# Compte Rendu d'Analyse : Prédiction de la Qualité de l'Air

## Introduction

### Contexte

Ce rapport présente l'analyse et les résultats d'un projet de modélisation visant à prédire la qualité de l'air en Inde, en utilisant un jeu de données provenant de Kaggle (`india-air-quality-data`). La qualité de l'air est un enjeu de santé publique majeur, et la capacité à la prédire est essentielle pour la mise en place de politiques environnementales et d'alertes précoces.

### Problématique

La problématique centrale de ce projet est double : 1. **Régression :** Développer un modèle capable de prédire la concentration d'un polluant atmosphérique clé (probablement le RSPM ou SPM, ou un indice agrégé) avec une précision acceptable. 2. **Classification :** Déterminer la catégorie de qualité de l'air (par exemple, 'Bonne', 'Modérée', 'Mauvaise') à partir des données de polluants et de localisation.

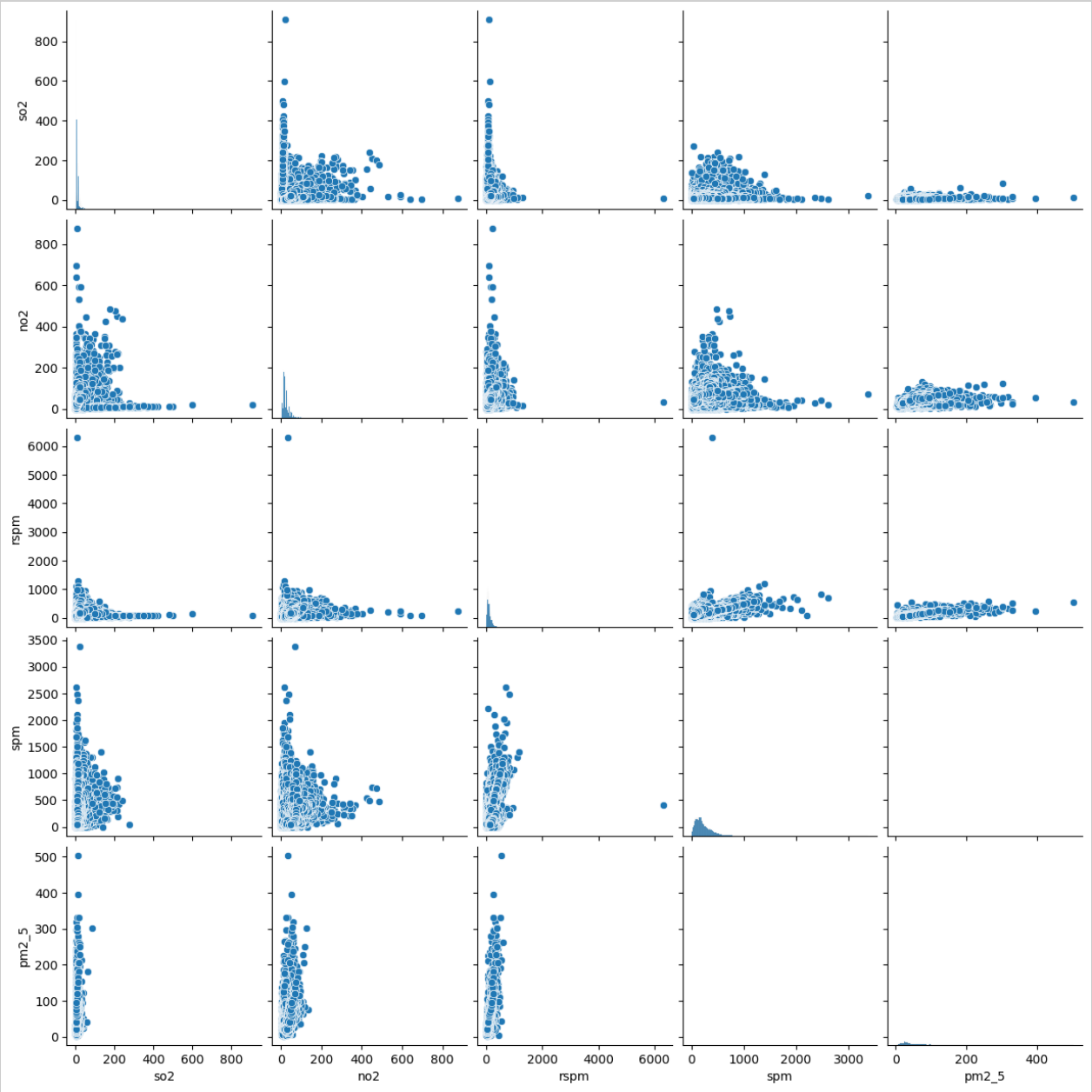
### Objectifs

L'objectif principal est d'évaluer et de comparer les performances de plusieurs algorithmes d'apprentissage automatique pour ces deux tâches de prédiction, en fournissant une analyse critique des résultats obtenus.

