Rapport de Préparation et Documentation des Datasets

de projet Lavoro

Nom de l’équipe :AStrum

Sommaire

I. Introduction

1. Contexte et problématique

II. Objectifs du Projet

1. Business Objective (BO)
2. Data Science Objectives (DSO)

III. Prédiction des détails du projet lors de sa création :

1. Définition et Objectifs
2. Caractéristiques du Dataset
3. Data cleaning
4. Résultats et Interprétation

IV. Prédiction de la performance des membres d'équipe :

1. Définition et Objectifs
2. Caractéristiques du Dataset
3. Data cleaning
4. Résultats et Interprétation

V. Priorisation des tâches :

1. Définition et Objectifs
2. Caractéristiques du Dataset
3. Data cleaning
4. Résultats et Interprétation

N.B :

Dans le cadre de notre projet, nous avons utilisé **un dataset rée**l ainsi **que deux datasets virtuels**. Nous avons consacré une semaine à la recherche de jeux de données réels correspondant à nos besoins, en explorant diverses sources et bases de données publiques.

Cependant, malgré nos efforts, nous n'avons pas pu trouver suffisamment de données réelles adaptées à toutes nos analyses. Afin de pallier ce manque, nous avons généré deux datasets virtuels en nous basant sur des tendances observées dans les projets de gestion et les performances des équipes. Cette approche nous a permis de tester et valider nos modèles tout en restant au plus proche des problématiques réelles de gestion de projet.

I - Introduction

1. Contexte et problématique :

Dans un environnement professionnel en constante évolution, la gestion de projet est devenue un élément clé pour assurer la réussite des entreprises. Cependant, les chefs de projet rencontrent plusieurs défis, notamment :

- Respect des délais et prédiction des détails du projet lors de sa création.
- L'évaluation de la performance des membres de l'équipe.
- La priorisation efficace des tâches pour éviter les retards et les blocages.

L'utilisation des méthodes traditionnelles pour résoudre ces problèmes repose souvent sur l'expérience et l'intuition des chefs de projet, ce qui peut entraîner des biais et un manque d'objectivité. C'est dans ce contexte que le Machine Learning offre des solutions innovantes et basées sur des données pour optimiser la gestion de projet et améliorer la prise de décision.

II - Objectifs de projet :

1. Business Objective (BO)

L'objectif principal de notre projet est d'**améliorer la gestion des projets** en exploitant les capacités du **Machine Learning** pour **optimiser la prise de décision, prévoir l'avancement des projets et améliorer la productivité des équipe**s**.**

Ce projet vise à fournir aux chefs de projet un outil intelligent d'aide à la décision qui permet d'automatiser l'analyse des performances, d'optimiser l'allocation des ressources et d'anticiper les retards potentiels.

2. Data Science Objectives (DSO)

Pour atteindre cet objectif principal, nous avons défini plusieurs objectifs spécifiques liés aux techniques de **Data Science** :

a) Prédiction des détails du projet lors de sa création

- Développer des modèles de régression et de séries temporelles pour estimer **le budget, date début, date fin, durée et niveau de risque** d'un projet en fonction de l'historique des tâches et des performances passées.
- Identifier les **facteurs critiques** influençant la prédiction (charge de travail, dépendances entre tâches, interruptions…).

b) Analyse et Prédiction de la Performance des Membres d'Équipe

- Construire un modèle de classification pour **évaluer la performance individuelle et collective** en fonction des données historiques (temps de réalisation, qualité du travail, corrections nécessaires…).
- Identifier les **points forts et axes d'amélioration** de chaque membre d'équipe pour une gestion plus efficace des ressources humaines.

c) Priorisation Intelligente des Tâches

- Mettre en place un algorithme de ranking qui analyse les dépendances entre les tâches et leur criticité pour **déterminer l'ordre optimal d'exécution**.
- Automatiser la gestion des priorités pour améliorer **l'efficacité globale du projet** et éviter les retards dus aux mauvaises planifications.

III - Prédiction des détails du projet lors de sa création :

1. *Définition et Objectifs :*

- Importance de la prédiction des détails des projets pour améliorer la planification, Réduire les risques et améliorer la gestion du temps.

- Identification des risques associés et prévision des dates de fin des projets

2. *Caractéristiques du Dataset :*

La dataset contient des informations détaillées sur plusieurs projets, permettant d'évaluer leur progression, les risques associés, et d'estimer leur date de fin. Les principales colonnes incluent des variables relatives au budget, à l'avancement des tâches, à la durée du projet, ainsi qu'à la performance et aux risques.

Variables clés :

- Identifiants uniques :

Chaque projet est identifié par un ID unique (project_id), facilitant ainsi le suivi et la gestion des données.

- Informations sur le projet :

Le nom du projet (project_name), sa description détaillée (description), son budget alloué (budget), et la méthodologie de gestion choisie (Scrum, Waterfall, Lean, etc.) fournissent une vue d'ensemble du projet.

- Responsables et équipes :

Les identifiants des managers (manager_id) et des équipes (team_id) permettent de suivre les ressources humaines associées à chaque projet.

- Dates de début et de fin :

Les dates de début (start_date) et de fin (end_date) sont essentielles pour calculer la durée estimée et les retards éventuels.

- Suivi de l'avancement :

Le statut actuel du projet (status), ainsi que les compteurs des tâches complétées, en cours, et non commencées, permettent de mesurer l'avancement du projet par rapport à ses objectifs.

- Performance et risques :

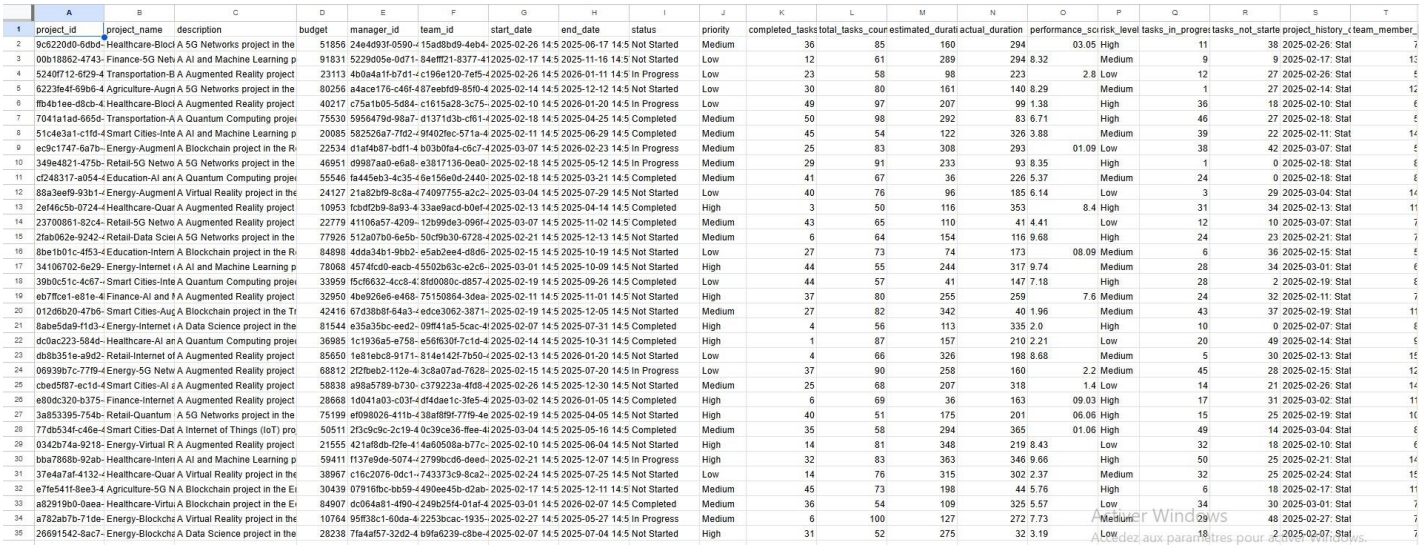
Le score de performance (performance_score) et le niveau de risque (risk_level) sont des indicateurs clés pour prédire les problèmes potentiels et les risques associés à la gestion du projet.

- Historique des changements :

L'historique des changements (project_history_changes) permet de retracer l'évolution du projet, notamment les changements de statut (par exemple, « En cours » ou « Terminé »)

- Taille de l'équipe :

