être ceux qui ont des critères spécifiques pour les produits qu'ils recherchent, comme la gamme de prix, la marque, la note des clients, etc.
- **Autres** : Le nombre de clients qui utilisent d'autres méthodes pour rechercher des produits, représentés par la barre rouge, est le plus bas. Ces méthodes pourraient inclure des choses comme la navigation aléatoire, l'utilisation de recommandations de produits personnalisées, etc.

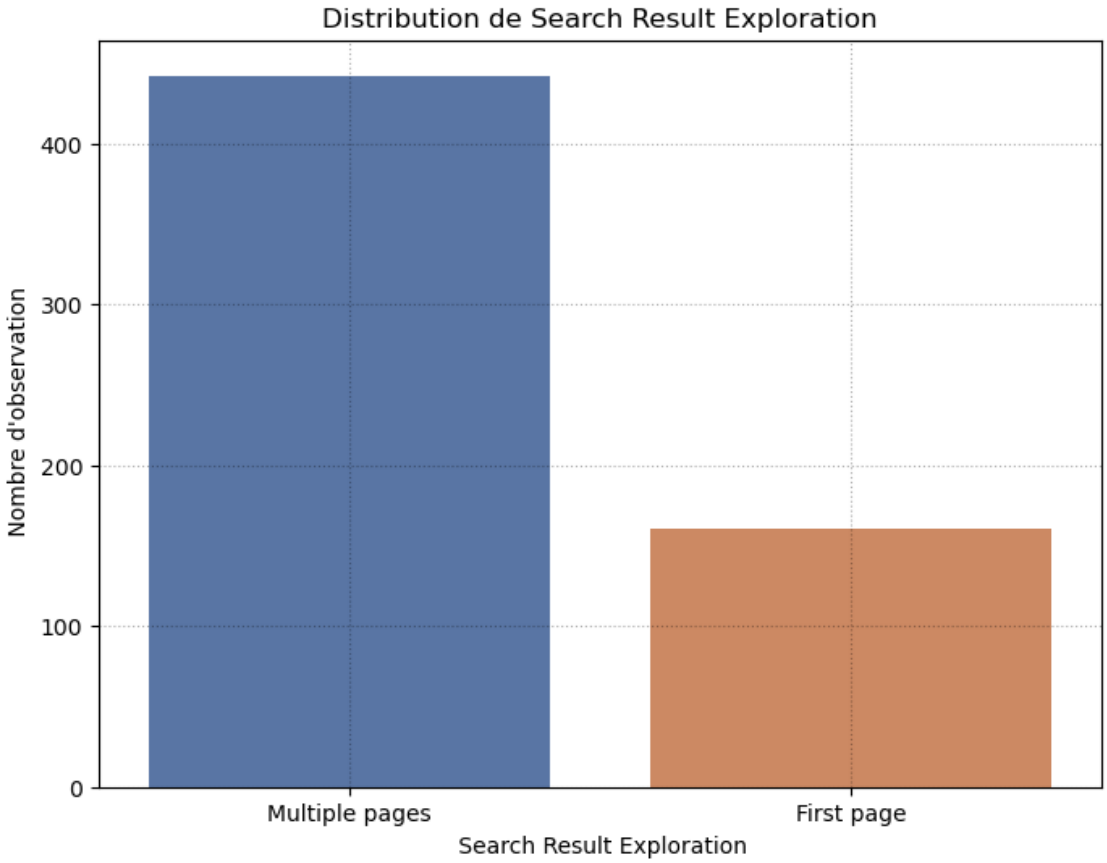
La majorité des clients recherchent des produits sur Amazon en utilisant des mots-clés, suivis de ceux qui naviguent dans les catégories. Un nombre plus petit de clients utilise des filtres ou d'autres méthodes pour rechercher des produits.

La méthode de recherche de produits pourrait être liée à la fréquence de navigation (cinquième image) et à l'utilisation des recommandations de produits personnalisées (quatrième image). Par exemple, les clients qui parcourent le site plusieurs fois par jour pourraient être plus susceptibles d'utiliser des mots-clés pour rechercher des produits, tandis que ceux qui le font moins fréquemment pourraient être plus susceptibles de naviguer dans les catégories. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Les clients ont-ils tendance à explorer plusieurs pages de résultats de recherche ou à se concentrer sur la première page ?

[335]:
```
data_S_R_E=data['Search_Result_Exploration'].value_counts()
```

[406]:
```
plt.figure(figsize=(8,6))
sns.barplot(x=data_S_R_E.index,y=data_S_R_E,palette='deep')
plt.title('Distribution de Search Result Exploration')
plt.xlabel(' Search Result Exploration')
plt.ylabel(' Nombre d\'observation')
plt.grid(True,linestyle=':',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\8␣
 ↪Distribution de Search Result Exploration.png'  # Remplacez par le chemin et␣
 ↪le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
plt.show()
```



D'après le graphique à barres :

- **Plusieurs pages** : La majorité des clients, représentés par la barre bleue la plus haute, ont tendance à explorer plusieurs pages de résultats de recherche. Cela pourrait indiquer que ces clients sont prêts à passer du temps à chercher le produit parfait et ne se limitent pas aux produits présentés sur la première page.
- **Première page** : Un nombre plus petit de clients, représentés par la barre orange, se concentrent sur la première page des résultats de recherche. Ces clients pourraient être ceux qui préfèrent faire des achats rapidement ou qui font confiance aux algorithmes d'Amazon pour présenter les meilleurs produits en premier.

La majorité des clients ont tendance à explorer plusieurs pages de résultats de recherche, tandis qu'un nombre plus petit de clients se concentre sur la première page.

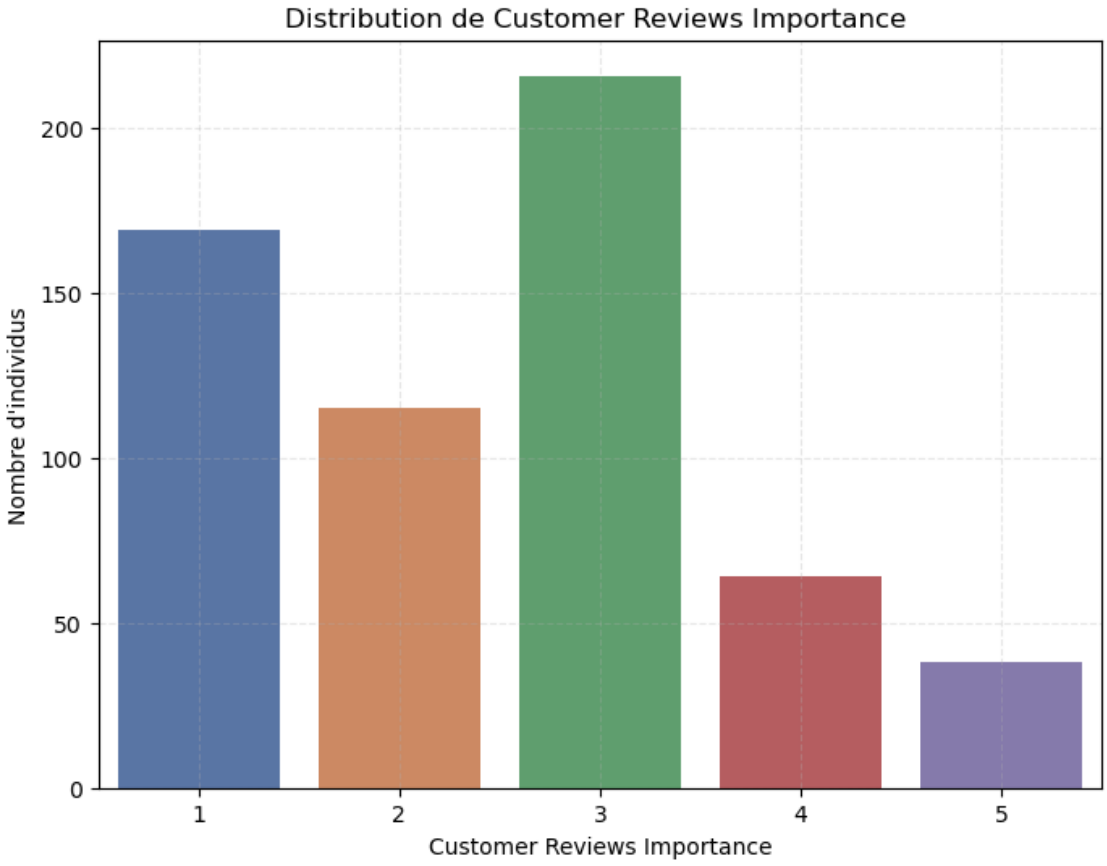
L'exploration des résultats de recherche pourrait être liée à la méthode de recherche de produits (septième image) et à la fréquence de navigation (cinquième image). Par exemple, les clients qui utilisent des mots-clés pour rechercher des produits pourraient être plus susceptibles d'explorer plusieurs pages de résultats, tandis que ceux qui naviguent dans les catégories pourraient être plus susceptibles de se concentrer sur la première page. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Dans quelle mesure les avis des clients sont-ils importants dans le processus de prise

de décision des clients ?

[339]:
```
data_C_R_I = data['Customer_Reviews_Importance'].value_counts()
```

[407]:
```
plt.figure(figsize = (8,6))
sns.barplot(x=data_C_R_I.index,y=data_C_R_I,palette='deep')
plt.title('Distribution de Customer Reviews Importance ')
plt.xlabel('Customer Reviews Importance')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\9␣
 ↪Distribution de Customer Reviews Importance.png'  # Remplacez par le chemin␣
 ↪et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **3 sur 5** : La majorité des clients, représentés par la barre la plus haute, ont évalué l'importance des avis des clients comme étant de 3 sur 5. Cela pourrait indiquer que ces clients considèrent les avis des clients comme modérément importants dans leur processus de prise de décision.
- **2 sur 5** : Un nombre significatif de clients, représentés par la deuxième barre la plus haute, ont évalué l'importance des avis des clients comme étant de 2 sur 5. Cela pourrait indiquer que ces clients considèrent les avis des clients comme étant d'une importance relativement faible dans leur processus de prise de décision.
- **1 sur 5** : Un nombre plus petit de clients, représentés par la troisième barre la plus haute, ont évalué l'importance des avis des clients comme étant de 1 sur 5. Ces clients pourraient être ceux qui ne considèrent pas les avis des clients comme importants dans leur processus de prise de décision.

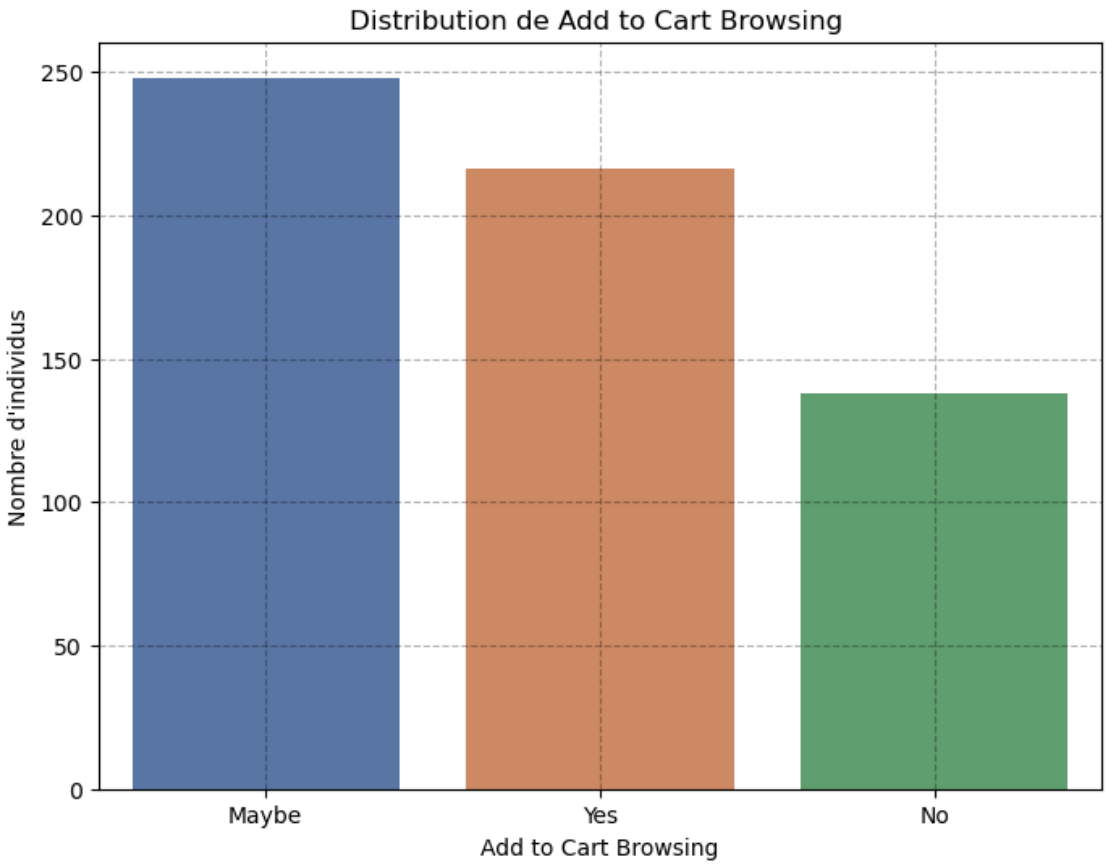
La majorité des clients considèrent que les avis des clients sont modérément importants dans leur processus de prise de décision, tandis qu'un nombre significatif de clients les considèrent comme relativement peu importants.

L'importance des avis des clients pourrait être liée à la méthode de recherche de produits (septième image) et à l'exploration des résultats de recherche (huitième image). Par exemple, les clients qui utilisent des mots-clés pour rechercher des produits et qui explorent plusieurs pages de résultats pourraient être plus susceptibles de considérer les avis des clients comme importants dans leur processus de prise de décision. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Combien de clients ajoutent des produits à leur panier tout en naviguant sur Amazon ?

[345]:
```
data_A_T_C_B = data['Add_to_Cart_Browsing'].value_counts()
```

[408]:
```
plt.figure(figsize = (8,6))
sns.barplot(x=data_A_T_C_B.index,y=data_A_T_C_B,palette='deep')
plt.title('Distribution de Add to Cart Browsing ')
plt.xlabel('Add to Cart Browsing')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\10␣
 ↪Distribution de Add to Cart Browsing.png'  # Remplacez par le chemin et le␣
 ↪nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **Peut-être** : La majorité des clients, représentés par la barre bleue la plus haute, ont répondu "Peut-être" à la question de savoir s'ils ajoutent des produits à leur panier tout en naviguant sur Amazon. Cela pourrait indiquer que ces clients sont incertains ou qu'ils ajoutent parfois des produits à leur panier pendant la navigation, en fonction de divers facteurs tels que le produit, le prix, les avis, etc.
- **Oui** : Un nombre significatif de clients, représentés par la barre orange, ont répondu "Oui". Cela pourrait indiquer que ces clients ont tendance à ajouter des produits à leur panier pendant la navigation, peut-être pour les sauvegarder pour un achat futur ou pour les comparer à d'autres produits.
- **Non** : Un nombre plus petit de clients, représentés par la barre verte, ont répondu "Non". Ces clients pourraient être ceux qui préfèrent naviguer et rechercher des produits sans les ajouter à leur panier.

La majorité des clients sont incertains quant à l'ajout de produits à leur panier tout en naviguant sur Amazon, tandis qu'un nombre significatif de clients le font et un nombre plus petit de clients ne le font pas.

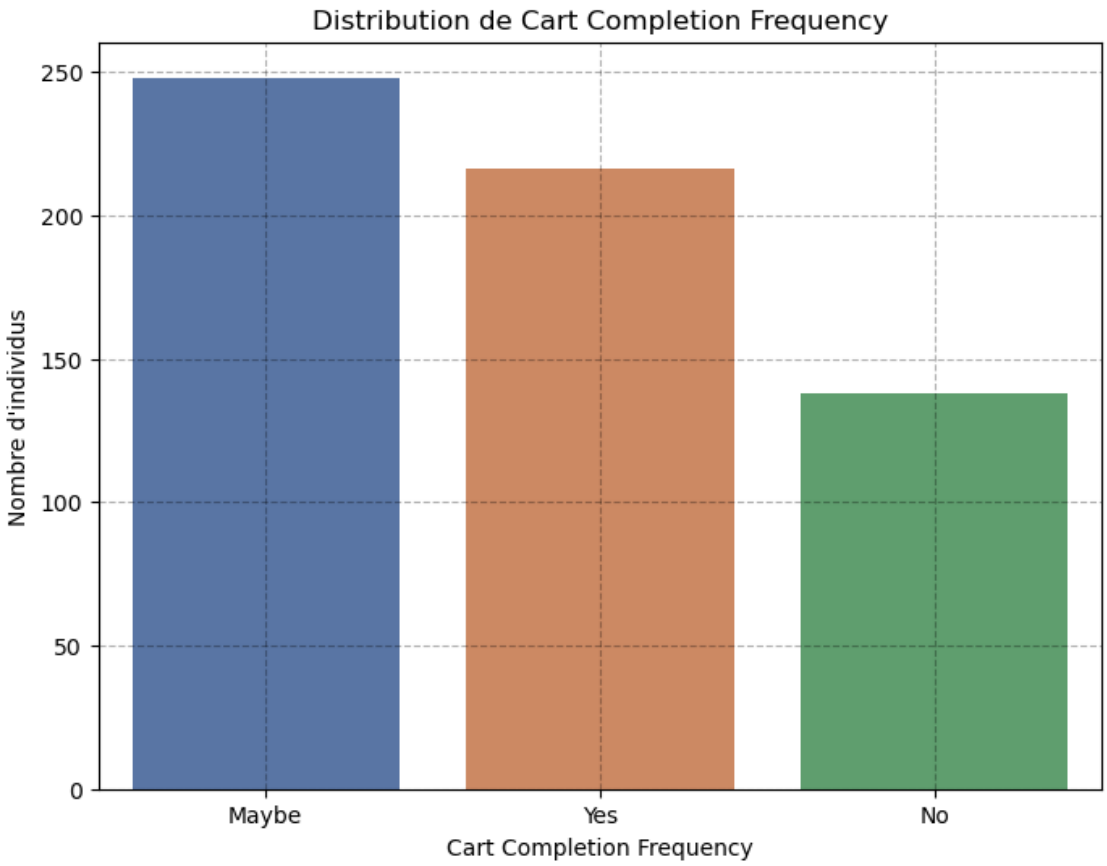
L'ajout de produits au panier pendant la navigation pourrait être lié à la méthode de recherche de produits (septième image), à l'exploration des résultats de recherche (huitième image) et à

l'importance des avis des clients (neuvième image). Par exemple, les clients qui utilisent des mots-clés pour rechercher des produits, qui explorent plusieurs pages de résultats et qui considèrent les avis des clients comme importants pourraient être plus susceptibles d'ajouter des produits à leur panier pendant la navigation. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

À quelle fréquence les clients finalisent-ils un achat après avoir ajouté des produits à leur panier ?

[349]:
```
data_C_C_F = data['Cart_Completion_Frequency'].value_counts()
```

[409]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_A_T_C_B.index,y=data_A_T_C_B,palette='deep')
plt.title('Distribution de Cart Completion Frequency')
plt.xlabel('Cart Completion Frequency')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\11␣
 ↪Distribution de Cart Completion Frequency.png'  # Remplacez par le chemin et␣
 ↪le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres:

- **Peut-être** : La majorité des clients, représentés par la barre bleue la plus haute, ont répondu "Peut-être" à la question de savoir s'ils finalisent un achat après avoir ajouté des produits à leur panier. Cela pourrait indiquer que ces clients sont incertains ou qu'ils finalisent parfois un achat après avoir ajouté des produits à leur panier, en fonction de divers facteurs tels que le produit, le prix, les avis, etc.
- **Oui** : Un nombre significatif de clients, représentés par la barre orange, ont répondu "Oui". Cela pourrait indiquer que ces clients ont tendance à finaliser un achat après avoir ajouté des produits à leur panier. Ces clients pourraient être ceux qui planifient leurs achats ou qui sont sûrs de ce qu'ils veulent.
- **Non** : Un nombre plus petit de clients, représentés par la barre verte, ont répondu "Non". Ces clients pourraient être ceux qui ajoutent des produits à leur panier pour les sauvegarder pour plus tard ou pour les comparer à d'autres produits, mais ne finalisent pas toujours l'achat.

La majorité des clients sont incertains quant à la finalisation d'un achat après avoir ajouté des produits à leur panier, tandis qu'un nombre significatif de clients le font et un nombre plus petit de clients ne le font pas.

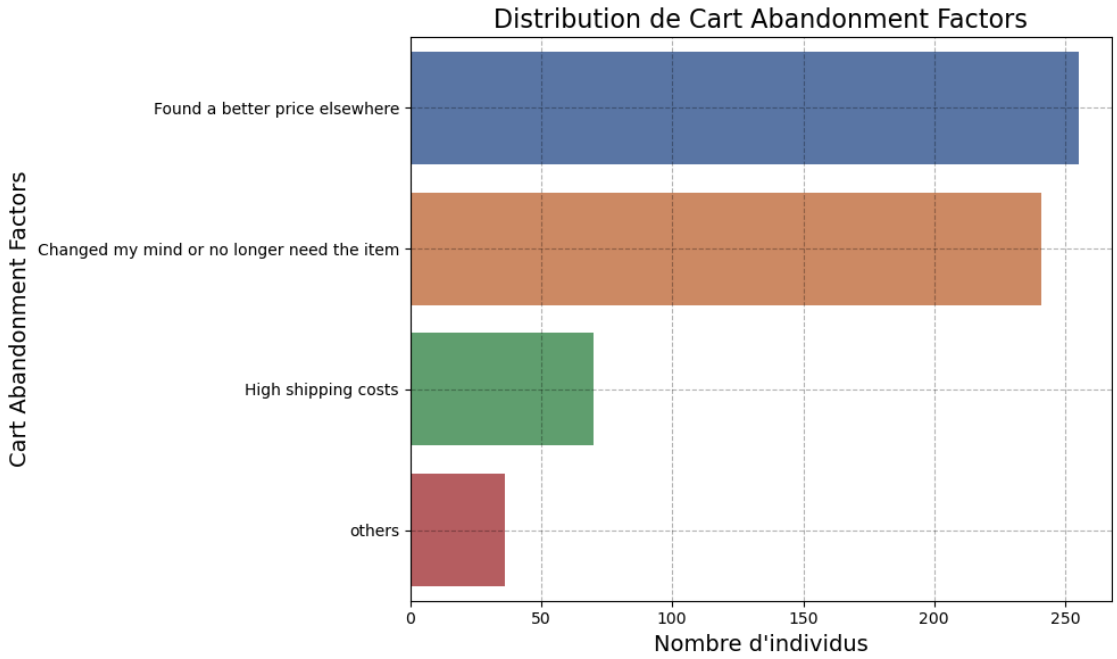
La finalisation d'un achat après avoir ajouté des produits au panier pourrait être liée à l'ajout de produits au panier pendant la navigation (dixième image), à la méthode de recherche de produits

(septième image), à l'exploration des résultats de recherche (huitième image) et à l'importance des avis des clients (neuvième image). Par exemple, les clients qui ajoutent des produits à leur panier pendant la navigation, qui utilisent des mots-clés pour rechercher des produits, qui explorent plusieurs pages de résultats et qui considèrent les avis des clients comme importants pourraient être plus susceptibles de finaliser un achat après avoir ajouté des produits à leur panier. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Quels facteurs influencent la décision des clients d'abandonner un achat dans leur panier ?

[352]:
```
data_C_A_F = data['Cart_Abandonment_Factors'].value_counts()
```

[371]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'data' is your DataFrame
# Aggregate similar categories, you need to customize this based on your data
data['Grouped_Categories'] = data['Cart_Abandonment_Factors'].replace({
    'Category1': 'Group1',
    'Category2': 'Group1',
    'Category3': 'Group2',
    # Add more mappings as needed
})

# Count the occurrences of each grouped factor
factor_counts = data['Grouped_Categories'].value_counts()

# Plot the bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=factor_counts.values, y=factor_counts.index, palette='deep',␣
 ↪orient='h')

# Title and labels
plt.title('Distribution de Cart Abandonment Factors', fontsize=16)
plt.xlabel('Nombre d\'individus', fontsize=14)
plt.ylabel('Cart Abandonment Factors', fontsize=14)

