Question 1 : Comme c'est souvent le cas dans les projets, le jeu de données peut nécessiter quelques manipulations pour être utilisable par une approche ML. Si tu rencontres des problèmes de qualité des données durant ta manipulation des données de Kickstarter, comment les as-tu résolus?

1. Dans le jeu de données de Kickstarter, nous avons identifié 632 observations présentant des décalages. Certains décalages étaient d'une colonne, tandis que d'autres étaient plus importants, dépassant une seule colonne. Pour résoudre ce problème, nous avons effectué des traitements, notamment la vérification du format des valeurs. En cas de non-conformité au format de la colonne, nous avons substitué la valeur par celle de la colonne suivante. Grâce à ces ajustements, nous avons pu corriger 620 observations, laissant toujours 12 observations présentant des décalages non résolus. Toutefois, pour des raisons de simplicité et compte tenu de leur faible importance relative dans la base de données, nous avons choisi de simplement les supprimer.

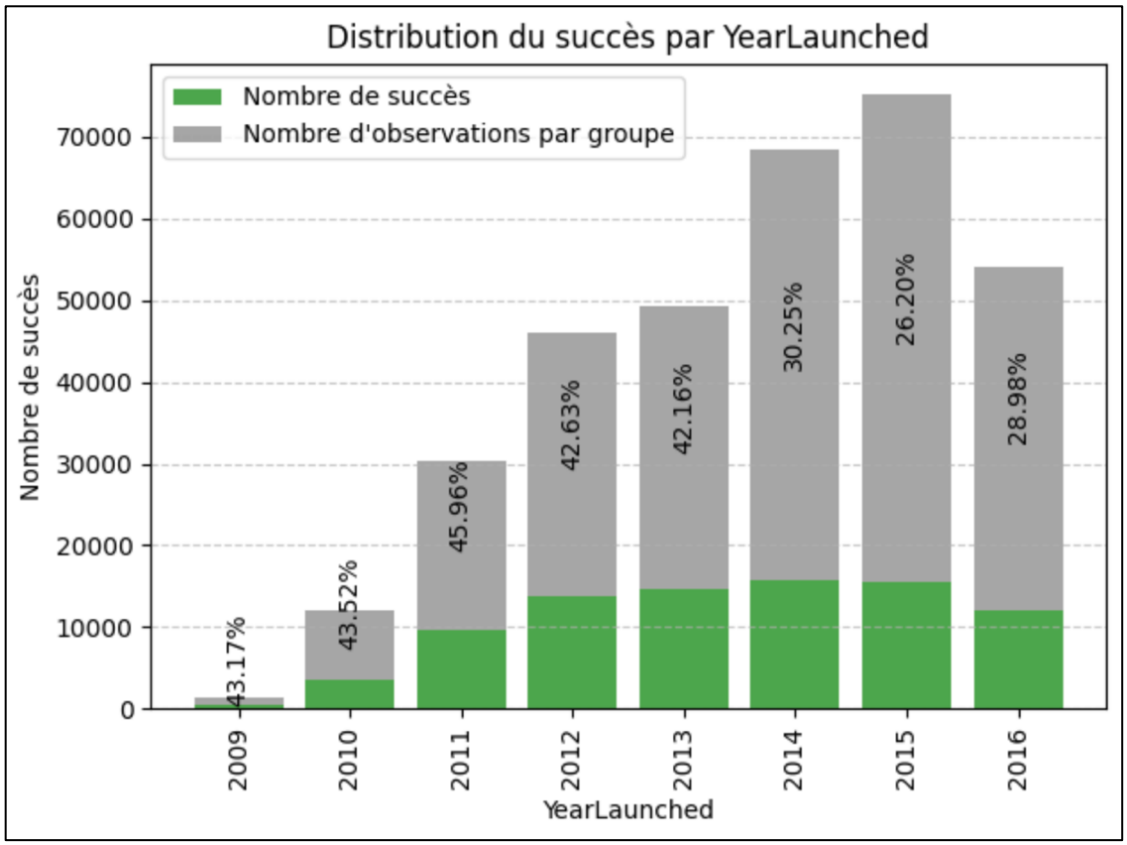
2. En ce qui concerne la variable "nom", nous avons constaté que de nombreuses données étaient entrées de manière incorrecte, présentant des caractères spéciaux, des espaces multiples et des termes inappropriés tels que « *canceled* » et « *suspended* », qui ne correspondent pas réellement à des noms. Afin de résoudre ce problème, nous avons utilisé les expressions régulières de Python et effectué une analyse de fréquence des mots pour déterminer les termes les plus couramment présents dans la colonne "nom". Par exemple, le terme "canceled" était identifié dans 23 110 observations. Nous avons donc supprimé les occurrences de "canceled" du nom de chaque observation. Ainsi, un projet intitulé « *My life project (Canceled)* » est devenu « *My life project* », ce qui reflète de manière plus précise la réalité.

