

REMARQUE : L'objectif se concentre légèrement plus sur l'analyse de la satisfaction client et la performance des produits plutôt que sur la gestion directe des fournisseurs. Plus tard, nous pourrions aborder les insights liés à la gestion des fournisseurs si cela devient crucial, en augmentant le jeu de données Kaggle avec des données synthétiques sur les stocks et les ventes.

Pour l'instant, travaillons avec :

<https://www.kaggle.com/datasets/promptcloud/walmart-product-review-dataset>

1. Exercice de Clustering :

Objectif :

Regrouper les produits en fonction de leurs performances en ventes pour identifier des articles à haute, moyenne et faible performance.

Affinage du jeu de données (si nécessaire) :

- **Utiliser un sous-ensemble d'environ 500 lignes** du jeu de données Kaggle Walmart contenant les colonnes suivantes :
 - Product_ID, Category, Price, Quantity_Sold, Inventory_Level.
- **Prétraitement des données :**
 - S'assurer qu'il n'y ait **aucune valeur manquante**.
 - Normaliser les colonnes numériques (par exemple, Price, Quantity_Sold) pour garantir des valeurs comparables (si cela fait du sens).

Instructions pas-à-pas avec un des outils :

Étapes :

1. **Charger le jeu de données dans un outil de clustering** (Orange ou Weka).
2. **Sélectionner les colonnes** Price, Quantity_Sold et Inventory_Level comme caractéristiques pour le clustering.
3. **Appliquer l'algorithme k-Means** avec $k=3$ pour regrouper les produits en trois clusters.
4. **Visualiser les clusters** à l'aide d'un graphique 2D (Orange/Weka), en affichant les relations entre deux caractéristiques comme Price et Quantity_Sold.

Résultat Attendu :

1. Visualisation des Clusters :

- Un graphique avec trois clusters distincts, représentant des articles à haute, moyenne et faible performance.

2. Résumé des Clusters : Exemple de tableau de synthèse :

	Cluster	Prix Moyen	Quantité Vendue Moyenne	Niveau de Stock Moyen
	Haut	200 \$	500	50
	Moyen	100 \$	250	30
	Bas	50 \$	100	20

Ou

Python

Instructions Étape par Étape

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt

# Étape 1 : Charger le Dataset
# Étape 2 : Prétraitement des Données
# Étape 3 : Appliquer k-Means Clustering
# Étape 4 : Associer les Clusters à des Labels
# Étape 5 : Visualisation des Clusters
# Étape 6 : Résumé des Clusters
```

1. Charger le Dataset

2. Prétraitement des Données :

- Seules les colonnes numériques (Price, Quantity_Sold, Inventory_Level) sont sélectionnées pour le clustering.
- Les données sont normalisées avec MinMaxScaler pour garantir un clustering équitable.

3. Clustering :

- L'algorithme k-Means est appliqué avec k=3 clusters pour regrouper les produits en trois catégories de performance : Élevé, Moyen, Faible.

4. Associer les Clusters à des Labels :

- Les clusters sont mappés à des étiquettes de performance (par ex., Faible, Moyen, Élevé).

5. Visualisation :

- Un diagramme de dispersion montre les clusters dans un espace 2D, mettant en évidence la relation entre Price et Quantity_Sold.

6. Résumé des Clusters :

- Les moyennes de Price, Quantity_Sold, et Inventory_Level pour chaque cluster sont calculées et affichées dans un tableau.

Sorties Attendues

Diagramme de Dispersion :

- Points regroupés en trois clusters, étiquetés comme produits Élevés, Moyens, ou Faibles en performance.

Tableau de Résumé des Clusters :

Exemple de sortie :

Cluster_Label	Avg_Price	Avg_Quantity_Sold	Avg_Inventory_Level
Élevé	\$350	700	400
Moyen	\$150	400	250
Faible	\$50	150	100

Modifier ou Étendre :

- Modifier le nombre de clusters (k) pour observer l'effet sur le regroupement.
- Ajouter des fonctionnalités supplémentaires (par ex., Category) pour explorer le clustering multidimensionnel.
- Visualiser les clusters en utilisant différentes paires de fonctionnalités.

Livrables Attendus

1. Diagramme de Dispersion :

- Points regroupés en clusters étiquetés comme Élevé, Moyen, et Faible.

2. Tableau de Résumé des Clusters :

- Tableau indiquant les moyennes de Price, Quantity_Sold, et Inventory_Level pour chaque cluster.