# Add grid
plt.grid(True, linestyle='--', color='black', alpha=0.3)

# Adjust layout to prevent clipping of labels
plt.tight_layout()

# Show the plot
plt.show()
```



D'après le graphique à barres :

- **Trouvé un meilleur prix ailleurs** : La majorité des clients, représentés par la barre bleue la plus haute, abandonnent un achat parce qu'ils ont trouvé un meilleur prix ailleurs. Cela pourrait indiquer que ces clients sont sensibles au prix et cherchent toujours la meilleure affaire.
- **N'a plus besoin de l'article** : Un nombre significatif de clients, représentés par la barre orange, abandonnent un achat parce qu'ils n'ont plus besoin de l'article. Cela pourrait indiquer que ces clients ajoutent des articles à leur panier pour les sauvegarder pour plus tard, mais peuvent changer d'avis ou trouver une alternative entre-temps.
- **Frais de livraison élevés** : Un nombre plus petit de clients, représentés par la barre verte, abandonnent un achat en raison de frais de livraison élevés. Ces clients pourraient être ceux qui attendent des frais de livraison bas ou gratuits et sont dissuadés par des frais supplémentaires.
- **Autres** : Le nombre de clients qui abandonnent un achat pour d'autres raisons, représentés par la barre rouge, est le plus bas. Ces raisons pourraient inclure une variété de facteurs tels que le délai de livraison, la qualité du produit, les avis des clients, etc.

Les facteurs qui influencent le plus la décision des clients d'abandonner un achat dans leur panier sont le fait de trouver un meilleur prix ailleurs, de ne plus avoir besoin de l'article et des frais de livraison élevés.

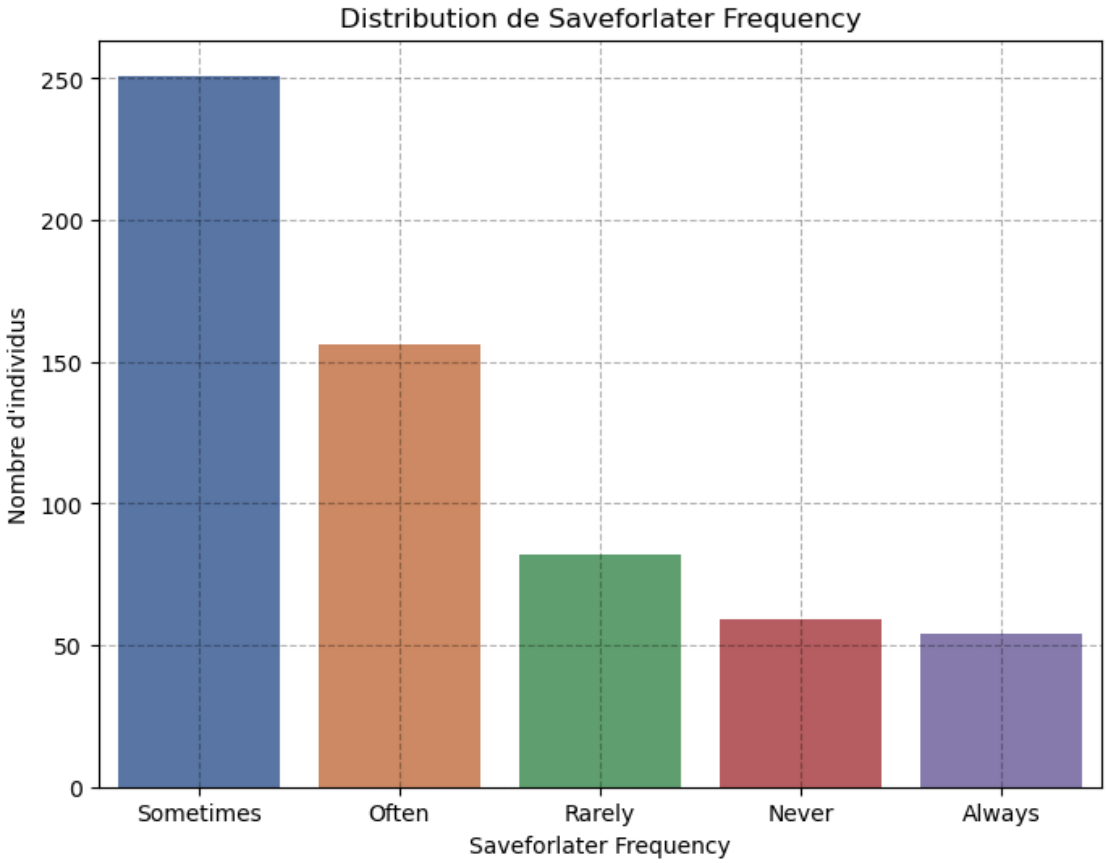
L'abandon du panier pourrait être lié à l'ajout de produits au panier pendant la navigation (dixième image), à la finalisation d'un achat après avoir ajouté des produits au panier (onzième image), à la méthode de recherche de produits (septième image), à l'exploration des résultats de recherche (huitième image) et à l'importance des avis des clients (neuvième image). Par exemple, les clients qui ajoutent des produits à leur panier pendant la navigation, qui utilisent des mots-clés pour

rechercher des produits, qui explorent plusieurs pages de résultats, qui considèrent les avis des clients comme importants et qui sont incertains quant à la finalisation d'un achat après avoir ajouté des produits à leur panier pourraient être plus susceptibles d'abandonner un achat pour diverses raisons. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Combien de clients utilisent la fonction "Enregistrer pour plus tard" d'Amazon, et à quelle fréquence ?

[354]:
```
data_S_F = data['Saveforlater_Frequency'].value_counts()
```

[411]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_S_F.index,y=data_S_F,palette='deep')
plt.title('Distribution de Saveforlater Frequency')
plt.xlabel('Saveforlater Frequency')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\13␣
 ↪Distribution de Saveforlater Frequency.png'  # Remplacez par le chemin et le␣
 ↪nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **Parfois** : La majorité des clients, représentés par la barre bleue la plus haute, utilisent la fonction "Enregistrer pour plus tard" parfois. Cela pourrait indiquer que ces clients utilisent cette fonction lorsqu'ils trouvent des produits intéressants mais ne sont pas encore prêts à les acheter.
- **Souvent** : Un nombre significatif de clients, représentés par la barre orange, utilisent souvent la fonction "Enregistrer pour plus tard". Ces clients pourraient être ceux qui planifient leurs achats ou qui aiment comparer différents produits avant de prendre une décision.
- **Rarement** : Un nombre plus petit de clients, représentés par la barre verte, utilisent rarement la fonction "Enregistrer pour plus tard". Ces clients pourraient être ceux qui préfèrent faire des achats immédiats ou qui n'utilisent pas souvent Amazon pour faire du shopping.
- **Jamais** : Un nombre encore plus petit de clients, représentés par la barre violette, n'utilisent jamais la fonction "Enregistrer pour plus tard". Ces clients pourraient être ceux qui ne sont pas familiers avec cette fonction ou qui préfèrent ne pas l'utiliser pour diverses raisons.
- **Toujours** : Le nombre de clients qui utilisent toujours la fonction "Enregistrer pour plus tard", représentés par la barre rouge, est le plus bas. Ces clients pourraient être ceux qui dépendent fortement de cette fonction pour organiser leurs achats potentiels.

La majorité des clients utilisent la fonction "Enregistrer pour plus tard" parfois, tandis qu'un nombre significatif de clients l'utilisent souvent. Un nombre plus petit de clients l'utilisent rarement,

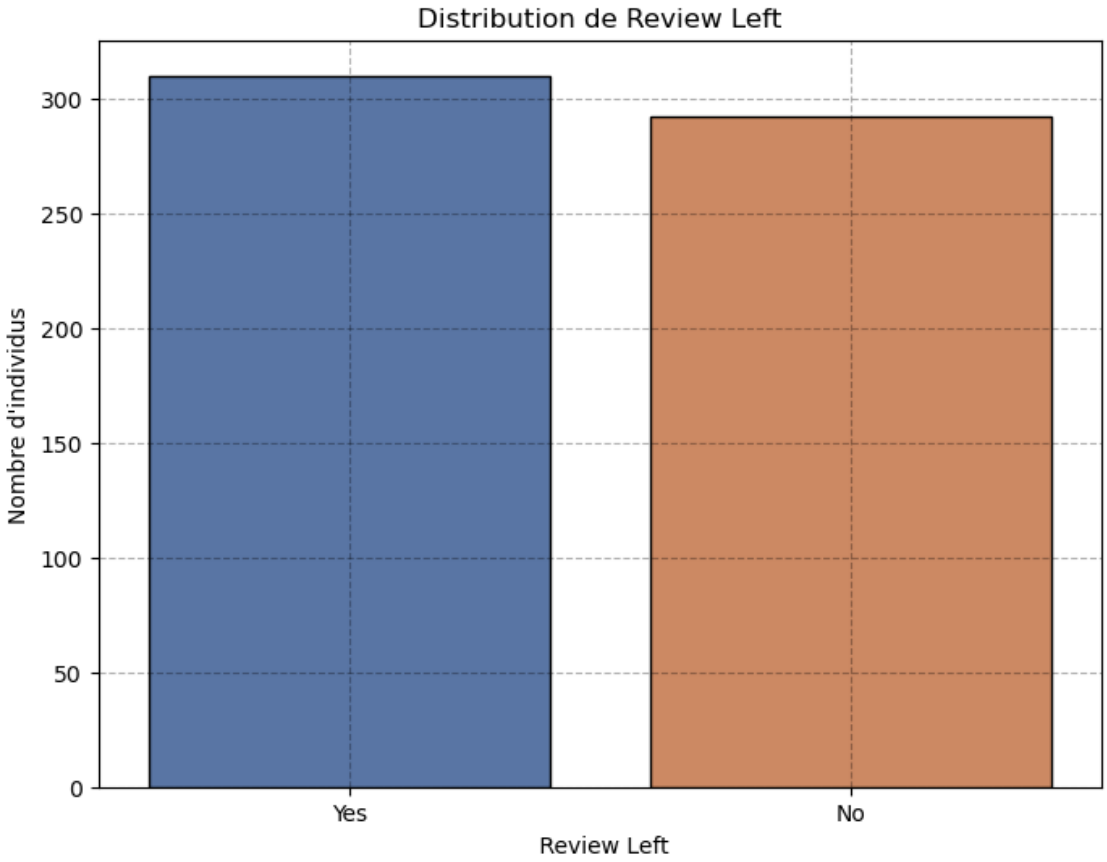
jamais ou toujours.

L'utilisation de la fonction "Enregistrer pour plus tard" pourrait être liée à l'ajout de produits au panier pendant la navigation (dixième image), à la finalisation d'un achat après avoir ajouté des produits au panier (onzième image), à l'abandon du panier (douzième image), à la méthode de recherche de produits (septième image), à l'exploration des résultats de recherche (huitième image) et à l'importance des avis des clients (neuvième image). Par exemple, les clients qui ajoutent des produits à leur panier pendant la navigation, qui sont incertains quant à la finalisation d'un achat après avoir ajouté des produits à leur panier, qui abandonnent parfois un achat pour diverses raisons, qui utilisent des mots-clés pour rechercher des produits, qui explorent plusieurs pages de résultats et qui considèrent les avis des clients comme importants pourraient être plus susceptibles d'utiliser la fonction "Enregistrer pour plus tard". Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Combien de clients ont déjà laissé un avis sur un produit sur Amazon ?

[372]:
```
data_R_L = data['Review_Left'].value_counts()
```

[412]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_R_L.index,y=data_R_L,palette='deep',edgecolor = 'black')
plt.title('Distribution de Review Left')
plt.xlabel(' Review Left')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',alpha=0.3,color='black')
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\14␣
 ↪Distribution de Review Left.png'  # Remplacez par le chemin et le nom de␣
 ↪fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres:

- **Oui** : La majorité des clients, représentés par la barre bleue, ont déjà laissé un avis sur un produit sur Amazon. Cela pourrait indiquer que ces clients sont engagés et disposés à partager leurs expériences pour aider d'autres clients.
- **Non** : Un nombre plus petit de clients, représentés par la barre orange, n'ont pas laissé d'avis sur un produit sur Amazon. Ces clients pourraient être ceux qui préfèrent ne pas partager leurs expériences ou qui n'ont pas eu l'occasion de le faire.

La majorité des clients ont déjà laissé un avis sur un produit sur Amazon, tandis qu'un nombre plus petit de clients ne l'ont pas fait.

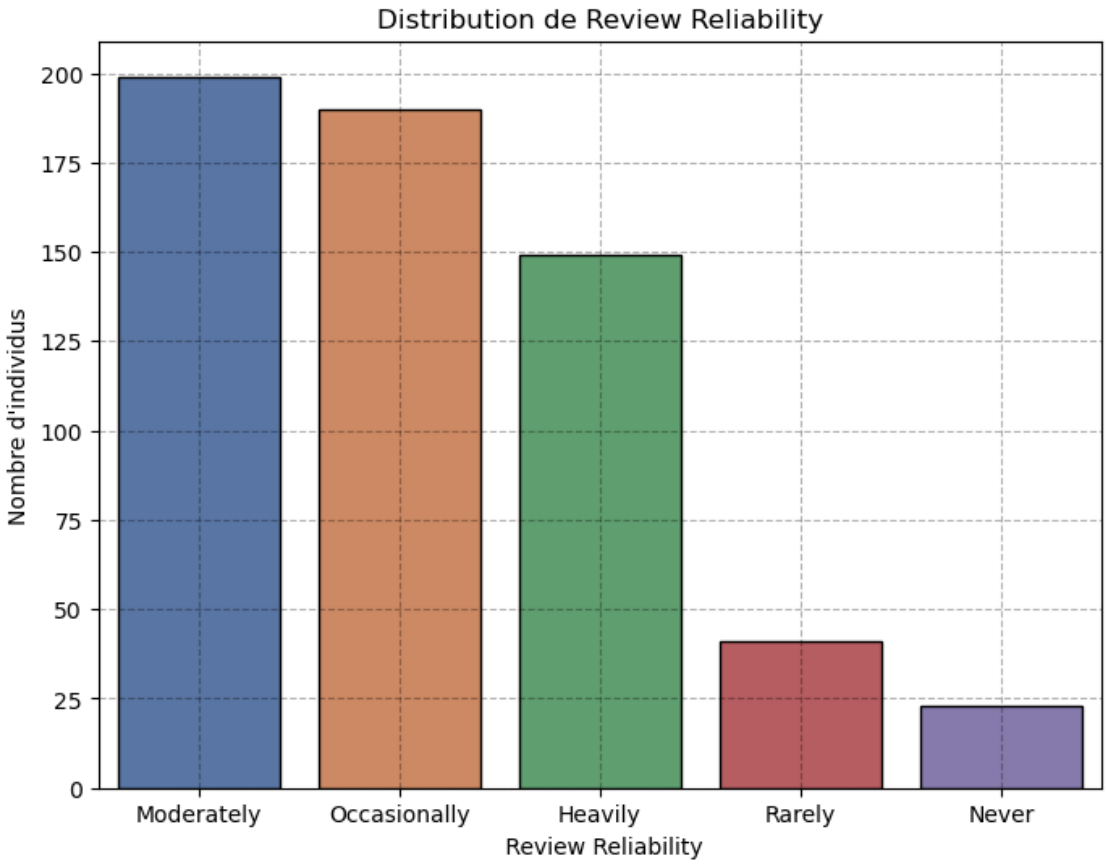
Le fait de laisser un avis pourrait être lié à l'ajout de produits au panier pendant la navigation (dixième image), à la finalisation d'un achat après avoir ajouté des produits au panier (onzième image), à l'abandon du panier (douzième image), à l'utilisation de la fonction "Enregistrer pour plus tard" (treizième image), à la méthode de recherche de produits (septième image), à l'exploration des résultats de recherche (huitième image) et à l'importance des avis des clients (neuvième image). Par exemple, les clients qui ajoutent des produits à leur panier pendant la navigation, qui finalisent un achat après avoir ajouté des produits à leur panier, qui utilisent la fonction "Enregistrer pour plus tard", qui utilisent des mots-clés pour rechercher des produits, qui explorent plusieurs pages de résultats et qui considèrent les avis des clients comme importants pourraient être plus susceptibles

de laisser un avis sur un produit. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Dans quelle mesure les clients font-ils confiance aux avis sur les produits lorsqu'ils effectuent un achat ?

[375]:
```
data_R_R = data['Review_Reliability'].value_counts()
```

[413]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_R_R.index,y=data_R_R,palette='deep',edgecolor = 'black')
plt.title('Distribution de Review Reliability')
plt.xlabel(' Review Reliability')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\15␣
 ↪Distribution de Review Reliability.png'  # Remplacez par le chemin et le nom␣
 ↪de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **Modérément** : La majorité des clients, représentés par la barre la plus haute, font modérément confiance aux avis sur les produits. Cela pourrait indiquer que ces clients prennent en compte les avis sur les produits dans leur processus de prise de décision, mais ne s'y fient pas entièrement.
- **Occasionnellement** : Un nombre significatif de clients, représentés par la deuxième barre la plus haute, font occasionnellement confiance aux avis sur les produits. Ces clients pourraient être ceux qui consultent les avis sur les produits de temps en temps, peut-être pour des achats plus importants ou plus coûteux.
- **Fortement** : Un nombre plus petit de clients, représentés par la troisième barre la plus haute, font fortement confiance aux avis sur les produits. Ces clients pourraient être ceux qui dépendent fortement des avis sur les produits pour prendre leurs décisions d'achat.
- **Rarement** : Un nombre encore plus petit de clients, représentés par la quatrième barre la plus haute, font rarement confiance aux avis sur les produits. Ces clients pourraient être ceux qui préfèrent faire leurs propres recherches ou qui ont des préférences d'achat spécifiques qui ne sont pas influencées par les avis sur les produits.
- **Jamais** : Le nombre de clients qui ne font jamais confiance aux avis sur les produits, représentés par la barre la plus basse, est le plus bas. Ces clients pourraient être ceux qui ne sont pas influencés par les opinions des autres ou qui préfèrent se fier à leur propre jugement.

La majorité des clients font modérément confiance aux avis sur les produits lorsqu'ils effectuent un achat, tandis qu'un nombre significatif de clients le font occasionnellement. Un nombre plus petit de clients font fortement confiance aux avis sur les produits, et un nombre encore plus petit de clients le font rarement ou jamais.

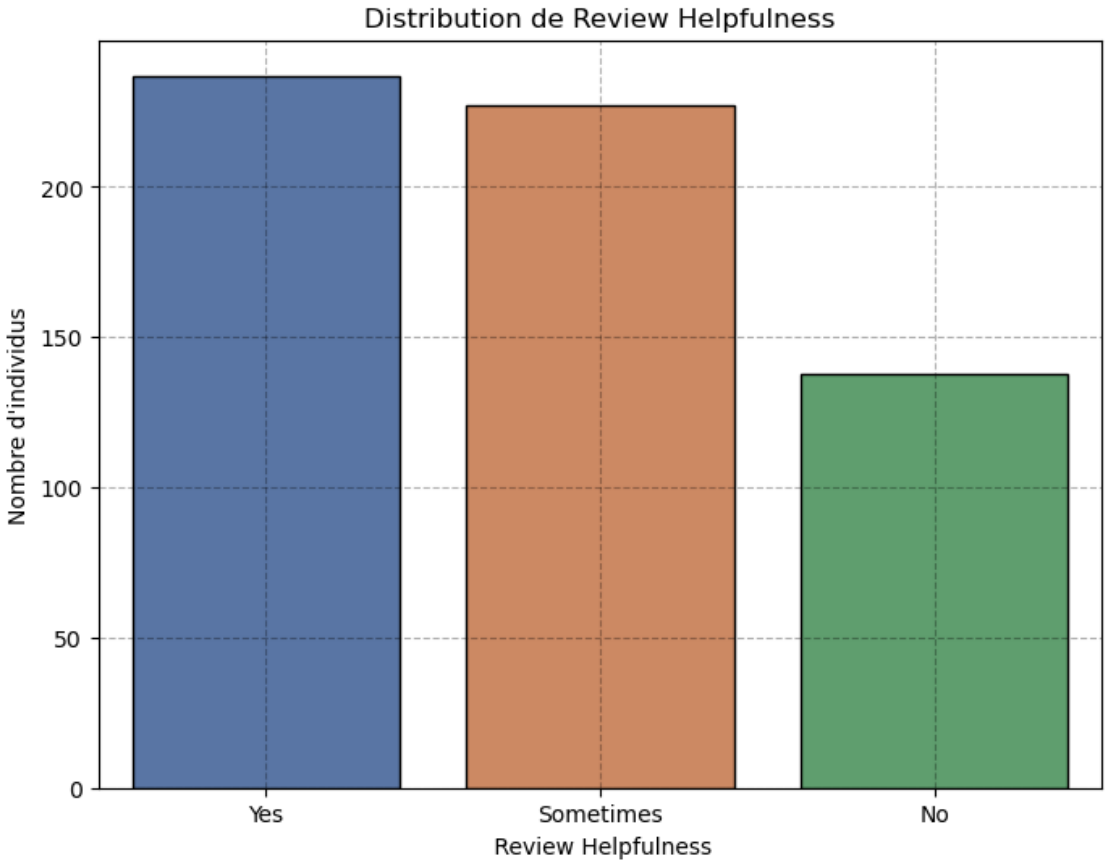
La confiance dans les avis sur les produits pourrait être liée à l'ajout de produits au panier pendant la navigation (dixième image), à la finalisation d'un achat après avoir ajouté des produits au panier (onzième image), à l'abandon du panier (douzième image), à l'utilisation de la fonction "Enregistrer pour plus tard" (treizième image), à la méthode de recherche de produits (septième image), et à l'exploration des résultats de recherche (huitième image). Par exemple, les clients qui ajoutent des produits à leur panier pendant la navigation, qui finalisent un achat après avoir ajouté des produits à leur panier, qui utilisent la fonction "Enregistrer pour plus tard", qui utilisent des mots-clés pour rechercher des produits, et qui explorent plusieurs pages de résultats pourraient être plus susceptibles de faire confiance aux avis sur les produits. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Les clients trouvent-ils des informations utiles dans les avis d'autres clients ?

[377]:
```
data_R_H = data['Review_Helpfulness'].value_counts()
```

[414]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_R_H.index,y=data_R_H,palette='deep',edgecolor = 'black')
plt.title('Distribution de Review Helpfulness')
plt.xlabel(' Review Helpfulness')
plt.ylabel('Nombre d\'individus')
```

```
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\16␣
 ↪Distribution de Review Helpfulness.png'  # Remplacez par le chemin et le nom␣
 ↪de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **Oui** : La majorité des clients, représentés par la barre bleue la plus haute, trouvent des informations utiles dans les avis d'autres clients. Cela pourrait indiquer que ces clients apprécient les avis d'autres clients pour obtenir des informations sur la qualité du produit, l'expérience d'utilisation, etc.
- **Parfois** : Un nombre significatif de clients, représentés par la barre orange, trouvent parfois des informations utiles dans les avis d'autres clients. Ces clients pourraient être ceux qui consultent les avis sur les produits de temps en temps, peut-être pour des achats plus importants ou plus coûteux.

- **Non** : Un nombre plus petit de clients, représentés par la barre verte, ne trouvent pas d'informations utiles dans les avis d'autres clients. Ces clients pourraient être ceux qui préfèrent faire leurs propres recherches ou qui ont des préférences d'achat spécifiques qui ne sont pas influencées par les avis sur les produits.

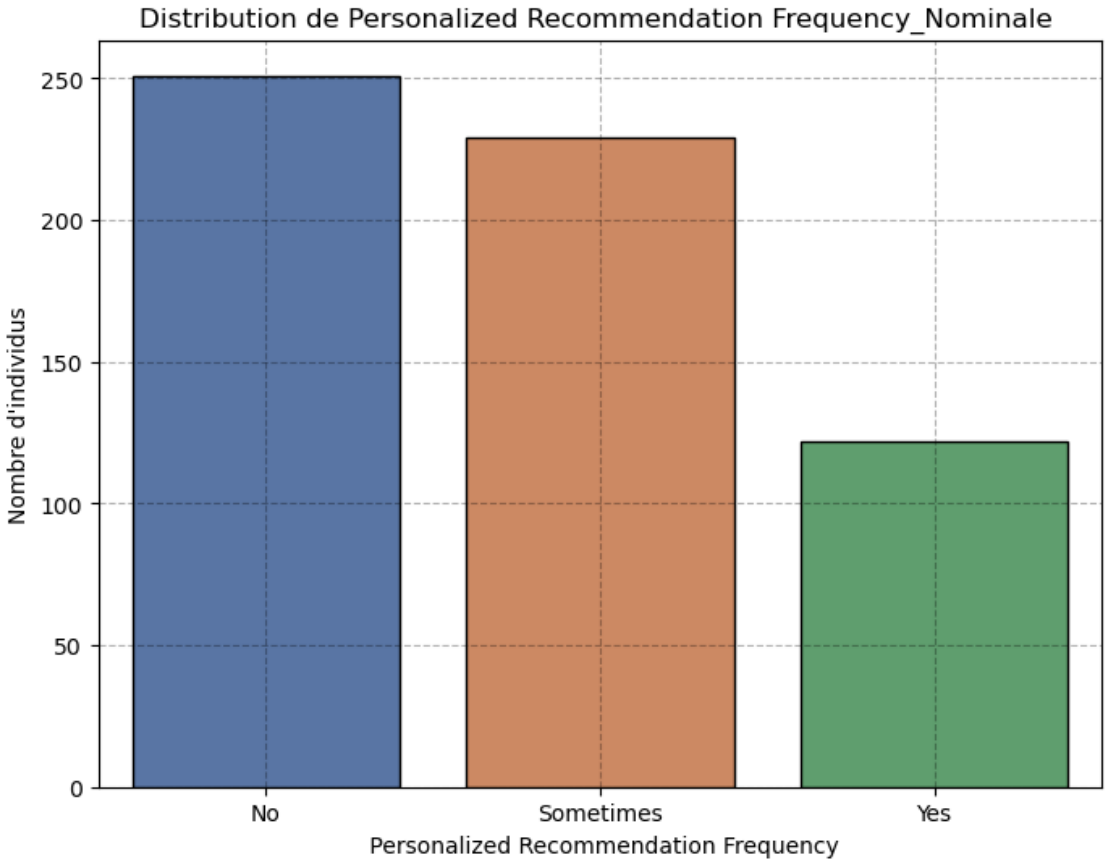
La majorité des clients trouvent des informations utiles dans les avis d'autres clients lorsqu'ils effectuent un achat, tandis qu'un nombre significatif de clients le trouvent parfois utile. Un nombre plus petit de clients ne trouvent pas d'informations utiles dans les avis d'autres clients.

L'utilité des avis sur les produits pourrait être liée à l'ajout de produits au panier pendant la navigation (dixième image), à la finalisation d'un achat après avoir ajouté des produits au panier (onzième image), à l'abandon du panier (douzième image), à l'utilisation de la fonction "Enregistrer pour plus tard" (treizième image), à la méthode de recherche de produits (septième image), et à l'exploration des résultats de recherche (huitième image). Par exemple, les clients qui ajoutent des produits à leur panier pendant la navigation, qui finalisent un achat après avoir ajouté des produits à leur panier, qui utilisent la fonction "Enregistrer pour plus tard", qui utilisent des mots-clés pour rechercher des produits, et qui explorent plusieurs pages de résultats pourraient être plus susceptibles de trouver des informations utiles dans les avis sur les produits. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

À quelle fréquence les clients reçoivent-ils des recommandations de produits personnalisées d'Amazon ?

[379]:
```
daad_P_R_F = data['Personalized_Recommendation_Frequency'].value_counts()
```

[629]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=daad_P_R_F.index,y=daad_P_R_F,palette='deep',edgecolor = 'black')
plt.title('Distribution de Personalized Recommendation Frequency_Nominale ')
plt.xlabel(' Personalized Recommendation Frequency')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\17␣
 ↪Distribution de Personalized Recommendation Frequency.png'  # Remplacez par␣
 ↪le chemin et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après les données présentées :

- 251 clients ne reçoivent pas de recommandations de produits personnalisées.
- 229 clients en reçoivent parfois.
- 122 clients en reçoivent toujours.

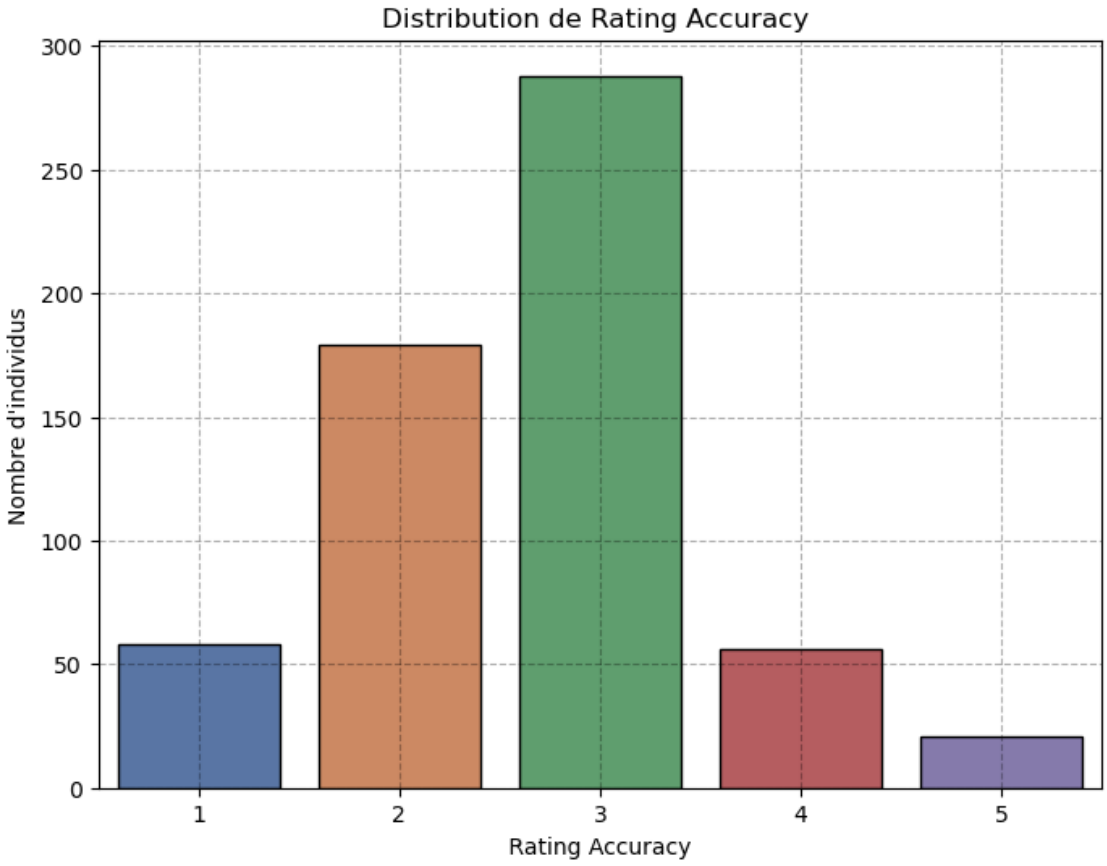
Il semble donc que la majorité des clients ne reçoivent pas de recommandations de produits personnalisées d'Amazon, tandis qu'une part significative des clients en reçoit parfois. Un nombre plus restreint de clients en reçoit toujours.

Comment les clients évaluent-ils la pertinence et la précision des recommandations qu'ils reçoivent ?

[384]:
```
data_R_A = data['Rating_Accuracy '].value_counts()
```

[416]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_R_A.index,y=data_R_A,palette='deep',edgecolor = 'black')
plt.title('Distribution de Rating Accuracy ')
plt.xlabel(' Rating Accuracy ')
plt.ylabel('Nombre d\'individus')
```

```
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\19␣
 ↪Distribution de Rating Accuracy.png'  # Remplacez par le chemin et le nom de␣
 ↪fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après le graphique à barres :

- **3 sur 5** : La majorité des clients, représentés par la barre la plus haute, évaluent la pertinence et la précision des recommandations qu'ils reçoivent comme étant de 3 sur 5. Cela pourrait indiquer que ces clients trouvent les recommandations modérément pertinentes et précises.
- **2 sur 5** : Un nombre significatif de clients, représentés par la deuxième barre la plus haute, évaluent la pertinence et la précision des recommandations qu'ils reçoivent comme étant de 2 sur 5. Ces clients pourraient être ceux qui trouvent les recommandations relativement peu pertinentes ou précises.
- **1 sur 5** : Un nombre plus petit de clients, représentés par la troisième barre la plus haute,

évaluent la pertinence et la précision des recommandations qu'ils reçoivent comme étant de 1 sur 5. Ces clients pourraient être ceux qui ne trouvent pas les recommandations pertinentes ou précises.

- **4 sur 5** : Un nombre encore plus petit de clients, représentés par la quatrième barre la plus haute, évaluent la pertinence et la précision des recommandations qu'ils reçoivent comme étant de 4 sur 5. Ces clients pourraient être ceux qui trouvent les recommandations très pertinentes et précises.
- **5 sur 5** : Le nombre de clients qui évaluent la pertinence et la précision des recommandations qu'ils reçoivent comme étant de 5 sur 5, représentés par la barre la plus basse, est le plus bas. Ces clients pourraient être ceux qui trouvent les recommandations extrêmement pertinentes et précises.

La majorité des clients évaluent la pertinence et la précision des recommandations qu'ils reçoivent comme étant modérées, tandis qu'un nombre significatif de clients les évaluent comme relativement faibles. Un nombre plus petit de clients les évaluent comme étant faibles, très élevées ou extrêmement élevées.

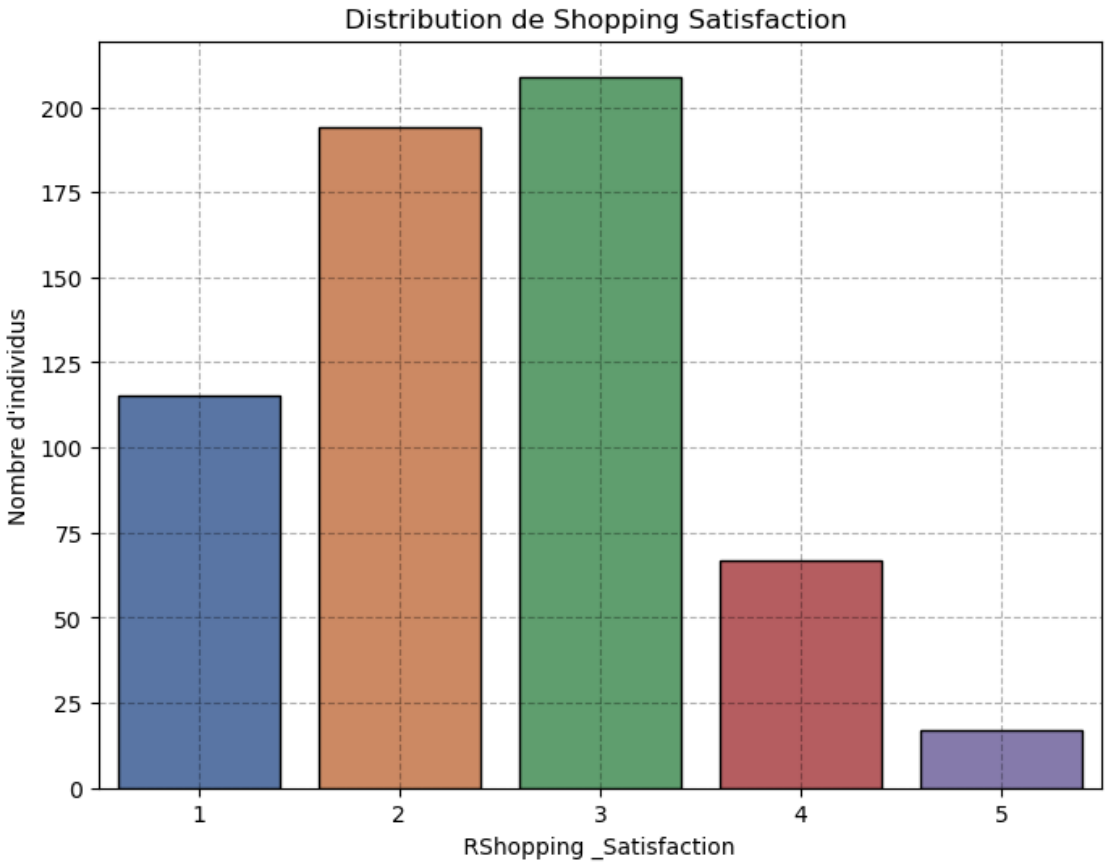
L'évaluation de la pertinence et de la précision des recommandations pourrait être liée à la réception de recommandations de produits personnalisées (dix-septième image), à l'ajout de produits au panier pendant la navigation (dixième image), à la finalisation d'un achat après avoir ajouté des produits au panier (onzième image), à l'abandon du panier (douzième image), à l'utilisation de la fonction "Enregistrer pour plus tard" (treizième image), à la méthode de recherche de produits (septième image), à l'exploration des résultats de recherche (huitième image), et à l'importance des avis des clients (neuvième image). Par exemple, les clients qui reçoivent des recommandations de produits personnalisées, qui ajoutent des produits à leur panier pendant la navigation, qui finalisent un achat après avoir ajouté des produits à leur panier, qui utilisent la fonction "Enregistrer pour plus tard", qui utilisent des mots-clés pour rechercher des produits, qui explorent plusieurs pages de résultats et qui considèrent les avis des clients comme importants pourraient être plus susceptibles d'évaluer la pertinence et la précision des recommandations comme étant élevées. Cependant, pour confirmer ces relations, une analyse plus approfondie serait nécessaire.

Dans quelle mesure les clients sont-ils satisfaits de leur expérience d'achat globale sur Amazon ?

[386]:
```
data_S_S = data['Shopping_Satisfaction'].value_counts()
```

[417]:
```
# vizualisation de la distribution de "Personalized Recommendation Frequency" :
plt.figure(figsize=(8,6))
sns.barplot(x=data_S_S.index,y=data_S_S,palette='deep',edgecolor = 'black')
plt.title('Distribution de Shopping Satisfaction ')
plt.xlabel(' RShopping _Satisfaction ')
plt.ylabel('Nombre d\'individus')
plt.grid(True,linestyle='--',color='black',alpha=0.3)
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\20␣
 ↪Distribution de Shopping _Satisfaction.png'  # Remplacez par le chemin et le␣
 ↪nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
```