3. En ce qui concerne les colonnes « catégorie » et « catégorie principale », nous avons remarqué que de nombreuses valeurs étaient très similaires les unes aux autres et auraient pu être regroupées. Cependant, afin d'éviter la complexité et la durée d'une telle tâche, nous avons opté pour une approche plus simple. Nous avons examiné le nombre d'observations pour chaque catégorie et catégorie principale, et nous avons supprimé les observations dont la catégorie était très rare (peu utilisée par les autres observations).

4. En ce qui concerne la colonne « pays », nous avons constaté que de nombreux pays étaient mal classés avec la valeur « N"0 ». Pour résoudre ce problème, nous avons dû inférer le pays à partir de la devise associée pour les observations concernées. Nous avons émis l'hypothèse qu'un projet utilisant le dollar américain comme devise provenait logiquement des États-Unis. Cependant, cette stratégie s'est avérée moins efficace pour les projets utilisant l'euro comme devise, car tous les pays de la zone euro partagent la même devise. Malgré cette difficulté, cette approche nous a permis de récupérer de nombreuses valeurs manquantes avec un niveau de confiance approprié.

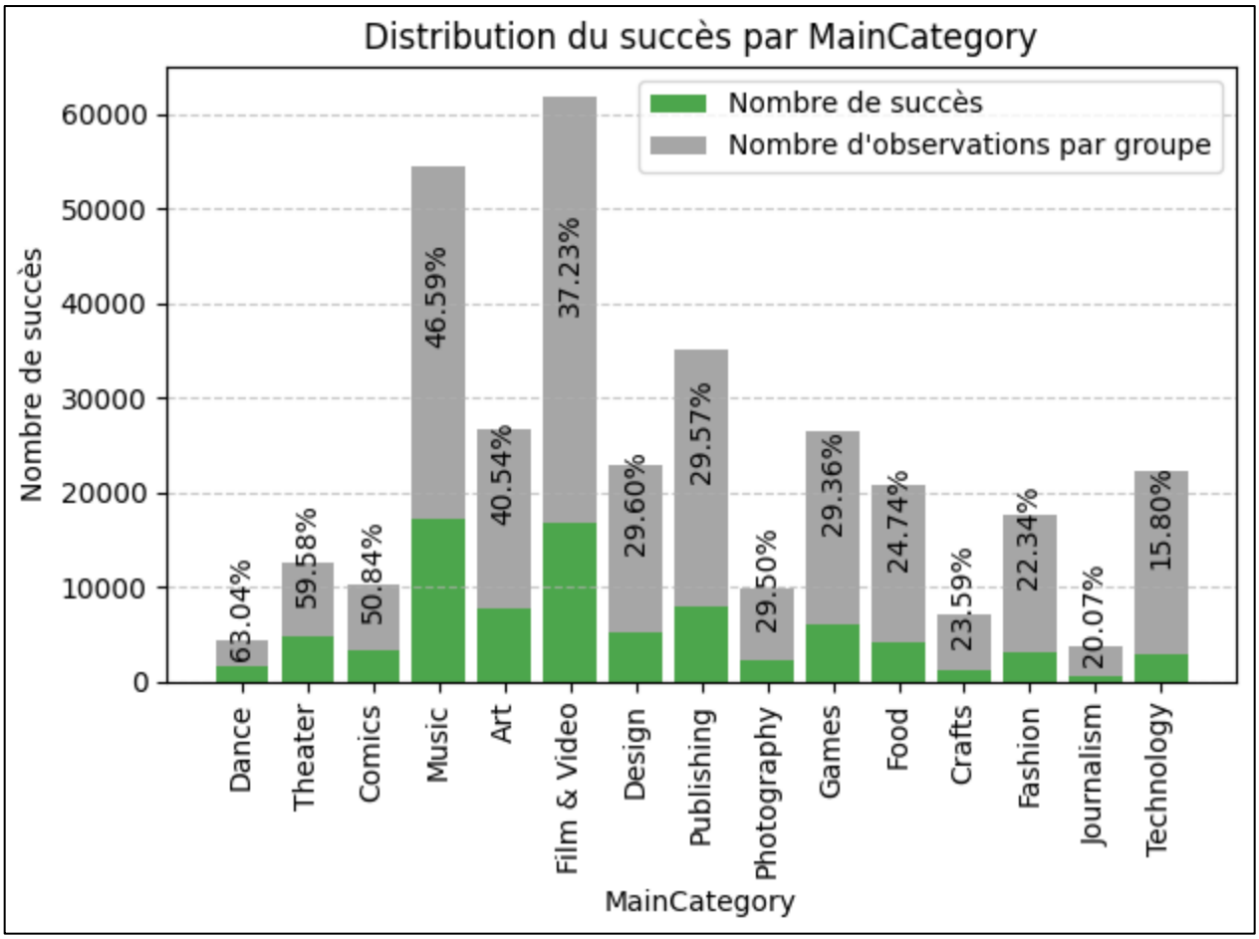
Question 2 : Identifie des « insights » qui, selon toi, peuvent contribuer à comprendre le succès ou non des campagnes.