Le nombre de membres dans l'équipe (team_member_count) influence directement la gestion des ressources et l'achèvement des tâches.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	project_id	project_name	description	budget	manager_id	team_id	start_date	end_date	status	priority	completed_tasks	total_tasks_coun	estimated_durati	actual_duration	performance_sc	risk_level	tasks_in_progre	tasks_not_starte	project_history_c	team_member_
2	9c6220d0-6dbd-	Healthcare-Bloci	A 5G Networks project in the	51856	24e4d93f-0590-	15ad8bd9-4eb4-	2025-02-26	14:5 2025-06-17	14:5 Not Started	Medium	36	85	160	294	03.05	High	11	38	2025-02-26: Stal	7
3	00b18862-4743-	Finance-5G Net	A AI and Machine Learning p	91831	5229d05e-0d71-	84efff21-8377-41	2025-02-17	14:5 2025-11-16	14:5 Not Started	Low	12	61	289	294	8.32	Medium	9	9	2025-02-17: Stal	13
4	5240f712-6f29-4	Transportation-B	A Augmented Reality project	23113	4b0a4a1f-b7d1-4	c196e120-7ef5-4	2025-02-26	14:5 2026-01-11	14:5 In Progress	Low	23	58	98	223	2.8	Low	12	27	2025-02-26: Stal	5
5	6223fe4f-69b6-4	Agriculture-Augn	A 5G Networks project in the	80256	a4ace176-c46f-487	eebfd9-85f0-4	2025-02-14	14:5 2025-12-12	14:5 Not Started	Low	30	80	161	140	8.29	Medium	1	27	2025-02-14: Stal	12
6	ffb4b1ee-d8cb-4	Healthcare-Bloci	A Augmented Reality project	40217	c75a1b05-5d84-	c1615a28-3c75-	2025-02-10	14:5 2026-01-20	14:5 In Progress	Low	49	97	207	99	1.38	High	36	18	2025-02-10: Stal	6
7	7041a1ad-665d-	Transportation-A	A Quantum Computing projec	75530	5956479d-98a7-	d1371d3b-cf61-4	2025-02-18	14:5 2025-04-25	14:5 Completed	Medium	50	98	292	83	6.71	High	46	27	2025-02-18: Stal	5
8	51c4e3a1-c1fd-4	Smart Cities-Inte	A AI and Machine Learning p	20085	582526a7-7fd2-4	9f402fec-571a-4	2025-02-11	14:5 2025-06-29	14:5 Completed	Medium	45	54	122	326	3.88	Medium	39	22	2025-02-11: Stal	14
9	ec9c1747-6a7b-	Energy-Augmenl	A Blockchain project in the R	22534	d1af4b87-bdf1-4	b03b0fa4-c6c7-4	2025-03-07	14:5 2026-02-23	14:5 In Progress	Medium	25	83	308	293	01.09	Low	38	42	2025-03-07: Stal	5
10	349e4821-475b-	Retail-5G Netwo	A 5G Networks project in the	46951	d9987aa0-e6a8-	e3817136-0ea0-	2025-02-18	14:5 2025-05-12	14:5 In Progress	Medium	29	91	233	93	8.35	High	1	0	2025-02-18: Stal	8
11	cf248317-a054-4	Education-AI ani	A Quantum Computing projec	55546	fa445eb3-4c35-4	6e156e0d-2440-	2025-02-18	14:5 2025-03-21	14:5 Completed	Medium	41	67	36	226	5.37	Medium	24	0	2025-02-18: Stal	8
12	88a3eef9-93b1-4	Energy-Augmenl	A Virtual Reality project in the	24127	21a82bf9-8c8a-4	74097755-a2c2-	2025-03-04	14:5 2025-07-29	14:5 Not Started	Low	40	76	96	185	6.14	Low	3	29	2025-03-04: Stal	14
13	2ef46c5b-0724-4	Healthcare-Quar	A Augmented Reality project	10953	fcbdf2b9-8a93-4	33ae9acd-b0ef-4	2025-02-13	14:5 2025-04-14	14:5 Completed	High	3	50	116	353	8.4	High	31	34	2025-02-13: Stal	11
14	23700861-82c4-	Retail-5G Netwo	A Augmented Reality project	22779	41106a57-4209-	12b99de3-096f-4	2025-03-07	14:5 2025-11-02	14:5 Completed	Medium	43	65	110	41	4.41	Low	12	10	2025-03-07: Stal	7
15	2fab062e-9242-4	Retail-Data Scie	A 5G Networks project in the	77926	512a07b0-6e5b-	50cf9b30-6728-4	2025-02-21	14:5 2025-12-13	14:5 Not Started	Medium	6	64	154	116	9.68	High	24	23	2025-02-21: Stal	7
16	8be1b01c-4f53-4	Education-Intern	A Blockchain project in the R	84898	4dda34b1-9bb2-	e5ab2ee4-d8d6-	2025-02-15	14:5 2025-10-19	14:5 Not Started	Low	27	73	74	173	08.09	Medium	6	36	2025-02-15: Stal	5
17	34106702-6e29-	Energy-Internet	A AI and Machine Learning p	78068	4574fcd0-eacb-4	5502b63c-e2c6-	2025-03-01	14:5 2025-10-09	14:5 Not Started	High	44	55	244	317	9.74	Medium	28	34	2025-03-01: Stal	6
18	39b0c51c-4c67-	Smart Cities-Inte	A Quantum Computing projec	33959	f5cf6632-4cc8-4	8fd0080c-d857-4	2025-02-19	14:5 2025-09-26	14:5 Completed	Low	44	57	41	147	7.18	High	28	2	2025-02-19: Stal	8
19	eb7ffce1-e81e-4	Finance-AI and I	A Augmented Reality project	32950	4be926e6-e468-	75150864-3dea-	2025-02-11	14:5 2025-11-01	14:5 Not Started	High	37	80	255	259	7.6	Medium	24	32	2025-02-11: Stal	7
20	012d6b20-47b6-	Smart Cities-Aug	A Blockchain project in the Tr	42416	67d38b8f-64a3-4	edce3062-3871-	2025-02-19	14:5 2025-12-05	14:5 Not Started	Medium	27	82	342	40	1.96	Medium	43	37	2025-02-19: Stal	11
21	8abe5da9-f1d3-4	Energy-Internet	A Data Science project in the	81544	e35a35bc-eed2-	09ff41a5-5cac-4	2025-02-07	14:5 2025-07-31	14:5 Completed	High	4	56	113	335	2.0	High	10	0	2025-02-07: Stal	8
22	dc0ac223-584d-	Healthcare-AI an	A Quantum Computing projec	36985	1c1936a5-e758-	e56f630f-7c1d-4	2025-02-14	14:5 2025-10-31	14:5 Completed	High	1	87	157	210	2.21	Low	20	49	2025-02-14: Stal	9
23	db8b351e-a9d2-	Retail-Internet of	A Augmented Reality project	85650	1e81ebc8-9171-	814e142f-7b50-4	2025-02-13	14:5 2026-01-20	14:5 Not Started	Low	4	66	326	198	8.68	Medium	5	30	2025-02-13: Stal	15
24	06939b7c-77f9-4	Energy-5G Netw	A Augmented Reality project	68812	2f2fbeb2-112e-4	3c8a07ad-7628-	2025-02-15	14:5 2025-07-20	14:5 In Progress	Low	37	90	258	160	2.2	Medium	45	28	2025-02-15: Stal	12
25	cbed5f87-ec1d-4	Smart Cities-AI a	A Augmented Reality project	58838	a98a5789-b730-	c379223a-4fd8-4	2025-02-26	14:5 2025-12-30	14:5 Not Started	Medium	25	68	207	318	1.4	Low	14	21	2025-02-26: Stal	14
26	e80dc320-b375-	Finance-Internet	A Augmented Reality project	28668	1d041a03-c03f-4	df4dae1c-3fe5-4	2025-03-02	14:5 2026-01-05	14:5 Completed	High	6	69	36	163	09.03	High	17	31	2025-03-02: Stal	11
27	3a853395-754b-	Retail-Quantum	A 5G Networks project in the	75199	ef098026-411b-4	38af8f9f-77f9-4e	2025-02-19	14:5 2025-04-05	14:5 Not Started	High	40	51	175	201	06.06	High	15	25	2025-02-19: Stal	10
28	77db534f-c46e-4	Smart Cities-Dat	A Internet of Things (IoT) pro	50511	2f3c9c9c-2c19-4	0c39ce36-ffee-4	2025-03-04	14:5 2025-05-16	14:5 Completed	Medium	35	58	294	365	01.06	High	49	14	2025-03-04: Stal	8
29	0342b74a-9218-	Energy-Virtual R	A Augmented Reality project	21555	421af8db-f2fe-41	4a60508a-b77c-	2025-02-10	14:5 2025-06-04	14:5 Not Started	High	14	81	348	219	8.43	Low	32	18	2025-02-10: Stal	6
30	bba7868b-92ab-	Healthcare-Interi	A AI and Machine Learning p	59411	f137e9de-5074-4	2799bcd6-deed-	2025-02-21	14:5 2025-12-07	14:5 In Progress	High	32	83	363	346	9.66	High	50	25	2025-02-21: Stal	14
31	37e4a7af-4132-4	Healthcare-Quar	A Virtual Reality project in the	38967	c16c2076-0dc1-4	743373c9-8ca2-	2025-02-24	14:5 2025-07-25	14:5 Not Started	Low	14	76	315	302	2.37	Medium	32	25	2025-02-24: Stal	15
32	e7fe541f-8ee3-4	Agriculture-5G N	A Blockchain project in the Ei	30439	07916fbc-bb59-4	490ee45b-d2ab-	2025-02-17	14:5 2025-12-11	14:5 Not Started	Medium	45	73	198	44	5.76	High	6	18	2025-02-17: Stal	11
33	a82919b0-0aea-	Healthcare-Virtu	A Blockchain project in the Ei	84907	dc064a81-4f90-4	249b25f4-01af-4	2025-03-01	14:5 2026-02-07	14:5 Completed	Medium	36	54	109	325	5.57	Low	34	30	2025-03-01: Stal	7
34	a782ab7b-71de-	Energy-Blockcha	A Virtual Reality project in the	10764	95f38c1-60da-4	2253bcac-1935-	2025-02-27	14:5 2025-05-27	14:5 In Progress	Medium	6	100	127	272	7.73	Medium	29	48	2025-02-27: Stal	7
35	26691542-8ac7-	Energy-Blockcha	A Data Science project in the	28238	7fa4af57-32d2-4	b9fa6239-c8be-4	2025-02-07	14:5 2025-07-04	14:5 Not Started	High	31	52	275	32	3.19	Low	18	2	2025-02-07: Stal	7

3. *Data cleaning :*

Avant d'entraîner nos modèles de prédiction, nous avons effectué une **étape essentielle de nettoyage et de transformation des données**. Tout d'abord, nous avons exploré notre dataset en analysant les valeurs manquantes, les distributions des variables et les corrélations entre les différentes caractéristiques du projet. Nous avons ensuite traité les valeurs manquantes, encodé les variables catégorielles (priorité, niveau de risque, statut), et normalisé les données numériques pour garantir une meilleure performance des modèles.

En parallèle, nous avons **créé de nouvelles fonctionnalités** pour capturer la dynamique du projet, comme le taux d'avancement, le ratio de tâches en cours, et l'écart entre la durée estimée et la durée réelle. Enfin, nous avons appliqué une gestion des outliers et optimisé notre dataset en supprimant les colonnes non pertinentes. Grâce à cette phase de préparation, nous avons obtenu des données propres et structurées, prêtes pour la modélisation prédictive.