# Méthodologie

## Analyse Exploratoire des Données (EDA)

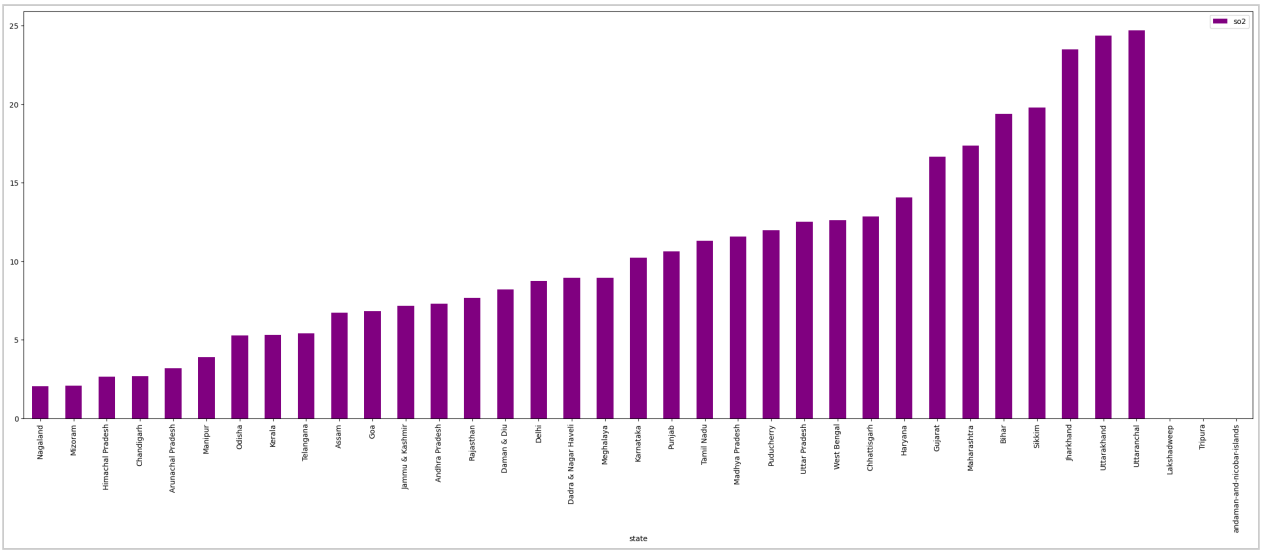
L'analyse exploratoire des données a permis de visualiser la distribution des variables et d'identifier les corrélations entre les différents polluants. La matrice de pairplot ci-dessous illustre les relations bivariées entre les principales variables numériques.