1. La proportion de succès semble être affectée par l'année de lancement du projet. Nous pensons que cela peut être attribué à des facteurs macroéconomiques externes, tels qu'un ralentissement de l'économie mondiale. Lorsque la situation financière est moins favorable, les individus sont moins enclins à financer des projets sur Kickstarter et deviennent plus prudents avec leurs finances. Par exemple, une diminution notable de la proportion de succès est observée de 2014 à 2016, correspondant à une période où le prix du pétrole, initialement élevé au début des années 2010, a entamé une forte baisse à partir de mi-2014.



Cela ajoute à la pertinence d'ajouter des données macroéconomiques, que nous n'avons pas fait puisqu'il s'agit d'un mandat fictif, dont le but est une évaluation de compétence.

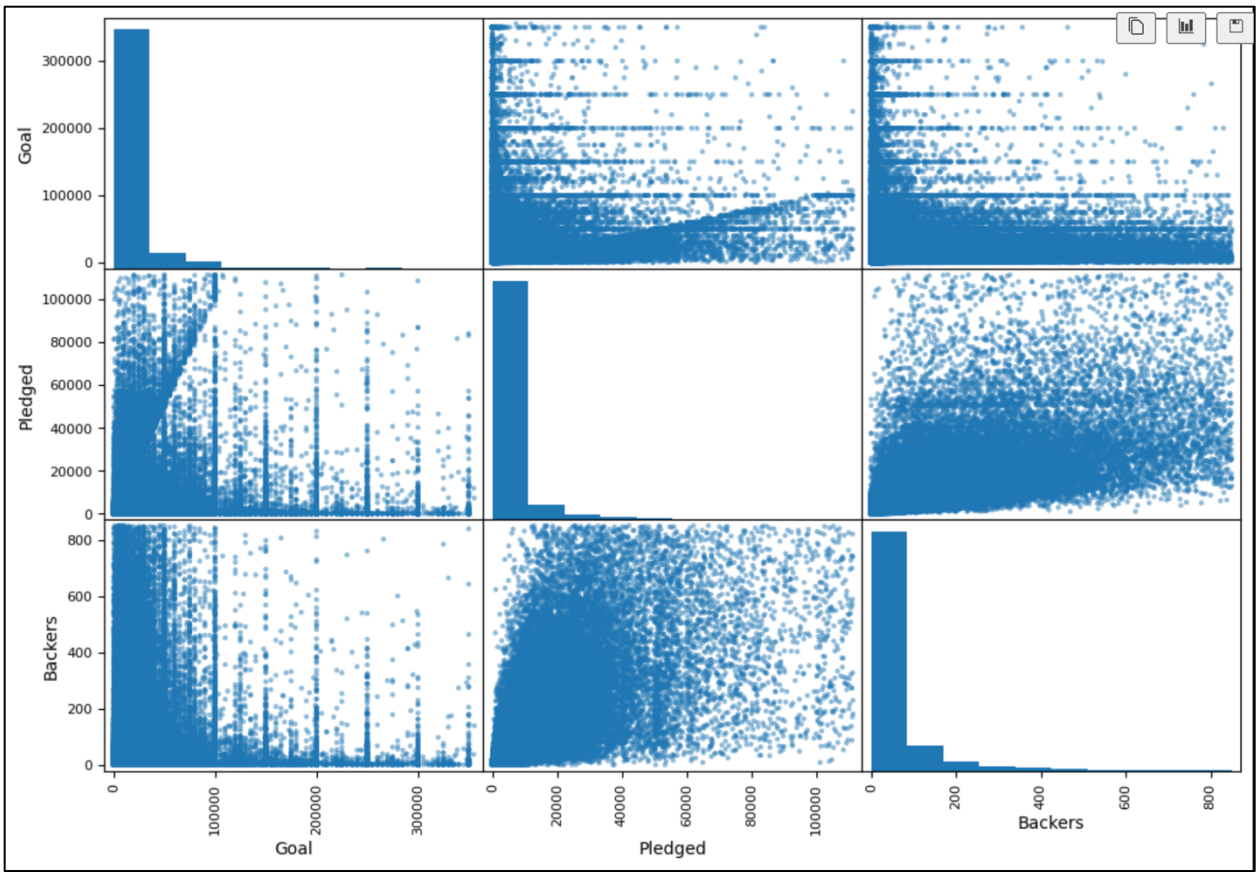
2. Les catégories principales telles que « musique », « art » et « film & vidéo » se démarquent par une proportion de succès notable par rapport aux autres catégories principales, tout en prenant en compte le fait qu'elles sont parmi les catégories les plus représentées en termes de volume. Cette observation est renforcée par la loi des grands nombres, ce qui accroît la crédibilité statistique de ce constat.