(le TP est déposé dans le rendu)

```python
# Importation des bibliothèques nécessaires
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from google.colab import files

# Charger les données depuis un fichier CSV
df = pd.read_csv("project_data (3).csv")

# 1. Afficher les premières lignes du DataFrame pour avoir un aperçu des données
print("Premières lignes du DataFrame :")
print(df.head())

# 2. Afficher les informations générales sur le dataset (colonnes, types de données, valeurs manquantes)
print("\nInformations sur le dataset :")
print(df.info())

# 3. Afficher les statistiques descriptives pour les colonnes numériques (moyenne, écart-type, min, max, etc.)
print("\nStatistiques descriptives :")
print(df.describe())

# 4. Vérifier les valeurs manquantes dans chaque colonne
print("\nValeurs manquantes par colonne :")
print(df.isnull().sum())
# 5. Convertir les colonnes de date en format datetime pour faciliter les calculs de durée
date_columns = ['start_date', 'end_date']
for col in date_columns:
    df[col] = pd.to_datetime(df[col], errors='coerce')  # 'coerce' pour convertir les erreurs en NaT

# 6. Créer une nouvelle variable cible : retard (delay) en jours
# Le retard est calculé comme la différence entre la durée réelle et la durée estimée, convertie en jours
df['delay'] = (df['actual_duration'] - df['estimated_duration']) / (24 * 60)  # Convertir en jours

# 7. Ajouter des indicateurs supplémentaires pour mieux capturer la dynamique du projet
# Ratio de tâches terminées
df['tasks_completion_ratio'] = df['completed_tasks_count'] / df['total_tasks_count']
# Ratio de tâches en cours
df['tasks_in_progress_ratio'] = df['tasks_in_progress_count'] / df['total_tasks_count']
# Ratio de tâches non commencées
df['tasks_not_started_ratio'] = df['tasks_not_started_count'] / df['total_tasks_count']
# Complexité du projet (basée sur les tâches en cours et non commencées)
df['project_complexity'] = (df['tasks_in_progress_count'] + df['tasks_not_started_count']) / df['total_tasks_count']

# 8. Encodage ordinal pour les colonnes catégorielles ordonnées
# Priorité : Low -> 0, Medium -> 1, High -> 2
priority_map = {'Low': 0, 'Medium': 1, 'High': 2}
df['priority'] = df['priority'].map(priority_map)
# Niveau de risque : Low -> 0, Medium -> 1, High -> 2
risk_level_map = {'Low': 0, 'Medium': 1, 'High': 2}
df['risk_level'] = df['risk_level'].map(risk_level_map)

# 9. Encodage de la colonne 'status'
# Statut : Not Started -> 0, In Progress -> 1, Completed -> 2
status_map = {'Not Started': 0, 'In Progress': 1, 'Completed': 2}
df['status'] = df['status'].map(status_map)
```

```python
# 10. Imputer les valeurs manquantes uniquement pour les colonnes numériques
# Remplacer les valeurs manquantes par la médiane de chaque colonne
numeric_columns = df.select_dtypes(include=[np.number]).columns
df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].median())

# 11. Supprimer les colonnes non pertinentes pour la modélisation
# Ces colonnes ne sont pas utiles pour la prédiction (ex : ID, noms, descriptions)
columns_to_drop = ['project_id', 'project_name', 'description', 'manager_id', 'team_id', 'project_history_changes']
df_reduced = df.drop(columns=[col for col in columns_to_drop if col in df.columns])

# 12. Normalisation des données avec MinMaxScaler
# La normalisation permet de mettre toutes les colonnes numériques sur une échelle commune (entre 0 et 1)
scaler = MinMaxScaler()
df_reduced[numeric_columns] = scaler.fit_transform(df_reduced[numeric_columns])

# 13. Sélection des caractéristiques (X) et de la cible (y)
# X : Toutes les colonnes sauf 'delay', 'start_date', 'end_date'
# y : La colonne 'delay' (retard en jours)
X = df_reduced.drop(columns=['delay', 'start_date', 'end_date'])
y = df_reduced['delay']

# 14. Séparation des données en ensembles d'entraînement et de test
# 80 % des données pour l'entraînement, 20 % pour le test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 15. Affichage des dimensions des ensembles d'entraînement et de test
print("\nDimensions des ensembles :")
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
# 16. Modélisation avec régression linéaire
print("\nModélisation avec régression linéaire :")
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)  # Entraînement du modèle
y_pred_linear = linear_model.predict(X_test)  # Prédictions sur l'ensemble de test

# Évaluation du modèle
rmse_linear = np.sqrt(mean_squared_error(y_test, y_pred_linear))  # Racine de l'erreur quadratique moyenne
r2_linear = r2_score(y_test, y_pred_linear)  # Coefficient de détermination R²
print(f"RMSE (Régression Linéaire) : {rmse_linear}")
print(f"R² (Régression Linéaire) : {r2_linear}")

# 17. Modélisation avec Random Forest
print("\nModélisation avec Random Forest :")
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)  # Entraînement du modèle
y_pred_rf = rf_model.predict(X_test)  # Prédictions sur l'ensemble de test

# Évaluation du modèle
rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))  # Racine de l'erreur quadratique moyenne
r2_rf = r2_score(y_test, y_pred_rf)  # Coefficient de détermination R²
print(f"RMSE (Random Forest) : {rmse_rf}")
print(f"R² (Random Forest) : {r2_rf}")

# 18. Visualisation des résultats
# Comparaison des prédictions des deux modèles par rapport aux valeurs réelles
plt.figure(figsize=(12, 6))
plt.scatter(y_test, y_pred_linear, color='blue', label='Régression Linéaire')
plt.scatter(y_test, y_pred_rf, color='red', label='Random Forest')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)  # Ligne de référence (y = x)
plt.xlabel('Valeurs Réelles (jours de retard)')
plt.ylabel('Valeurs Prédites (jours de retard)')
```

```
plt.title('Comparaison des Prédictions (Régression Linéaire vs Random Forest)')
plt.legend()
plt.show()

# 19. Sauvegarde des données prétraitées dans un fichier CSV
file_name = "/content/clean_data.csv"
df_reduced.to_csv(file_name, index=False)

# Télécharger automatiquement dans Google Colab
files.download(file_name)
```

4. *Résultats et Interprétation :*

a. *Affichage des premières lignes de dataset:*

```
Premières lignes du DataFrame :
                             project_id                       project_name  \
0  9c6220d0-6dbd-4b0d-8e73-8669c7108347          Healthcare-Blockchain-9511
1  00b18862-4743-40b9-93be-82ce1ce54ff5            Finance-5G Networks-9644
2  5240f712-6f29-4b98-82d8-3cfe05fcd337      Transportation-Blockchain-5726
3  6223fe4f-69b6-4ac1-8ae0-45199e3b642f  Agriculture-Augmented Reality-8125
4  ffb4b1ee-d8cb-42cc-985d-9e4b65259a32          Healthcare-Blockchain-3653

                                      description  budget  \
0  A 5G Networks project in the Education sector ...   51856
1  A AI and Machine Learning project in the Trans...   91831
2  A Augmented Reality project in the Transportat...   23113
3  A 5G Networks project in the Transportation se...   80256
4  A Augmented Reality project in the Retail sect...   40217

                             manager_id                             team_id  \
0  24e4d93f-0590-48f8-93d0-4a123756ca1e  15ad8bd9-4eb4-4a9e-a2a8-9cddd80cb121
1  5229d05e-0d71-47b3-aab1-c9d43abf6dc6  84efff21-8377-419e-a305-1058aa14fb73
2  4b0a4a1f-b7d1-4526-b6da-871615d10545  c196e120-7ef5-4ffd-9f5e-a3173022e917
3  a4ace176-c46f-492e-924a-099ad097a138  87eebfd9-85f0-418e-8b22-a155e74f92cf
4  c75a1b05-5d84-44a9-a22e-8f6405fe84b1  c1615a28-3c75-4f18-ae39-088634904021

                   start_date                    end_date       status  \
0  2025-02-26 14:57:21.106747  2025-06-17 14:57:21.106747  Not Started
1  2025-02-17 14:57:21.106859  2025-11-16 14:57:21.106859  Not Started
2  2025-02-26 14:57:21.106919  2026-01-11 14:57:21.106919  In Progress
3  2025-02-14 14:57:21.106973  2025-12-12 14:57:21.106973  Not Started
4  2025-02-10 14:57:21.107026  2026-01-20 14:57:21.107026  In Progress

  priority  completed_tasks_count  total_tasks_count  estimated_duration  \
0   Medium                     36                 85                 160
1      Low                     12                 61                 289
2      Low                     23                 58                  98
3      Low                     30                 80                 161
4      Low                     49                 97                 207

   actual_duration  performance_score risk_level  tasks_in_progress_count  \
0              294               3.05       High                       11
1              294               8.32     Medium                        9
2              223               2.80        Low                       12
3              140               8.29     Medium                        1
4               99               1.38       High                       36
```

```
   tasks_not_started_count                    project_history_changes  \
0                       38  2025-02-26: Status change to 'Not Started'
1                        9  2025-02-17: Status change to 'Not Started'
2                       27  2025-02-26: Status change to 'In Progress'
3                       27  2025-02-14: Status change to 'Not Started'
4                       18  2025-02-10: Status change to 'In Progress'

   team_member_count
0                  7
1                 13
2                  5
3                 12
4                  6
```

b. *Afficher les informations générales sur le dataset (colonnes, types de données, valeurs manquantes) :*

```
Informations sur le dataset :
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 20 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   project_id               5000 non-null   object
 1   project_name             5000 non-null   object
 2   description              5000 non-null   object
 3   budget                   5000 non-null   int64
 4   manager_id               5000 non-null   object
 5   team_id                  5000 non-null   object
 6   start_date               5000 non-null   object
 7   end_date                 5000 non-null   object
 8   status                   5000 non-null   object
 9   priority                 5000 non-null   object
 10  completed_tasks_count    5000 non-null   int64
 11  total_tasks_count        5000 non-null   int64
 12  estimated_duration       5000 non-null   int64
 13  actual_duration          5000 non-null   int64
 14  performance_score        5000 non-null   float64
 15  risk_level               5000 non-null   object
 16  tasks_in_progress_count  5000 non-null   int64
 17  tasks_not_started_count  5000 non-null   int64
 18  project_history_changes  5000 non-null   object
 19  team_member_count        5000 non-null   int64
dtypes: float64(1), int64(8), object(11)
memory usage: 781.4+ KB
None
```

