Activité 1 : Analyse de Données GMAO et Reporting Décisionnel

Contexte Professionnel

Vous êtes technicien de maintenance dans une usine de production. Votre responsable vous demande d'analyser les interventions de maintenance de l'année 2024 pour identifier les machines problématiques et optimiser le plan de maintenance.

Le fichier `interventions_2024.csv` contient **5000 interventions** extraites de la GMAO (Gestion de Maintenance Assistée par Ordinateur), mais les données sont brutes et contiennent des erreurs de saisie.

Objectifs Pédagogiques

1. Apprendre à utiliser l'IA pour nettoyer et normaliser des données
2. Calculer des indicateurs clés de maintenance (MTBF, MTTR)
3. Générer du code Python pour visualiser les données
4. Produire un rapport de synthèse pour le management

Durée

60 minutes

Livrables Attendus

1. Un script Python fonctionnel (`analyse_gmao.py`)
2. Des graphiques de visualisation (PNG ou dans le script)
3. Un rapport de synthèse au format Markdown ou PDF

Étapes de Travail

Étape 1 : Préparation (5 min)

1. **Générer les données** :

   ```
   python generer_donnees_gmao.py
   ```

   Cela va créer le fichier `interventions_2024.csv`.

2. **Examiner les données** :

   - Ouvrez le fichier CSV dans Excel ou un éditeur de texte
   - Identifiez les problèmes de qualité des données

Étape 2 : Nettoyage des Données avec l'IA (15 min)

Objectif : Normaliser les données sales.

Prompt suggéré pour Gemini :

```
J'ai un fichier CSV d'interventions de maintenance avec les colonnes :
- ID_Intervention, Date, ID_Machine, Type_Panne, Duree_Arret_h, Technicien, Pieces_Changees

Les données contiennent des problèmes :
- Fautes de frappe dans Type_Panne (ex: "Panne Mecanique" au lieu de "Panne Mécanique")
- Formats de dates hétérogènes (DD/MM/YYYY, YYYY-MM-DD, etc.)
- Formats de durées variés (3.5, 3h30, 3:30, etc.)
- Noms de techniciens incohérents (M. Dupont, Martin Dupont, etc.)
- Données manquantes

Peux-tu me générer le code Python avec pandas pour :
1. Charger le fichier "interventions_2024.csv"
2. Normaliser tous ces champs
3. Gérer les valeurs manquantes
4. Sauvegarder le fichier nettoyé "interventions_2024_clean.csv"
```

À faire : - Copier le code généré par Gemini - L'exécuter et vérifier le résultat - Ajuster si nécessaire

Étape 3 : Calcul des Indicateurs (15 min)

Indicateurs à calculer :

- **MTBF** (Mean Time Between Failures) : Temps moyen entre pannes pour chaque machine
- **MTTR** (Mean Time To Repair) : Temps moyen de réparation
- **Nombre total de pannes par machine**
- **Temps d'arrêt total par machine**

Prompt suggéré :

```
À partir du fichier nettoyé "interventions_2024_clean.csv",
peux-tu enrichir le script Python pour calculer :

1. Pour chaque machine (ID_Machine) :
   - Nombre total d'interventions
   - MTTR : durée moyenne des arrêts (en heures)
   - Temps d'arrêt cumulé sur l'année
   - MTBF : temps moyen entre pannes (en jours, sachant que l'année = 365 jours)

2. Créer un DataFrame récapitulatif avec ces métriques
3. Trier les machines par temps d'arrêt cumulé décroissant
```

```
4. Afficher le top 10 des machines les plus critiques
```

Étape 4 : Visualisation - Diagramme de Pareto (15 min)

Le **diagramme de Pareto** permet d'identifier les 20% de machines qui causent 80% des problèmes.

Prompt suggéré :

```
Peux-tu ajouter au script la génération d'un diagramme de Pareto en utilisant matplotlib ?

Le graphique doit montrer :
- En barres : le temps d'arrêt cumulé par machine (en ordre décroissant)
- En courbe : le pourcentage cumulé du temps d'arrêt total
- Une ligne horizontale à 80% pour identifier la zone critique

Titre : "Diagramme de Pareto - Temps d'arrêt par machine (2024)"
Sauvegarde : "pareto_machines.png"
```

Questions à se poser : - Combien de machines représentent 80% des arrêts ? - Quels types de pannes sont les plus fréquents sur ces machines ?

Étape 5 : Rapport de Synthèse (10 min)

Objectif : Produire une note managériale avec recommandations.

Prompt suggéré :

```
Sur la base des résultats d'analyse, rédige une note de synthèse professionnelle
pour mon responsable de maintenance avec :

**Contexte** : Analyse des 5000 interventions de 2024

**Principaux constats** :
- Top 3 des machines les plus critiques (ID, temps d'arrêt total, nb interventions)
- Types de pannes dominants
- Estimation du coût (si arrêt = 500€/heure)

**Recommandations** :
- Investissements à prévoir (remplacement, modernisation)
- Actions de maintenance préventive renforcée
- Formations techniciens si nécessaire

Format : Markdown avec sections claires et chiffres concrets.
```

Personnalisez : - Ajoutez votre analyse personnelle - Vérifiez la cohérence des recommandations - Exportez en PDF si demandé

Critères d'Évaluation

Critère	Points
Script fonctionnel : Le code s'exécute sans erreur	30%
Nettoyage des données : Normalisation efficace	20%
Calculs corrects : MTBF, MTTR calculés correctement	20%
Visualisation : Diagramme de Pareto clair et pertinent	15%
Rapport : Note de synthèse professionnelle et argumentée	15%

Conseils

1. **Itérez avec Gemini** : Si le code ne fonctionne pas, copiez l'erreur et demandez une correction
2. **Vérifiez les résultats** : Ne faites pas confiance aveuglément à l'IA, vérifiez la cohérence des chiffres
3. **Soyez critique** : L'IA peut faire des erreurs de calcul ou d'interprétation
4. **Contextualisez** : Ajoutez votre expertise métier dans les recommandations

Ressources Complémentaires

Formules de référence :

- **MTBF** = Temps total disponible / Nombre de pannes
- **MTTR** = Temps total de réparation / Nombre de pannes
- **Disponibilité** = MTBF / (MTBF + MTTR)

Bibliothèques Python utilisées : - `pandas` : manipulation de données - `matplotlib` : visualisation - `numpy` : calculs numériques

---

Aller Plus Loin (Optionnel)

Si vous terminez en avance :

1. **Analyse par type de panne** : Quel type de panne génère le plus de temps d'arrêt ?
2. **Analyse temporelle** : Y a-t-il des périodes de l'année plus critiques ?
3. **Analyse par technicien** : Identifier les temps de réparation moyens par technicien
4. **Carte de criticité** : Matrice (Fréquence × Gravité) pour prioriser les actions

---

Bonne analyse !