3. Une corrélation significative est observée entre les variables « *Pledged* » (montant amassé) et « *Backers* » (supporteurs), ainsi qu'avec le taux de réussite (« *State* » égal à « *successful* ». Il est raisonnable de supposer que plus un projet bénéficie d'un grand nombre de supporteurs, plus il accumule de fonds, ce qui augmente les chances de réussite. Voici une représentation de la corrélation avec la variable Succès, avec les variables concernées et leur transformation logarithmiques et racine carrée.

Success	1.000000
log_Backers	0.642009
sqrt_Backers	0.621533
log_Pledged	0.584147
sqrt_Pledged	0.574444
Backers	0.459074
Pledged	0.388007

De même, nous avons également testé la corrélation entre « *Pledged* » et « *Backers* ». Cette corrélation s'élève à 74,98%. Voici, une représentation visuelle qui le démontre.



Question 2.1 : Basé sur les insights mentionnés plus haut, y a-t-il un risque que des « *confounding variables* » (i.e. facteur de confusion) viennent affecter l'interprétation de tes observations ?

En effet, les variables « Pledged » (montant amassé) et « Backers » (supporteurs) présentent une forte multicolinéarité, ce qui signifie qu'elles fournissent potentiellement la même information et que leur explication de la variable indépendante est redondante. L'inclusion des deux variables dans le modèle pourrait entraîner une complexité inutile et conduire à un surajustement (*overfitting*). C'est pourquoi il est essentiel de repérer et de traiter les variables de confusion potentielles dans un projet d'apprentissage automatique de classification binaire, afin de construire un modèle à la fois robuste et interprétable. Dans notre modèle, nous avons d'abord intégré un facteur d'interaction et avons supprimé les variables individuelles de notre jeu de données. Nous les avons ensuite complètement retiré du modèle puisque leur incidence sur la variable indépendante était trop forte.

Question 2.2 : Est-ce que les « *insights* » trouvés peuvent être transformés en « *features* » qui faciliteront l'apprentissage du modèle ML?

Les « *insights* » trouvés peuvent être transformés en « *features* », par le biais d'un custom class que nous implémentons dans notre code et qui va comme suit :