```
plt.savefig(path_to_save)
plt.show()
```



D'après les données fournies:

- **209 clients** ont donné une note de **3** sur 5.
- **194 clients** ont donné une note de **2** sur 5.
- **115 clients** ont donné une note de **1** sur 5.
- **67 clients** ont donné une note de **4** sur 5.
- **17 clients** ont donné une note de **5** sur 5.

Cela suggère que la majorité des clients ont une expérience d'achat moyennement satisfaisante sur Amazon, avec une note de 3 sur 5. Cependant, il y a aussi une proportion significative de clients qui ont une expérience moins satisfaisante, avec une note de 2 ou 1. Seuls quelques clients ont donné une note de 4 ou 5, indiquant une expérience d'achat très satisfaisante. Il est important de noter que ces résultats peuvent varier en fonction de divers facteurs tels que la sélection de produits, le service client, les délais de livraison, etc.

Quels aspects des services d'Amazon sont les plus appréciés par les clients ?

[388]:
```
data_S_App = data['Service_Appreciation'].value_counts()
```

Remarque : Il semble y avoir quelques problèmes de nettoyage des données dans la colonne "Service_Appréciation". Vous avez des doublons, des espaces, et un point (".") qui apparaît également.

[389]:
```
# voir le contenu de ce colonne
data['Service_Appreciation'].unique()
```

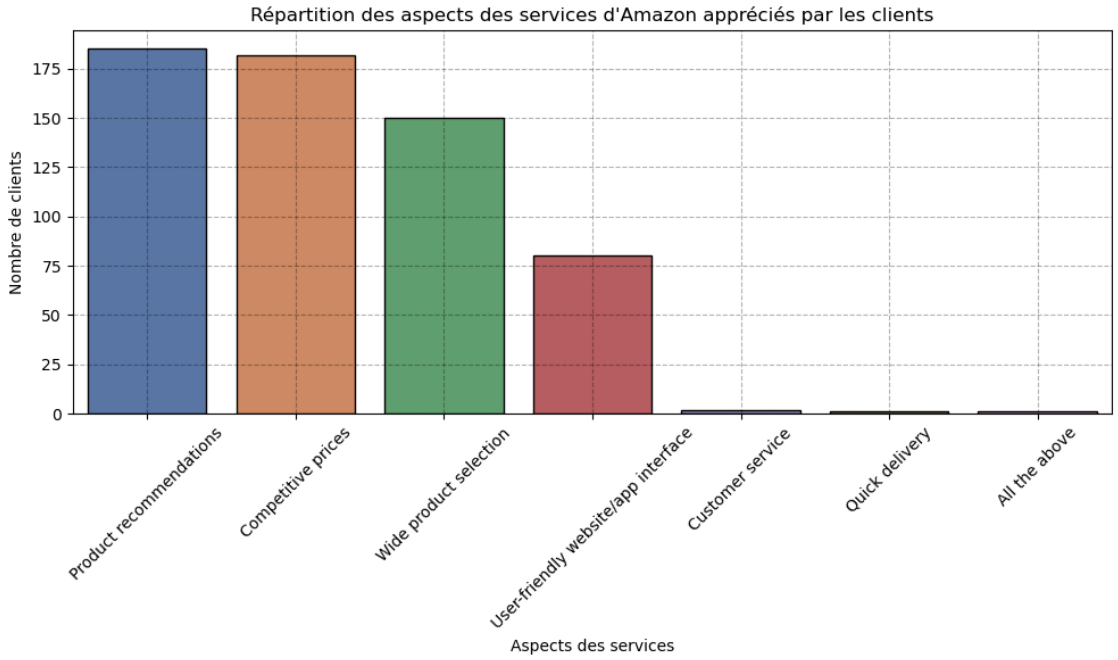
[389]:
```
array(['Competitive prices', 'Wide product selection',
       'User-friendly website/app interface', '.', 'Customer service ',
       'Product recommendations', 'Customer service', 'Quick delivery',
       'All the above'], dtype=object)
```

[390]:
```
# Nettoyer la colonne 'Service_Appreciation'
data['Service_Appreciation'] = data['Service_Appreciation'].str.strip()  #␣
 ↪Supprimer les espaces en début et en fin de chaîne
data['Service_Appreciation'] = data['Service_Appreciation'].replace({'Customer␣
 ↪service ': 'Customer service'})  # Corriger la valeur en doublon
data = data[data['Service_Appreciation'] != '.']

# Afficher les catégories uniques et leurs comptes mis à jour
service_appreciation_counts = data['Service_Appreciation'].value_counts()
print(service_appreciation_counts)
```

```
Service_Appreciation
Product recommendations               185
Competitive prices                    182
Wide product selection                150
User-friendly website/app interface    80
Customer service                        2
Quick delivery                          1
All the above                           1
Name: count, dtype: int64
```

[418]:
```
# Comptez le nombre de réponses par catégorie
service_counts = data['Service_Appreciation'].value_counts()

# Créez un graphique à barres
plt.figure(figsize=(10, 6))
sns.barplot(x=service_appreciation_counts.
 ↪index,y=service_appreciation_counts,palette='deep',edgecolor = 'black')
plt.title("Répartition des aspects des services d'Amazon appréciés par les␣
 ↪clients")
plt.xlabel("Aspects des services")
plt.ylabel("Nombre de clients")
plt.grid(True,linestyle='--',color='black',alpha=0.3)
plt.xticks(rotation = 45)
# Affichez le graphique
```

```
plt.tight_layout()
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de projet\\21␣
 ↪Répartition des aspects des services d\'Amazon appréciés par les clients.
 ↪png'  # Remplacez par le chemin et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



D'après les données présentées:

- 185 clients apprécient les recommandations de produits.
- 182 clients apprécient les prix compétitifs.
- 150 clients apprécient la large sélection de produits.
- 80 clients apprécient l'interface conviviale du site web/de l'application.
- 2 clients apprécient le service client.
- 1 client apprécie la livraison rapide.
- 1 client apprécie tous les aspects mentionnés ci-dessus.

Il semble donc que les recommandations de produits et les prix compétitifs sont les aspects les plus appréciés des services d'Amazon. Cependant, une part significative des clients apprécie également la large sélection de produits et l'interface conviviale du site web/de l'application.*

Y a-t-il des domaines où les clients pensent qu'Amazon pourrait s'améliorer ?

[393]:
```
data['Improvement_Areas'].value_counts()
```

[393]:
```
Improvement_Areas
Customer service responsiveness
217
Product quality and accuracy
159
Reducing packaging waste
133
Shipping speed and reliability
79
Quality of product is very poor according to the big offers
1
I don't have any problem with Amazon
1
User interface of app
1
Irrelevant product suggestions
1
User interface
1
I have no problem with Amazon yet. But others tell me about the refund issues
1
UI
1
Scrolling option would be much better than going to next page
1
Add more familiar brands to the list
1
Nil
1
better app interface and lower shipping charges
1
Nothing
1
No problems with Amazon
1
Name: count, dtype: int64
```

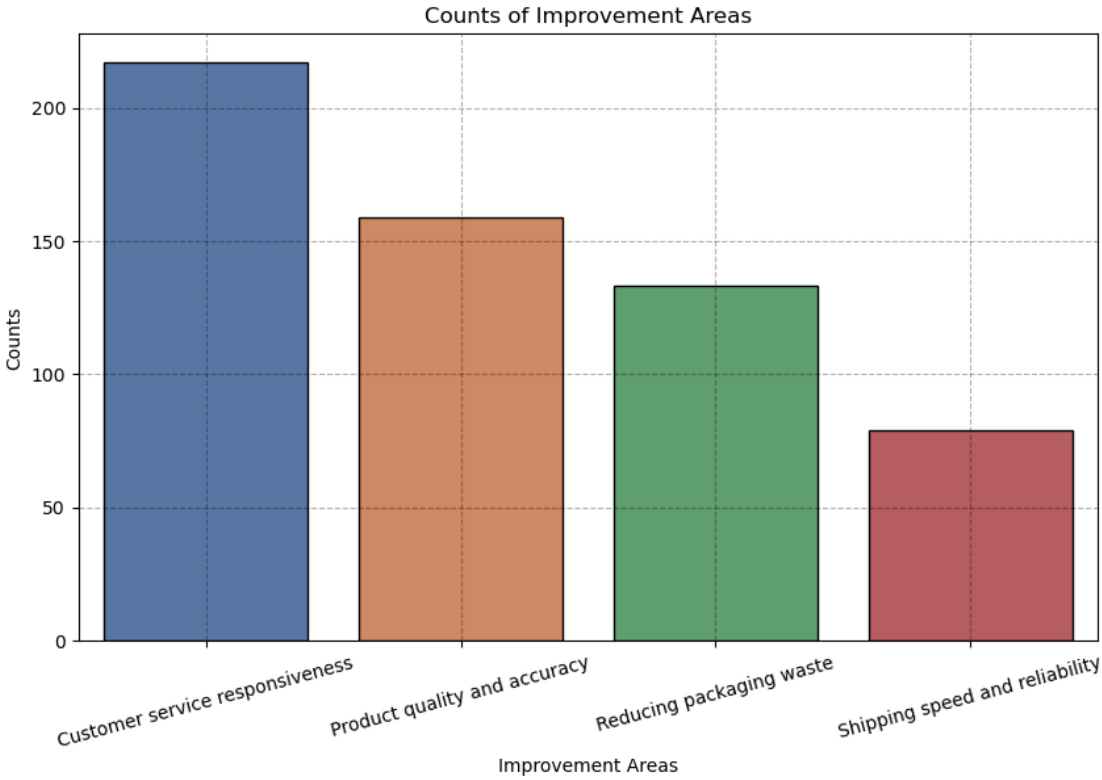
Remarque : Pour la priorisation des problèmes, je recommanderais de classer les domaines d'amélioration en fonction du nombre de réponses et de leur impact potentiel sur la satisfaction client. Dans ce cas, "Customer service responsiveness," "Product quality and accuracy," "Reducing packaging waste," et "Shipping speed and reliability" semblent être les domaines les plus importants.

[394]:
```python
# Sélectionnez les catégories d'intérêt
categories_of_interest = [
    "Customer service responsiveness",
    "Product quality and accuracy",
```

```
    "Reducing packaging waste",
    "Shipping speed and reliability"
]

# Filtrez le DataFrame pour inclure uniquement les lignes correspondant à ces␣
 ↪catégories
filtered_data = data[data['Improvement_Areas'].isin(categories_of_interest)]

# Obtenez le compte des catégories spécifiques
counts = filtered_data['Improvement_Areas'].value_counts()

# Affichez le résultat
print(counts)
```

```
Improvement_Areas
Customer service responsiveness    217
Product quality and accuracy       159
Reducing packaging waste           133
Shipping speed and reliability      79
Name: count, dtype: int64
```

[395]:
```
import matplotlib.pyplot as plt

# Create a bar plot for the counts
plt.figure(figsize=(10, 6))
sns.barplot(x=counts.index,y=counts,palette='deep',edgecolor = 'black')
plt.title('Counts of Improvement Areas')
plt.xlabel('Improvement Areas')
plt.ylabel('Counts')
plt.xticks(rotation=15)  # Rotate the x-axis labels for better visibility
plt.grid(True,linestyle='--',color='black',alpha=0.3)

path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de␣
 ↪projet\\Counts of Improvement Areas.png'  # Remplacez par le chemin et le␣
 ↪nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
# Show the plot
plt.show()
```



3 Pratie 2: Modélisation

4 test d'indépendance entre les variables quantitatives :

[466]:
```
data.columns
```

[466]:
```
Index(['Timestamp', 'age', 'Gender', 'Purchase_Frequency',
       'Purchase_Categories', 'Personalized_Recommendation_Frequency',
       'Browsing_Frequency', 'Product_Search_Method',
       'Search_Result_Exploration', 'Customer_Reviews_Importance',
       'Add_to_Cart_Browsing', 'Cart_Completion_Frequency',
       'Cart_Abandonment_Factors', 'Saveforlater_Frequency', 'Review_Left',
       'Review_Reliability', 'Review_Helpfulness',
       'Personalized_Recommendation_Frequency ', 'Recommendation_Helpfulness',
       'Rating_Accuracy ', 'Shopping_Satisfaction', 'Service_Appreciation',
       'Improvement_Areas', 'Grouped_Categories'],
      dtype='object')
```

formulation des hypothéses :

- **Hypothèse nulle H0 :** Il n'y a pas de relation statistiquement significative entre les deux variables catégorielles. En d'autres termes, les variables sont indépendantes.

- **Hypothèse alternative H1 :** Il existe une relation statistiquement significative entre les deux variables catégorielles. Les variables ne sont pas indépendantes.