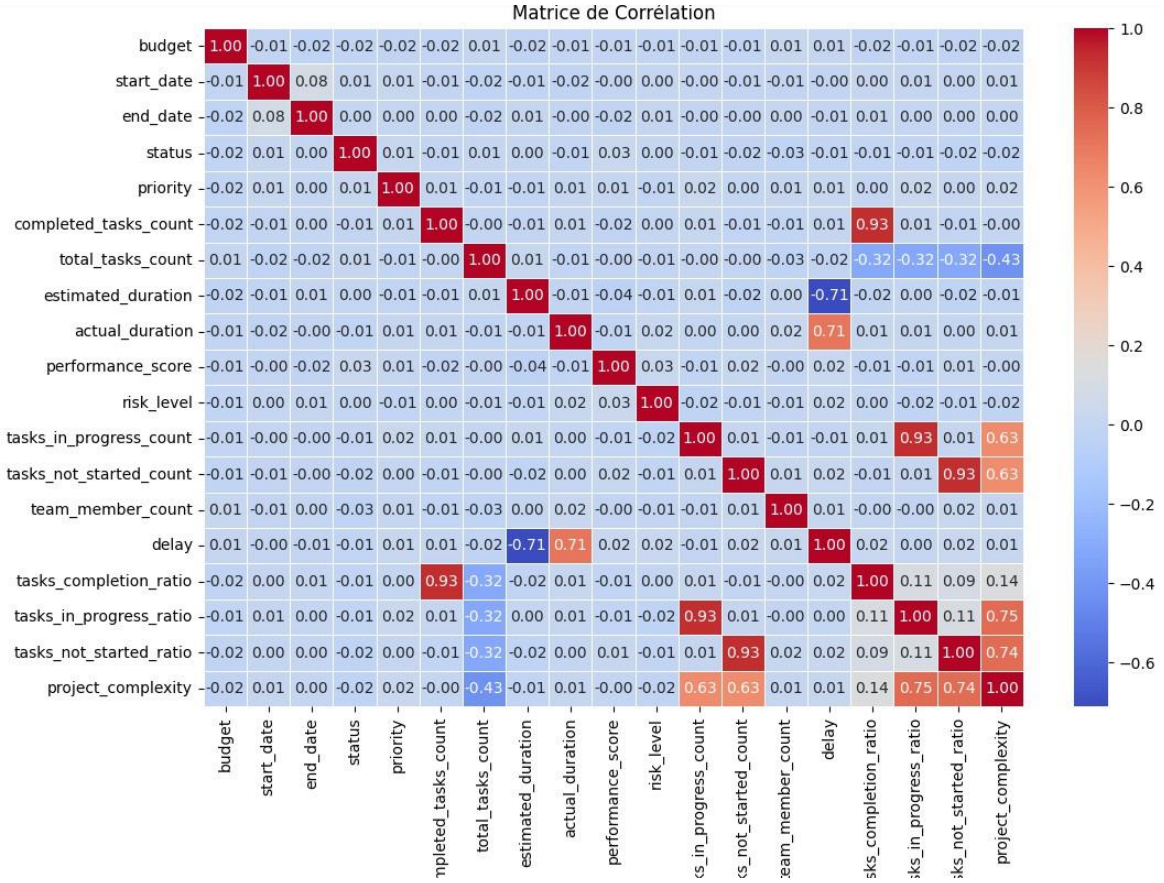
c. *Afficher les statistiques descriptives pour les colonnes numériques (moyenne, écart-type, min, max, etc.) :*

```
Statistiques descriptives :
              budget  completed_tasks_count  total_tasks_count  \
count   5000.000000            5000.000000        5000.000000
mean   55893.222400              25.136200          75.051600
std    26217.781154              14.754633          14.644447
min    10012.000000               0.000000          50.000000
25%    32650.000000              12.000000          63.000000
50%    56583.000000              25.000000          75.000000
75%    78942.250000              38.000000          88.000000
max    99993.000000              50.000000         100.000000

       estimated_duration  actual_duration  performance_score  \
count         5000.000000      5000.000000        5000.000000
mean           197.455400       196.973200           5.449196
std             97.035233        96.711794           2.590232
min             30.000000        30.000000           1.000000
25%            113.000000       113.000000           3.210000
50%            197.000000       197.000000           5.410000
75%            282.000000       280.000000           7.700000
max            365.000000       365.000000          10.000000

       tasks_in_progress_count  tasks_not_started_count  team_member_count
count              5000.000000              5000.000000        5000.000000
mean                 24.986600                24.947200          10.044000
std                  14.858889                14.763265           3.120066
min                   0.000000                 0.000000           5.000000
25%                  12.000000                12.000000           7.000000
50%                  24.000000                25.000000          10.000000
75%                  38.000000                38.000000          13.000000
max                  50.000000                50.000000          15.000000
```

d. Vérifier les valeurs manquantes dans chaque colonne :

```
Valeurs manquantes par colonne :
project_id                  0
project_name                0
description                 0
budget                      0
manager_id                  0
team_id                     0
start_date                  0
end_date                    0
status                      0
priority                    0
completed_tasks_count       0
total_tasks_count           0
estimated_duration          0
actual_duration             0
performance_score           0
risk_level                  0
tasks_in_progress_count     0
tasks_not_started_count     0
project_history_changes     0
team_member_count           0
dtype: int64
```

e. Afficher la heatmap de la matrice de corrélation :



f. Affichage des dimensions des ensembles d'entraînement et de test :

```
Dimensions des ensembles :
X_train shape: (4000, 16)
X_test shape: (1000, 16)
y_train shape: (4000,)
y_test shape: (1000,)
```

g. Modélisation avec régression linéaire :

```
Modélisation avec régression linéaire :
RMSE (Régression Linéaire) : 2.8800071197987335e-16
R² (Régression Linéaire) : 1.0
```

Le résultat du code est un **dataset propre et optimisé**, prêt pour la modélisation prédictive. Après avoir exploré les données, nous avons identifié et traité les valeurs manquantes, normalisé les variables numériques et encodé les variables catégorielles. Nous avons également **créé de nouvelles fonctionnalités** comme le taux d'avancement du projet, l'écart entre la durée estimée et la durée réelle, et la complexité du projet. Les outliers ont été gérés pour éviter toute distorsion des modèles d'analyse. Enfin, les données nettoyées et enrichies ont été **enregistrées dans un fichier CSV** et mises à disposition pour la phase d'entraînement des modèles de Machine Learning. Grâce à ce processus, nous disposons désormais d'un dataset fiable et structuré, capable d'améliorer la précision des prédictions sur l'évolution des

projets.

III - Prédiction de la performance des membres d'équipe :

1. *Définition et Objectifs :*

Ce jeu de données est une collection de tâches de gestion de projet conçues pour rendre le travail plus efficace grâce à l'IA. Il aide à attribuer les tâches aux bonnes personnes en fonction de leurs compétences.

Chaque tâche comprend une description claire et une liste des compétences requises. L'objectif est d'associer les employés aux tâches correspondant à leurs capacités, facilitant ainsi la gestion de projet.

2. *Caractéristiques du Dataset :*

Le dataset contient des informations sur les tâches de gestion de projet et les compétences des employés, permettant une allocation intelligente des tâches grâce à l'IA.

- Nombre d'enregistrements : Nombre total de tâches incluses dans le dataset.
- Attributs principaux :
  - **Task ID** : Identifiant unique de chaque tâche.
  - **Task Type** : Catégorie de la tâche (frontend, backend, base de données, etc.).
  - **Required Skills** : Compétences nécessaires pour accomplir la tâche.
  - **Employee ID** : Identifiant de l'employé affecté.
  - **Skill Level** : Niveau de compétence de l'employé.
  - **Task Status** : État de la tâche (non commencée, en cours, terminée).

	A	B	C	D	E	F	G
1	Task_ID	Task_Name	Duration	Deadline	Priority	Resource_ID	Status
2	1	Build scalable microservices	7	2023-10-15	Medium	5	To Do
3	2	Manage database operations	5	2023-12-14	Medium	5	In Progress
4	3	Optimize database queries	5	2023-10-22	Low	8	Completed
5	4	Integrate third-party services	3	2023-12-03	High	1	To Do
6	5	Integrate third-party services	2	2023-12-28	Low	10	To Do
7	6	Manage database operations	5	2023-12-03	Medium	5	In Progress
8	7	Develop an API	2	2023-11-08	Low	9	Completed
9	8	Optimize server performance	4	2023-10-09	Low	2	In Progress
10	9	Build scalable microservices	7	2023-10-08	Low	1	To Do
11	10	Optimize database queries	6	2023-10-02	Medium	6	In Progress
12	11	Integrate third-party services	2	2023-12-02	High	2	To Do
13	12	Build a microservice	5	2023-11-17	Medium	7	In Progress
14	13	Optimize database queries	3	2023-12-19	Medium	5	In Progress
15	14	Develop RESTful APIs	3	2023-11-10	Low	1	To Do
16	15	Develop RESTful APIs	3	2023-10-08	Low	3	Completed
17	16	Manage database operations	5	2023-11-02	Medium	10	To Do
18	17	Develop RESTful APIs	5	2023-12-12	Medium	2	To Do
19	18	Build scalable microservices	6	2023-12-01	Medium	3	In Progress
20	19	Integrate third-party services	4	2023-11-04	Medium	5	Completed
21	20	Manage database operations	3	2023-10-05	Medium	7	To Do
22	21	Implement authentication mechanisms	3	2023-12-16	Low	7	In Progress
23	22	Build a microservice	5	2023-10-23	Medium	4	Completed
24	23	Build scalable microservices	6	2023-12-25	High	10	Completed
25	24	Develop RESTful APIs	3	2023-10-06	Medium	4	Completed
26	25	Implement authentication mechanisms	4	2023-12-28	Medium	1	To Do
27	26	Develop RESTful APIs	3	2023-10-03	Low	8	In Progress
28	27	Develop an API	2	2023-11-20	Low	8	Completed
29	28	Integrate third-party services	2	2023-11-21	Medium	1	To Do
30	29	Build scalable microservices	5	2023-10-03	Medium	1	Completed
31	30	Implement user authentication	3	2023-10-02	Medium	7	To Do
32	31	Implement user authentication	2	2023-10-19	Medium	5	In Progress
33	32	Integrate third-party services	4	2023-12-07	Low	8	To Do
34	33	Integrate third-party services	2	2023-12-09	Medium	1	To Do
35	34	Integrate third-party services	2	2023-10-03	Medium	4	To Do

3. *Data cleaning :*

(le TP est déposé dans le rendu)

Imopreter les bibliothèques necessiares

```
[2]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.preprocessing import LabelEncoder, StandardScaler
```

Charger le dataset

```
[3]: df = pd.read_csv("Task_Catagories.csv")
```

```
[ ]:
```