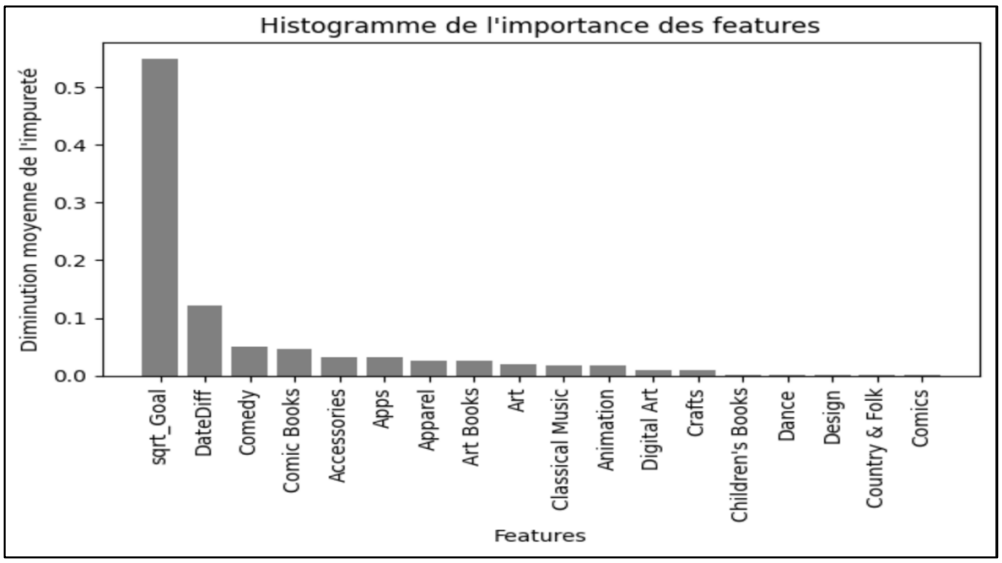
```python
# Crée une classe personnalisée pour effectuer les transformation nécessaires et créer des features numériques.
from sklearn.base import BaseEstimator, TransformerMixin

class FeaturesGenerator(BaseEstimator, TransformerMixin):
    def __init__(self, add_date_diff = True, sqrt_goal = True, sqrt_backers = True, interaction_sqrt=True):
        self.add_date_diff = add_date_diff
        self.sqrt_goal = sqrt_goal
        self.sqrt_backers = sqrt_backers
        self.interaction_sqrt = interaction_sqrt

    def fit(self, X, y=None):
        return self

    def transform(self, X, y=None):
        # Ajoute la variable DateDiff, si True
        if self.add_date_diff:
            for column in ['Deadline','Launched']:
                X[column] = pd.to_datetime(X[column], format='%Y-%m-%d %H:%M:%S')

            X['DateDiff'] = (X['Deadline'] - X['Launched']).dt.days

        # Ajoute la variable sqrt_Goal, si True
        if self.sqrt_goal:
            X['sqrt_Goal'] = np.sqrt(X['Goal'].astype(int))

        # Ajoute la variable sqrt_Backers, si True
        if self.sqrt_backers:
            X['sqrt_Backers'] = np.sqrt(X['Backers'].astype(int))

        # Ajoute la variable sqrt_Pledged_Backers, si True
        if self.interaction_sqrt:
            X['sqrt_Pledged_Backers'] = np.sqrt(X['Backers'].astype(int))*np.sqrt(X['Pledged'].astype(int))

        X = X.drop(columns=['ID','Name','Deadline','Goal','Launched','Pledged'])

        return X
```

Dans cette classe nous créons quatre variables numériques, soit la différence entre les dates de lancement et les dates d'échéance « *DateDiff* », la racine carrée de « *Goal* », la racine carrée de « *Backers* » et le produit d'interaction entre les racines carrées de « *Backers* » et « *Pledged* ». Ces transformations ont pour but de normaliser les valeurs extrêmes par une transformation en racine carrée, de réduire les facteurs de confusion par le produit d'interaction et de donner une profondeur supplémentaire à l'analyse en testant des variables potentiellement significatives pour expliquer la variable indépendante.

Question 3 : En tenant compte des parties prenantes visées par ta solution, comment interprètes-tu les résultats produits par ta solution ML ? Comment cette solution ajoute-t-elle de la valeur pour ces parties prenantes? Selon toi, comment envisage-tu que les parties prenantes vont utiliser ta solution pour tenter de comprendre comment lancer des campagnes à haut taux de succès?

Étant donné que notre solution d'apprentissage machine visait à offrir des informations claires aux créateurs de campagnes sur Kickstarter, il était essentiel de choisir un modèle facilement explicable. Dans notre cas, le *DecisionTreeClassifier* s'est révélé être le choix optimal, étant à la fois le modèle le plus performant et le plus simple. Le modèle a une précision de 0,62, ce qui signifie que parmi les campagnes qu'il prédit comme étant réussies, 62 % d'entre elles le sont réellement. Cela suggère que le modèle est relativement précis lorsqu'il prédit le succès des campagnes, mais il y a encore une marge d'erreur de 38 %. Le rappel est de 0,33, ce qui signifie que le modèle ne parvient à détecter que 33 % des campagnes réellement réussies. Il manque donc un certain nombre de campagnes qui auraient dû être identifiées comme réussies. Cela peut indiquer que le modèle est moins sensible à la détection des campagnes réussies. Une solution potentielle à ce genre de problème pourrait être le rééquilibrage des classes de la variable indépendante. Sachant qu'uniquement 35% des observations sont des succès, nous pourrions *oversample* cette sous-classe pour accroître leur représentativité et ainsi améliorer le rappel.