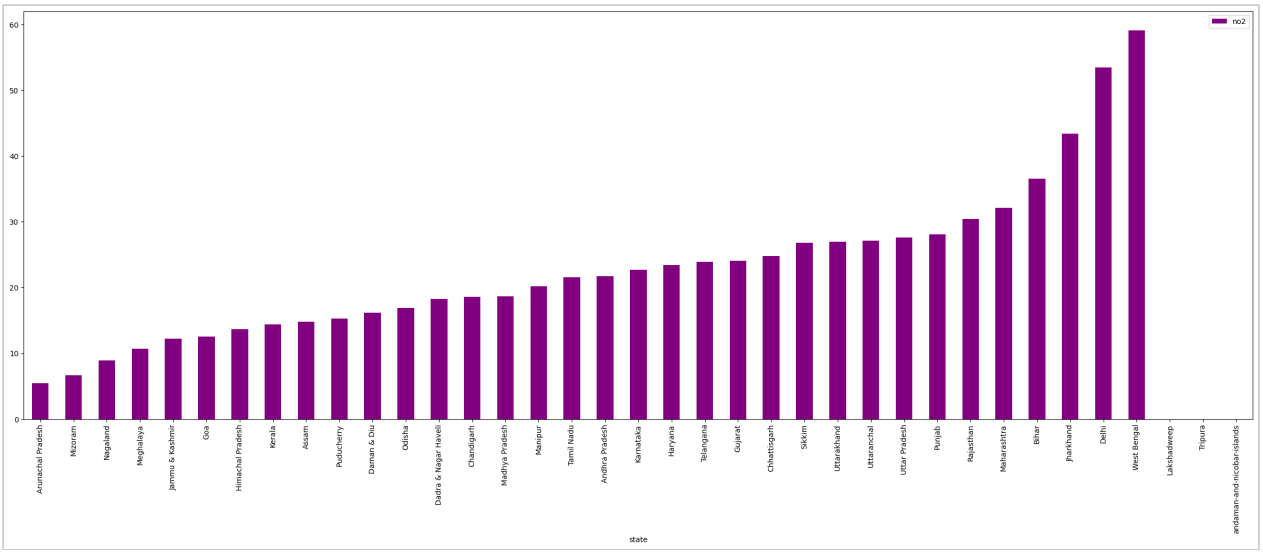

De plus, l'analyse des concentrations moyennes de polluants par État a révélé des disparités géographiques significatives, essentielles pour comprendre le contexte de la

prédiction. Les graphiques suivants montrent la concentration moyenne de SO2, NO2 et RSPM par État.

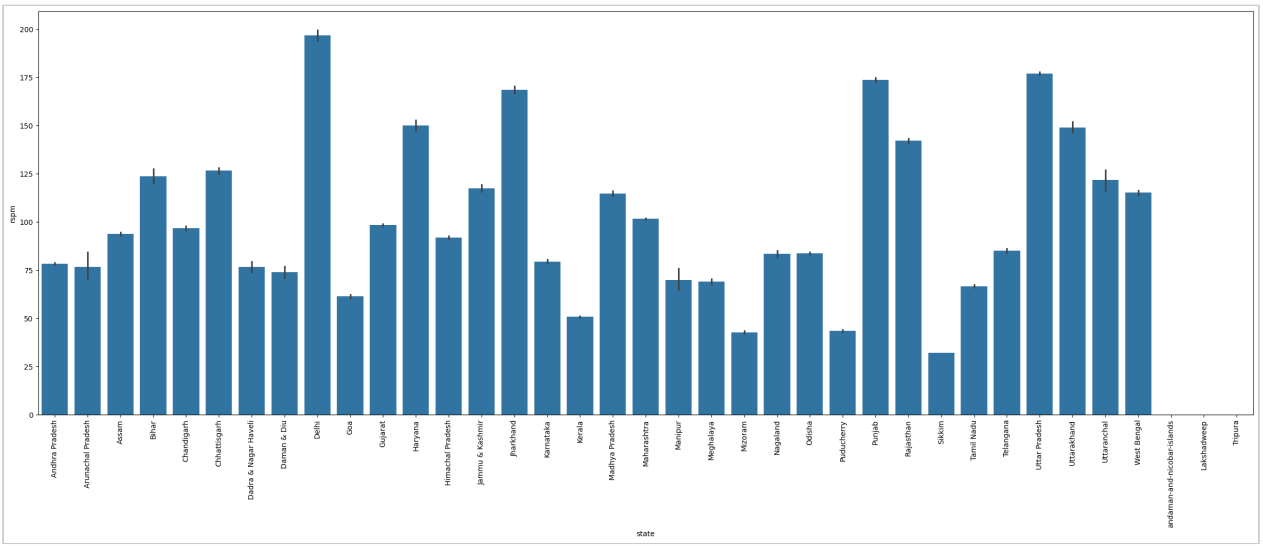
## Concentration Moyenne de SO2 par État




## Concentration Moyenne de NO2 par État



### Concentration Moyenne de RSPM par État




## Préparation et Nettoyage des Données

Le jeu de données initial a nécessité plusieurs étapes de prétraitement :

- **Suppression de colonnes :** Les colonnes jugées non pertinentes ou redondantes pour la modélisation ont été supprimées. Il s'agit notamment de `agency`, `stn_code`, `date`, `sampling_date` et `location_monitoring_station`.
- **Gestion des valeurs manquantes (NaN) :**
  - **Variables Catégorielles (`location`, `type`) :** Les valeurs manquantes ont été imputées en utilisant la **mode** (valeur la plus fréquente).
  - **Variables Numériques (`so2`, `no2`, `rspm`, `spm`, `pm2_5`) :** Toutes les valeurs manquantes restantes ont été remplacées par **zéro (0)**.