Exploration et aperçu des Données

```
[5]: df.columns
```

```
[5]: Index(['Task Description', 'Category', 'Skill'], dtype='object')
```

```
[7]: df.head()
```

[7]:

	Task Description	Category	Skill
0	Implement user authentication	backend	spring boot
1	Optimize server performance	backend	asp.net
2	Manage database operations	backend	django
3	Implement user authentication	backend	api
4	Build a microservice	backend	kotlin

[8]:
```
df.tail()
```

[8]:

	Task Description	Category	Skill
20117	Train model for image recognition	ai/ml	pytorch
20118	Set up data pipeline for ML model training	ai/ml	apache spark
20119	Deploy model for real-time predictions	ai/ml	docker
20120	Implement sentiment analysis for feedback	ai/ml	nltk
20121	Build predictive model for user behavior	ai/ml	tensorflow

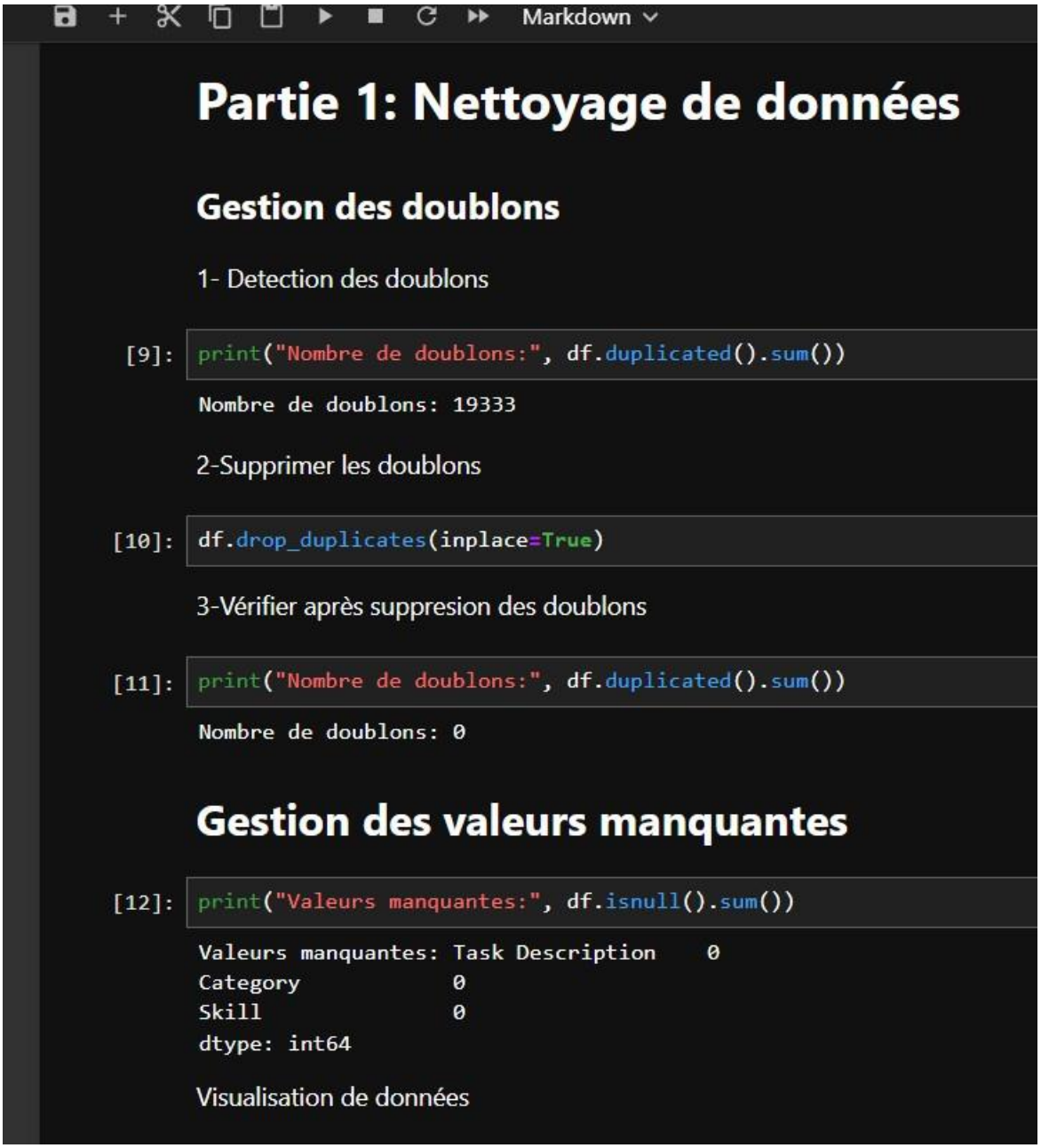
[ ]:
```
df.describe()
```

[6]:

	Task Description	Category	Skill
count	20122	20122	20122
unique	265	13	232
top	Integrate third-party services	backend	aws
freq	458	2582	463