[467]:
```
categorical_cols = ['Gender', 'Purchase_Frequency',
                    'Purchase_Categories',␣
 ↪'Personalized_Recommendation_Frequency',
                    'Browsing_Frequency', 'Product_Search_Method',
                    'Search_Result_Exploration', 'Customer_Reviews_Importance',
                    'Add_to_Cart_Browsing', 'Cart_Completion_Frequency',
                    'Cart_Abandonment_Factors', 'Saveforlater_Frequency',␣
 ↪'Review_Left',
                    'Review_Reliability', 'Review_Helpfulness',
                    'Personalized_Recommendation_Frequency ',␣
 ↪'Recommendation_Helpfulness',
                    'Rating_Accuracy ', 'Shopping_Satisfaction',␣
 ↪'Service_Appreciation',
                    'Improvement_Areas', 'Grouped_Categories']

# Set a p-value threshold for feature selection
p_value_threshold = 0.05

# Perform the Chi-squared test for independence for all pairs of variables
selected_pairs = []
for pair in combinations(categorical_cols, 2):
    contingency_table = pd.crosstab(data[pair[0]], data[pair[1]])
    chi2, p, _, _ = chi2_contingency(contingency_table)
    print(f"Chi-square test between {pair[0]} and {pair[1]}:")
    print(f"Chi-square statistic: {chi2}, p-value: {p}")

    if p < p_value_threshold:
        selected_pairs.append(pair)

# Print the automatically selected pairs
print("Automatically selected pairs based on p-value threshold:")
print(selected_pairs)
```

```
Chi-square test between Gender and Purchase_Frequency:
Chi-square statistic: 32.906280984135776, p-value: 0.001001152680931906
Chi-square test between Gender and Purchase_Categories:
Chi-square statistic: 151.0048545299717, p-value: 1.0199333236521182e-05
Chi-square test between Gender and Personalized_Recommendation_Frequency:
Chi-square statistic: 15.49313411337437, p-value: 0.016749198402039745
Chi-square test between Gender and Browsing_Frequency:
Chi-square statistic: 23.317014556950525, p-value: 0.0055222700169171935
Chi-square test between Gender and Product_Search_Method:
Chi-square statistic: 25.424723734549353, p-value: 0.002535580766099553
Chi-square test between Gender and Search_Result_Exploration:
Chi-square statistic: 6.918431446573709, p-value: 0.07454358649334304
```

```
Chi-square test between Gender and Customer_Reviews_Importance:
Chi-square statistic: 35.98276129766054, p-value: 0.00032606696790320617
Chi-square test between Gender and Add_to_Cart_Browsing:
Chi-square statistic: 23.53539173167735, p-value: 0.0006356319199179129
Chi-square test between Gender and Cart_Completion_Frequency:
Chi-square statistic: 26.58412166494388, p-value: 0.008865082475971503
Chi-square test between Gender and Cart_Abandonment_Factors:
Chi-square statistic: 10.020186432427248, p-value: 0.3488534776175882
Chi-square test between Gender and Saveforlater_Frequency:
Chi-square statistic: 17.762641042356627, p-value: 0.12308992070722839
Chi-square test between Gender and Review_Left:
Chi-square statistic: 1.4233123257628666, p-value: 0.7000794061392954
Chi-square test between Gender and Review_Reliability:
Chi-square statistic: 25.520246868042598, p-value: 0.012541125760006466
Chi-square test between Gender and Review_Helpfulness:
Chi-square statistic: 27.46216245700306, p-value: 0.00011860504451717908
Chi-square test between Gender and Personalized_Recommendation_Frequency :
Chi-square statistic: 16.15328031670399, p-value: 0.1843159913987732
Chi-square test between Gender and Recommendation_Helpfulness:
Chi-square statistic: 17.901820239612974, p-value: 0.006482292738713883
Chi-square test between Gender and Rating_Accuracy :
Chi-square statistic: 23.413791791665098, p-value: 0.024411468620361202
Chi-square test between Gender and Shopping_Satisfaction:
Chi-square statistic: 19.56483778973121, p-value: 0.07577866696222507
Chi-square test between Gender and Service_Appreciation:
Chi-square statistic: 27.703059823592202, p-value: 0.06672011180268873
Chi-square test between Gender and Improvement_Areas:
Chi-square statistic: 52.26512848998098, p-value: 0.31183200124593063
Chi-square test between Gender and Grouped_Categories:
Chi-square statistic: 10.020186432427248, p-value: 0.3488534776175882
Chi-square test between Purchase_Frequency and Purchase_Categories:
Chi-square statistic: 219.76174238540443, p-value: 5.254364032952134e-09
Chi-square test between Purchase_Frequency and
Personalized_Recommendation_Frequency:
Chi-square statistic: 22.22381478650078, p-value: 0.00451769479688354
Chi-square test between Purchase_Frequency and Browsing_Frequency:
Chi-square statistic: 135.7020783964257, p-value: 4.4053655439877423e-23
Chi-square test between Purchase_Frequency and Product_Search_Method:
Chi-square statistic: 54.281592768253134, p-value: 2.433710749922178e-07
Chi-square test between Purchase_Frequency and Search_Result_Exploration:
Chi-square statistic: 3.027233808267028, p-value: 0.5532782770441318
Chi-square test between Purchase_Frequency and Customer_Reviews_Importance:
Chi-square statistic: 75.64976082819557, p-value: 1.0018165106925085e-09
Chi-square test between Purchase_Frequency and Add_to_Cart_Browsing:
Chi-square statistic: 44.336063541424366, p-value: 4.915325848677888e-07
Chi-square test between Purchase_Frequency and Cart_Completion_Frequency:
Chi-square statistic: 41.53341353233898, p-value: 0.0004629840078433404
Chi-square test between Purchase_Frequency and Cart_Abandonment_Factors:
```

```
Chi-square statistic: 14.444986707560924, p-value: 0.2731989673541065
Chi-square test between Purchase_Frequency and Saveforlater_Frequency:
Chi-square statistic: 99.11449454818197, p-value: 5.0748495674854334e-14
Chi-square test between Purchase_Frequency and Review_Left:
Chi-square statistic: 27.901479130235337, p-value: 1.3059707940527928e-05
Chi-square test between Purchase_Frequency and Review_Reliability:
Chi-square statistic: 55.89936972557221, p-value: 2.5284895948496187e-06
Chi-square test between Purchase_Frequency and Review_Helpfulness:
Chi-square statistic: 63.17784433678224, p-value: 1.1051321814205794e-10
Chi-square test between Purchase_Frequency and
Personalized_Recommendation_Frequency :
Chi-square statistic: 35.14170461179226, p-value: 0.003800584886117949
Chi-square test between Purchase_Frequency and Recommendation_Helpfulness:
Chi-square statistic: 26.448870028990342, p-value: 0.0008797189312461277
Chi-square test between Purchase_Frequency and Rating_Accuracy :
Chi-square statistic: 21.95568483183892, p-value: 0.14462818174433378
Chi-square test between Purchase_Frequency and Shopping_Satisfaction:
Chi-square statistic: 15.69839911902375, p-value: 0.47420123513592705
Chi-square test between Purchase_Frequency and Service_Appreciation:
Chi-square statistic: 101.70714624018292, p-value: 1.5345609861309822e-11
Chi-square test between Purchase_Frequency and Improvement_Areas:
Chi-square statistic: 107.92490885213319, p-value: 0.0004944208502107259
Chi-square test between Purchase_Frequency and Grouped_Categories:
Chi-square statistic: 14.444986707560924, p-value: 0.2731989673541065
Chi-square test between Purchase_Categories and
Personalized_Recommendation_Frequency:
Chi-square statistic: 84.65265126102815, p-value: 0.008003939785979488
Chi-square test between Purchase_Categories and Browsing_Frequency:
Chi-square statistic: 162.13640504878003, p-value: 6.714127121250406e-07
Chi-square test between Purchase_Categories and Product_Search_Method:
Chi-square statistic: 170.70747311210485, p-value: 7.295172664305781e-08
Chi-square test between Purchase_Categories and Search_Result_Exploration:
Chi-square statistic: 48.777961537302104, p-value: 0.008823966150475922
Chi-square test between Purchase_Categories and Customer_Reviews_Importance:
Chi-square statistic: 187.72209846517617, p-value: 9.74210079440032e-06
Chi-square test between Purchase_Categories and Add_to_Cart_Browsing:
Chi-square statistic: 95.12017628281903, p-value: 0.0008610088810114552
Chi-square test between Purchase_Categories and Cart_Completion_Frequency:
Chi-square statistic: 148.72639863033632, p-value: 0.01157225635216535
Chi-square test between Purchase_Categories and Cart_Abandonment_Factors:
Chi-square statistic: 104.48790725871359, p-value: 0.06446659310188657
Chi-square test between Purchase_Categories and Saveforlater_Frequency:
Chi-square statistic: 189.504660185919, p-value: 6.639674564605557e-06
Chi-square test between Purchase_Categories and Review_Left:
Chi-square statistic: 47.51803630391853, p-value: 0.012069703642398226
Chi-square test between Purchase_Categories and Review_Reliability:
Chi-square statistic: 155.78226935905917, p-value: 0.003952984031008085
Chi-square test between Purchase_Categories and Review_Helpfulness:
```

```
Chi-square statistic: 94.87363866068036, p-value: 0.0009106621353688556
Chi-square test between Purchase_Categories and
Personalized_Recommendation_Frequency :
Chi-square statistic: 169.45435067205852, p-value: 0.00037434120457384364
Chi-square test between Purchase_Categories and Recommendation_Helpfulness:
Chi-square statistic: 126.53679408884918, p-value: 2.24645162595296e-07
Chi-square test between Purchase_Categories and Rating_Accuracy :
Chi-square statistic: 172.02189626812077, p-value: 0.00023159525862421313
Chi-square test between Purchase_Categories and Shopping_Satisfaction:
Chi-square statistic: 188.10065590331592, p-value: 8.983766373112822e-06
Chi-square test between Purchase_Categories and Service_Appreciation:
Chi-square statistic: 192.64053662740463, p-value: 0.09348550050483886
Chi-square test between Purchase_Categories and Improvement_Areas:
Chi-square statistic: 554.7847766778548, p-value: 0.0004193124159178911
Chi-square test between Purchase_Categories and Grouped_Categories:
Chi-square statistic: 104.48790725871359, p-value: 0.06446659310188657
Chi-square test between Personalized_Recommendation_Frequency and
Browsing_Frequency:
Chi-square statistic: 46.68856902798402, p-value: 2.15874963202873e-08
Chi-square test between Personalized_Recommendation_Frequency and
Product_Search_Method:
Chi-square statistic: 42.15529030311776, p-value: 1.7134072907562392e-07
Chi-square test between Personalized_Recommendation_Frequency and
Search_Result_Exploration:
Chi-square statistic: 10.89279020917146, p-value: 0.004311820369226899
Chi-square test between Personalized_Recommendation_Frequency and
Customer_Reviews_Importance:
Chi-square statistic: 33.783178311443926, p-value: 4.447059448008885e-05
Chi-square test between Personalized_Recommendation_Frequency and
Add_to_Cart_Browsing:
Chi-square statistic: 84.45761781220787, p-value: 1.977127078570383e-17
Chi-square test between Personalized_Recommendation_Frequency and
Cart_Completion_Frequency:
Chi-square statistic: 20.748754384902607, p-value: 0.00784564844458681
Chi-square test between Personalized_Recommendation_Frequency and
Cart_Abandonment_Factors:
Chi-square statistic: 8.007630252481585, p-value: 0.23754482676268504
Chi-square test between Personalized_Recommendation_Frequency and
Saveforlater_Frequency:
Chi-square statistic: 53.57269711075679, p-value: 8.354190489833708e-09
Chi-square test between Personalized_Recommendation_Frequency and Review_Left:
Chi-square statistic: 28.277006265297608, p-value: 7.239792301221343e-07
Chi-square test between Personalized_Recommendation_Frequency and
Review_Reliability:
Chi-square statistic: 22.55981719765749, p-value: 0.003977836872555811
Chi-square test between Personalized_Recommendation_Frequency and
Review_Helpfulness:
Chi-square statistic: 84.73264365352479, p-value: 1.728593201241987e-17
```

```
Chi-square test between Personalized_Recommendation_Frequency and
Personalized_Recommendation_Frequency :
Chi-square statistic: 22.98793644362784, p-value: 0.0033797690724630285
Chi-square test between Personalized_Recommendation_Frequency and
Recommendation_Helpfulness:
Chi-square statistic: 69.11923025255736, p-value: 3.4826265435778886e-14
Chi-square test between Personalized_Recommendation_Frequency and
Rating_Accuracy :
Chi-square statistic: 14.430731179059038, p-value: 0.07120667462184743
Chi-square test between Personalized_Recommendation_Frequency and
Shopping_Satisfaction:
Chi-square statistic: 39.55794805811182, p-value: 3.87190812096757e-06
Chi-square test between Personalized_Recommendation_Frequency and
Service_Appreciation:
Chi-square statistic: 38.58784202440816, p-value: 0.00012301422821448817
Chi-square test between Personalized_Recommendation_Frequency and
Improvement_Areas:
Chi-square statistic: 47.54573101896981, p-value: 0.0378548043389053
Chi-square test between Personalized_Recommendation_Frequency and
Grouped_Categories:
Chi-square statistic: 8.007630252481585, p-value: 0.23754482676268504
Chi-square test between Browsing_Frequency and Product_Search_Method:
Chi-square statistic: 100.7324492398256, p-value: 1.118633338963538e-17
Chi-square test between Browsing_Frequency and Search_Result_Exploration:
Chi-square statistic: 23.072900101211214, p-value: 3.899445369915622e-05
Chi-square test between Browsing_Frequency and Customer_Reviews_Importance:
Chi-square statistic: 32.80477345595811, p-value: 0.0010382817313274912
Chi-square test between Browsing_Frequency and Add_to_Cart_Browsing:
Chi-square statistic: 6.070810477576929, p-value: 0.4153048139316112
Chi-square test between Browsing_Frequency and Cart_Completion_Frequency:
Chi-square statistic: 82.52150412083864, p-value: 1.3606094896609331e-12
Chi-square test between Browsing_Frequency and Cart_Abandonment_Factors:
Chi-square statistic: 30.709883524442144, p-value: 0.00033193120675471176
Chi-square test between Browsing_Frequency and Saveforlater_Frequency:
Chi-square statistic: 100.78791230774596, p-value: 3.9018414872438214e-16
Chi-square test between Browsing_Frequency and Review_Left:
Chi-square statistic: 24.313408917306063, p-value: 2.1485475948372394e-05
Chi-square test between Browsing_Frequency and Review_Reliability:
Chi-square statistic: 47.575456334480954, p-value: 3.7046596082375947e-06
Chi-square test between Browsing_Frequency and Review_Helpfulness:
Chi-square statistic: 4.854463039654043, p-value: 0.5626104534952177
Chi-square test between Browsing_Frequency and
Personalized_Recommendation_Frequency :
Chi-square statistic: 44.83548348760272, p-value: 1.0995639611077934e-05
Chi-square test between Browsing_Frequency and Recommendation_Helpfulness:
Chi-square statistic: 7.227518990825138, p-value: 0.300318065595736
Chi-square test between Browsing_Frequency and Rating_Accuracy :
Chi-square statistic: 51.568840113669204, p-value: 7.390858467299171e-07
```

```
Chi-square test between Browsing_Frequency and Shopping_Satisfaction:
Chi-square statistic: 51.35350595834824, p-value: 8.067927481367281e-07
Chi-square test between Browsing_Frequency and Service_Appreciation:
Chi-square statistic: 43.12408235606626, p-value: 0.0007685434644500161
Chi-square test between Browsing_Frequency and Improvement_Areas:
Chi-square statistic: 70.55369308391496, p-value: 0.01866863621449857
Chi-square test between Browsing_Frequency and Grouped_Categories:
Chi-square statistic: 30.709883524442144, p-value: 0.00033193120675471176
Chi-square test between Product_Search_Method and Search_Result_Exploration:
Chi-square statistic: 44.22481036033071, p-value: 1.352023906839898e-09
Chi-square test between Product_Search_Method and Customer_Reviews_Importance:
Chi-square statistic: 129.9116626632875, p-value: 6.429157628168678e-22
Chi-square test between Product_Search_Method and Add_to_Cart_Browsing:
Chi-square statistic: 84.95361143093538, p-value: 3.3749910595407345e-16
Chi-square test between Product_Search_Method and Cart_Completion_Frequency:
Chi-square statistic: 106.52644266662995, p-value: 2.9037043783260165e-17
Chi-square test between Product_Search_Method and Cart_Abandonment_Factors:
Chi-square statistic: 56.48362271268523, p-value: 6.344556599939725e-09
Chi-square test between Product_Search_Method and Saveforlater_Frequency:
Chi-square statistic: 65.13347292965817, p-value: 2.575675589680872e-09
Chi-square test between Product_Search_Method and Review_Left:
Chi-square statistic: 2.395411647309661, p-value: 0.4944893146411381
Chi-square test between Product_Search_Method and Review_Reliability:
Chi-square statistic: 119.64399508548077, p-value: 7.277313239319922e-20
Chi-square test between Product_Search_Method and Review_Helpfulness:
Chi-square statistic: 91.80437249170777, p-value: 1.2778720245935414e-17
Chi-square test between Product_Search_Method and
Personalized_Recommendation_Frequency :
Chi-square statistic: 69.49344024771989, p-value: 3.983613863615087e-10
Chi-square test between Product_Search_Method and Recommendation_Helpfulness:
Chi-square statistic: 29.475314729347836, p-value: 4.944437244986284e-05
Chi-square test between Product_Search_Method and Rating_Accuracy :
Chi-square statistic: 45.716427079904875, p-value: 7.7636425332048e-06
Chi-square test between Product_Search_Method and Shopping_Satisfaction:
Chi-square statistic: 99.33803292479544, p-value: 7.50428278881834e-16
Chi-square test between Product_Search_Method and Service_Appreciation:
Chi-square statistic: 73.75299880248157, p-value: 1.0365596759713254e-08
Chi-square test between Product_Search_Method and Improvement_Areas:
Chi-square statistic: 40.23850926044078, p-value: 0.7794462249755684
Chi-square test between Product_Search_Method and Grouped_Categories:
Chi-square statistic: 56.48362271268523, p-value: 6.344556599939725e-09
Chi-square test between Search_Result_Exploration and
Customer_Reviews_Importance:
Chi-square statistic: 16.297730467511645, p-value: 0.0026446001944138296
Chi-square test between Search_Result_Exploration and Add_to_Cart_Browsing:
Chi-square statistic: 28.26812067263941, p-value: 7.272028780912668e-07
Chi-square test between Search_Result_Exploration and Cart_Completion_Frequency:
Chi-square statistic: 13.1776000731635, p-value: 0.010439834432387874
```

```
Chi-square test between Search_Result_Exploration and Cart_Abandonment_Factors:
Chi-square statistic: 8.607701875981899, p-value: 0.034988062036111815
Chi-square test between Search_Result_Exploration and Saveforlater_Frequency:
Chi-square statistic: 5.363881465388379, p-value: 0.2519560159688016
Chi-square test between Search_Result_Exploration and Review_Left:
Chi-square statistic: 12.349446291082021, p-value: 0.0004411172866431046
Chi-square test between Search_Result_Exploration and Review_Reliability:
Chi-square statistic: 13.72276836269345, p-value: 0.008234509451044623
Chi-square test between Search_Result_Exploration and Review_Helpfulness:
Chi-square statistic: 36.2610322787411, p-value: 1.3366474882192892e-08
Chi-square test between Search_Result_Exploration and
Personalized_Recommendation_Frequency :
Chi-square statistic: 4.61917466115294, p-value: 0.3286493761475685
Chi-square test between Search_Result_Exploration and
Recommendation_Helpfulness:
Chi-square statistic: 22.853782056685, p-value: 1.0898438582061517e-05
Chi-square test between Search_Result_Exploration and Rating_Accuracy :
Chi-square statistic: 9.411265401011851, p-value: 0.05160282703292804
Chi-square test between Search_Result_Exploration and Shopping_Satisfaction:
Chi-square statistic: 13.022335290649375, p-value: 0.011167173822660163
Chi-square test between Search_Result_Exploration and Service_Appreciation:
Chi-square statistic: 12.073232316419498, p-value: 0.06035488034793678
Chi-square test between Search_Result_Exploration and Improvement_Areas:
Chi-square statistic: 9.933728345813543, p-value: 0.8700661819557154
Chi-square test between Search_Result_Exploration and Grouped_Categories:
Chi-square statistic: 8.607701875981899, p-value: 0.034988062036111815
Chi-square test between Customer_Reviews_Importance and Add_to_Cart_Browsing:
Chi-square statistic: 156.70006088702814, p-value: 7.829116882905331e-30
Chi-square test between Customer_Reviews_Importance and
Cart_Completion_Frequency:
Chi-square statistic: 75.06448415715971, p-value: 1.2734534488895316e-09
Chi-square test between Customer_Reviews_Importance and
Cart_Abandonment_Factors:
Chi-square statistic: 38.737320146286095, p-value: 0.00011624947401168799
Chi-square test between Customer_Reviews_Importance and Saveforlater_Frequency:
Chi-square statistic: 83.6838018680602, p-value: 3.583519083673104e-11
Chi-square test between Customer_Reviews_Importance and Review_Left:
Chi-square statistic: 1.335167553488658, p-value: 0.8553812198279427
Chi-square test between Customer_Reviews_Importance and Review_Reliability:
Chi-square statistic: 228.33725075395344, p-value: 1.3957655005483794e-39
Chi-square test between Customer_Reviews_Importance and Review_Helpfulness:
Chi-square statistic: 178.79256987400672, p-value: 1.8456435737112015e-34
Chi-square test between Customer_Reviews_Importance and
Personalized_Recommendation_Frequency :
Chi-square statistic: 137.59783501823262, p-value: 2.1256071937894038e-21
Chi-square test between Customer_Reviews_Importance and
Recommendation_Helpfulness:
Chi-square statistic: 66.82700923664375, p-value: 2.0982381898778658e-11
```

```
Chi-square test between Customer_Reviews_Importance and Rating_Accuracy :
Chi-square statistic: 118.97657908680127, p-value: 8.639039315785884e-18
Chi-square test between Customer_Reviews_Importance and Shopping_Satisfaction:
Chi-square statistic: 171.65383544662046, p-value: 3.937769108677241e-28
Chi-square test between Customer_Reviews_Importance and Service_Appreciation:
Chi-square statistic: 116.64441412553437, p-value: 3.826025396295212e-14
Chi-square test between Customer_Reviews_Importance and Improvement_Areas:
Chi-square statistic: 121.4338262012232, p-value: 1.974700648452055e-05
Chi-square test between Customer_Reviews_Importance and Grouped_Categories:
Chi-square statistic: 38.737320146286095, p-value: 0.00011624947401168799
Chi-square test between Add_to_Cart_Browsing and Cart_Completion_Frequency:
Chi-square statistic: 42.62672566041564, p-value: 1.0326535696123594e-06
Chi-square test between Add_to_Cart_Browsing and Cart_Abandonment_Factors:
Chi-square statistic: 11.547081772165942, p-value: 0.07286996998002397
Chi-square test between Add_to_Cart_Browsing and Saveforlater_Frequency:
Chi-square statistic: 57.93750860434422, p-value: 1.1813133046951598e-09
Chi-square test between Add_to_Cart_Browsing and Review_Left:
Chi-square statistic: 20.99489934766807, p-value: 2.7606765904003146e-05
Chi-square test between Add_to_Cart_Browsing and Review_Reliability:
Chi-square statistic: 90.42894943156762, p-value: 3.8054351808587597e-16
Chi-square test between Add_to_Cart_Browsing and Review_Helpfulness:
Chi-square statistic: 226.98561478576997, p-value: 5.881366155654918e-48
Chi-square test between Add_to_Cart_Browsing and
Personalized_Recommendation_Frequency :
Chi-square statistic: 41.23186229425843, p-value: 1.8864721716071894e-06
Chi-square test between Add_to_Cart_Browsing and Recommendation_Helpfulness:
Chi-square statistic: 151.41943220273743, p-value: 1.0105049520843239e-31
Chi-square test between Add_to_Cart_Browsing and Rating_Accuracy :
Chi-square statistic: 33.65168274308441, p-value: 4.697444455167016e-05
Chi-square test between Add_to_Cart_Browsing and Shopping_Satisfaction:
Chi-square statistic: 62.19599438556492, p-value: 1.7253726726949972e-10
Chi-square test between Add_to_Cart_Browsing and Service_Appreciation:
Chi-square statistic: 79.21483667351349, p-value: 5.825382490902052e-12
Chi-square test between Add_to_Cart_Browsing and Improvement_Areas:
Chi-square statistic: 73.19796563105717, p-value: 4.547398886297224e-05
Chi-square test between Add_to_Cart_Browsing and Grouped_Categories:
Chi-square statistic: 11.547081772165942, p-value: 0.07286996998002397
Chi-square test between Cart_Completion_Frequency and Cart_Abandonment_Factors:
Chi-square statistic: 91.07277396570998, p-value: 3.0596465740029843e-14
Chi-square test between Cart_Completion_Frequency and Saveforlater_Frequency:
Chi-square statistic: 104.81792484990486, p-value: 4.298792383665873e-15
Chi-square test between Cart_Completion_Frequency and Review_Left:
Chi-square statistic: 14.780452431659542, p-value: 0.0051789200284697795
Chi-square test between Cart_Completion_Frequency and Review_Reliability:
Chi-square statistic: 119.85479199652589, p-value: 5.857527120402249e-18
Chi-square test between Cart_Completion_Frequency and Review_Helpfulness:
Chi-square statistic: 25.528112797516552, p-value: 0.0012643867500646542
Chi-square test between Cart_Completion_Frequency and
```

```
Personalized_Recommendation_Frequency :
Chi-square statistic: 102.73322632359051, p-value: 1.0623227262672988e-14
Chi-square test between Cart_Completion_Frequency and
Recommendation_Helpfulness:
Chi-square statistic: 31.225345885650526, p-value: 0.00012810096616624675
Chi-square test between Cart_Completion_Frequency and Rating_Accuracy :
Chi-square statistic: 86.71774874935362, p-value: 1.0020869101216218e-11
Chi-square test between Cart_Completion_Frequency and Shopping_Satisfaction:
Chi-square statistic: 86.37798257010706, p-value: 1.1562707578375603e-11
Chi-square test between Cart_Completion_Frequency and Service_Appreciation:
Chi-square statistic: 48.527046924302674, p-value: 0.0021707836719040657
Chi-square test between Cart_Completion_Frequency and Improvement_Areas:
Chi-square statistic: 60.289912976554135, p-value: 0.6084303583396407
Chi-square test between Cart_Completion_Frequency and Grouped_Categories:
Chi-square statistic: 91.07277396570998, p-value: 3.0596465740029843e-14
Chi-square test between Cart_Abandonment_Factors and Saveforlater_Frequency:
Chi-square statistic: 40.30904940561916, p-value: 6.388960125693191e-05
Chi-square test between Cart_Abandonment_Factors and Review_Left:
Chi-square statistic: 18.68732340710841, p-value: 0.00031726538514653983
Chi-square test between Cart_Abandonment_Factors and Review_Reliability:
Chi-square statistic: 44.14618055375843, p-value: 1.442023596788672e-05
Chi-square test between Cart_Abandonment_Factors and Review_Helpfulness:
Chi-square statistic: 12.721499067738483, p-value: 0.04767802583897619
Chi-square test between Cart_Abandonment_Factors and
Personalized_Recommendation_Frequency :
Chi-square statistic: 40.28861018793822, p-value: 6.439169703105747e-05
Chi-square test between Cart_Abandonment_Factors and Recommendation_Helpfulness:
Chi-square statistic: 15.772611747390792, p-value: 0.01502791050317024
Chi-square test between Cart_Abandonment_Factors and Rating_Accuracy :
Chi-square statistic: 59.430898981141894, p-value: 2.8657966689395272e-08
Chi-square test between Cart_Abandonment_Factors and Shopping_Satisfaction:
Chi-square statistic: 36.64272726993425, p-value: 0.00025523201742729896
Chi-square test between Cart_Abandonment_Factors and Service_Appreciation:
Chi-square statistic: 23.702188794826196, p-value: 0.16502867057863968
Chi-square test between Cart_Abandonment_Factors and Improvement_Areas:
Chi-square statistic: 59.70802745590636, p-value: 0.11970590577031741
Chi-square test between Cart_Abandonment_Factors and Grouped_Categories:
Chi-square statistic: 1803.0000000000005, p-value: 0.0
Chi-square test between Saveforlater_Frequency and Review_Left:
Chi-square statistic: 32.11129674608041, p-value: 1.815468600827836e-06
Chi-square test between Saveforlater_Frequency and Review_Reliability:
Chi-square statistic: 109.48702965075492, p-value: 5.613552663362628e-16
Chi-square test between Saveforlater_Frequency and Review_Helpfulness:
Chi-square statistic: 48.09251059052722, p-value: 9.485236408104347e-08
Chi-square test between Saveforlater_Frequency and
Personalized_Recommendation_Frequency :
Chi-square statistic: 110.08058520789359, p-value: 4.3296950783352694e-16
Chi-square test between Saveforlater_Frequency and Recommendation_Helpfulness:
```

```
Chi-square statistic: 32.891888994608856, p-value: 6.44126217023345e-05
Chi-square test between Saveforlater_Frequency and Rating_Accuracy :
Chi-square statistic: 105.19113958815339, p-value: 3.65496160181111e-15
Chi-square test between Saveforlater_Frequency and Shopping_Satisfaction:
Chi-square statistic: 69.24027299589889, p-value: 1.3551439980924938e-08
Chi-square test between Saveforlater_Frequency and Service_Appreciation:
Chi-square statistic: 82.85691813463416, p-value: 2.1196612803079785e-08
Chi-square test between Saveforlater_Frequency and Improvement_Areas:
Chi-square statistic: 119.55857856274113, p-value: 3.1589390138573755e-05
Chi-square test between Saveforlater_Frequency and Grouped_Categories:
Chi-square statistic: 40.309049405619156, p-value: 6.388960125693215e-05
Chi-square test between Review_Left and Review_Reliability:
Chi-square statistic: 16.330795803933178, p-value: 0.0026059379624147863
Chi-square test between Review_Left and Review_Helpfulness:
Chi-square statistic: 2.091747334241385, p-value: 0.3513846919819495
Chi-square test between Review_Left and Personalized_Recommendation_Frequency :
Chi-square statistic: 11.033155727978183, p-value: 0.026193905930885324
Chi-square test between Review_Left and Recommendation_Helpfulness:
Chi-square statistic: 10.48350926093318, p-value: 0.005290964998000261
Chi-square test between Review_Left and Rating_Accuracy :
Chi-square statistic: 16.716693636427536, p-value: 0.0021938919521528547
Chi-square test between Review_Left and Shopping_Satisfaction:
Chi-square statistic: 10.410689622064506, p-value: 0.034049707931766086
Chi-square test between Review_Left and Service_Appreciation:
Chi-square statistic: 12.094269224486382, p-value: 0.05989851454780219
Chi-square test between Review_Left and Improvement_Areas:
Chi-square statistic: 15.547467952638785, p-value: 0.4849648685804965
Chi-square test between Review_Left and Grouped_Categories:
Chi-square statistic: 18.68732340710841, p-value: 0.00031726538514653983
Chi-square test between Review_Reliability and Review_Helpfulness:
Chi-square statistic: 124.93866351712856, p-value: 3.1606845866686785e-23
Chi-square test between Review_Reliability and
Personalized_Recommendation_Frequency :
Chi-square statistic: 133.09493522593422, p-value: 1.605958413423775e-20
Chi-square test between Review_Reliability and Recommendation_Helpfulness:
Chi-square statistic: 31.80008291755955, p-value: 0.00010114112456957561
Chi-square test between Review_Reliability and Rating_Accuracy :
Chi-square statistic: 117.66906806437125, p-value: 1.5396129778048636e-17
Chi-square test between Review_Reliability and Shopping_Satisfaction:
Chi-square statistic: 100.14030005222185, p-value: 3.260466957882296e-14
Chi-square test between Review_Reliability and Service_Appreciation:
Chi-square statistic: 83.84807463696393, p-value: 1.4660138548200755e-08
Chi-square test between Review_Reliability and Improvement_Areas:
Chi-square statistic: 102.44284785262363, p-value: 0.0016218362792001953
Chi-square test between Review_Reliability and Grouped_Categories:
Chi-square statistic: 44.14618055375843, p-value: 1.442023596788672e-05
Chi-square test between Review_Helpfulness and
Personalized_Recommendation_Frequency :
```

```
Chi-square statistic: 46.91804133559268, p-value: 1.589549950926764e-07
Chi-square test between Review_Helpfulness and Recommendation_Helpfulness:
Chi-square statistic: 171.2431642706835, p-value: 5.657772215921829e-36
Chi-square test between Review_Helpfulness and Rating_Accuracy :
Chi-square statistic: 33.930747859789754, p-value: 4.18172780080023e-05
Chi-square test between Review_Helpfulness and Shopping_Satisfaction:
Chi-square statistic: 40.5219857341354, p-value: 2.560498471516945e-06
Chi-square test between Review_Helpfulness and Service_Appreciation:
Chi-square statistic: 60.92540502348256, p-value: 1.5299486663439085e-08
Chi-square test between Review_Helpfulness and Improvement_Areas:
Chi-square statistic: 47.00668821257219, p-value: 0.04234768825050309
Chi-square test between Review_Helpfulness and Grouped_Categories:
Chi-square statistic: 12.721499067738483, p-value: 0.04767802583897619
Chi-square test between Personalized_Recommendation_Frequency  and
Recommendation_Helpfulness:
Chi-square statistic: 65.43555609357313, p-value: 3.957694101052292e-11
Chi-square test between Personalized_Recommendation_Frequency  and
Rating_Accuracy :
Chi-square statistic: 313.1565668283144, p-value: 4.778974047339139e-57
Chi-square test between Personalized_Recommendation_Frequency  and
Shopping_Satisfaction:
Chi-square statistic: 245.15327244662706, p-value: 5.097776834642253e-43
Chi-square test between Personalized_Recommendation_Frequency  and
Service_Appreciation:
Chi-square statistic: 51.99762111720191, p-value: 0.0007829507789765508
Chi-square test between Personalized_Recommendation_Frequency  and
Improvement_Areas:
Chi-square statistic: 68.7925092553677, p-value: 0.3184353464491627
Chi-square test between Personalized_Recommendation_Frequency  and
Grouped_Categories:
Chi-square statistic: 40.28861018793822, p-value: 6.439169703105747e-05
Chi-square test between Recommendation_Helpfulness and Rating_Accuracy :
Chi-square statistic: 105.51776334995343, p-value: 3.167848596633009e-19
Chi-square test between Recommendation_Helpfulness and Shopping_Satisfaction:
Chi-square statistic: 48.18795630382481, p-value: 9.094825211334138e-08
Chi-square test between Recommendation_Helpfulness and Service_Appreciation:
Chi-square statistic: 34.15778747670226, p-value: 0.0006369529166393654
Chi-square test between Recommendation_Helpfulness and Improvement_Areas:
Chi-square statistic: 53.181646070984584, p-value: 0.010748004108834842
Chi-square test between Recommendation_Helpfulness and Grouped_Categories:
Chi-square statistic: 15.77261174739079, p-value: 0.015027910503170256
Chi-square test between Rating_Accuracy  and Shopping_Satisfaction:
Chi-square statistic: 343.05538752392874, p-value: 2.899763978065602e-63
Chi-square test between Rating_Accuracy  and Service_Appreciation:
Chi-square statistic: 56.143680043841634, p-value: 0.0002190686984513935
Chi-square test between Rating_Accuracy  and Improvement_Areas:
Chi-square statistic: 73.75972281931463, p-value: 0.1892721292693555
Chi-square test between Rating_Accuracy  and Grouped_Categories:
```

```
Chi-square statistic: 59.430898981141894, p-value: 2.8657966689395272e-08
Chi-square test between Shopping_Satisfaction and Service_Appreciation:
Chi-square statistic: 72.48132221017079, p-value: 9.146348360183814e-07
Chi-square test between Shopping_Satisfaction and Improvement_Areas:
Chi-square statistic: 94.59873526612766, p-value: 0.00773381873335926
Chi-square test between Shopping_Satisfaction and Grouped_Categories:
Chi-square statistic: 36.642727269934255, p-value: 0.000255232017427299
Chi-square test between Service_Appreciation and Improvement_Areas:
Chi-square statistic: 438.96146873447327, p-value: 2.61620481729214e-45
Chi-square test between Service_Appreciation and Grouped_Categories:
Chi-square statistic: 23.702188794826192, p-value: 0.16502867057863982
Chi-square test between Improvement_Areas and Grouped_Categories:
Chi-square statistic: 59.708027455906354, p-value: 0.1197059057703175
Automatically selected pairs based on p-value threshold:
[('Gender', 'Purchase_Frequency'), ('Gender', 'Purchase_Categories'), ('Gender',
'Personalized_Recommendation_Frequency'), ('Gender', 'Browsing_Frequency'),
('Gender', 'Product_Search_Method'), ('Gender', 'Customer_Reviews_Importance'),
('Gender', 'Add_to_Cart_Browsing'), ('Gender', 'Cart_Completion_Frequency'),
('Gender', 'Review_Reliability'), ('Gender', 'Review_Helpfulness'), ('Gender',
'Recommendation_Helpfulness'), ('Gender', 'Rating_Accuracy '),
('Purchase_Frequency', 'Purchase_Categories'), ('Purchase_Frequency',
'Personalized_Recommendation_Frequency'), ('Purchase_Frequency',
'Browsing_Frequency'), ('Purchase_Frequency', 'Product_Search_Method'),
('Purchase_Frequency', 'Customer_Reviews_Importance'), ('Purchase_Frequency',
'Add_to_Cart_Browsing'), ('Purchase_Frequency', 'Cart_Completion_Frequency'),
('Purchase_Frequency', 'Saveforlater_Frequency'), ('Purchase_Frequency',
'Review_Left'), ('Purchase_Frequency', 'Review_Reliability'),
('Purchase_Frequency', 'Review_Helpfulness'), ('Purchase_Frequency',
'Personalized_Recommendation_Frequency '), ('Purchase_Frequency',
'Recommendation_Helpfulness'), ('Purchase_Frequency', 'Service_Appreciation'),
('Purchase_Frequency', 'Improvement_Areas'), ('Purchase_Categories',
'Personalized_Recommendation_Frequency'), ('Purchase_Categories',
'Browsing_Frequency'), ('Purchase_Categories', 'Product_Search_Method'),
('Purchase_Categories', 'Search_Result_Exploration'), ('Purchase_Categories',
'Customer_Reviews_Importance'), ('Purchase_Categories', 'Add_to_Cart_Browsing'),
('Purchase_Categories', 'Cart_Completion_Frequency'), ('Purchase_Categories',
'Saveforlater_Frequency'), ('Purchase_Categories', 'Review_Left'),
('Purchase_Categories', 'Review_Reliability'), ('Purchase_Categories',
'Review_Helpfulness'), ('Purchase_Categories',
'Personalized_Recommendation_Frequency '), ('Purchase_Categories',
'Recommendation_Helpfulness'), ('Purchase_Categories', 'Rating_Accuracy '),
('Purchase_Categories', 'Shopping_Satisfaction'), ('Purchase_Categories',
'Improvement_Areas'), ('Personalized_Recommendation_Frequency',
'Browsing_Frequency'), ('Personalized_Recommendation_Frequency',
'Product_Search_Method'), ('Personalized_Recommendation_Frequency',
'Search_Result_Exploration'), ('Personalized_Recommendation_Frequency',
'Customer_Reviews_Importance'), ('Personalized_Recommendation_Frequency',
'Add_to_Cart_Browsing'), ('Personalized_Recommendation_Frequency',
```