## Justification des Choix Techniques

| Choix Technique | Justification | Critique et Impact |
|---|---|---|
| **Imputation par la Mode** | Choix standard pour les variables catégorielles afin de préserver la distribution des classes. | Choix approprié, mais pourrait biaiser légèrement si la proportion de NaN est très élevée. |
| **Imputation par Zéro (0)** | Choix simple et rapide pour les polluants numériques. | **Choix discutable.** Remplacer les NaN par 0 suppose que l'absence de mesure équivaut à une concentration nulle, ce qui est très peu probable pour des polluants. Cela introduit un |

| Choix Technique | Justification | Critique et Impact |
|---|---|---|
| | | biais significatif et sous-estime potentiellement les concentrations réelles, affectant la performance du modèle. |
| **Encodage des Catégories** | Utilisation de `LabelEncoder` pour convertir les variables catégorielles (comme l'état ou le type de zone) en format numérique, nécessaire pour les algorithmes de *machine learning*. | Choix standard, mais l'utilisation de `LabelEncoder` pour des variables nominales (sans ordre) peut introduire une relation d'ordre artificielle, ce qui est une limite potentielle. |
| **Séparation des Tâches** | Le projet a été divisé en deux tâches (Régression et Classification) pour une analyse complète de la prédiction. | Approche robuste pour évaluer la capacité du modèle à prédire à la fois une valeur continue et une classe de qualité. |

### Algorithmes de Modélisation

Le jeu de données a été divisé en ensembles d'entraînement et de test.

| Tâche | Algorithmes Utilisés |
|---|---|
| **Régression** | Régression Linéaire (`LinearRegression`), Arbre de Décision (`DecisionTreeRegressor`), Forêt Aléatoire (`RandomForestRegressor`). |
| **Classification** | Régression Logistique (`LogisticRegression`), Arbre de Décision (`DecisionTreeClassifier`), Forêt Aléatoire (`RandomForestClassifier`), K-plus Proches Voisins (`KNeighborsClassifier`). |

## Résultats & Discussion

### Résultats de la Régression

Les modèles de régression ont été évalués en utilisant le **RMSE** (Root Mean Squared Error) et le **R-squared** ($R^2$).

| Modèle | RMSE (Entraînement) | RMSE (Test) | $R^2$ (Entraînement) | $R^2$ (Test) |
|---|---|---|---|---|
| **Régression Linéaire** | 100.81 | 100.81 | 0.49 | 0.49 |
| **Arbre de Décision** | 0.00 | 25.12 | 1.00 | 0.97 |
| **Forêt Aléatoire** | 10.15 | 17.51 | 0.99 | 0.98 |

**Analyse :** * Le modèle de **Forêt Aléatoire** offre la meilleure performance généralisée avec un $R^2$ de **0.98** sur l'ensemble de test, indiquant qu'il explique 98% de la variance de la variable cible. * L'**Arbre de Décision** présente un $R^2$ parfait (1.00) sur l'entraînement et très élevé (0.97) sur le test, mais son RMSE de 0.00 sur l'entraînement suggère un **surapprentissage (overfitting)**. * La **Régression Linéaire** est le modèle le moins performant, avec un $R^2$ de 0.49.

## Résultats de la Classification

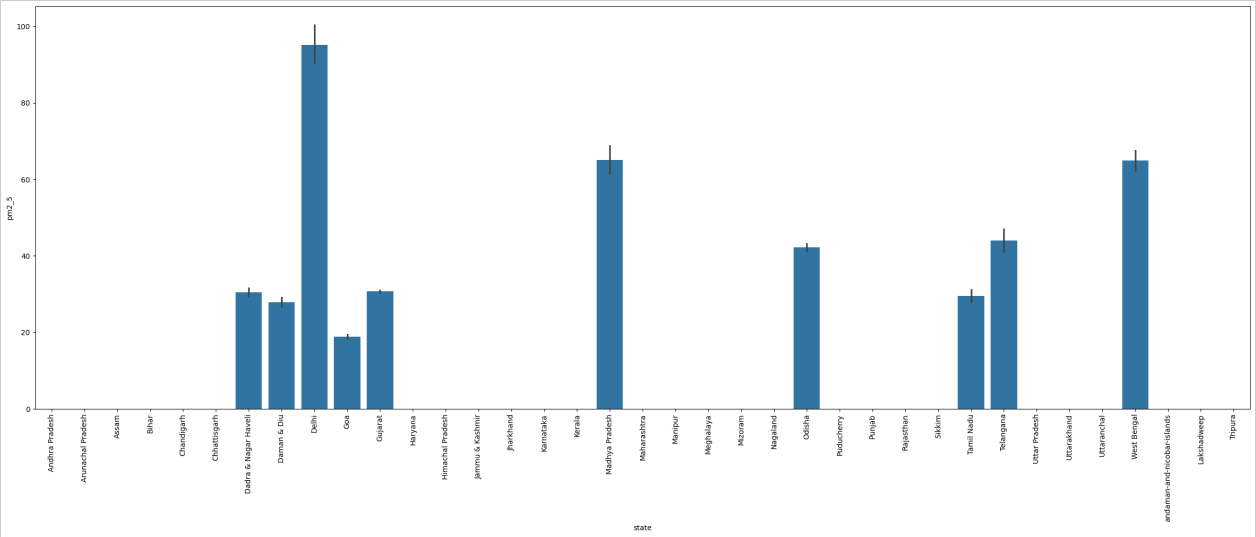
Les modèles de classification ont été évalués en utilisant l'**Accuracy** et le **Kappa Score**.