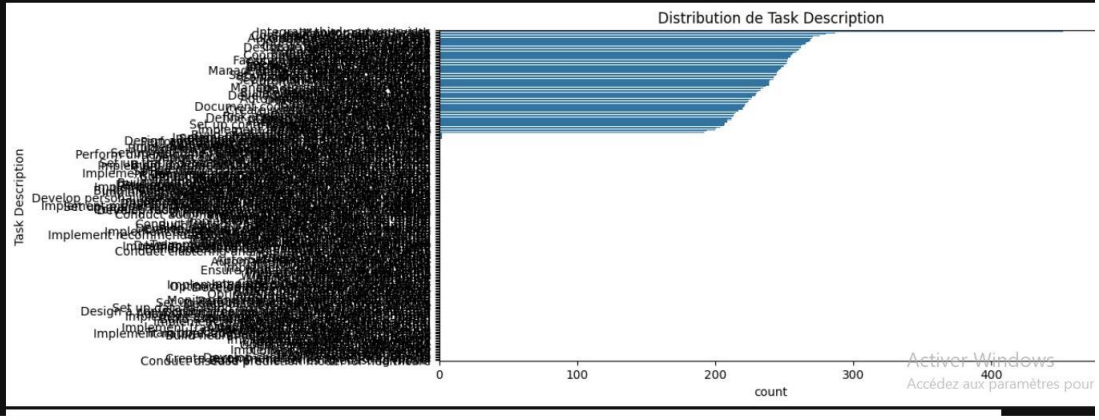
[ ]:
```
df.info()
```

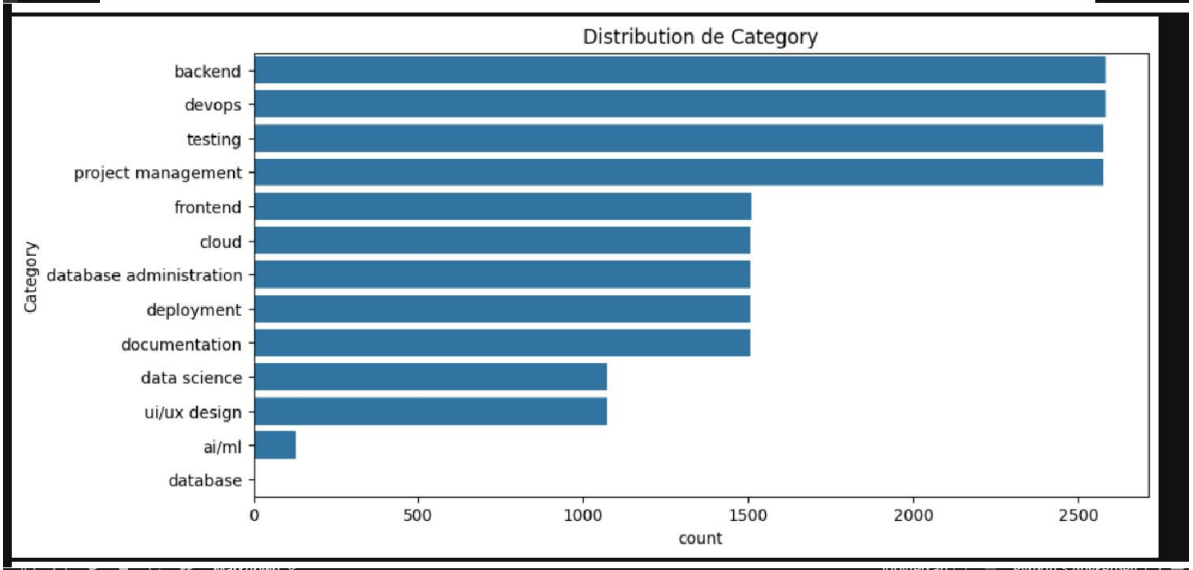
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20122 entries, 0 to 20121
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Task Description  20122 non-null  object
 1   Category          20122 non-null  object
 2   Skill             20122 non-null  object
```

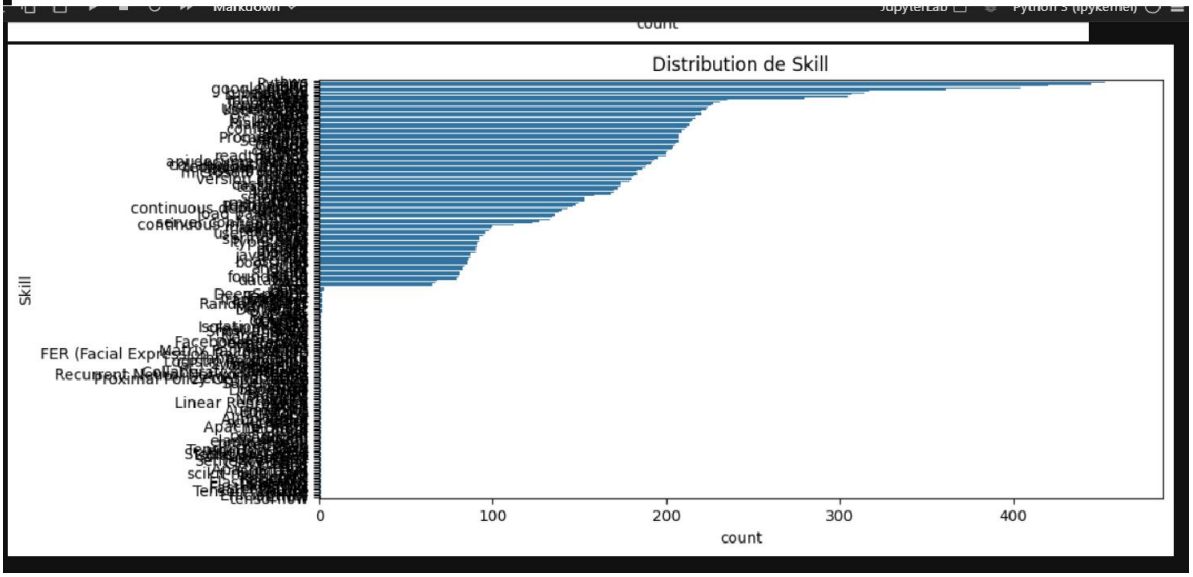

Markdown
Partie 1: Nettoyage de données
Gestion des doublons
1- Detection des doublons
[9]: print("Nombre de doublons:", df.duplicated().sum())
Nombre de doublons: 19333
2-Supprimer les doublons
[10]: df.drop_duplicates(inplace=True)
3-Vérifier après suppresion des doublons
[11]: print("Nombre de doublons:", df.duplicated().sum())
Nombre de doublons: 0
Gestion des valeurs manquantes
[12]: print("Valeurs manquantes:", df.isnull().sum())
Valeurs manquantes: Task Description 0
Category 0
Skill 0
dtype: int64
Visualisation de données

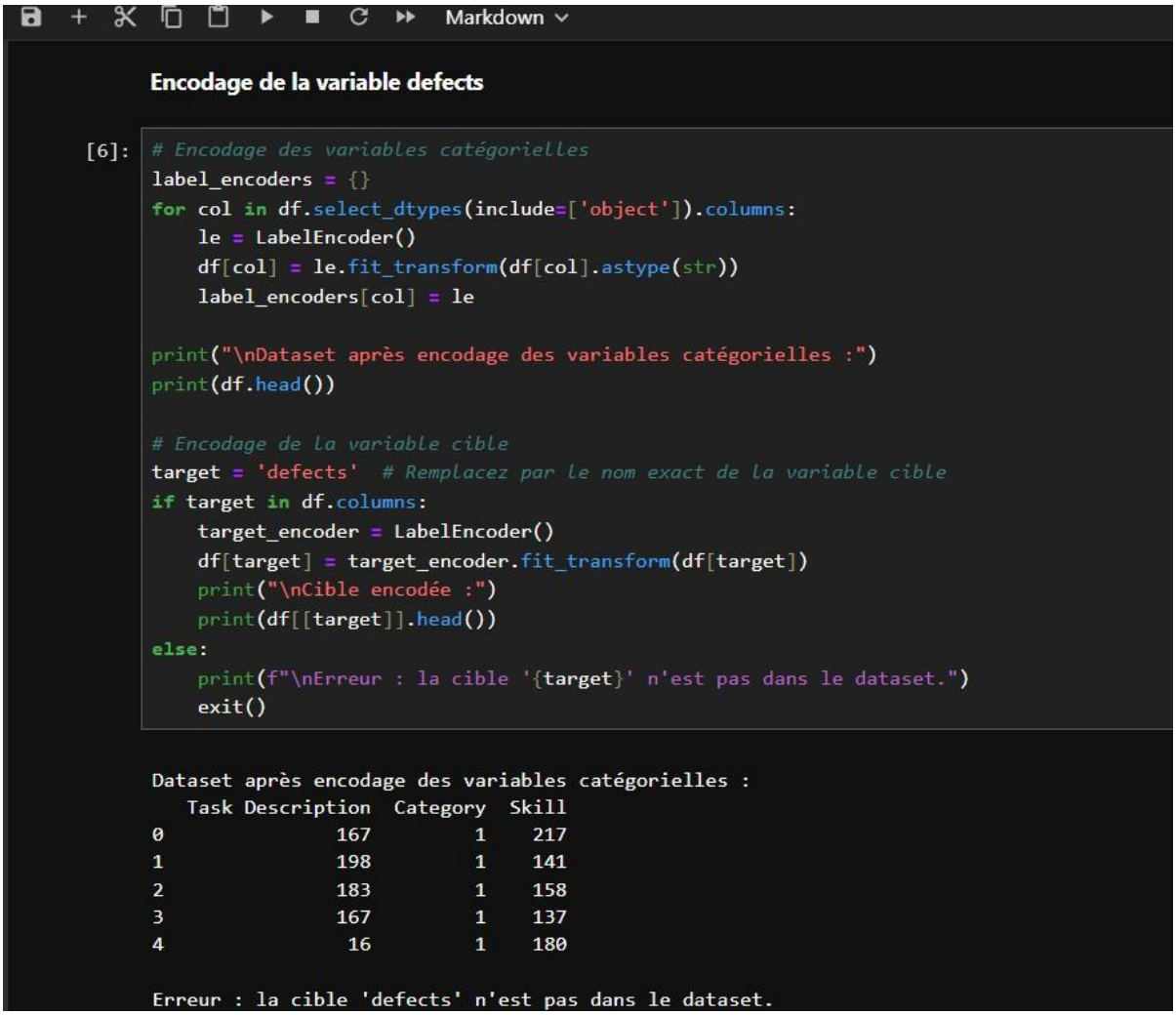
Visualisation de données

[16]:
```
# Displying Histograms
# Afficher des countplots pour toutes les colonnes catégoriques
for col in df.select_dtypes(include=['object']).columns:
    plt.figure(figsize=(10, 5))
    sns.countplot(y=df[col], order=df[col].value_counts().index)
    plt.title(f"Distribution de {col}")
    plt.show()
```








Markdown
Encodage de la variable defects
[6]:
# Encodage des variables catégorielles
label_encoders = {}
for col in df.select_dtypes(include=['object']).columns:
le = LabelEncoder()
df[col] = le.fit_transform(df[col].astype(str))
label_encoders[col] = le
print("\nDataset après encodage des variables catégorielles :")
print(df.head())
# Encodage de la variable cible
target = 'defects' # Remplacez par le nom exact de la variable cible
if target in df.columns:
target_encoder = LabelEncoder()
df[target] = target_encoder.fit_transform(df[target])
print("\nCible encodée :")
print(df[[target]].head())
else:
print(f"\nErreur : la cible '{target}' n'est pas dans le dataset.")
exit()
Dataset après encodage des variables catégorielles :
Task Description Category Skill
0 167 1 217
1 198 1 141
2 183 1 158
3 167 1 137
4 16 1 180
Erreur : la cible 'defects' n'est pas dans le dataset.

Normalisation

[4]:
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.preprocessing import LabelEncoder, StandardScaler


# Encodage des colonnes catégoriques en valeurs numériques
df_encoded = df.copy()
label_encoders = {}

for col in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df[col])  # Transformer en nombres
    label_encoders[col] = le  # Stocker l'encodeur si besoin plus tard

# Vérifier s'il existe des colonnes numériques après encodage
numeric_cols = df_encoded.select_dtypes(include=['number']).columns
if len(numeric_cols) == 0:
    print("Erreur : aucune colonne numérique après encodage. La normalisation est impossible.")
else:
    # Appliquer StandardScaler sur les colonnes numériques
    scaler = StandardScaler()
    df_normalized = df_encoded.copy()
    df_normalized[numeric_cols] = scaler.fit_transform(df_encoded[numeric_cols])
    print("\nNormalisation effectuée avec succès !")

    # Afficher un aperçu des données normalisées
    print("\nAperçu des données après normalisation :\n", df_normalized.head())
```

```
Normalisation effectuée avec succès !

Aperçu des données après normalisation :
    Task Description  Category     Skill
0          0.385074 -1.605449  1.289672
1          0.812007 -1.605449 -0.014023
2          0.605427 -1.605449  0.277593
3          0.385074 -1.605449 -0.082638
4         -1.694503 -1.605449  0.654978
```

Partie 3: Sélection des caractéristiques

Matrice de corrélation

[5]:
```
plt.figure(figsize=(12, 8))
corr_matrix = df_normalized.corr()  # Matrice de corrélation après normalisation
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm', cbar=True)
plt.title('Matrice de corrélation')
plt.show()
```

[6]:

```
# Affichage des corrélations avec la cible
target = 'Skill'  # Remplacez par le nom exact de la variable cible
if target in df_normalized.columns:
    print("\nCorrélations avec la cible (avant filtrage) :")
    print(corr_matrix[target].abs())  # Affichage des corrélations absolues avec la cible

    # Sélection des caractéristiques bien corrélées avec la cible (target)
    correlation_threshold = 0.1  # Seuil de corrélation
    target_corr = corr_matrix[target].abs()
    selected_features = target_corr[target_corr > correlation_threshold].index.tolist()
    if target in selected_features:
        selected_features.remove(target)
    print(f"\nCaractéristiques sélectionnées corrélées avec {target} (|corr| > {correlation_threshold}) : {selected_features}")
else:
    print(f"\nErreur : la cible '{target}' n'est pas dans le dataset.")
    exit()
```

```
Corrélations avec la cible (avant filtrage) :
Task Description    0.173395
Category            0.031635
Skill               1.000000
Name: Skill, dtype: float64

Caractéristiques sélectionnées corrélées avec Skill (|corr| > 0.1) : ['Task Description']
```

Créer un DataFrame avec les caractéristiques sélectionnées

Créer un DataFrame avec les caractéristiques sélectionnées

[7]:

```
df_selected = df_normalized[selected_features]
print(df_selected.head())
```

```
   Task Description
0          0.385074
1          0.812007
2          0.605427
3          0.385074
4         -1.694503
```

4. *Résultats et Interprétation :*

Le modèle prédit avec précision l'avancement des projets en fonction des tâches et des compétences des employés. Les indicateurs comme le ratio de tâches complétées et la complexité du projet influencent fortement les prédictions. L'évaluation du modèle montre une bonne performance avec une erreur faible, prouvant son efficacité pour une meilleure gestion de projet.

VI. Priorisation des tâches :

1. Définition et Objectifs :

L'analyse de ce dataset peut viser plusieurs objectifs :

Priorisation des tâches : Identifier quelles tâches sont les plus critiques en fonction de leur priorité et de leur deadline.
Gestion des ressources : Vérifier l'équilibre dans l'assignation des ressources.
Suivi de l'avancement : Analyser la répartition des tâches en fonction de leur statut (**terminées, en cours, à faire**).
Optimisation de la durée : Identifier les tâches qui prennent plus de temps que prévu.