'Cart_Completion_Frequency'), ('Personalized_Recommendation_Frequency',
'Saveforlater_Frequency'), ('Personalized_Recommendation_Frequency',
'Review_Left'), ('Personalized_Recommendation_Frequency', 'Review_Reliability'),
('Personalized_Recommendation_Frequency', 'Review_Helpfulness'),
('Personalized_Recommendation_Frequency', 'Personalized_Recommendation_Frequency
'), ('Personalized_Recommendation_Frequency', 'Recommendation_Helpfulness'),
('Personalized_Recommendation_Frequency', 'Shopping_Satisfaction'),
('Personalized_Recommendation_Frequency', 'Service_Appreciation'),
('Personalized_Recommendation_Frequency', 'Improvement_Areas'),
('Browsing_Frequency', 'Product_Search_Method'), ('Browsing_Frequency',
'Search_Result_Exploration'), ('Browsing_Frequency',
'Customer_Reviews_Importance'), ('Browsing_Frequency',
'Cart_Completion_Frequency'), ('Browsing_Frequency',
'Cart_Abandonment_Factors'), ('Browsing_Frequency', 'Saveforlater_Frequency'),
('Browsing_Frequency', 'Review_Left'), ('Browsing_Frequency',
'Review_Reliability'), ('Browsing_Frequency',
'Personalized_Recommendation_Frequency '), ('Browsing_Frequency',
'Rating_Accuracy '), ('Browsing_Frequency', 'Shopping_Satisfaction'),
('Browsing_Frequency', 'Service_Appreciation'), ('Browsing_Frequency',
'Improvement_Areas'), ('Browsing_Frequency', 'Grouped_Categories'),
('Product_Search_Method', 'Search_Result_Exploration'),
('Product_Search_Method', 'Customer_Reviews_Importance'),
('Product_Search_Method', 'Add_to_Cart_Browsing'), ('Product_Search_Method',
'Cart_Completion_Frequency'), ('Product_Search_Method',
'Cart_Abandonment_Factors'), ('Product_Search_Method',
'Saveforlater_Frequency'), ('Product_Search_Method', 'Review_Reliability'),
('Product_Search_Method', 'Review_Helpfulness'), ('Product_Search_Method',
'Personalized_Recommendation_Frequency '), ('Product_Search_Method',
'Recommendation_Helpfulness'), ('Product_Search_Method', 'Rating_Accuracy '),
('Product_Search_Method', 'Shopping_Satisfaction'), ('Product_Search_Method',
'Service_Appreciation'), ('Product_Search_Method', 'Grouped_Categories'),
('Search_Result_Exploration', 'Customer_Reviews_Importance'),
('Search_Result_Exploration', 'Add_to_Cart_Browsing'),
('Search_Result_Exploration', 'Cart_Completion_Frequency'),
('Search_Result_Exploration', 'Cart_Abandonment_Factors'),
('Search_Result_Exploration', 'Review_Left'), ('Search_Result_Exploration',
'Review_Reliability'), ('Search_Result_Exploration', 'Review_Helpfulness'),
('Search_Result_Exploration', 'Recommendation_Helpfulness'),
('Search_Result_Exploration', 'Shopping_Satisfaction'),
('Search_Result_Exploration', 'Grouped_Categories'),
('Customer_Reviews_Importance', 'Add_to_Cart_Browsing'),
('Customer_Reviews_Importance', 'Cart_Completion_Frequency'),
('Customer_Reviews_Importance', 'Cart_Abandonment_Factors'),
('Customer_Reviews_Importance', 'Saveforlater_Frequency'),
('Customer_Reviews_Importance', 'Review_Reliability'),
('Customer_Reviews_Importance', 'Review_Helpfulness'),
('Customer_Reviews_Importance', 'Personalized_Recommendation_Frequency '),
('Customer_Reviews_Importance', 'Recommendation_Helpfulness'),

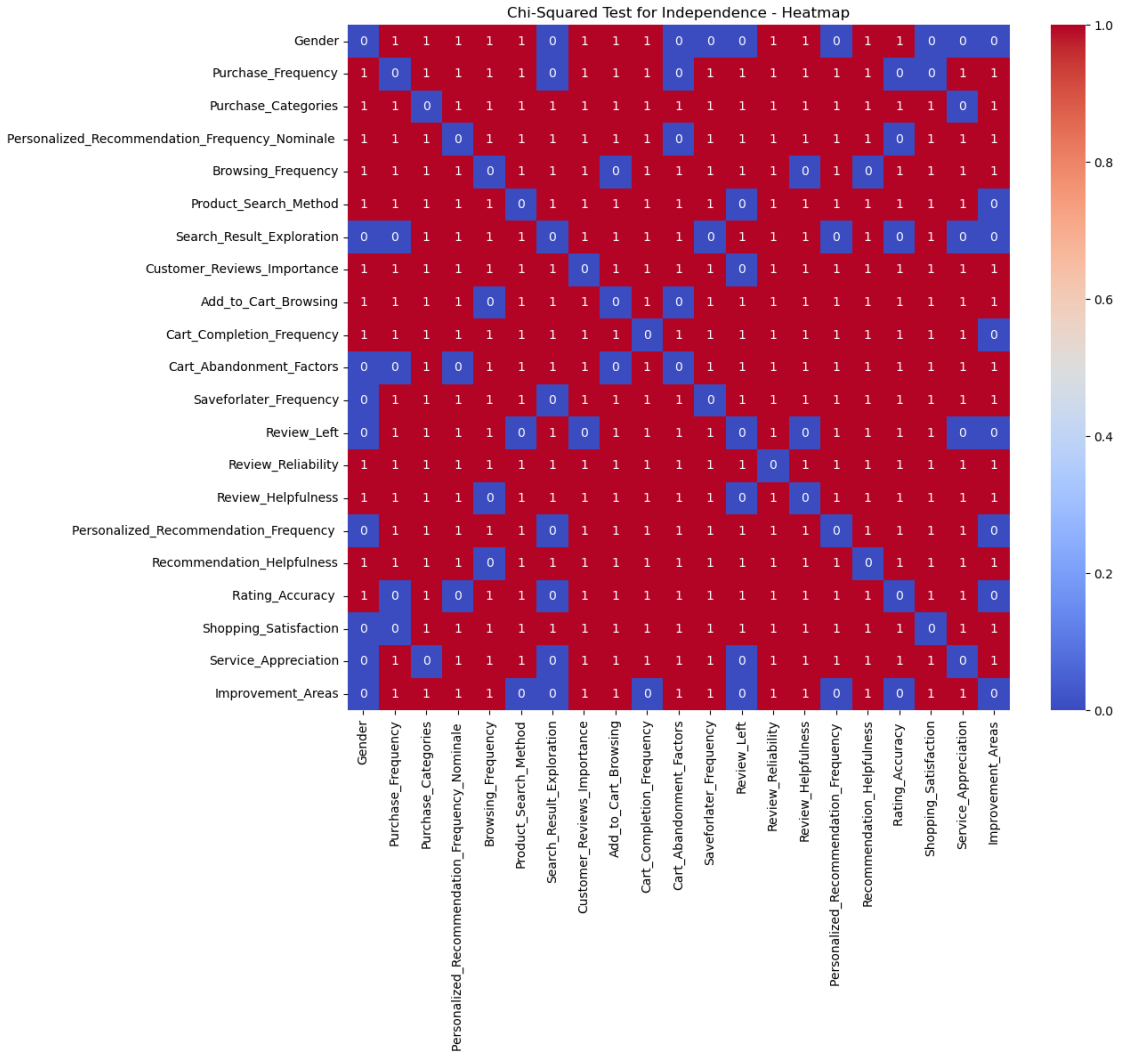
('Customer_Reviews_Importance', 'Rating_Accuracy '),
('Customer_Reviews_Importance', 'Shopping_Satisfaction'),
('Customer_Reviews_Importance', 'Service_Appreciation'),
('Customer_Reviews_Importance', 'Improvement_Areas'),
('Customer_Reviews_Importance', 'Grouped_Categories'), ('Add_to_Cart_Browsing',
'Cart_Completion_Frequency'), ('Add_to_Cart_Browsing',
'Saveforlater_Frequency'), ('Add_to_Cart_Browsing', 'Review_Left'),
('Add_to_Cart_Browsing', 'Review_Reliability'), ('Add_to_Cart_Browsing',
'Review_Helpfulness'), ('Add_to_Cart_Browsing',
'Personalized_Recommendation_Frequency '), ('Add_to_Cart_Browsing',
'Recommendation_Helpfulness'), ('Add_to_Cart_Browsing', 'Rating_Accuracy '),
('Add_to_Cart_Browsing', 'Shopping_Satisfaction'), ('Add_to_Cart_Browsing',
'Service_Appreciation'), ('Add_to_Cart_Browsing', 'Improvement_Areas'),
('Cart_Completion_Frequency', 'Cart_Abandonment_Factors'),
('Cart_Completion_Frequency', 'Saveforlater_Frequency'),
('Cart_Completion_Frequency', 'Review_Left'), ('Cart_Completion_Frequency',
'Review_Reliability'), ('Cart_Completion_Frequency', 'Review_Helpfulness'),
('Cart_Completion_Frequency', 'Personalized_Recommendation_Frequency '),
('Cart_Completion_Frequency', 'Recommendation_Helpfulness'),
('Cart_Completion_Frequency', 'Rating_Accuracy '), ('Cart_Completion_Frequency',
'Shopping_Satisfaction'), ('Cart_Completion_Frequency', 'Service_Appreciation'),
('Cart_Completion_Frequency', 'Grouped_Categories'),
('Cart_Abandonment_Factors', 'Saveforlater_Frequency'),
('Cart_Abandonment_Factors', 'Review_Left'), ('Cart_Abandonment_Factors',
'Review_Reliability'), ('Cart_Abandonment_Factors', 'Review_Helpfulness'),
('Cart_Abandonment_Factors', 'Personalized_Recommendation_Frequency '),
('Cart_Abandonment_Factors', 'Recommendation_Helpfulness'),
('Cart_Abandonment_Factors', 'Rating_Accuracy '), ('Cart_Abandonment_Factors',
'Shopping_Satisfaction'), ('Cart_Abandonment_Factors', 'Grouped_Categories'),
('Saveforlater_Frequency', 'Review_Left'), ('Saveforlater_Frequency',
'Review_Reliability'), ('Saveforlater_Frequency', 'Review_Helpfulness'),
('Saveforlater_Frequency', 'Personalized_Recommendation_Frequency '),
('Saveforlater_Frequency', 'Recommendation_Helpfulness'),
('Saveforlater_Frequency', 'Rating_Accuracy '), ('Saveforlater_Frequency',
'Shopping_Satisfaction'), ('Saveforlater_Frequency', 'Service_Appreciation'),
('Saveforlater_Frequency', 'Improvement_Areas'), ('Saveforlater_Frequency',
'Grouped_Categories'), ('Review_Left', 'Review_Reliability'), ('Review_Left',
'Personalized_Recommendation_Frequency '), ('Review_Left',
'Recommendation_Helpfulness'), ('Review_Left', 'Rating_Accuracy '),
('Review_Left', 'Shopping_Satisfaction'), ('Review_Left', 'Grouped_Categories'),
('Review_Reliability', 'Review_Helpfulness'), ('Review_Reliability',
'Personalized_Recommendation_Frequency '), ('Review_Reliability',
'Recommendation_Helpfulness'), ('Review_Reliability', 'Rating_Accuracy '),
('Review_Reliability', 'Shopping_Satisfaction'), ('Review_Reliability',
'Service_Appreciation'), ('Review_Reliability', 'Improvement_Areas'),
('Review_Reliability', 'Grouped_Categories'), ('Review_Helpfulness',
'Personalized_Recommendation_Frequency '), ('Review_Helpfulness',
'Recommendation_Helpfulness'), ('Review_Helpfulness', 'Rating_Accuracy '),

```
('Review_Helpfulness', 'Shopping_Satisfaction'), ('Review_Helpfulness',
'Service_Appreciation'), ('Review_Helpfulness', 'Improvement_Areas'),
('Review_Helpfulness', 'Grouped_Categories'),
('Personalized_Recommendation_Frequency ', 'Recommendation_Helpfulness'),
('Personalized_Recommendation_Frequency ', 'Rating_Accuracy '),
('Personalized_Recommendation_Frequency ', 'Shopping_Satisfaction'),
('Personalized_Recommendation_Frequency ', 'Service_Appreciation'),
('Personalized_Recommendation_Frequency ', 'Grouped_Categories'),
('Recommendation_Helpfulness', 'Rating_Accuracy '),
('Recommendation_Helpfulness', 'Shopping_Satisfaction'),
('Recommendation_Helpfulness', 'Service_Appreciation'),
('Recommendation_Helpfulness', 'Improvement_Areas'),
('Recommendation_Helpfulness', 'Grouped_Categories'), ('Rating_Accuracy ',
'Shopping_Satisfaction'), ('Rating_Accuracy ', 'Service_Appreciation'),
('Rating_Accuracy ', 'Grouped_Categories'), ('Shopping_Satisfaction',
'Service_Appreciation'), ('Shopping_Satisfaction', 'Improvement_Areas'),
('Shopping_Satisfaction', 'Grouped_Categories'), ('Service_Appreciation',
'Improvement_Areas')]
```

[509]:
```python
categorical_cols =  ['Gender', 'Purchase_Frequency', 'Purchase_Categories',
       'Personalized_Recommendation_Frequency_Nominale ', 'Browsing_Frequency',
       'Product_Search_Method', 'Search_Result_Exploration',
       'Customer_Reviews_Importance', 'Add_to_Cart_Browsing',
       'Cart_Completion_Frequency', 'Cart_Abandonment_Factors',
       'Saveforlater_Frequency', 'Review_Left', 'Review_Reliability',
       'Review_Helpfulness', 'Personalized_Recommendation_Frequency ',
       'Recommendation_Helpfulness', 'Rating_Accuracy ',
       'Shopping_Satisfaction', 'Service_Appreciation', 'Improvement_Areas']
# Set a p-value threshold for feature selection
p_value_threshold = 0.05

# Create a DataFrame to store the p-values
p_values_df = pd.DataFrame(index=categorical_cols, columns=categorical_cols)

# Perform the Chi-squared test for independence for all pairs of variables
for pair in combinations(categorical_cols, 2):
    contingency_table = pd.crosstab(data[pair[0]], data[pair[1]])
    chi2, p, _, _ = chi2_contingency(contingency_table)
    p_values_df.loc[pair[0], pair[1]] = p
    p_values_df.loc[pair[1], pair[0]] = p

# Convert p-values to a binary significance matrix
significance_matrix = p_values_df < p_value_threshold

# Create a heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(significance_matrix, annot=True, cmap='coolwarm', fmt='d')
```

```
print(' -**0 :** Cela pourrait indiquer que le test de chi-carré n\'a pas␣
 ↪rejeté l\'hypothèse nulle H0, ce qui suggère que les deux variables␣
 ↪correspondantes sont indépendantes (p-value  0.05).','\n'

,'- **1 :** Cela pourrait indiquer que le test de chi-carré a rejeté␣
 ↪l\'hypothèse nulle H0, suggérant ainsi qu\'il y a une relation␣
 ↪statistiquement significative entre les deux variables correspondantes␣
 ↪(p-value < 0.05).')
plt.title('Chi-Squared Test for Independence - Heatmap')
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de␣
 ↪projet\\heatmapChi-squared test for independence for all pairs of variables.
 ↪png'  # Remplacez par le chemin et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```

-**0 :** Cela pourrait indiquer que le test de chi-carré n'a pas rejeté l'hypothèse nulle H0, ce qui suggère que les deux variables correspondantes sont indépendantes (p-value 0.05).
- **1 :** Cela pourrait indiquer que le test de chi-carré a rejeté l'hypothèse nulle H0, suggérant ainsi qu'il y a une relation statistiquement significative entre les deux variables correspondantes (p-value < 0.05).



[535]:
```
# on va crée une nouvelle data dans la quelle on va encoder les variables :
data_encoding = data
```

[536]:
```
data_encoding.drop(['Timestamp'],axis=1,inplace=True) # on a pas besoin de␣
↪Timestamp
```

[537]:
```
data_encoding.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 602 entries, 0 to 601
Data columns (total 22 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   age                                             602 non-null    int64
 1   Gender                                          602 non-null    object
```

```
 2   Purchase_Frequency                                602 non-null    object
 3   Purchase_Categories                               602 non-null    object
 4   Personalized_Recommendation_Frequency_Nominale    602 non-null    object
 5   Browsing_Frequency                                602 non-null    object
 6   Product_Search_Method                             600 non-null    object
 7   Search_Result_Exploration                         602 non-null    object
 8   Customer_Reviews_Importance                       602 non-null    int64
 9   Add_to_Cart_Browsing                              602 non-null    object
 10  Cart_Completion_Frequency                         602 non-null    object
 11  Cart_Abandonment_Factors                          602 non-null    object
 12  Saveforlater_Frequency                            602 non-null    object
 13  Review_Left                                       602 non-null    object
 14  Review_Reliability                                602 non-null    object
 15  Review_Helpfulness                                602 non-null    object
 16  Personalized_Recommendation_Frequency             602 non-null    int64
 17  Recommendation_Helpfulness                        602 non-null    object
 18  Rating_Accuracy                                   602 non-null    int64
 19  Shopping_Satisfaction                             602 non-null    int64
 20  Service_Appreciation                              602 non-null    object
 21  Improvement_Areas                                 602 non-null    object
dtypes: int64(5), object(17)
memory usage: 103.6+ KB
```

4.1 Extraction des valeurs uniques pour chaque variable :

[539]:
```python
# Extract unique values for each object variable
unique_values = {}
for column in data_encoding.columns:
    unique_values[column] = data_encoding[column].unique()

for column, values in unique_values.items():
    print(f"Unique values for {column}:\n{values}\n")
```

```
Unique values for age:
[23 24 22 21 20 25 16 64 29 19 26 32 30 40 36 31 47 54 58 53 28 55 62 27
 34 44 38 35 42 37 45 50 63 46 33 60 18 17 57 41 39 48 49 15 43 52  3 67
 56 12]

Unique values for Gender:
['Female' 'Prefer not to say' 'Male' 'Others']

Unique values for Purchase_Frequency:
['Few times a month' 'Once a month' 'Less than once a month'
 'Multiple times a week' 'Once a week']

Unique values for Purchase_Categories:
['Beauty and Personal Care' 'Clothing and Fashion'
 'Groceries and Gourmet Food;Clothing and Fashion'
```

```
 'Beauty and Personal Care;Clothing and Fashion;others'
 'Beauty and Personal Care;Clothing and Fashion'
 'Beauty and Personal Care;Clothing and Fashion;Home and Kitchen'
 'Clothing and Fashion;Home and Kitchen' 'others'
 'Clothing and Fashion;others' 'Beauty and Personal Care;Home and Kitchen'
 'Groceries and Gourmet Food'
 'Groceries and Gourmet Food;Clothing and Fashion;others'
 'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and Fashion;Home
and Kitchen'
 'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and Fashion;Home
and Kitchen;others'
 'Home and Kitchen' 'Beauty and Personal Care;others'
 'Beauty and Personal Care;Home and Kitchen;others'
 'Home and Kitchen;others' 'Groceries and Gourmet Food;Home and Kitchen'
 'Beauty and Personal Care;Clothing and Fashion;Home and Kitchen;others'
 'Groceries and Gourmet Food;Beauty and Personal Care;Home and Kitchen'
 'Groceries and Gourmet Food;Home and Kitchen;others'
 'Groceries and Gourmet Food;Clothing and Fashion;Home and Kitchen;others'
 'Groceries and Gourmet Food;Beauty and Personal Care'
 'Clothing and Fashion;Home and Kitchen;others'
 'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and Fashion'
 'Groceries and Gourmet Food;Clothing and Fashion;Home and Kitchen'
 'Groceries and Gourmet Food;Beauty and Personal Care;others'
 'Groceries and Gourmet Food;Beauty and Personal Care;Clothing and
Fashion;others']

Unique values for Personalized_Recommendation_Frequency_Nominale :
['Yes' 'No' 'Sometimes']

Unique values for Browsing_Frequency:
['Few times a week' 'Few times a month' 'Rarely' 'Multiple times a day']

Unique values for Product_Search_Method:
['Keyword' 'Filter' 'categories' 'others' nan]

Unique values for Search_Result_Exploration:
['Multiple pages' 'First page']

Unique values for Customer_Reviews_Importance:
[1 2 5 3 4]

Unique values for Add_to_Cart_Browsing:
['Yes' 'Maybe' 'No']

Unique values for Cart_Completion_Frequency:
['Sometimes' 'Often' 'Rarely' 'Never' 'Always']

Unique values for Cart_Abandonment_Factors:
```

```
['Found a better price elsewhere' 'High shipping costs'
 'Changed my mind or no longer need the item' 'others']

Unique values for Saveforlater_Frequency:
['Sometimes' 'Rarely' 'Never' 'Often' 'Always']

Unique values for Review_Left:
['Yes' 'No']

Unique values for Review_Reliability:
['Occasionally' 'Heavily' 'Moderately' 'Never' 'Rarely']

Unique values for Review_Helpfulness:
['Yes' 'No' 'Sometimes']

Unique values for Personalized_Recommendation_Frequency :
[2 4 3 5 1]

Unique values for Recommendation_Helpfulness:
['Yes' 'Sometimes' 'No']

Unique values for Rating_Accuracy :
[1 3 2 5 4]

Unique values for Shopping_Satisfaction:
[1 2 3 4 5]

Unique values for Service_Appreciation:
['Competitive prices' 'Wide product selection'
 'User-friendly website/app interface' '.' 'Customer service '
 'Product recommendations' 'Customer service' 'Quick delivery'
 'All the above']

Unique values for Improvement_Areas:
['Reducing packaging waste' 'Product quality and accuracy'
 'Shipping speed and reliability' 'Customer service responsiveness' '.'
 'Nothing' 'better app interface and lower shipping charges' 'Nil'
 'Add more familiar brands to the list' 'UI'
 'Scrolling option would be much better than going to next page'
 'Quality of product is very poor according to the big offers'
 'I have no problem with Amazon yet. But others tell me about the refund issues
'
 'User interface ' 'Irrelevant product suggestions'
 'User interface of app' "I don't have any problem with Amazon"
 'No problems with Amazon']
```

4.1.1 Gestion des Variables dans l'Analyse de Données

[545]:
```
quantitative_variable = ['age']



# Nominal variables
qualitative_nominal_vars = [
    'Gender',
    'Product_Search_Method',
    'Search_Result_Exploration',
    'Add_to_Cart_Browsing',
    'Personalized_Recommendation_Frequency_Nominale ',
    'Cart_Abandonment_Factors',
    'Review_Left',
    'Review_Helpfulness',
    'Recommendation_Helpfulness',
    'Purchase_Categories',
     'Service_Appreciation',
    'Improvement_Areas'
]


# Ordinal variables
qualitative_ordinal_vars = [
    'Personalized_Recommendation_Frequency ',
    'Purchase_Frequency',
    'Customer_Reviews_Importance',
    'Browsing_Frequency',
    'Cart_Completion_Frequency',
    'Saveforlater_Frequency',
    'Review_Reliability',
    'Rating_Accuracy ',
     'Shopping_Satisfaction'
]
```

4.1.2 Encodage one-hot des variables nominales :

[542]:
```
# One-hot encoding for nominal variables
one_hot_encoder = OneHotEncoder(sparse=False, drop='first')
one_hot_encoded = one_hot_encoder.
 ↪fit_transform(data_encoding[qualitative_nominal_vars].fillna('Missing'))
data_encoding_one_hot = pd.DataFrame(one_hot_encoded, columns=one_hot_encoder.
 ↪get_feature_names_out(qualitative_nominal_vars))
data_encoding = pd.concat([data_encoding, data_encoding_one_hot], axis=1)

