Report

Introduction

Ce rapport présente l'analyse et le traitement des données démographiques et de dépenses des utilisateurs pour l'entreprise Sell4All. Le projet consiste à explorer, nettoyer, analyser et visualiser les données pour préparer un ensemble de données propre, utilisable pour des fonctionnalités d'IA sur le site de l'entreprise.

Étape 1 : Chargement et Conversion des Données

- **Lecture du fichier Excel :**
  - Le fichier `data-sell4all.xlsx` est lu dans un DataFrame Pandas. Ce fichier contient les données brutes sur les utilisateurs de Sell4All.
- **Conversion en fichier CSV :**
  - Le DataFrame est converti en un fichier CSV nommé `data-sell4all.csv`. Cette conversion facilite les opérations de traitement et d'analyse ultérieures.

```
#  'data-sell4all.xlsx'
file_path = 'data-sell4all.xlsx'

# Read the Excel file
df = pd.read_excel(file_path)

# Convert the DataFrame to a CSV file
csv_file_path = 'data-sell4all.csv'
df.to_csv(csv_file_path, index=False)
```

Étape 2 : Exploration des Données

- **Lecture du fichier CSV :**
  - Le fichier CSV `data-sell4all.csv` est chargé dans un DataFrame pour une première exploration.

```
# Lire le fichier CSV
df = pd.read_csv('data-sell4all.csv')
```

- **Affichage des 5 premières lignes :**
  - Les cinq premières lignes du DataFrame sont affichées pour un aperçu rapide des données. Cela permet de vérifier la structure et le contenu des données.

```
# Afficher les 5 premières lignes du DataFrame
print(df.head())
```

```
      Pays  Age    Genre Dépenses des clients
0   France    32  Female                150.50
1  Germany    45    Male                200.75
2    Spain    28  Female                 75.25
3    Italy    39    Male                180.00
4       UK    52  Female                250.30
```

- **Résumé Technique :**
  - Un résumé technique (`df.info()`) est généré pour donner un aperçu complet des données, y compris :
    - **Nombre de lignes :** Indique le nombre total d'enregistrements (utilisateurs) présents dans le dataset.
    - **Colonnes :** Liste les noms des colonnes, représentant les attributs de chaque utilisateur.
    - **Types de données :** Indique les types de données associés à chaque colonne (par exemple, `int64` pour les entiers, `float64` pour les nombres décimaux, `object` pour les chaînes de caractères).

```
# Afficher un résumé technique des données
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65 entries, 0 to 64
Data columns (total 4 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Pays                  65 non-null     object
 1    Age                  62 non-null     object
 2   Genre                 64 non-null     object
 3   Dépenses des clients  63 non-null     object
dtypes: object(4)
memory usage: 2.2+ KB
None
```

```
print(df.columns)
```

```
Index(['Pays', ' Age ', 'Genre', 'Dépenses des clients'], dtype='object')
```

Étape 3 : Nettoyage des Données

- **Conversion de la Colonne "Âge" en Numérique :**
  - La colonne "Âge" est convertie en format numérique. Les valeurs non convertibles sont remplacées par `NaN`. Les lignes contenant des valeurs `NaN` sont ensuite supprimées pour garantir la validité des données.

```
# Nettoyer la colonne "Age"
df[' Age '] = pd.to_numeric(df[' Age '], errors='coerce')  # Convertir en numérique, les erreurs seront remplacées par NaN

# Supprimer les lignes où l'âge est manquant ou invalide
df = df.dropna(subset=[' Age '])
```

- **Nettoyage de la Colonne "Dépenses des Clients" :**
  - De même, la colonne "Dépenses des clients" est nettoyée en convertissant les valeurs en format numérique. Les erreurs sont traitées en remplaçant les valeurs invalides par `NaN`, puis en supprimant les lignes correspondantes.

```
# Nettoyer la colonne "Dépenses des clients"
df['Dépenses des clients'] = pd.to_numeric(df['Dépenses des clients'], errors='coerce')

# Supprimer les lignes où les dépenses des clients sont manquantes ou invalides
df = df.dropna(subset=['Dépenses des clients'])
```

- **Calcul des Statistiques :**
  - Après le nettoyage, les statistiques suivantes sont calculées :
    - **Médiane de l'âge :** Le point médian de la distribution des âges des utilisateurs restants.
    - **Médiane des dépenses des clients :** Le point médian des dépenses effectuées par les clients.
    - **Moyenne de l'âge :** L'âge moyen des utilisateurs après nettoyage.
    - **Moyenne des dépenses des clients :** La dépense moyenne des clients restants.

```python
# Recalculer la médiane et la moyenne après le nettoyage
median_age = df[' Age '].median()
median_spending = df['Dépenses des clients'].median()

mean_age = df[' Age '].mean()
mean_spending = df['Dépenses des clients'].mean()

print(f'Médiane de l\'âge: {median_age}')
print(f'Médiane des dépenses des clients: {median_spending}')
print(f'Moyenne de l\'âge: {mean_age}')
print(f'Moyenne des dépenses des clients: {mean_spending}')
```

```
Médiane de l'âge: 36.5
Médiane des dépenses des clients: 167.5
Moyenne de l'âge: 54.232142857142854
Moyenne des dépenses des clients: 18014.616964285717
```

Étape 4 : Visualisation des Données

- **Dépenses des Clients par Pays :**
  - Un graphique à barres est créé pour visualiser les dépenses totales des clients par pays. Ce graphique permet de comprendre la répartition des dépenses en fonction des différentes localisations géographiques des utilisateurs.