Pour parvenir à notre modèle final nous avons utilisé la méthode SelectFromModel de Scikit-learn, qui permet une sélection des n nombre de variables explicatives les plus pertinentes sur la base d'un *threshold* préétabli. Pour ce faire, nous avons testé 100 *threshold* différents et en avons sélectionné un qui maximise la performance du modèle. Ainsi, notre modèle final comporte 18 variables qui furent classées, en ordre d'importance, par la fonction *feature_importances_*. Cette fonction indique à quel point une variable explicative réduit en moyenne l'incertitude ou l'impureté relative à la variable indépendante. Voici une représentation visuelle des 18 variables sélectionnées :



Cette solution ajoute la valeur aux parties prenantes justement par l'analyse des variables explicatives les plus importantes, du point de vue de leur impact sur la variable indépendante, comme représenté dans la visualisation précédente. Les parties prenantes peuvent prendre des décisions sur ce genre de visualisation, en assumant que la caractéristique la plus importante pour la prédiction du succès des campagnes Kickstarter est la racine carrée du montant cible de financement. Plus précisément, une valeur plus élevée de *sqrt_Goal* (et de *Goal* par conséquent) tend à être associée à une campagne plus susceptible de réussir. De plus, La durée de la campagne en jours est la deuxième caractéristique la plus importante, mais elle a un poids moins élevé que *sqrt_Goal*. Une durée de campagne plus longue (valeur plus élevée de *DateDiff*) semble contribuer positivement à la prédiction du succès. Ce qui est particulier avec cette variable est qu'elle est extraite de notre étape de *feature engineering*, où nous avons créé des variables supplémentaires à partir des données disponibles. Finalement, il y a une liste de catégories avec leurs importances relatives. En considérant les catégories les plus importantes pour la prédiction du succès, comme « *Comédie* » par exemple, un créateur de campagne pourrait être incité à lancer davantage de projet dans cette catégorie pour maximiser ses chances de réussite.

Question 4 : Imaginons que ta solution est déployée et roule maintenant en production. Tu remarques que la performance de ton modèle se dégrade progressivement depuis les derniers mois. De plus, tu identifies également certaines variables dont les valeurs semblent avoir évoluées durant la même période. Selon toi, quel serait une raison qui explique cette situation et comment la ressouderais-tu ?

Pour assurer la qualité pérenne d'un modèle, nous devons rédiger du code de surveillance pour vérifier la performance en direct et à des intervalles réguliers, puis déclencher des alertes en cas de baisse de performance. Il s'agit d'une pratique assez courante car les modèles ont tendance à se « dégrader » à mesure que les données évoluent avec le temps, à moins qu'ils ne soient régulièrement entraînés avec des données fraîches. Voici quelques étapes que je ferais :

1. **Analyse des données de performance** : Je commencerais par examiner de près les données de performance du modèle au fil du temps. Cela inclurait l'analyse des métriques clés telles que la précision, le rappel, la F1-score, etc., pour identifier la période exacte de dégradation et les variables spécifiques qui semblent affectées.

2. **Analyse des variables** : J'examinerais ensuite les variables dont les valeurs ont évolué pendant la même période. Je vérifierais s'il s'agit de variables d'entrée directe pour le modèle ou de variables liées à l'environnement ou aux données. Si les variables semblent avoir évolué de manière significative par rapport à leur comportement antérieur, j'utiliserais des techniques de détection d'anomalies pour identifier des points de données ou des périodes qui pourraient être à l'origine du problème.

3. **Réévaluation du modèle** : Si j'identifie des variables liées au modèle lui-même, je réévaluerais le modèle pour voir s'il est toujours adapté aux données actuelles. Cela pourrait nécessiter une révision de l'architecture du modèle, une mise à jour des hyperparamètres ou une révision de la stratégie de prétraitement des données.