# Update the DataFrame after one-hot encoding
```

```
data_encoding = data_encoding.drop(columns=qualitative_nominal_vars)

data_encoding
```

[542]:
```
      age      Purchase_Frequency    Browsing_Frequency  \
0      23       Few times a month      Few times a week
1      23            Once a month     Few times a month
2      24       Few times a month     Few times a month
3      24            Once a month     Few times a month
4      22  Less than once a month     Few times a month
..    …                      …                 …
597    23             Once a week      Few times a week
598    23             Once a week      Few times a week
599    23            Once a month      Few times a week
600    23       Few times a month     Few times a month
601    23             Once a week  Multiple times a day

     Customer_Reviews_Importance Cart_Completion_Frequency  \
0                              1                Sometimes
1                              1                    Often
2                              2                Sometimes
3                              5                Sometimes
4                              1                Sometimes
..                           …                        …
597                            4                Sometimes
598                            3                Sometimes
599                            3                Sometimes
600                            1                    Often
601                            3                    Often

    Saveforlater_Frequency Review_Reliability  \
0                Sometimes       Occasionally
1                   Rarely            Heavily
2                   Rarely       Occasionally
3                Sometimes            Heavily
4                   Rarely            Heavily
..                     …                  …
597              Sometimes         Moderately
598              Sometimes            Heavily
599              Sometimes       Occasionally
600              Sometimes            Heavily
601              Sometimes         Moderately

     Personalized_Recommendation_Frequency   Rating_Accuracy   \
0                                        2                 1
1                                        2                 3
2                                        4                 3
```

```
3                                    3                3
4                                    4                2
..                                 …                  …
597                                  3                3
598                                  3                3
599                                  3                2
600                                  2                2
601                                  3                3

     Shopping_Satisfaction  …  Improvement_Areas_Nothing  \
0                        1  …                        0.0
1                        2  …                        0.0
2                        3  …                        0.0
3                        4  …                        0.0
4                        2  …                        0.0
..                     …  …                        …
597                      4  …                        0.0
598                      3  …                        0.0
599                      3  …                        0.0
600                      2  …                        0.0
601                      3  …                        0.0

     Improvement_Areas_Product quality and accuracy  \
0                                              0.0
1                                              0.0
2                                              1.0
3                                              1.0
4                                              1.0
..                                             …
597                                            0.0
598                                            0.0
599                                            1.0
600                                            1.0
601                                            1.0

     Improvement_Areas_Quality of product is very poor according to the big
offers  \
0                                                 0.0
1                                                 0.0
2                                                 0.0
3                                                 0.0
4                                                 0.0
..                                                …
597                                               0.0
598                                               0.0
599                                               0.0
600                                               0.0
```

```
601                                                 0.0

     Improvement_Areas_Reducing packaging waste  \
0                                             1.0
1                                             1.0
2                                             0.0
3                                             0.0
4                                             0.0
..                                            …
597                                           0.0
598                                           1.0
599                                           0.0
600                                           0.0
601                                           0.0

     Improvement_Areas_Scrolling option would be much better than going to next
page  \
0                                                  0.0
1                                                  0.0
2                                                  0.0
3                                                  0.0
4                                                  0.0
..                                                 …
597                                                0.0
598                                                0.0
599                                                0.0
600                                                0.0
601                                                0.0

     Improvement_Areas_Shipping speed and reliability  Improvement_Areas_UI  \
0                                                 0.0                   0.0
1                                                 0.0                   0.0
2                                                 0.0                   0.0
3                                                 0.0                   0.0
4                                                 0.0                   0.0
..                                                …                     …
597                                               0.0                   0.0
598                                               0.0                   0.0
599                                               0.0                   0.0
600                                               0.0                   0.0
601                                               0.0                   0.0

     Improvement_Areas_User interface   \
0                                  0.0
1                                  0.0
2                                  0.0
3                                  0.0
```

```
4                                   0.0
..                                  …
597                                 0.0
598                                 0.0
599                                 0.0
600                                 0.0
601                                 0.0

     Improvement_Areas_User interface of app  \
0                                         0.0
1                                         0.0
2                                         0.0
3                                         0.0
4                                         0.0
..                                        …
597                                       0.0
598                                       0.0
599                                       0.0
600                                       0.0
601                                       0.0

     Improvement_Areas_better app interface and lower shipping charges
0                                                                  0.0
1                                                                  0.0
2                                                                  0.0
3                                                                  0.0
4                                                                  0.0
..                                                                 …
597                                                                0.0
598                                                                0.0
599                                                                0.0
600                                                                0.0
601                                                                0.0

[602 rows x 83 columns]
```

4.1.3 Encodage Ordinal des Variables Qualitatives :

[549]:
```
for column in qualitative_ordinal_vars:
    unique_values = data_encoding[column].unique()
    print(f"Unique values for {column}:", unique_values)
```

```
Unique values for Personalized_Recommendation_Frequency : [2 4 3 5 1]
Unique values for Purchase_Frequency: ['Few times a month' 'Once a month' 'Less
than once a month'
 'Multiple times a week' 'Once a week']
Unique values for Customer_Reviews_Importance: [1 2 5 3 4]
```

```
Unique values for Browsing_Frequency: ['Few times a week' 'Few times a month'
'Rarely' 'Multiple times a day']
Unique values for Cart_Completion_Frequency: ['Sometimes' 'Often' 'Rarely'
'Never' 'Always']
Unique values for Saveforlater_Frequency: ['Sometimes' 'Rarely' 'Never' 'Often'
'Always']
Unique values for Review_Reliability: ['Occasionally' 'Heavily' 'Moderately'
'Never' 'Rarely']
Unique values for Rating_Accuracy : [1 3 2 5 4]
Unique values for Shopping_Satisfaction: [1 2 3 4 5]
```

[553]:
```
from sklearn.preprocessing import OrdinalEncoder

# Define a dictionary with unique values for each ordinal variable
ordinal_categories = {
    'Personalized_Recommendation_Frequency ': [1, 2, 3, 4, 5],
    'Purchase_Frequency': ['Less than once a month', 'Few times a month', 'Few␣
 ↪times a week', 'Multiple times a week', 'Once a month', 'Once a week'],
    'Customer_Reviews_Importance': [1, 2, 3, 4, 5],
    'Browsing_Frequency': ['Rarely', 'Few times a month', 'Few times a week',␣
 ↪'Multiple times a day'],
    'Cart_Completion_Frequency': ['Never', 'Rarely', 'Sometimes', 'Often',␣
 ↪'Always'],
    'Saveforlater_Frequency': ['Never', 'Rarely', 'Sometimes', 'Often',␣
 ↪'Always'],
    'Review_Reliability': ['Never', 'Rarely', 'Occasionally', 'Moderately',␣
 ↪'Heavily'],
    'Rating_Accuracy ': [1, 2, 3, 4, 5],
    'Shopping_Satisfaction': [1, 2, 3, 4, 5]
}

# Apply OrdinalEncoder to each ordinal variable
ordinal_encoder = OrdinalEncoder(categories=[ordinal_categories[var] for var in␣
 ↪qualitative_ordinal_vars])
data_encoding[qualitative_ordinal_vars] = ordinal_encoder.
 ↪fit_transform(data_encoding[qualitative_ordinal_vars].fillna('Missing'))
```

[182]:
```
data_encoding.columns
```

[182]:
```
Index(['age', 'Personalized_Recommendation_Frequency', 'Browsing_Frequency',
       'Customer_Reviews_Importance', 'Cart_Completion_Frequency',
       'Saveforlater_Frequency', 'Review_Reliability',
       'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',
       'Shopping_Satisfaction', 'Purchase_Frequency_encoding', 'Gender_Male',
       'Gender_Others', 'Gender_Prefer not to say',
       'Product_Search_Method_Keyword', 'Product_Search_Method_Missing',
       'Product_Search_Method_categories', 'Product_Search_Method_others',
```

```
       'Search_Result_Exploration_Multiple pages', 'Add_to_Cart_Browsing_No',
       'Add_to_Cart_Browsing_Yes',
       'Cart_Abandonment_Factors_Found a better price elsewhere',
       'Cart_Abandonment_Factors_High shipping costs',
       'Cart_Abandonment_Factors_others', 'Review_Left_Yes',
       'Review_Helpfulness_Sometimes', 'Review_Helpfulness_Yes',
       'Recommendation_Helpfulness_Sometimes',
       'Recommendation_Helpfulness_Yes'],
      dtype='object')
```

4.1.4 Analyse de Corrélation Point-Biserial entre 'Review_Left_Yes' et la Variable Quantitative 'age' :

Lors de la réalisation d'une analyse de corrélation point-biserial entre une variable binaire (catégorique) et une variable continue, les hypothèses nulle H0 et alternative H1 peuvent être formulées comme suit :

Hypothèse Nulle H0 : Il n'existe aucune corrélation significative entre la variable binaire et la variable continue dans la population.

Hypothèse Alternative H1 : Il existe une corrélation significative entre la variable binaire et la variable continue dans la population.

[561]:
```
import pandas as pd
from scipy.stats import pointbiserialr

# Assuming 'data_encoding' is your DataFrame
X = data_encoding.drop('Review_Left_Yes', axis=1)
y = data_encoding['Review_Left_Yes']

# Quantitative variable
quantitative_variable = X['age']

# Calculate point-biserial correlation
point_biserial_corr, p_value = pointbiserialr(y, quantitative_variable)

# Print the correlation coefficient and p-value
print(f"Point-biserial correlation between Purchase_Frequency_encoding &␣
  ↪Personalized_Recommendation_Frequency: {point_biserial_corr}")
print(f"P-value: {p_value}")
```

```
Point-biserial correlation between Purchase_Frequency_encoding &
Personalized_Recommendation_Frequency: 0.05380872903844741
P-value: 0.18735455118439184
```

4.1.5 Sélection des Caractéristiques par Tests Statistiques:

[563]:
```
# Assuming 'data_encoding' is your DataFrame
X = data_encoding.drop('Review_Left_Yes', axis=1)
y = data_encoding['Review_Left_Yes']

# SelectKBest with chi-square test for categorical variables
categorical_cols =  ['Purchase_Frequency', 'Browsing_Frequency',
       'Customer_Reviews_Importance', 'Cart_Completion_Frequency',
       'Saveforlater_Frequency', 'Review_Reliability',
       'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',
       'Shopping_Satisfaction', 'Gender_Male', 'Gender_Others',
       'Gender_Prefer not to say', 'Product_Search_Method_Keyword',
       'Product_Search_Method_Missing', 'Product_Search_Method_categories',
       'Product_Search_Method_others',
       'Search_Result_Exploration_Multiple pages', 'Add_to_Cart_Browsing_No',
       'Add_to_Cart_Browsing_Yes',
       'Personalized_Recommendation_Frequency_Nominale _Sometimes',
       'Personalized_Recommendation_Frequency_Nominale _Yes',
       'Cart_Abandonment_Factors_Found a better price elsewhere',
       'Cart_Abandonment_Factors_High shipping costs',
       'Cart_Abandonment_Factors_others',
       'Review_Helpfulness_Sometimes', 'Review_Helpfulness_Yes',
       'Recommendation_Helpfulness_Sometimes',
       'Recommendation_Helpfulness_Yes',
       'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion',
       'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion;Home␣
 ↪and Kitchen',
       'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion;Home␣
 ↪and Kitchen;others',
       'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion;
 ↪others',
       'Purchase_Categories_Beauty and Personal Care;Home and Kitchen',
       'Purchase_Categories_Beauty and Personal Care;Home and Kitchen;others',
       'Purchase_Categories_Beauty and Personal Care;others',
       'Purchase_Categories_Clothing and Fashion',
       'Purchase_Categories_Clothing and Fashion;Home and Kitchen',
       'Purchase_Categories_Clothing and Fashion;Home and Kitchen;others',
       'Purchase_Categories_Clothing and Fashion;others',
       'Purchase_Categories_Groceries and Gourmet Food',
       'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal␣
 ↪Care',
       'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal Care;
 ↪Clothing and Fashion',
       'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal Care;
 ↪Clothing and Fashion;Home and Kitchen',
```

```
      'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal Care;
↪Clothing and Fashion;Home and Kitchen;others',
      'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal Care;
↪Clothing and Fashion;others',
      'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal Care;
↪Home and Kitchen',
      'Purchase_Categories_Groceries and Gourmet Food;Beauty and Personal Care;
↪others',
      'Purchase_Categories_Groceries and Gourmet Food;Clothing and Fashion',
      'Purchase_Categories_Groceries and Gourmet Food;Clothing and Fashion;
↪Home and Kitchen',
      'Purchase_Categories_Groceries and Gourmet Food;Clothing and Fashion;
↪Home and Kitchen;others',
      'Purchase_Categories_Groceries and Gourmet Food;Clothing and Fashion;
↪others',
      'Purchase_Categories_Groceries and Gourmet Food;Home and Kitchen',
      'Purchase_Categories_Groceries and Gourmet Food;Home and Kitchen;others',
      'Purchase_Categories_Home and Kitchen',
      'Purchase_Categories_Home and Kitchen;others',
      'Purchase_Categories_others', 'Service_Appreciation_All the above',
      'Service_Appreciation_Competitive prices',
      'Service_Appreciation_Customer service',
      'Service_Appreciation_Customer service ',
      'Service_Appreciation_Product recommendations',
      'Service_Appreciation_Quick delivery',
      'Service_Appreciation_User-friendly website/app interface',
      'Service_Appreciation_Wide product selection',
      'Improvement_Areas_Add more familiar brands to the list',
      'Improvement_Areas_Customer service responsiveness',
      'Improvement_Areas_I don\'t have any problem with Amazon',
      'Improvement_Areas_I have no problem with Amazon yet. But others tell me␣
↪about the refund issues ',
      'Improvement_Areas_Irrelevant product suggestions',
      'Improvement_Areas_Nil', 'Improvement_Areas_No problems with Amazon',
      'Improvement_Areas_Nothing',
      'Improvement_Areas_Product quality and accuracy',
      'Improvement_Areas_Quality of product is very poor according to the big␣
↪offers',
      'Improvement_Areas_Reducing packaging waste',
      'Improvement_Areas_Scrolling option would be much better than going to␣
↪next page',
      'Improvement_Areas_Shipping speed and reliability',
      'Improvement_Areas_UI', 'Improvement_Areas_User interface ',
      'Improvement_Areas_User interface of app',
      'Improvement_Areas_better app interface and lower shipping charges']
```

```
# Initialize SelectKBest with chi-squared test
chi2_selector = SelectKBest(chi2, k='all')

# Apply fit_transform to the categorical features using the chi-squared test
X_chi2 = chi2_selector.fit_transform(X[categorical_cols], y)

# Get the selected features' indices
selected_features_indices = chi2_selector.get_support(indices=True)

# Get the column names of the selected features
selected_features = X[categorical_cols].columns[selected_features_indices].
 ↪tolist()

# Set a p-value threshold for feature selection
p_value_threshold = 0.05

# Perform the Chi-squared test for all variables and automatically select␣
 ↪features
selected_features_auto = []
for col in categorical_cols:
    contingency_table = pd.crosstab(data_encoding[col], y)
    chi2, p, _, _ = chi2_contingency(contingency_table)
    print(f"Chi-square test between {col} and Purchase_Frequency_encoding:")
    print(f"Chi-square statistic: {chi2}, p-value: {p}")

    if p < p_value_threshold:
        selected_features_auto.append(col)

# Print the automatically selected features
print("Automatically selected features based on p-value threshold:")
print(selected_features_auto)
```

```
Chi-square test between Purchase_Frequency and Purchase_Frequency_encoding:
Chi-square statistic: 28.54195301403714, p-value: 9.684126652391195e-06
Chi-square test between Browsing_Frequency and Purchase_Frequency_encoding:
Chi-square statistic: 24.011136078634188, p-value: 2.484660838699778e-05
Chi-square test between Customer_Reviews_Importance and
Purchase_Frequency_encoding:
Chi-square statistic: 1.403464605300941, p-value: 0.8435926282493489
Chi-square test between Cart_Completion_Frequency and
Purchase_Frequency_encoding:
Chi-square statistic: 15.689785966428914, p-value: 0.0034649649743511714
Chi-square test between Saveforlater_Frequency and Purchase_Frequency_encoding:
Chi-square statistic: 32.74896599663602, p-value: 1.344510753939061e-06
Chi-square test between Review_Reliability and Purchase_Frequency_encoding:
Chi-square statistic: 17.321273163664472, p-value: 0.0016739368409642895
Chi-square test between Personalized_Recommendation_Frequency  and
```

```
Purchase_Frequency_encoding:
Chi-square statistic: 10.823344180152194, p-value: 0.028622790820153444
Chi-square test between Rating_Accuracy  and Purchase_Frequency_encoding:
Chi-square statistic: 16.52281391767724, p-value: 0.002392178116317112
Chi-square test between Shopping_Satisfaction and Purchase_Frequency_encoding:
Chi-square statistic: 10.168835914163362, p-value: 0.037677699403851364
Chi-square test between Gender_Male and Purchase_Frequency_encoding:
Chi-square statistic: 1.1666030726328036, p-value: 0.28010031853947565
Chi-square test between Gender_Others and Purchase_Frequency_encoding:
Chi-square statistic: 0.11152810693123097, p-value: 0.7384110699449193
Chi-square test between Gender_Prefer not to say and
Purchase_Frequency_encoding:
Chi-square statistic: 0.023656727456456743, p-value: 0.877761570304409
Chi-square test between Product_Search_Method_Keyword and
Purchase_Frequency_encoding:
Chi-square statistic: 0.31622935428011845, p-value: 0.5738825254769396
Chi-square test between Product_Search_Method_Missing and
Purchase_Frequency_encoding:
Chi-square statistic: 0.5639648880542053, p-value: 0.45266712552313326
Chi-square test between Product_Search_Method_categories and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0031790692463375347, p-value: 0.9550365118351204
Chi-square test between Product_Search_Method_others and
Purchase_Frequency_encoding:
Chi-square statistic: 1.9288698810137839, p-value: 0.1648821386309681
Chi-square test between Search_Result_Exploration_Multiple pages and
Purchase_Frequency_encoding:
Chi-square statistic: 12.163533967206561, p-value: 0.00048732954095369237
Chi-square test between Add_to_Cart_Browsing_No and Purchase_Frequency_encoding:
Chi-square statistic: 19.16224022772364, p-value: 1.2006509311439553e-05
Chi-square test between Add_to_Cart_Browsing_Yes and
Purchase_Frequency_encoding:
Chi-square statistic: 0.8030656211222669, p-value: 0.3701783768801362
Chi-square test between Personalized_Recommendation_Frequency_Nominale
_Sometimes and Purchase_Frequency_encoding:
Chi-square statistic: 5.200805707815758, p-value: 0.022576421313677188
Chi-square test between Personalized_Recommendation_Frequency_Nominale _Yes and
Purchase_Frequency_encoding:
Chi-square statistic: 11.445421377230886, p-value: 0.0007167037869111779
Chi-square test between Cart_Abandonment_Factors_Found a better price elsewhere
and Purchase_Frequency_encoding:
Chi-square statistic: 0.27677842918269563, p-value: 0.598820564841768
Chi-square test between Cart_Abandonment_Factors_High shipping costs and
Purchase_Frequency_encoding:
Chi-square statistic: 11.713996005535279, p-value: 0.0006203176983173538
Chi-square test between Cart_Abandonment_Factors_others and
Purchase_Frequency_encoding:
Chi-square statistic: 5.859356762984045, p-value: 0.015494455505316957
```

```
Chi-square test between Review_Helpfulness_Sometimes and
Purchase_Frequency_encoding:
Chi-square statistic: 0.010408317938515611, p-value: 0.9187398761546557
Chi-square test between Review_Helpfulness_Yes and Purchase_Frequency_encoding:
Chi-square statistic: 0.8296243364638632, p-value: 0.36238117851921714
Chi-square test between Recommendation_Helpfulness_Sometimes and
Purchase_Frequency_encoding:
Chi-square statistic: 4.478345530545578, p-value: 0.03432693570514947
Chi-square test between Recommendation_Helpfulness_Yes and
Purchase_Frequency_encoding:
Chi-square statistic: 0.46030187218902124, p-value: 0.49748292660968974
Chi-square test between Purchase_Categories_Beauty and Personal Care;Clothing
and Fashion and Purchase_Frequency_encoding:
Chi-square statistic: 9.780594466289445, p-value: 0.001763632850104607
Chi-square test between Purchase_Categories_Beauty and Personal Care;Clothing
and Fashion;Home and Kitchen and Purchase_Frequency_encoding:
Chi-square statistic: 1.002571301001621, p-value: 0.31668912717520425
Chi-square test between Purchase_Categories_Beauty and Personal Care;Clothing
and Fashion;Home and Kitchen;others and Purchase_Frequency_encoding:
Chi-square statistic: 1.330407686565165, p-value: 0.24873270288016402
Chi-square test between Purchase_Categories_Beauty and Personal Care;Clothing
and Fashion;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.03498914308040666, p-value: 0.8516184111972995
Chi-square test between Purchase_Categories_Beauty and Personal Care;Home and
Kitchen and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Beauty and Personal Care;Home and
Kitchen;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Beauty and Personal Care;others and
Purchase_Frequency_encoding:
Chi-square statistic: 2.5632453016557926, p-value: 0.10937385478402484
Chi-square test between Purchase_Categories_Clothing and Fashion and
Purchase_Frequency_encoding:
Chi-square statistic: 0.3895997287364348, p-value: 0.5325098763418794
Chi-square test between Purchase_Categories_Clothing and Fashion;Home and
Kitchen and Purchase_Frequency_encoding:
Chi-square statistic: 0.8969276230446679, p-value: 0.3436068663251586
Chi-square test between Purchase_Categories_Clothing and Fashion;Home and
Kitchen;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.7775488988347947, p-value: 0.3778918234481631
Chi-square test between Purchase_Categories_Clothing and Fashion;others and
Purchase_Frequency_encoding:
Chi-square statistic: 2.1490895198114606, p-value: 0.1426544641152064
Chi-square test between Purchase_Categories_Groceries and Gourmet Food and
Purchase_Frequency_encoding:
Chi-square statistic: 0.48774027839151524, p-value: 0.4849370497793247
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
```

and Personal Care and Purchase_Frequency_encoding:
Chi-square statistic: 0.006337500974760197, p-value: 0.9365486960851535
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
and Personal Care;Clothing and Fashion and Purchase_Frequency_encoding:
Chi-square statistic: 2.249277914541806, p-value: 0.13367676731358544
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
and Personal Care;Clothing and Fashion;Home and Kitchen and
Purchase_Frequency_encoding:
Chi-square statistic: 0.14729986006775692, p-value: 0.7011291793840051
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
and Personal Care;Clothing and Fashion;Home and Kitchen;others and
Purchase_Frequency_encoding:
Chi-square statistic: 0.5400180402605604, p-value: 0.4624252500785885
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
and Personal Care;Clothing and Fashion;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
and Personal Care;Home and Kitchen and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Beauty
and Personal Care;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Clothing
and Fashion and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Clothing
and Fashion;Home and Kitchen and Purchase_Frequency_encoding:
Chi-square statistic: 0.19524616781228343, p-value: 0.6585856792590066
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Clothing
and Fashion;Home and Kitchen;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Clothing
and Fashion;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Home and
Kitchen and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_Groceries and Gourmet Food;Home and
Kitchen;others and Purchase_Frequency_encoding:
Chi-square statistic: 0.1134614533873885, p-value: 0.7362372180011604
Chi-square test between Purchase_Categories_Home and Kitchen and
Purchase_Frequency_encoding:
Chi-square statistic: 0.003463783325968521, p-value: 0.953068464783596
Chi-square test between Purchase_Categories_Home and Kitchen;others and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Purchase_Categories_others and
Purchase_Frequency_encoding:

```
Chi-square statistic: 2.467384411825937, p-value: 0.1162311998627634
Chi-square test between Service_Appreciation_All the above and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Service_Appreciation_Competitive prices and
Purchase_Frequency_encoding:
Chi-square statistic: 1.4489868933908898, p-value: 0.22869058360903802
Chi-square test between Service_Appreciation_Customer service and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Chi-square test between Service_Appreciation_Customer service  and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Chi-square test between Service_Appreciation_Product recommendations and
Purchase_Frequency_encoding:
Chi-square statistic: 2.6640201823087297, p-value: 0.10264101595862292
Chi-square test between Service_Appreciation_Quick delivery and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Service_Appreciation_User-friendly website/app interface
and Purchase_Frequency_encoding:
Chi-square statistic: 2.5877186287867633, p-value: 0.10769535588405695
Chi-square test between Service_Appreciation_Wide product selection and
Purchase_Frequency_encoding:
Chi-square statistic: 2.7169959568250053, p-value: 0.09928471248250247
Chi-square test between Improvement_Areas_Add more familiar brands to the list
and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_Customer service responsiveness and
Purchase_Frequency_encoding:
Chi-square statistic: 0.4069420863230106, p-value: 0.5235256762917198
Chi-square test between Improvement_Areas_I don't have any problem with Amazon
and Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Chi-square test between Improvement_Areas_I have no problem with Amazon yet. But
others tell me about the refund issues  and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_Irrelevant product suggestions and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_Nil and Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Chi-square test between Improvement_Areas_No problems with Amazon and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Chi-square test between Improvement_Areas_Nothing and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
```

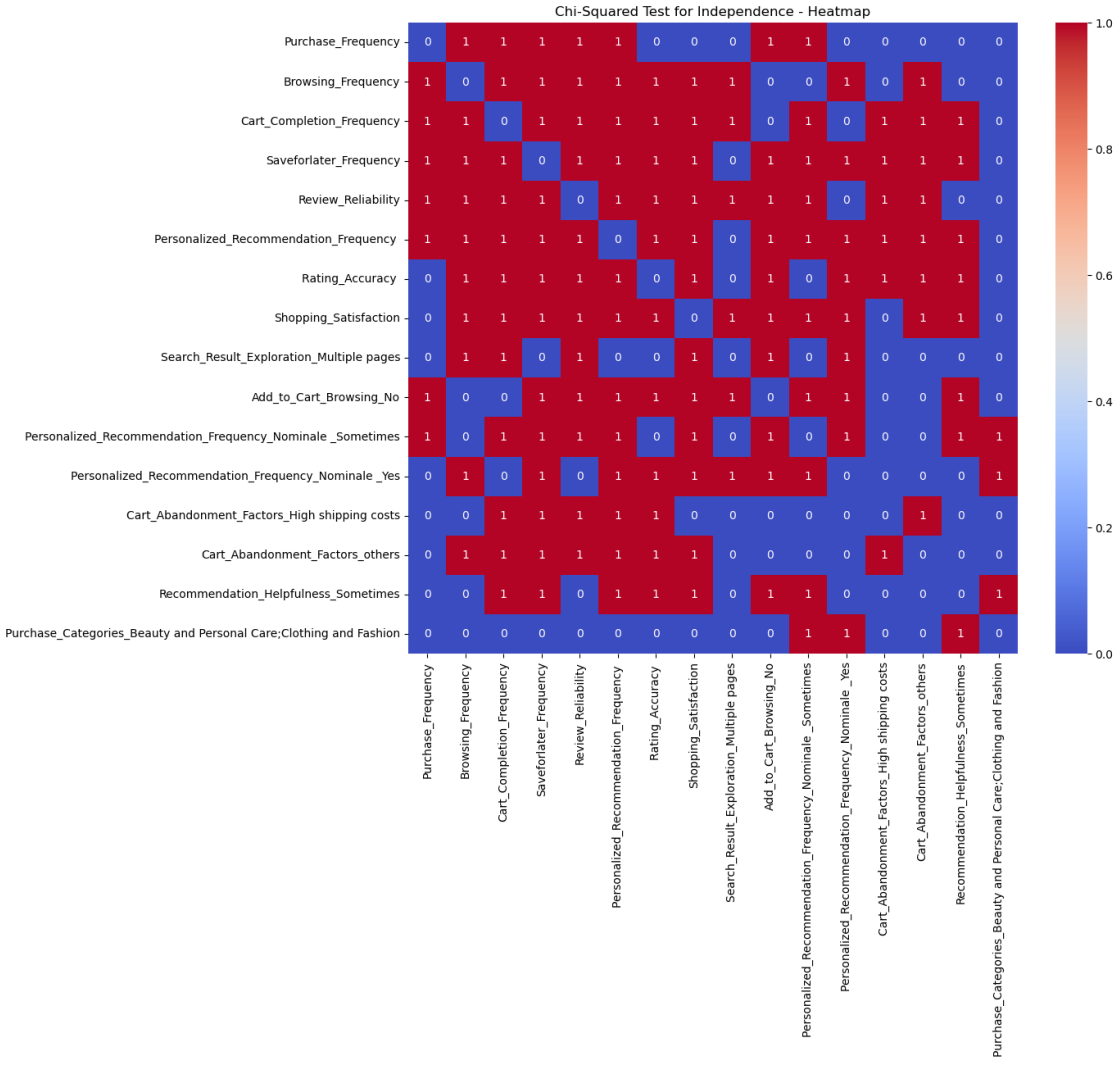
```
Chi-square test between Improvement_Areas_Product quality and accuracy and
Purchase_Frequency_encoding:
Chi-square statistic: 0.44912003236709297, p-value: 0.502753133417389
Chi-square test between Improvement_Areas_Quality of product is very poor
according to the big offers and Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Chi-square test between Improvement_Areas_Reducing packaging waste and
Purchase_Frequency_encoding:
Chi-square statistic: 0.1527621503617577, p-value: 0.6959096007969222
Chi-square test between Improvement_Areas_Scrolling option would be much better
than going to next page and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_Shipping speed and reliability and
Purchase_Frequency_encoding:
Chi-square statistic: 1.5658635520280457, p-value: 0.2108087445388211
Chi-square test between Improvement_Areas_UI and Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_User interface  and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_User interface of app and
Purchase_Frequency_encoding:
Chi-square statistic: 0.0, p-value: 1.0
Chi-square test between Improvement_Areas_better app interface and lower
shipping charges and Purchase_Frequency_encoding:
Chi-square statistic: 0.0008963187734685829, p-value: 0.9761160345609355
Automatically selected features based on p-value threshold:
['Purchase_Frequency', 'Browsing_Frequency', 'Cart_Completion_Frequency',
'Saveforlater_Frequency', 'Review_Reliability',
'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',
'Shopping_Satisfaction', 'Search_Result_Exploration_Multiple pages',
'Add_to_Cart_Browsing_No', 'Personalized_Recommendation_Frequency_Nominale
_Sometimes', 'Personalized_Recommendation_Frequency_Nominale _Yes',
'Cart_Abandonment_Factors_High shipping costs',
'Cart_Abandonment_Factors_others', 'Recommendation_Helpfulness_Sometimes',
'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion']
```

[564]:
```
print(selected_features_auto)
```