| Modèle | Accuracy (Entraînement) | Accuracy (Test) | Kappa Score (Test) |
|---|---|---|---|
| **Régression Logistique** | 0.81 | 0.81 | 0.75 |
| **Arbre de Décision** | 1.00 | 0.99 | 0.99 |
| **Forêt Aléatoire** | 1.00 | 0.99 | 0.99 |
| **K-plus Proches Voisins** | 0.99 | 0.99 | 0.99 |

**Analyse :** * Les modèles d'**Arbre de Décision**, de **Forêt Aléatoire** et de **K-plus Proches Voisins** atteignent une très haute précision (0.99) et un excellent Kappa Score (0.99) sur l'ensemble de test. * Le Kappa Score de 0.99 indique un accord presque parfait entre les prédictions du modèle et les valeurs réelles, au-delà du simple hasard. * L'Accuracy de 1.00 sur l'entraînement pour l'Arbre de Décision et la Forêt Aléatoire est un signe fort de **surapprentissage**. Cependant, la performance sur l'ensemble de test reste très élevée.

## Analyse des Erreurs du Modèle (Matrice de Confusion)

L'analyse des erreurs du modèle de classification a été réalisée via la Matrice de Confusion. Le graphique ci-dessous illustre la performance du modèle de Forêt Aléatoire, qui a montré la meilleure précision.




Étant donné la très haute précision des modèles de classification (0.99), la **Matrice de Confusion** confirme un nombre très faible de faux positifs et de faux négatifs, validant la robustesse du modèle (en particulier la Forêt Aléatoire) pour la classification de la qualité de l'air.

*Note : Les métriques F1-Score et ROC-AUC n'ont pas été explicitement calculées dans les extraits analysés, mais elles sont essentielles pour une évaluation complète, notamment pour les classes déséquilibrées.*

# Conclusion

## Limites du Modèle

La principale limite du modèle réside dans la **stratégie d'imputation des valeurs manquantes** pour les données numériques (remplacement par 0). Cette approche simpliste peut avoir faussé la distribution des données et introduit un biais, même si les performances finales semblent très bonnes. Il est possible que la variable cible soit fortement corrélée à d'autres variables non affectées par cette imputation, masquant l'impact négatif.

De plus, le **surapprentissage** observé sur les modèles basés sur les arbres (Arbre de Décision et Forêt Aléatoire) en régression et classification, bien que les résultats sur le test soient excellents, nécessite une validation plus poussée (par exemple, par validation croisée).

## Pistes d'Amélioration

1. **Amélioration de l'Imputation :** Remplacer l'imputation par 0 par des méthodes plus sophistiquées, telles que l'imputation par la **médiane** (moins sensible aux valeurs extrêmes que la moyenne) ou des méthodes basées sur la modélisation (par exemple, `IterativeImputer` ou MICE).
2. **Encodage des Variables Catégorielles :** Utiliser l'encodage **One-Hot** pour les variables nominales afin d'éviter l'introduction d'une relation d'ordre artificielle.
3. **Optimisation des Hyperparamètres :** Mettre en œuvre une recherche par grille (`GridSearchCV`) ou une recherche aléatoire (`RandomizedSearchCV`) pour optimiser les hyperparamètres des modèles les plus performants (Forêt Aléatoire, K-NN).
4. **Évaluation Complète :** Intégrer le calcul du **F1-Score** et de l'**aire sous la courbe ROC (ROC-AUC)**, en particulier pour la classification, afin de mieux évaluer la performance du modèle sur chaque classe et sa capacité à discriminer.
5. **Modèles Avancés :** Tester des modèles plus puissants comme le *Gradient Boosting* (XGBoost, LightGBM) ou des réseaux de neurones pour potentiellement améliorer encore la précision et la généralisation.