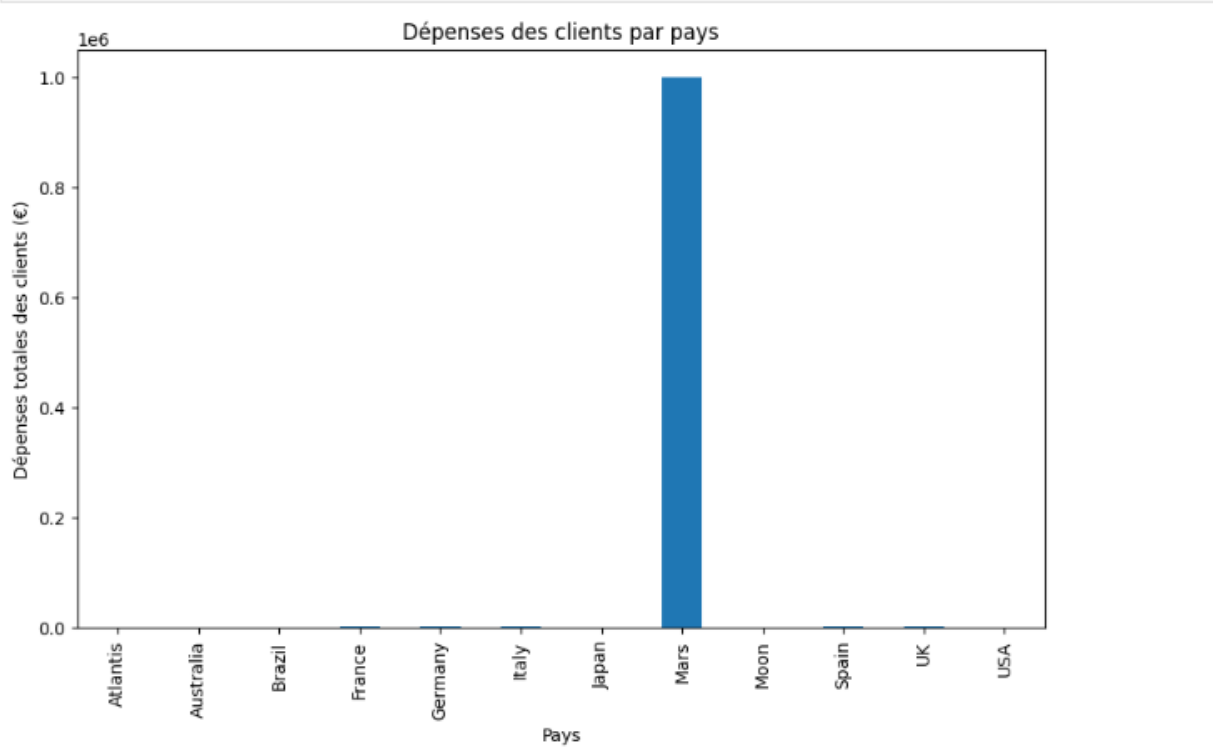
```python
import matplotlib.pyplot as plt

# Calculer les dépenses totales par pays
spending_by_country = df.groupby('Pays')['Dépenses des clients'].sum()

# Créer un graphique à barres
plt.figure(figsize=(10, 6))
spending_by_country.plot(kind='bar')
plt.title('Dépenses des clients par pays')
plt.xlabel('Pays')
plt.ylabel('Dépenses totales des clients (€)')
plt.show()
```

Résultats :

Le graphique révèle les pays où les clients dépensent le plus. Cela peut aider l'entreprise à cibler des campagnes marketing spécifiques ou à explorer de nouveaux marchés.



- • Le graphique montre une disparité significative dans les dépenses des clients par pays.
- • Le pays (ou entité géographique) "Mars" se distingue par des dépenses extrêmement élevées, atteignant près d'un million d'euros, tandis que les autres pays affichent des dépenses négligeables en comparaison.

Étape 5 : Nettoyage Avancé

- **Suppression des Lignes avec des Dépenses Faibles :**
  - Les utilisateurs ayant dépensé moins de 10 € sont supprimés du dataset. Cette étape élimine les transactions mineures qui pourraient fausser l'analyse.

```
# Supprimer les lignes avec des dépenses client inférieures à 10 €
df = df[df['Dépenses des clients'] >= 10]
```

- **Suppression des Doublons :**
  - Les doublons dans le dataset sont supprimés pour garantir l'unicité des enregistrements, essentielle pour une analyse précise.

```
# Supprimer les doublons
df = df.drop_duplicates()
```

Étape 6 : Enregistrement des Données Nettoyées

- **Sauvegarde des Données Nettoyées :**
  - Les données nettoyées, avec les colonnes "Pays", "Âge", "Genre" et "Dépenses des clients", sont exportées dans un nouveau fichier CSV nommé `data-sell4all-cleaned.csv`.

Résultats :

- Le fichier CSV final contient un ensemble de données propre et structuré, prêt à être utilisé pour des analyses plus poussées ou pour alimenter des algorithmes d'IA.