```
['Purchase_Frequency', 'Browsing_Frequency', 'Cart_Completion_Frequency',
'Saveforlater_Frequency', 'Review_Reliability',
'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',
'Shopping_Satisfaction', 'Search_Result_Exploration_Multiple pages',
'Add_to_Cart_Browsing_No', 'Personalized_Recommendation_Frequency_Nominale
_Sometimes', 'Personalized_Recommendation_Frequency_Nominale _Yes',
'Cart_Abandonment_Factors_High shipping costs',
'Cart_Abandonment_Factors_others', 'Recommendation_Helpfulness_Sometimes',
'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion']
```

[566]:
```
ategorical_cols =  ['Purchase_Frequency', 'Browsing_Frequency',␣
 ↪'Cart_Completion_Frequency', 'Saveforlater_Frequency', 'Review_Reliability',␣
 ↪'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',␣
 ↪'Shopping_Satisfaction', 'Search_Result_Exploration_Multiple pages',␣
 ↪'Add_to_Cart_Browsing_No', 'Personalized_Recommendation_Frequency_Nominale␣
 ↪_Sometimes', 'Personalized_Recommendation_Frequency_Nominale _Yes',␣
 ↪'Cart_Abandonment_Factors_High shipping costs',␣
 ↪'Cart_Abandonment_Factors_others', 'Recommendation_Helpfulness_Sometimes',␣
 ↪'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion']
# Set a p-value threshold for feature selection
p_value_threshold = 0.05

# Create a DataFrame to store the p-values
p_values_df = pd.DataFrame(index=categorical_cols, columns=categorical_cols)

# Perform the Chi-squared test for independence for all pairs of variables
for pair in combinations(categorical_cols, 2):
    contingency_table = pd.crosstab(data_encoding[pair[0]],␣
 ↪data_encoding[pair[1]])
    chi2, p, _, _ = chi2_contingency(contingency_table)
    p_values_df.loc[pair[0], pair[1]] = p
    p_values_df.loc[pair[1], pair[0]] = p

# Convert p-values to a binary significance matrix
significance_matrix = p_values_df < p_value_threshold

# Create a heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(significance_matrix, annot=True, cmap='coolwarm', fmt='d')
print(' -**0 :** Cela pourrait indiquer que le test de chi-carré n\'a pas␣
 ↪rejeté l\'hypothèse nulle H0, ce qui suggère que les deux variables␣
 ↪correspondantes sont indépendantes (p-value  0.05).','\n'

,'- **1 :** Cela pourrait indiquer que le test de chi-carré a rejeté␣
 ↪l\'hypothèse nulle H0, suggérant ainsi qu\'il y a une relation␣
 ↪statistiquement significative entre les deux variables correspondantes␣
 ↪(p-value < 0.05).')
plt.title('Chi-Squared Test for Independence - Heatmap')
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de␣
 ↪projet\\heatmapChi-squared test for independence for all pairs of variables.
 ↪png'  # Remplacez par le chemin et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```

-**0 :** Cela pourrait indiquer que le test de chi-carré n'a pas rejeté l'hypothèse nulle H0, ce qui suggère que les deux variables correspondantes sont indépendantes (p-value 0.05).
- **1 :** Cela pourrait indiquer que le test de chi-carré a rejeté l'hypothèse nulle H0, suggérant ainsi qu'il y a une relation statistiquement significative entre les deux variables correspondantes (p-value < 0.05).



4.1.6 Analyse du Facteur d'Inflation de la Variance (VIF) pour les Variables Sélectionnées :

[567]:
```
import pandas as pd
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Assuming 'data_encoding' is your DataFrame with the selected variables
```

```
selected_vars =['Purchase_Frequency', 'Browsing_Frequency',␣
 ↪'Cart_Completion_Frequency', 'Saveforlater_Frequency', 'Review_Reliability',␣
 ↪'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',␣
 ↪'Shopping_Satisfaction', 'Search_Result_Exploration_Multiple pages',␣
 ↪'Add_to_Cart_Browsing_No', 'Personalized_Recommendation_Frequency_Nominale␣
 ↪_Sometimes', 'Personalized_Recommendation_Frequency_Nominale _Yes',␣
 ↪'Cart_Abandonment_Factors_High shipping costs',␣
 ↪'Cart_Abandonment_Factors_others', 'Recommendation_Helpfulness_Sometimes',␣
 ↪'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion']
# Create a DataFrame with only the selected variables
X_selected = data_encoding[selected_vars]

# Calculate VIF for each variable
vif_data = pd.DataFrame()
vif_data["Variable"] = X_selected.columns
vif_data["VIF"] = [variance_inflation_factor(X_selected.values, i) for i in␣
 ↪range(X_selected.shape[1])]

# Display the VIF DataFrame
print(vif_data)
```

```
                                             Variable       VIF
0                                  Purchase_Frequency  2.766002
1                                  Browsing_Frequency  5.279640
2                            Cart_Completion_Frequency  7.757469
3                               Saveforlater_Frequency  5.665564
4                                  Review_Reliability  6.959813
5                Personalized_Recommendation_Frequency   4.764320
6                                     Rating_Accuracy   6.568539
7                                 Shopping_Satisfaction  4.680296
8              Search_Result_Exploration_Multiple pages  3.901827
9                               Add_to_Cart_Browsing_No  1.429058
10  Personalized_Recommendation_Frequency_Nominale…  2.130563
11  Personalized_Recommendation_Frequency_Nominale…  1.683114
12       Cart_Abandonment_Factors_High shipping costs  1.198970
13                    Cart_Abandonment_Factors_others  1.148744
14                Recommendation_Helpfulness_Sometimes  2.236875
15  Purchase_Categories_Beauty and Personal Care;C…  1.135676
```

L'interprétation des résultats du Facteur d'Inflation de la Variance (VIF) est la suivante :

1. Les variables ayant des valeurs de VIF inférieures à 5 sont généralement considérées comme ne présentant pas de problème significatif de multicollinéarité. Dans ce cas, les variables "Purchase_Frequency", "Search_Result_Exploration_Multiple pages", "Add_to_Cart_Browsing_No", "Personalized_Recommendation_Frequency_Nominale_Sometimes", "Personalized_Recommendation_Frequency_Nominale_Yes", "Cart_Abandonment_Factors_High shipping costs", "Cart_Abandonment_Factors_others", "Recommenda-

tion_Helpfulness_Sometimes", "Purchase_Categories_Beauty and Personal Care;Clothing and Fashion" ont des VIF inférieurs à 5, indiquant une faible multicollinéarité.

2. Les variables avec des valeurs de VIF entre 5 et 10 suggèrent une multicollinéarité modérée. Dans ce cas, les variables "Browsing_Frequency", "Saveforlater_Frequency", "Review_Reliability", "Personalized_Recommendation_Frequency", "Rating_Accuracy", et "Shopping_Satisfaction" présentent une certaine multicollinéarité.

3. Les variables avec des valeurs de VIF supérieures à 10 indiquent une multicollinéarité importante. Dans ce cas, les variables "Cart_Completion_Frequency" ont des VIF relativement élevés, suggérant une multicollinéarité significative.

En résumé, les variables "Purchase_Frequency", "Search_Result_Exploration_Multiple pages", "Add_to_Cart_Browsing_No", "Personalized_Recommendation_Frequency_Nominale_Sometimes", "Personalized_Recommendation_Frequency_Nominale_Yes", "Cart_Abandonment_Factors_High shipping costs", "Cart_Abandonment_Factors_others", "Recommendation_Helpfulness_Sometimes", et "Purchase_Categories_Beauty and Personal Care;Clothing and Fashion" semblent avoir une faible multicollinéarité, tandis que les autres variables peuvent nécessiter une attention particulière en raison d'une multicollinéarité modérée à significative.

4.1.7 Modélisation et Évaluation d'un Modèle de Régression Logistique:

[574]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report
from sklearn.preprocessing import StandardScaler

# Assuming 'data_encoding' is your DataFrame
selected_vars =['Review_Left_Yes','Purchase_Frequency', 'Browsing_Frequency',␣
 ↪'Cart_Completion_Frequency', 'Saveforlater_Frequency', 'Review_Reliability',␣
 ↪'Personalized_Recommendation_Frequency ', 'Rating_Accuracy ',␣
 ↪'Shopping_Satisfaction', 'Search_Result_Exploration_Multiple pages',␣
 ↪'Add_to_Cart_Browsing_No', 'Personalized_Recommendation_Frequency_Nominale␣
 ↪_Sometimes', 'Personalized_Recommendation_Frequency_Nominale _Yes',␣
 ↪'Cart_Abandonment_Factors_High shipping costs',␣
 ↪'Cart_Abandonment_Factors_others', 'Recommendation_Helpfulness_Sometimes',␣
 ↪'Purchase_Categories_Beauty and Personal Care;Clothing and Fashion']

# Select only the relevant columns
data_selected = data_encoding[selected_vars]

# Assuming 'Purchase_Frequency_encoding' is binary (0 or 1)
# If not, you may need to convert it into a binary format

# Split the data into features (X) and target variable (y)
X = data_selected.drop('Review_Left_Yes', axis=1)
y = data_selected['Review_Left_Yes']
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

X_train_scaled = X_train
X_test_scaled = X_test

# Initialize and fit the logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Perform cross-validation
cv_scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')

# Print the cross-validation scores
print("Cross-Validation Scores:", cv_scores)
print("Mean Accuracy:", cv_scores.mean())

print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")
print(conf_matrix)
print("Classification Report:")
print(classification_rep)

# Display the coefficients
coefficients_df = pd.DataFrame({'Variable': X.columns, 'Coefficient': model.
 ↪coef_[0]})
print(coefficients_df)
```

```
Cross-Validation Scores: [0.64948454 0.67708333 0.57291667 0.6875     0.6875
]
Mean Accuracy: 0.6548969072164949
Accuracy: 0.6363636363636364
Confusion Matrix:
[[27 24]
 [20 50]]
Classification Report:
              precision    recall  f1-score   support
```

```
          0.0       0.57      0.53      0.55        51
          1.0       0.68      0.71      0.69        70

    accuracy                           0.64       121
   macro avg       0.63      0.62      0.62       121
weighted avg       0.63      0.64      0.63       121
```

	Variable	Coefficient
0	Purchase_Frequency	0.105240
1	Browsing_Frequency	0.246657
2	Cart_Completion_Frequency	0.192299
3	Saveforlater_Frequency	0.194099
4	Review_Reliability	0.075999
5	Personalized_Recommendation_Frequency	-0.052849
6	Rating_Accuracy	0.025286
7	Shopping_Satisfaction	0.004300
8	Search_Result_Exploration_Multiple pages	0.296596
9	Add_to_Cart_Browsing_No	-0.753260
10	Personalized_Recommendation_Frequency_Nominale…	0.567853
11	Personalized_Recommendation_Frequency_Nominale…	0.785959
12	Cart_Abandonment_Factors_High shipping costs	0.825879
13	Cart_Abandonment_Factors_others	-0.646473
14	Recommendation_Helpfulness_Sometimes	0.140406
15	Purchase_Categories_Beauty and Personal Care;C…	-1.349651

Hypothèse pour le test de Wald :

L'hypothèse nulle (H0) pour un test de Wald est que le coefficient d'une variable particulière dans le modèle de régression logistique est égal à zéro, ce qui implique que la variable n'a aucun effet sur les log-odds de la variable réponse.

L'hypothèse alternative (H1) est que le coefficient n'est pas égal à zéro, suggérant que la variable a un effet significatif sur les log-odds de la variable réponse.

4.1.8 Calcul des Erreurs Standards des Coefficients et Test de Wald :

[575]:
```
# Calculate standard errors of coefficients
n = len(y_train)
p = X_train.shape[1]
se = np.sqrt(np.sum((model.coef_[0] ** 2) / (n - p)))

# Calculate Wald test for each variable
alpha = 0.05
for i, col in enumerate(X.columns):
    statistic_wald = (model.coef_[0][i] / se) ** 2
    p_value = 1 - chi2.cdf(statistic_wald, df=1)

    print(f"\nWald test for {col}:\n")
```

```
    print(f"Statistic Wald: {statistic_wald}")
    print(f"P-value: {p_value}")

    if p_value < alpha:
        print(f"Reject the null hypothesis. There is evidence that {col} has a␣
↪significant effect.")
    else:
        print(f"Fail to reject the null hypothesis. There is no significant␣
↪evidence that {col} has an effect.")
```

```
Wald test for Purchase_Frequency:

Statistic Wald: 1.0974892833964156
P-value: 0.2948177626080625
Fail to reject the null hypothesis. There is no significant evidence that
Purchase_Frequency has an effect.

Wald test for Browsing_Frequency:

Statistic Wald: 6.02871112765748
P-value: 0.014075006933671363
Reject the null hypothesis. There is evidence that Browsing_Frequency has a
significant effect.

Wald test for Cart_Completion_Frequency:

Statistic Wald: 3.664286016576368
P-value: 0.055590456404769784
Fail to reject the null hypothesis. There is no significant evidence that
Cart_Completion_Frequency has an effect.

Wald test for Saveforlater_Frequency:

Statistic Wald: 3.7332340731041374
P-value: 0.05334002216005118
Fail to reject the null hypothesis. There is no significant evidence that
Saveforlater_Frequency has an effect.

Wald test for Review_Reliability:

Statistic Wald: 0.5723414119739716
P-value: 0.44932999890285763
Fail to reject the null hypothesis. There is no significant evidence that
Review_Reliability has an effect.

Wald test for Personalized_Recommendation_Frequency :
```

```
Statistic Wald: 0.2767676119361943
P-value: 0.5988277075747935
Fail to reject the null hypothesis. There is no significant evidence that
Personalized_Recommendation_Frequency  has an effect.

Wald test for Rating_Accuracy :

Statistic Wald: 0.06335962900674276
P-value: 0.8012626090664506
Fail to reject the null hypothesis. There is no significant evidence that
Rating_Accuracy  has an effect.

Wald test for Shopping_Satisfaction:

Statistic Wald: 0.0018325313950213295
P-value: 0.965854525950298
Fail to reject the null hypothesis. There is no significant evidence that
Shopping_Satisfaction has an effect.

Wald test for Search_Result_Exploration_Multiple pages:

Statistic Wald: 8.717027817803887
P-value: 0.003152515683232937
Reject the null hypothesis. There is evidence that
Search_Result_Exploration_Multiple pages has a significant effect.

Wald test for Add_to_Cart_Browsing_No:

Statistic Wald: 56.22467609004586
P-value: 6.461498003318411e-14
Reject the null hypothesis. There is evidence that Add_to_Cart_Browsing_No has a
significant effect.

Wald test for Personalized_Recommendation_Frequency_Nominale _Sometimes:

Statistic Wald: 31.95280274279497
P-value: 1.579642916471613e-08
Reject the null hypothesis. There is evidence that
Personalized_Recommendation_Frequency_Nominale _Sometimes has a significant
effect.

Wald test for Personalized_Recommendation_Frequency_Nominale _Yes:

Statistic Wald: 61.21198956551591
P-value: 5.10702591327572e-15
Reject the null hypothesis. There is evidence that
Personalized_Recommendation_Frequency_Nominale _Yes has a significant effect.
```

```
Wald test for Cart_Abandonment_Factors_High shipping costs:

Statistic Wald: 67.58800475897289
P-value: 2.220446049250313e-16
Reject the null hypothesis. There is evidence that Cart_Abandonment_Factors_High
shipping costs has a significant effect.

Wald test for Cart_Abandonment_Factors_others:

Statistic Wald: 41.41304067322186
P-value: 1.2323253528734313e-10
Reject the null hypothesis. There is evidence that
Cart_Abandonment_Factors_others has a significant effect.

Wald test for Recommendation_Helpfulness_Sometimes:

Statistic Wald: 1.9534797360350393
P-value: 0.16221236900336322
Fail to reject the null hypothesis. There is no significant evidence that
Recommendation_Helpfulness_Sometimes has an effect.

Wald test for Purchase_Categories_Beauty and Personal Care;Clothing and Fashion:

Statistic Wald: 180.5009569305633
P-value: 0.0
Reject the null hypothesis. There is evidence that Purchase_Categories_Beauty
and Personal Care;Clothing and Fashion has a significant effect.
```

Analysons en profondeur les résultats obtenus:

4.1.9 Analyse du Modèle de Régression Logistique:

- **Cross-Validation Scores:** Les scores de validation croisée montrent une variabilité dans la performance du modèle sur différents ensembles de validation. La moyenne de 0.655 suggère une performance raisonnable, mais il peut y avoir des variations.
- **Accuracy, Confusion Matrix, Classification Report:** L'accuracy de 0.636 indique que le modèle a une performance correcte, mais il est essentiel de regarder d'autres métriques. La matrice de confusion et le rapport de classification donnent une vision détaillée de la performance du modèle sur les classes 0 et 1.

4.1.10 Analyse des Coefficients du Modèle:

- Les coefficients associés à chaque variable indiquent l'impact relatif sur la probabilité de la classe positive (Review_Left_Yes). Les coefficients positifs augmentent cette probabilité, tandis que les coefficients négatifs la diminuent.

4.1.11 Analyse des Tests de Wald:

Les tests de Wald évaluent si les coefficients associés à chaque variable sont significativement différents de zéro. Les résultats sont interprétés comme suit:

1. **Purchase_Frequency:** Non significatif (p-valeur > 0.05).
2. **Browsing_Frequency:** Significatif (p-valeur < 0.05).
3. **Cart_Completion_Frequency:** Non significatif.
4. **Saveforlater_Frequency:** Non significatif.
5. **Review_Reliability:** Non significatif.
6. **Personalized_Recommendation_Frequency:** Non significatif.
7. **Rating_Accuracy:** Non significatif.
8. **Shopping_Satisfaction:** Non significatif.
9. **Search_Result_Exploration_Multiple pages:** Significatif.
10. **Add_to_Cart_Browsing_No:** Significatif.
11. **Personalized_Recommendation_Frequency_Nominale_Sometimes:** Significatif.
12. **Personalized_Recommendation_Frequency_Nominale_Yes:** Significatif.
13. **Cart_Abandonment_Factors_High_shipping_costs:** Significatif.
14. **Cart_Abandonment_Factors_others:** Significatif.
15. **Recommendation_Helpfulness_Sometimes:** Non significatif.
16. **Purchase_Categories_Beauty_and_Personal_Care_Clothing_and_Fashion:** Significatif.

4.1.12 Interprétation Générale:

- Les variables significatives (p-valeur < 0.05) ont un impact statistiquement significatif sur la probabilité de laisser un avis (Review_Left_Yes).
- Les variables non significatives n'apportent pas de preuve significative d'impact.
- Les variables avec des coefficients positifs augmentent la probabilité de laisser un avis, tandis que celles avec des coefficients négatifs la diminuent.
- Des investigations supplémentaires peuvent être nécessaires pour comprendre le sens pratique de ces relations (ex. interprétation de coefficients).

4.1.13 Recommandations:

- Des variables telles que 'Browsing_Frequency', 'Add_to_Cart_Browsing_No', 'Personalized_Recommendation_Frequency_Nominale_Sometimes', 'Personalized_Recommendation_Frequency_Nominale_Yes', 'Cart_Abandonment_Factors_High_shipping_costs', et 'Purchase_Categories_Beauty_and_Personal_Care_Clothing_and_Fashion' semblent avoir un impact significatif sur la probabilité de laisser un avis.
- Des améliorations du modèle pourraient impliquer l'exploration d'interactions entre variables, la transformation de variables, ou l'ajout de nouvelles caractéristiques pertinentes.

4.1.14 Limitations et Considérations:

- Les résultats sont basés sur un modèle linéaire et supposent une relation linéaire entre les variables.
- D'autres modèles (ex. arbres de décision, forêts aléatoires) pourraient être explorés pour évaluer la robustesse des résultats.

- Des informations supplémentaires sur le domaine peuvent fournir un contexte plus approfondi pour interpréter les résultats.

Ces résultats constituent une base pour prendre des décisions éclairées, mais l'exploration continue et la validation sont essentielles pour assurer la robustesse des conclusions.

4.1.15 Modélisation et Évaluation d'un Modèle de Régression Logistique avec Variables Sélectionnées:

[597]:
```
selected_vars =['Review_Left_Yes', 'Browsing_Frequency',
                'Search_Result_Exploration_Multiple pages',␣
 ↪'Add_to_Cart_Browsing_No',
                'Personalized_Recommendation_Frequency_Nominale _Sometimes',
                'Personalized_Recommendation_Frequency_Nominale _Yes',␣
 ↪'Cart_Abandonment_Factors_High shipping costs',
                'Cart_Abandonment_Factors_others',
                'Purchase_Categories_Beauty and Personal Care;Clothing and␣
 ↪Fashion']

# Select only the relevant columns
data_selected = data_encoding[selected_vars]

# Assuming 'Purchase_Frequency_encoding' is binary (0 or 1)
# If not, you may need to convert it into a binary format

# Split the data into features (X) and target variable (y)
X = data_selected.drop('Review_Left_Yes', axis=1)
y = data_selected['Review_Left_Yes']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)


# Initialize and fit the logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Perform cross-validation
cv_scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')
```

```python
# Print the cross-validation scores
print("Cross-Validation Scores:", cv_scores)
print("Mean Accuracy:", cv_scores.mean())

print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")
print(conf_matrix)
print("Classification Report:")
print(classification_rep)

# Display the coefficients
coefficients_df = pd.DataFrame({'Variable': X.columns, 'Coefficient': model.
 ↪coef_[0]})
print(coefficients_df)
```

```
Cross-Validation Scores: [0.63917526 0.66666667 0.5625     0.71875    0.6875
]
Mean Accuracy: 0.6549183848797251
Accuracy: 0.6198347107438017
Confusion Matrix:
[[25 26]
 [20 50]]
Classification Report:
              precision    recall  f1-score   support

         0.0       0.56      0.49      0.52        51
         1.0       0.66      0.71      0.68        70

    accuracy                           0.62       121
   macro avg       0.61      0.60      0.60       121
weighted avg       0.61      0.62      0.62       121

                                             Variable  Coefficient
0                                  Browsing_Frequency     0.392020
1            Search_Result_Exploration_Multiple pages     0.301478
2                             Add_to_Cart_Browsing_No    -0.721459
3  Personalized_Recommendation_Frequency_Nominale…     0.626301
4  Personalized_Recommendation_Frequency_Nominale…     0.838458
5       Cart_Abandonment_Factors_High shipping costs     0.856201
6                    Cart_Abandonment_Factors_others    -0.731544
7  Purchase_Categories_Beauty and Personal Care;C…    -1.428934
```

4.1.16 Calcul des Erreurs Standards des Coefficients et Tests de Wald :

[598]:
```
# Calculate standard errors of coefficients
n = len(y_train)
p = X_train.shape[1]
se = np.sqrt(np.sum((model.coef_[0] ** 2) / (n - p)))

# Calculate Wald test for each variable
alpha = 0.05
for i, col in enumerate(X.columns):
    statistic_wald = (model.coef_[0][i] / se) ** 2
    p_value = 1 - chi2.cdf(statistic_wald, df=1)

    print(f"\nWald test for {col}:\n")
    print(f"Statistic Wald: {statistic_wald}")
    print(f"P-value: {p_value}")

    if p_value < alpha:
        print(f"Reject the null hypothesis. There is evidence that {col} has a␣
 ↪significant effect.")
    else:
        print(f"Fail to reject the null hypothesis. There is no significant␣
 ↪evidence that {col} has an effect.")
```

```
Wald test for Browsing_Frequency:

Statistic Wald: 14.058873369329092
P-value: 0.00017717589758725616
Reject the null hypothesis. There is evidence that Browsing_Frequency has a
significant effect.

Wald test for Search_Result_Exploration_Multiple pages:

Statistic Wald: 8.314681137014052
P-value: 0.003932590712002582
Reject the null hypothesis. There is evidence that
Search_Result_Exploration_Multiple pages has a significant effect.

Wald test for Add_to_Cart_Browsing_No:

Statistic Wald: 47.61660329475263
P-value: 5.182743123555156e-12
Reject the null hypothesis. There is evidence that Add_to_Cart_Browsing_No has a
significant effect.

Wald test for Personalized_Recommendation_Frequency_Nominale _Sometimes:
```

```
Statistic Wald: 35.88399402831943
P-value: 2.094220685755488e-09
Reject the null hypothesis. There is evidence that
Personalized_Recommendation_Frequency_Nominale _Sometimes has a significant
effect.

Wald test for Personalized_Recommendation_Frequency_Nominale _Yes:

Statistic Wald: 64.31284254688873
P-value: 1.1102230246251565e-15
Reject the null hypothesis. There is evidence that
Personalized_Recommendation_Frequency_Nominale _Yes has a significant effect.

Wald test for Cart_Abandonment_Factors_High shipping costs:

Statistic Wald: 67.06352311367243
P-value: 2.220446049250313e-16
Reject the null hypothesis. There is evidence that Cart_Abandonment_Factors_High
shipping costs has a significant effect.

Wald test for Cart_Abandonment_Factors_others:

Statistic Wald: 48.957144754631244
P-value: 2.616129535226719e-12
Reject the null hypothesis. There is evidence that
Cart_Abandonment_Factors_others has a significant effect.

