Pandas Seaborn

Introduction à Pandas

Pandas est une bibliothèque de manipulation de données qui fournit des structures de données comme **DataFrame** et **Series** pour travailler efficacement avec des données étiquetées.

Charger des données avec Pandas

Pandas permet de lire facilement des fichiers CSV, Excel, et autres formats courants.

Exemple :

```
import pandas as pd
# Charger un fichier CSV
df = pd.read_csv("data.csv")
```

Filtrage des Données avec Pandas

Le préprocessing des données commence souvent par un **filtrage des données** : enlever les lignes avec des valeurs manquantes, filtrer des colonnes spécifiques, ou exclure des données non pertinentes.

1. Supprimer les lignes avec des valeurs manquantes

```
# Supprimer les lignes contenant des valeurs manquantes
df = df.dropna()
```

2. Filtrer les données selon des critères spécifiques

Parfois, il est nécessaire de garder seulement certaines lignes de données selon un critère spécifique (ex. : garder seulement les personnes de plus de 30 ans).

```
# Filtrer les lignes où l'âge est supérieur à 30
df_filtered = df[df['age'] > 30]
```

3. Sélectionner des colonnes spécifiques

Souvent, tu n'as besoin que de certaines colonnes pour ton modèle.

```
# Garder uniquement certaines colonnes
df = df[['age', 'revenu_estime_mois', 'niveau_etude']]
```

4. Remplacer les valeurs manquantes par une valeur par défaut

Si tu veux garder des lignes avec des valeurs manquantes, mais les remplacer par une valeur par défaut (par exemple, la moyenne).

```
# Remplacer les valeurs manquantes par la moyenne de la colonne
df['revenu_estime_mois'].fillna(df['revenu_estime_mois'].mean(), inplace=True)
```

Fonction importante à explorer soi-même

- `df.describe`
- `df.info`

Seaborn pour la Visualisation et le Préprocessing des Données

Seaborn est une bibliothèque de visualisation qui fonctionne bien avec Pandas. Elle permet de créer facilement des graphiques pour explorer les données et mieux comprendre leur structure avant d'entraîner un modèle.

Visualiser la distribution des données

La visualisation des données permet de détecter des anomalies, des outliers, et d'avoir un aperçu général des distributions.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Visualisation de la distribution des âges
sns.histplot(df['age'], kde=True)
plt.show()
```

Boxplot pour détecter les outliers

Les **boxplots** sont utiles pour détecter des **outliers** dans les données, comme des valeurs anormalement élevées ou faibles.

```
# Boxplot pour détecter les outliers dans les revenus
sns.boxplot(x=df['revenu_estime_mois'])
plt.show()
```

Nettoyage des Données avec Pandas et Seaborn

Avant de passer à l'entraînement du modèle, il est essentiel de nettoyer les données : suppression des doublons, gestion des valeurs aberrantes, et traitement des variables catégorielles.

1. Supprimer les doublons

Il est important de s'assurer que les données ne contiennent pas de doublons, ce qui pourrait fausser les résultats.

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

2. Traiter les outliers

Les **outliers** peuvent avoir un impact négatif sur les performances des modèles d'IA. Un moyen courant de les détecter est d'utiliser les **boxplots**, puis de les filtrer.

```
# Filtrer les outliers (par exemple, couper les revenus à plus de 3 fois l'écart interquartile)
Q1 = df['revenu_estime_mois'].quantile(0.25)
Q3 = df['revenu_estime_mois'].quantile(0.75)
IQR = Q3 - Q1
df = df[(df['revenu_estime_mois'] >= (Q1 - 1.5 * IQR)) & (df['revenu_estime_mois'] <= (Q3 + 1.5 * IQR))]
```

3. Traitement des variables catégorielles

Les modèles de machine learning nécessitent souvent que les variables catégorielles soient **transformées en variables numériques**. Seaborn est utile pour visualiser les relations entre ces variables avant de les transformer.

```
# Conversion des variables catégorielles en variables numériques avec "Label Encoding"
df['niveau_etude'] = df['niveau_etude'].map({
    'aucun': 0,
    'bac': 1,
    'bac+2': 2,
    'master': 3,
    'doctorat': 4
})
```