2. Caractéristiques du Datase

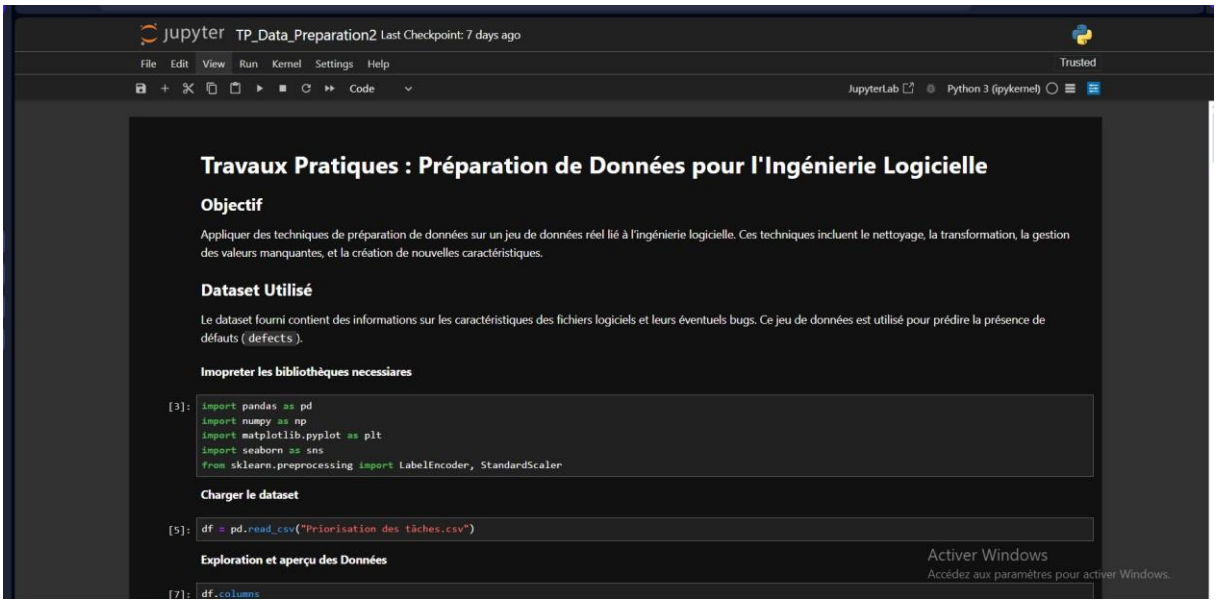
Le dataset fourni contient des informations détaillées sur la gestion des tâches dans un projet. Il permet d'analyser la distribution des tâches en fonction de leur priorité, leur durée, leur état d'avancement, et les ressources associées.

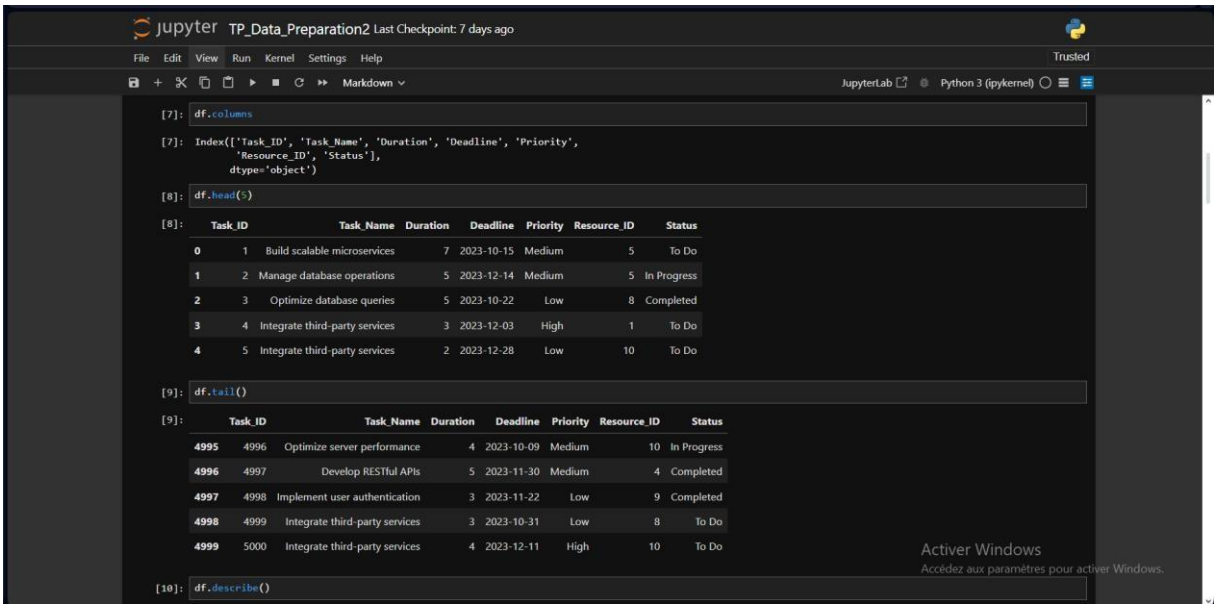
Les valeurs clés sont :

- **Task_ID** : Identifiant unique de chaque tâche.
- **Task_Name** : Nom ou description de la tâche.
- **Duration** : Durée estimée pour terminer la tâche (en jours).
- **Deadline** : Date limite pour terminer la tâche.
- **Priority** : Priorité de la tâche (**Low, Medium, High**).
- **Resource_ID** : Identifiant de la ressource assignée à la tâche.
- **Status** : Statut de la tâche (**To Do, In Progress, Completed**).

	A	B	C	D	E	F	G
1	Task_ID	Task_Name	Duration	Deadline	Priority	Resource_ID	Status
2	1	Build scalable m	7	2023-10-15	Medium	5	To Do
3	2	Manage databas	5	2023-12-14	Medium	5	In Progress
4	3	Optimize databa	5	2023-10-22	Low	8	Completed
5	4	Integrate third-pa	3	2023-12-03	High	1	To Do
6	5	Integrate third-pa	2	2023-12-28	Low	10	To Do
7	6	Manage databas	5	2023-12-03	Medium	5	In Progress
8	7	Develop an API	2	2023-11-08	Low	9	Completed
9	8	Optimize server	4	2023-10-09	Low	2	In Progress
10	9	Build scalable m	7	2023-10-08	Low	1	To Do
11	10	Optimize databa	6	2023-10-02	Medium	6	In Progress
12	11	Integrate third-pa	2	2023-12-02	High	2	To Do
13	12	Build a microser	5	2023-11-17	Medium	7	In Progress
14	13	Optimize databa	3	2023-12-19	Medium	5	In Progress
15	14	Develop RESTfu	3	2023-11-10	Low	1	To Do
16	15	Develop RESTfu	3	2023-10-08	Low	3	Completed
17	16	Manage databas	5	2023-11-02	Medium	10	To Do
18	17	Develop RESTfu	5	2023-12-12	Medium	2	To Do
19	18	Build scalable m	6	2023-12-01	Medium	3	In Progress
20	19	Integrate third-pa	4	2023-11-04	Medium	5	Completed
21	20	Manage databas	3	2023-10-05	Medium	7	To Do
22	21	Implement authe	3	2023-12-16	Low	7	In Progress
23	22	Build a microser	5	2023-10-23	Medium	4	Completed
24	23	Build scalable m	6	2023-12-25	High	10	Completed
25	24	Develop RESTfu	3	2023-10-06	Medium	4	Completed
26	25	Implement authe	4	2023-12-28	Medium	1	To Do
27	26	Develop RESTfu	3	2023-10-03	Low	8	In Progress
28	27	Develop an API	2	2023-11-20	Low	8	Completed
29	28	Integrate third-pa	2	2023-11-21	Medium	1	To Do
30	29	Build scalable m	5	2023-10-03	Medium	1	Completed
31	30	Implement user a	3	2023-10-02	Medium	7	To Do
32	31	Implement user a	2	2023-10-19	Medium	5	In Progress
33	32	Integrate third-pa	4	2023-12-07	Low	8	To Do
34	33	Integrate third-pa	2	2023-12-09	Medium	1	To Do
35	34	Integrate third-pa	2	2023-10-03	Medium	4	To Do

3. *Data cleaning :*






```
[7]: df.columns
```

```
[7]: Index(['Task_ID', 'Task_Name', 'Duration', 'Deadline', 'Priority',
       'Resource_ID', 'Status'],
      dtype='object')
```

```
[8]: df.head(5)
```

[8]:

	Task_ID	Task_Name	Duration	Deadline	Priority	Resource_ID	Status
0	1	Build scalable microservices	7	2023-10-15	Medium	5	To Do
1	2	Manage database operations	5	2023-12-14	Medium	5	In Progress
2	3	Optimize database queries	5	2023-10-22	Low	8	Completed
3	4	Integrate third-party services	3	2023-12-03	High	1	To Do
4	5	Integrate third-party services	2	2023-12-28	Low	10	To Do

```
[9]: df.tail()
```

[9]:

	Task_ID	Task_Name	Duration	Deadline	Priority	Resource_ID	Status
4995	4996	Optimize server performance	4	2023-10-09	Medium	10	In Progress
4996	4997	Develop RESTful APIs	5	2023-11-30	Medium	4	Completed
4997	4998	Implement user authentication	3	2023-11-22	Low	9	Completed
4998	4999	Integrate third-party services	3	2023-10-31	Low	8	To Do
4999	5000	Integrate third-party services	4	2023-12-11	High	10	To Do


```
[10]: df.describe()
```

[10]:

	Task_ID	Duration	Resource_ID
count	5000.000000	5000.000000	5000.000000
mean	2500.500000	3.998600	5.416000
std	1443.520003	1.251763	2.866529
min	1.000000	2.000000	1.000000
25%	1250.750000	3.000000	3.000000
50%	2500.500000	4.000000	5.000000
75%	3750.250000	5.000000	8.000000
max	5000.000000	7.000000	10.000000

```
[11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Task_ID      5000 non-null   int64
 1   Task_Name    5000 non-null   object
 2   Duration     5000 non-null   int64
 3   Deadline     5000 non-null   object
 4   Priority     5000 non-null   object
 5   Resource_ID  5000 non-null   int64
 6   Status       5000 non-null   object
dtypes: int64(3), object(4)
memory usage: 273.6+ KB
```

Partie 1: Nettoyage de données

Gestion des doublons

1- Detection des doublons

```
[15]: print("Nombre de doublons:", df.duplicated().sum())
```

```
Nombre de doublons: 0
```

2-Supprimer les doublons

```
[17]: df.drop_duplicates(inplace=True)
```

3-Vérifier après suppresion des doublons

```
[19]: print("Nombre de doublons:", df.duplicated().sum())
```

```
Nombre de doublons: 0
```

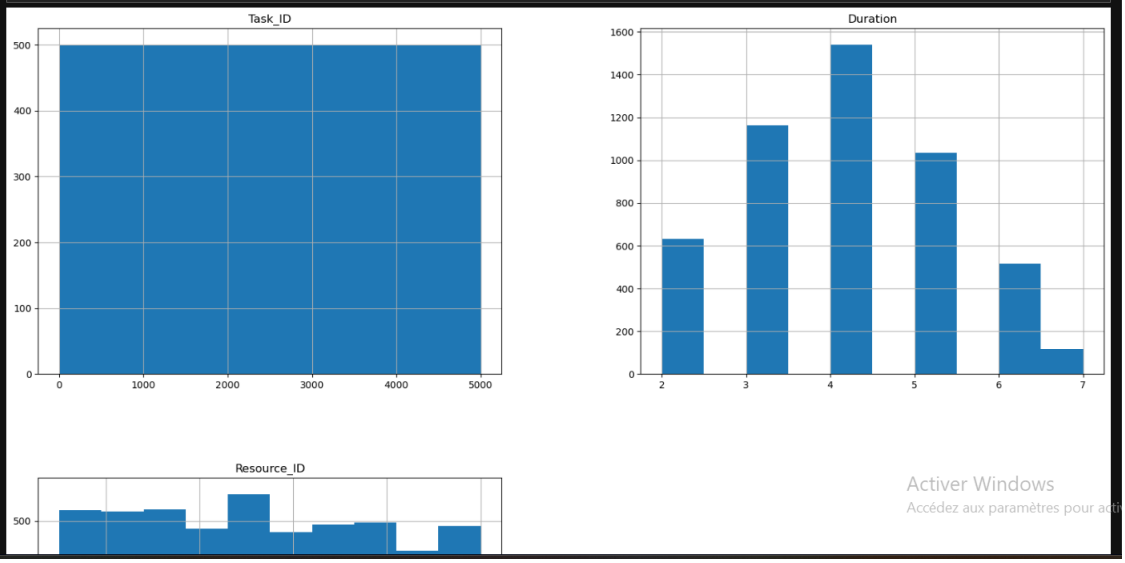
Gestion des valeurs manquantes

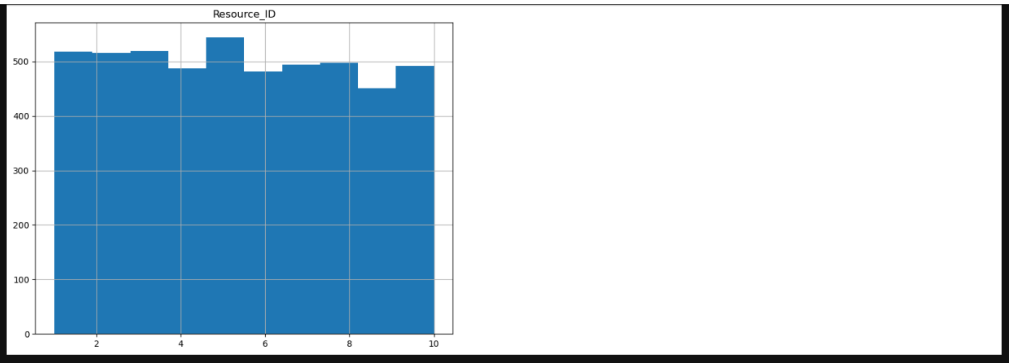
```
[21]: print("Valeurs manquantes:", df.isnull().sum())
```

```
Valeurs manquantes: Task_ID        0
Task_Name      0
Duration       0
Deadline       0
Priority       0
Resource_ID    0
Status         0
```



```
[23]: # Displying Histograms
df.hist(figsize=(20,15))
plt.show()
```





Partie 2: Transformation de données

Encodage de la variable defects

[26]:
```
# Encodage des variables catégorielles
label_encoders = {}
for col in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))
    label_encoders[col] = le
```



```
# Encodage de la variable cible
target = 'defects'  # Remplacez par le nom exact de la variable cible
if target in df.columns:
    target_encoder = LabelEncoder()
    df[target] = target_encoder.fit_transform(df[target])
    print("\nCible encodée :")
    print(df[[target]].head())
else:
    print(f"\nErreur : la cible '{target}' n'est pas dans le dataset.")
    exit()
```

```
Dataset après encodage des variables catégorielles :
   Task_ID  Task_Name  Duration  Deadline  Priority  Resource_ID  Status
0        1          1         7        14         2            5       2
1        2          7         5        74         2            5       1
2        3          8         5        21         1            8       0
3        4          6         3        63         0            1       2
4        5          6         2        88         1           10       2

Erreur : la cible 'defects' n'est pas dans le dataset.
```

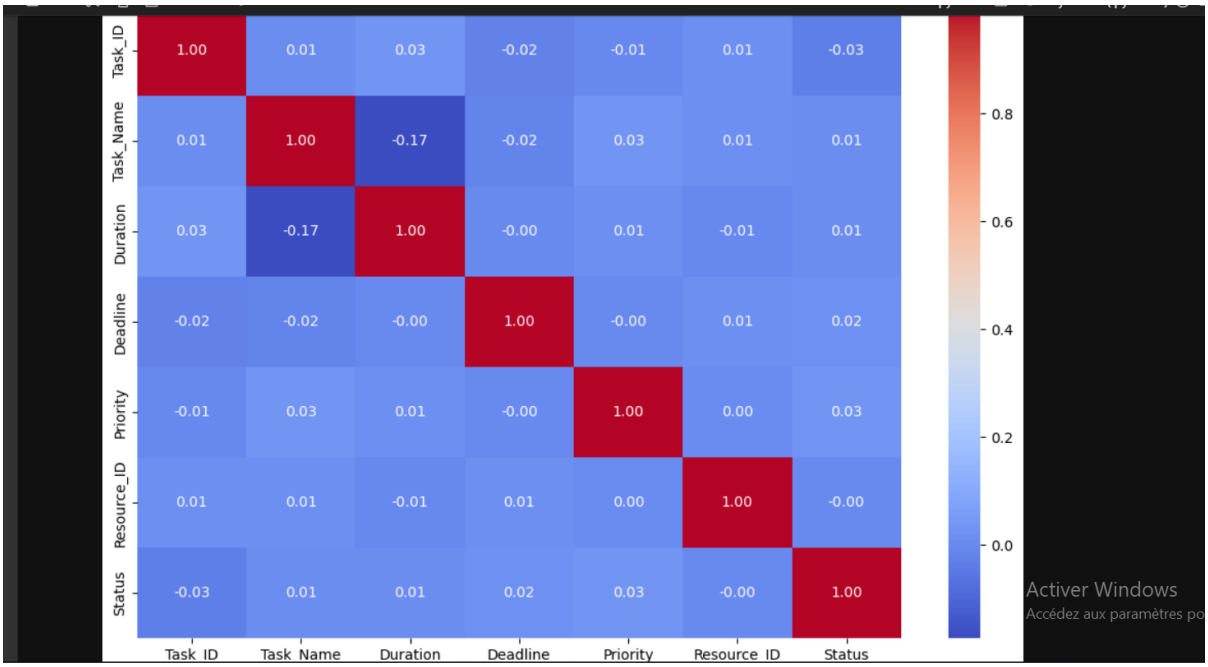
Normalisation

[28]:
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import pandas as pd
# Normalisation des données avant la matrice decorrélation
scaler = StandardScaler()
df_normalized = df.copy()
df_normalized[df.select_dtypes(include=[np.number]).columns] = scaler.fit_transform(df[df.select_dtypes(include=[np.number]).columns])
```

Partie 3: Sélection des caractéristiques

Matrice de corrélation

[31]:
```
plt.figure(figsize=(12, 8))
corr_matrix = df_normalized.corr()  # Matrice de corrélation après normalisation
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm', cbar=True)
plt.title('Matrice de corrélation')
plt.show()
```




```
[33]: # Affichage des corrélations avec la cible
target = 'defects'  # Remplacez par le nom exact de la variable cible
if target in df_normalized.columns:
    print("\nCorrélations avec la cible (avant filtrage) :")
    print(corr_matrix[target].abs())  # Affichage des corrélations absolues avec la cible

    # Sélection des caractéristiques bien corrélées avec la cible (target)
    correlation_threshold = 0.1  # Seuil de corrélation
    target_corr = corr_matrix[target].abs()
    selected_features = target_corr[target_corr > correlation_threshold].index.tolist()
    if target in selected_features:
        selected_features.remove(target)
    print(f"\nCaractéristiques sélectionnées corrélées avec {target} (|corr| > {correlation_threshold}) : {selected_features}")
else:
    print(f"\nErreur : la cible '{target}' n'est pas dans le dataset.")
    exit()

Erreur : la cible 'defects' n'est pas dans le dataset.
```

```
Erreur : la cible 'defects' n'est pas dans le dataset.

Créer un DataFrame avec les caractéristiques sélectionnées

[35]: from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df_normalized = df.copy()
df_normalized[df.select_dtypes(include=[np.number]).columns] = scaler.fit_transform(df[df.select_dtypes(include=[np.number]).columns])

print(df_normalized.head())  # All values will be between 0 and 1

   Task_ID  Task_Name  Duration  Deadline  Priority  Resource_ID  Status
0   0.0000   0.111111       1.0  0.155556       1.0     0.444444     1.0
1   0.0002   0.777778       0.6  0.822222       1.0     0.444444     0.5
2   0.0004   0.888889       0.6  0.233333       0.5     0.777778     0.0
3   0.0006   0.666667       0.2  0.700000       0.0     0.000000     1.0
4   0.0008   0.666667       0.0  0.977778       0.5     1.000000     1.0

[ ]:
```

4. *Résultats et Interprétation :*

Meilleure gestion du temps et des ressources et réduction des retards et des conflits de priorité.