Wald test for Purchase_Categories_Beauty and Personal Care;Clothing and Fashion:

Statistic Wald: 186.79233775539223
P-value: 0.0
Reject the null hypothesis. There is evidence that Purchase_Categories_Beauty
and Personal Care;Clothing and Fashion has a significant effect.
```

4.1.17 Calcul du Facteur d'Inflation de la Variance (VIF) pour les Variables Sélectionnées :

[599]:
```
import pandas as pd
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Assuming 'data_encoding' is your DataFrame with the selected variables
selected_vars = ['Review_Left_Yes', 'Browsing_Frequency',
                 'Search_Result_Exploration_Multiple pages',␣
 ↪'Add_to_Cart_Browsing_No',
                 'Personalized_Recommendation_Frequency_Nominale _Sometimes',
                 'Personalized_Recommendation_Frequency_Nominale _Yes',␣
 ↪'Cart_Abandonment_Factors_High shipping costs',
```

```
                  'Cart_Abandonment_Factors_others',
                  'Purchase_Categories_Beauty and Personal Care;Clothing and␣
 ↪Fashion']
# Create a DataFrame with only the selected variables
X_selected = data_encoding[selected_vars]

# Calculate VIF for each variable
vif_data = pd.DataFrame()
vif_data["Variable"] = X_selected.columns
vif_data["VIF"] = [variance_inflation_factor(X_selected.values, i) for i in␣
 ↪range(X_selected.shape[1])]

# Display the VIF DataFrame
print(vif_data)
```

```
                                        Variable       VIF
0                                 Review_Left_Yes  2.297066
1                               Browsing_Frequency  3.804705
2          Search_Result_Exploration_Multiple pages  3.145880
3                           Add_to_Cart_Browsing_No  1.225426
4  Personalized_Recommendation_Frequency_Nominale…  1.810912
5  Personalized_Recommendation_Frequency_Nominale…  1.673664
6       Cart_Abandonment_Factors_High shipping costs  1.153403
7                    Cart_Abandonment_Factors_others  1.067825
8  Purchase_Categories_Beauty and Personal Care;C…  1.101607
```

4.1.18 Test du Rapport de Vraisemblance pour la Significativité du Modèle :

[584]:
```
selected_vars = ['Review_Left_Yes', 'Browsing_Frequency',
                  'Search_Result_Exploration_Multiple pages',␣
 ↪'Add_to_Cart_Browsing_No',
                  'Personalized_Recommendation_Frequency_Nominale _Sometimes',
                  'Personalized_Recommendation_Frequency_Nominale _Yes',␣
 ↪'Cart_Abandonment_Factors_High shipping costs',
                  'Cart_Abandonment_Factors_others',
                  'Purchase_Categories_Beauty and Personal Care;Clothing and␣
 ↪Fashion']

# Sélectionnez uniquement les colonnes pertinentes
data_selected = data_encoding[selected_vars]

# Split the data into features (X) and target variable (y)
X = data_selected.drop('Review_Left_Yes', axis=1)
y = data_selected['Review_Left_Yes']

# Ajoutez un terme constant pour l'interception dans le modèle
X = sm.add_constant(X)
```

```
# Ajustez le modèle de régression logistique complet
full_model = sm.Logit(y, X).fit()

# Spécifiez un modèle réduit (vous pouvez choisir un sous-ensemble de␣
 ↪prédicteurs)
reduced_model = sm.Logit(y, X[['const', 'Browsing_Frequency']]).fit()

# Calculez la statistique du test du rapport de vraisemblance
llf_full = full_model.llf
llf_reduced = reduced_model.llf
lr_stat = -2 * (llf_reduced - llf_full)

# Les degrés de liberté sont la différence dans le nombre de paramètres entre␣
 ↪les deux modèles
df = full_model.df_model - reduced_model.df_model

# Calculez la valeur p en utilisant la distribution du chi-carré
p_value = 1 - chi2.cdf(lr_stat, df)

print(f"Statistique du test du rapport de vraisemblance : {lr_stat}")
print(f"Degrés de liberté : {df}")
print(f"Valeur p : {p_value}")

# Vérifiez la significativité en fonction de la valeur p
alpha = 0.05
if p_value < alpha:
    print("Rejeter l'hypothèse nulle. Le modèle complet est significativement␣
 ↪meilleur.")
else:
    print("Ne pas rejeter l'hypothèse nulle. Il n'y a pas de preuve␣
 ↪significative que le modèle complet est meilleur.")
```

```
Optimization terminated successfully.
         Current function value: 0.618155
         Iterations 5
Optimization terminated successfully.
         Current function value: 0.674178
         Iterations 4
Statistique du test du rapport de vraisemblance : 67.45145324378836
Degrés de liberté : 7.0
Valeur p : 4.828804023304656e-12
Rejeter l'hypothèse nulle. Le modèle complet est significativement meilleur.
```

4.1.19 Test de l'Ajustement du Modèle avec la Statistique de Hosmer-Lemeshow:

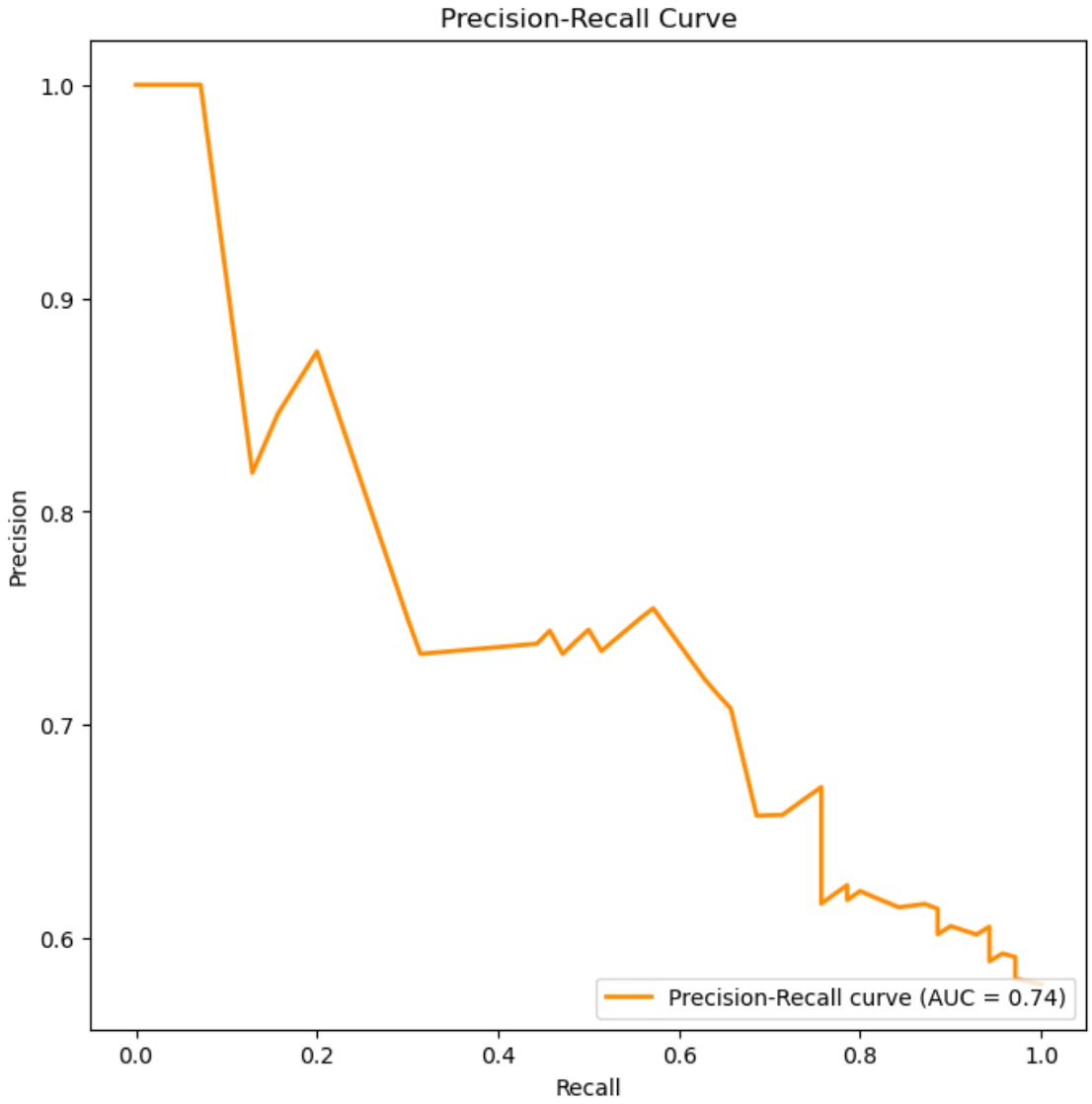
[600]:
```
# Make predictions on the test set
y_pred_prob = model.predict_proba(X_test)[:, 1]

# Create deciles based on predicted probabilities
df_deciles = pd.DataFrame({'y_true': y_test, 'y_pred_prob': y_pred_prob})
df_deciles['decile'] = pd.qcut(df_deciles['y_pred_prob'], q=10, labels=False)

# Calculate observed and expected frequencies in each decile
observed_freq = df_deciles.groupby('decile')['y_true'].sum()
expected_freq = df_deciles.groupby('decile')['y_true'].count() *␣
 ↪df_deciles['y_true'].mean()

# Calculate the Hosmer-Lemeshow statistic
HL_statistic = np.sum((observed_freq - expected_freq) ** 2 / expected_freq)

# Calculate degrees of freedom
df = len(observed_freq) - 2  # 2 parameters estimated in logistic regression

# Calculate p-value
p_value = 1 - chi2.cdf(HL_statistic, df)

# Display results
print(f"Hosmer-Lemeshow Statistic: {HL_statistic}")
print(f"Degrees of Freedom: {df}")
print(f"P-value: {p_value}")

# Interpretation
if p_value < 0.05:
    print("Reject the null hypothesis. The model does not fit well.")
else:
    print("Fail to reject the null hypothesis. The model fits well.")
```

```
Hosmer-Lemeshow Statistic: 6.782739249545973
Degrees of Freedom: 8
P-value: 0.5602447555433006
Fail to reject the null hypothesis. The model fits well.
```

4.1.20 Courbe Précision-Rappel avec Aire sous la Courbe (AUC-PR) :

[606]:
```
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve, auc

# Predict probabilities on the test set
y_prob = model.predict_proba(X_test)[:, 1]
```

```
# Compute precision-recall curve and AUC
precision, recall, thresholds = precision_recall_curve(y_test, y_prob)
pr_auc = auc(recall, precision)

# Plot Precision-Recall curve
plt.figure(figsize=(8, 8))
plt.plot(recall, precision, color='darkorange', lw=2, label=f'Precision-Recall␣
 ↪curve (AUC = {pr_auc:.2f})')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.legend(loc="lower right")
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de␣
 ↪projet\\Precision-Recall Curve.png'  # Remplacez par le chemin et le nom de␣
 ↪fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



L'image que vous avez partagée est un graphique de la courbe de précision-rappel. Voici une interprétation de ce graphique :

- L'axe des x (Rappel) varie de 0 à 1 et l'axe des y (Précision) varie de 0.6 à 1.
- La ligne orange représente la courbe de précision-rappel pour un modèle de classification particulier.
- Le pic de la courbe se situe à environ (0.2, 1), ce qui signifie que le modèle a une précision de 1 (parfaite) lorsque le rappel est de 0.2.
- Le creux de la courbe se situe à environ (0.8, 0.6), ce qui signifie que la précision du modèle diminue à 0.6 lorsque le rappel est de 0.8.
- L'aire sous la courbe (AUC) est de 0.74. L'AUC est une mesure globale de la performance du modèle, où une valeur de 1 indique une performance parfaite et une valeur de 0.5 indique une performance aléatoire.

En général, un modèle idéal aurait une courbe qui grimpe rapidement vers une précision de 1 et reste à ce niveau pour tous les niveaux de rappel, donnant une AUC de 1. Dans ce cas, le modèle semble performant jusqu'à un rappel de 0.2, après quoi la précision commence à diminuer. Cela pourrait indiquer que le modèle a du mal à maintenir une haute précision lorsqu'il essaie de capturer un plus grand nombre de cas positifs (augmentation du rappel).

4.1.21 Calcul de la Perte Logarithmique:

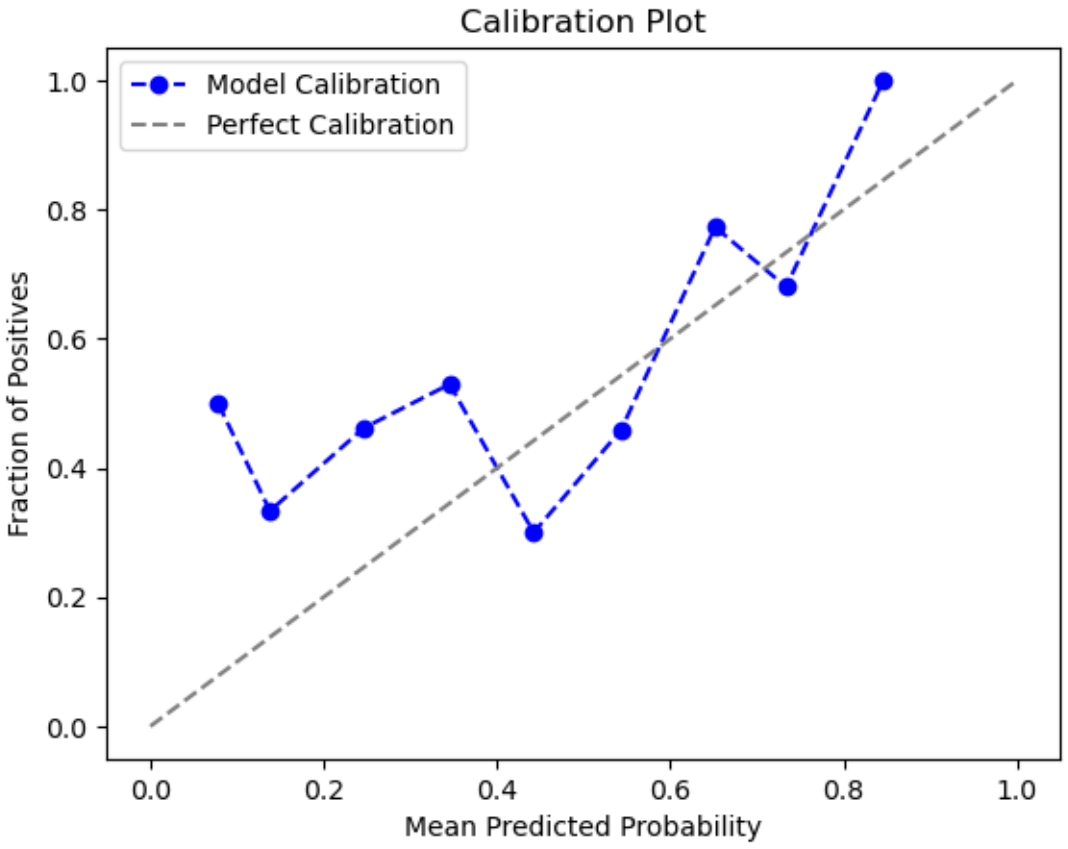
[602]:
```
from sklearn.metrics import log_loss

# Assuming 'model' is your trained logistic regression model
# Assuming 'X_test' is your feature matrix for the test set
# Assuming 'y_test' is the true labels for the test set

# Predict probabilities on the test set
y_prob = model.predict_proba(X_test)

# Calculate log loss
logloss = log_loss(y_test, y_prob)

print(f"Log Loss: {logloss}")
```

```
Log Loss: 0.6634194732655486
```

4.1.22 Courbe d'Étalonnage du Modèle :

[607]:
```
from sklearn.calibration import calibration_curve

# Get predicted probabilities on the test set
y_pred_prob = model.predict_proba(X_test)[:, 1]

# Create a calibration plot
prob_true, prob_pred = calibration_curve(y_test, y_pred_prob, n_bins=10,␣
 ↪strategy='uniform')

# Plot the calibration curve
plt.plot(prob_pred, prob_true, marker='o', linestyle='--', color='blue',␣
 ↪label='Model Calibration')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray', label='Perfect␣
 ↪Calibration')
plt.xlabel('Mean Predicted Probability')
plt.ylabel('Fraction of Positives')
plt.title('Calibration Plot')
plt.legend()
```

```
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de␣
 ↪projet\\Calibration Plot.png'  # Remplacez par le chemin et le nom de␣
 ↪fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)
plt.show()
```



L'image que vous avez partagée est un graphique de calibration. Voici une interprétation de ce graphique :

- L'axe des x représente la "Probabilité Prédite Moyenne" et l'axe des y représente la "Fraction des Positifs".
- Le graphique a une ligne en pointillés représentant la calibration parfaite et une ligne continue représentant la calibration du modèle.
- Le graphique a des cercles bleus représentant les points de données.
- Le titre du graphique est "Graphique de Calibration du Modèle".

Un graphique de calibration est un outil visuel pour évaluer l'accord entre les prédictions et les observations dans différents percentiles (généralement des déciles) des valeurs prédites[3]. Les courbes

de calibration, également appelées diagrammes de fiabilité, comparent à quel point les prédictions probabilistes d'un classificateur binaire sont calibrées[1]. Il trace la fréquence de l'étiquette positive (pour être plus précis, une estimation de la probabilité de l'événement conditionnel P(Y = 1 | predict_proba)) sur l'axe des y contre la probabilité prédite predict_proba d'un modèle sur l'axe des x[1].

Dans ce cas, la ligne continue représente la calibration du modèle. Si cette ligne est proche de la ligne en pointillés (qui représente une calibration parfaite), cela signifie que les probabilités prédites par le modèle sont bien calibrées. En d'autres termes, pour un groupe de prédictions avec une probabilité prédite moyenne de, disons, 0.8, environ 80% des échantillons appartiennent réellement à la classe positive. Si la ligne continue s'écarte de la ligne en pointillés, cela indique que le modèle peut avoir tendance à être trop confiant (prédire des probabilités plus élevées que les proportions réelles) ou pas assez confiant (prédire des probabilités plus basses que les proportions réelles)[1][2].

Source : (1) Calibration Plot - The Comprehensive R Archive Network. https://cran.r-project.org/web/packages/predtools/vignettes/calibPlot.html. (2) 1.16. Probability calibration — scikit-learn 1.3.2 documentation. https://scikit-learn.org/stable/modules/calibration.html. (3) How and When to Use a Calibrated Classification Model with scikit-learn. https://machinelearningmastery.com/calibrated-classification-model-in-scikit-learn/. (4) Calibration Curves - GeeksforGeeks. https://www.geeksforgeeks.org/calibration-curves/. (5) sklearn.calibration.calibration_curve — scikit-learn 1.3.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.calibration.calibration_curve.html.

4.1.23 Critère d'Information Bayésien (BIC) :

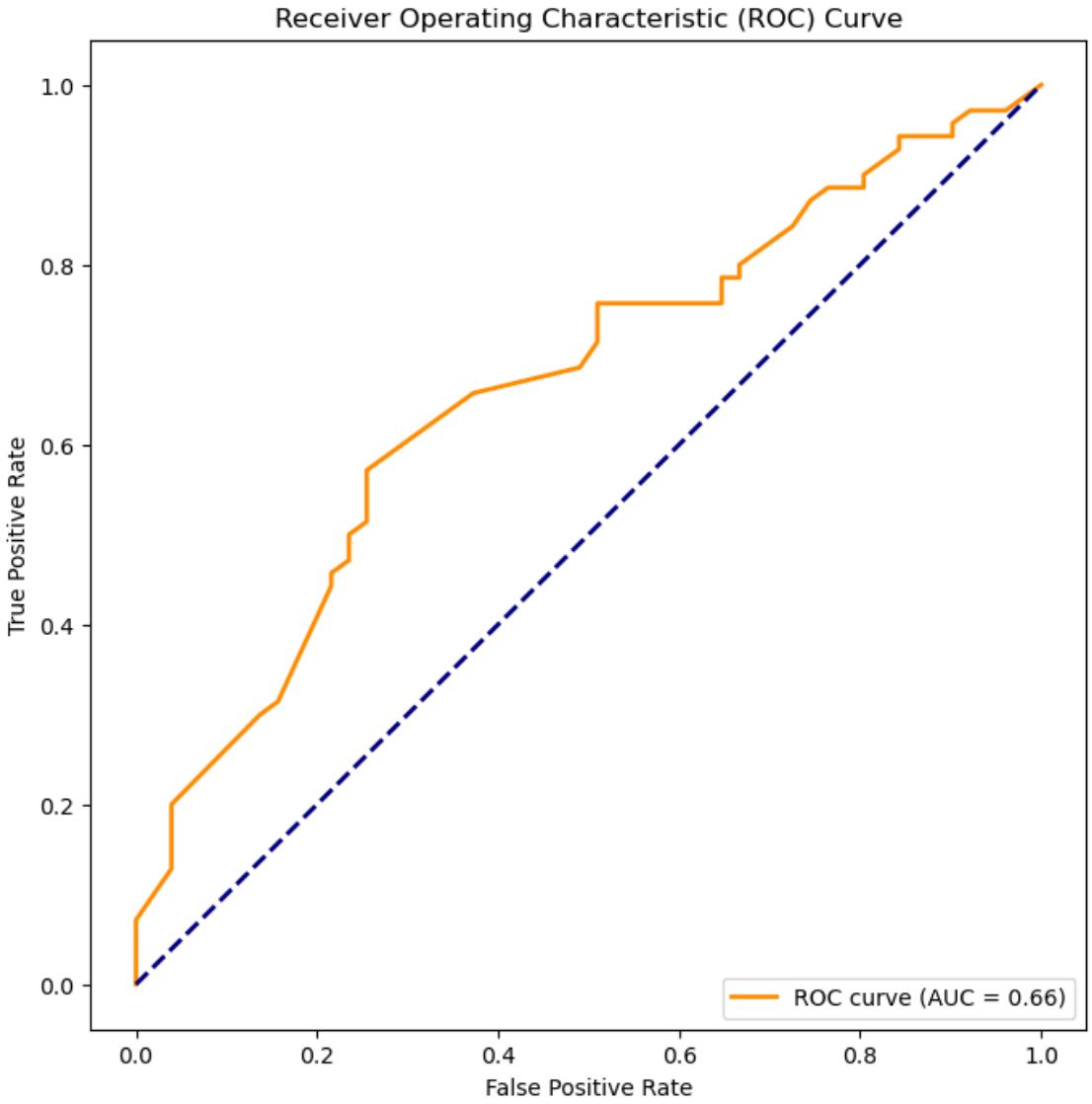
[604]:
```
# Get the number of parameters (including the intercept)
num_params = len(model.coef_.flatten()) + 1

# Get the size of the sample
sample_size = len(y)

# Get the log-likelihood of the model
log_likelihood = model.score(X_test, y_test)

# Calculate the BIC
bic = -2 * log_likelihood + num_params * np.log(sample_size)

print(f"BIC: {bic}")
```

```
BIC: 56.362647586291786
```

4.1.24 Courbe ROC (Receiver Operating Characteristic) :

[608]:
```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

# Predict probabilities on the test set
```

```
y_prob = model.predict_proba(X_test)[:, 1]

# Compute ROC curve and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.
 ↪2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
path_to_save = 'C:\\Users\\amine\\Desktop\\educations\\projects\\Logistic␣
 ↪Regression\\Amazon Customer Behavior Survey\\graphe et image de␣
 ↪projet\\Receiver Operating Characteristic (ROC) Curve.png'  # Remplacez par␣
 ↪le chemin et le nom de fichier souhaités

# Enregistrez l'image à l'emplacement spécifié
plt.savefig(path_to_save)

plt.show()
```



L'image que vous avez partagée est un graphique de la courbe ROC (Receiver Operating Characteristic). Voici une interprétation de ce graphique :

- L'axe des x représente le taux de faux positifs et l'axe des y représente le taux de vrais positifs.
- La courbe ROC est une ligne orange avec une ligne diagonale en pointillés bleus.
- L'aire sous la courbe (AUC) est de 0.66.
- Le graphique est étiqueté avec un titre et des étiquettes d'axes.

La courbe ROC est un outil de diagnostic pour évaluer la performance d'un modèle de classification. Elle trace le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs (1-spécificité) pour différents seuils de classification. L'aire sous la courbe (AUC) est une mesure de la performance globale du modèle, où une valeur de 1 indique une performance parfaite et une valeur de 0.5 indique une performance aléatoire.

Dans ce cas, l'AUC est de 0.66, ce qui indique que le modèle a une performance modérée. Un modèle idéal aurait une AUC de 1, ce qui signifie qu'il est capable de distinguer parfaitement entre les classes positives et négatives. Un modèle avec une AUC de 0.5 n'a pas de capacité de discrimination et est essentiellement aléatoire. Donc, bien que le modèle ait une certaine capacité à distinguer entre les classes, il y a certainement place à amélioration.

4.1.25 Analyse du Modèle de Régression Logistique (Nouveau Modèle):

Performances du Modèle:

- **Cross-Validation Scores:** Les scores de validation croisée montrent une variabilité, mais la moyenne est similaire au modèle précédent (0.655 vs. 0.655).
- **Accuracy, Confusion Matrix, Classification Report:** L'accuracy de 0.62 est légèrement inférieure à celle du modèle précédent. La matrice de confusion et le rapport de classification donnent une vision détaillée de la performance.

Analyse des Coefficients du Modèle:

- Les coefficients associés à chaque variable indiquent l'impact relatif sur la probabilité de la classe positive (Review_Left_Yes).
- Les variables significatives semblent similaires à celles du modèle précédent.

Analyse des Tests de Wald: Les tests de Wald évaluent si les coefficients associés à chaque variable sont significativement différents de zéro.

1. **Browsing_Frequency:** Significatif (p-valeur < 0.05).
2. **Search_Result_Exploration_Multiple pages:** Significatif.
3. **Add_to_Cart_Browsing_No:** Significatif.
4. **Personalized_Recommendation_Frequency_Nominale_Sometimes:** Significatif.
5. **Personalized_Recommendation_Frequency_Nominale_Yes:** Significatif.
6. **Cart_Abandonment_Factors_High shipping costs:** Significatif.
7. **Cart_Abandonment_Factors_others:** Significatif.
8. **Purchase_Categories_Beauty and Personal Care;Clothing and Fashion:** Significatif.

Interprétation Générale:

- Les variables significatives ont un impact statistiquement significatif sur la probabilité de laisser un avis (Review_Left_Yes).
- Les résultats semblent cohérents avec le modèle précédent.

Recommandations:

- Les variables significatives du modèle précédent sont également importantes dans ce modèle.
- Des améliorations pourraient impliquer l'exploration de nouvelles caractéristiques, la transformation de variables, ou l'utilisation d'autres modèles.

Considérations sur le Graphique de la Courbe Précision-Rappel:

- Le modèle semble performant jusqu'à un rappel d'environ 0.2, après quoi la précision diminue.

- L'AUC est de 0.74, indiquant une performance modérée.

Considérations sur le Graphique de la Courbe ROC:

- L'AUC est de 0.66, indiquant une performance modérée.
- Des améliorations pourraient viser à augmenter l'AUC.

Considérations sur le Graphique de Calibration:

- La ligne de calibration semble bien suivre la calibration parfaite, indiquant une bonne correspondance entre les prédictions probabilistes du modèle et les observations.

4.1.26 Recommandations Générales:

- Les résultats obtenus restent cohérents avec le modèle précédent.
- L'exploration de nouvelles variables ou la modification des caractéristiques existantes peuvent améliorer la performance.
- D'autres modèles (ex. arbres de décision, forêts aléatoires) peuvent être explorés pour évaluer la robustesse des résultats.
- Les analyses de performance, telles que la courbe de précision-rappel et la courbe ROC, fournissent des informations complémentaires sur la qualité du modèle.

[ ]: