Prédiction du Niveau de la Pollution Atmosphérique des Villes

Rapport

de

Projet d'Application de 4ème année de l'ESME Sudria, Spécialité IA

Effectué par :

Antoine Ghidini

Laetitia Kamwag

Sebila Doubaeva

Encadrant :

Alessandro Leite

Remerciements

Nous tenons à remercier Alessandro Leite, notre encadrant, pour nous avoir accompagné tout au long du projet. Malgré un début balbutiant, il a été d'une grande aide en nous documentant et en prenant le temps de nous expliquer les concepts necessaires pour le projet. Il a su nous remettre sur le droit chemin et nous permettre de finaliser le projet.

Nous remercions également M.Maidi, notre professeur, pour cette introduction au Machine Learning et à la construction d'un modèle.

Résumé

Ce rapport représente le résultat d'un projet d'application réalisé dans le cadre de la quatrième année visant à utiliser l'apprentissage automatique pour prédire les niveaux de pollution atmosphérique en combinant des données sur les polluants atmosphériques et macroéconomiques. Étant donné que toute la problématique est basée sur l'utilisation de techniques d'apprentissage automatique, une grande partie du contenu de ce rapport est axée sur les questions d'exploitation des données, dont la compréhension est la clé pour le choix ultérieur des méthodes. Diverses méthodes d'apprentissage automatique seront ensuite examinées, notamment la forêt aléatoire, les arbres de décision et les techniques de regroupement utilisant des ensembles de données sur les émissions atmosphériques.

- Projet GitHub : `air-pollution-prediction`
- Application Heroku : `air-pollution-prediction-app`

Table des matières

Acronymes

AEE Agence européenne pour l'environnement. 2, 5, 7, 17, 30

AI Artificial Intelligence. 4

clrtap_data CLRTAP_NVFR14_V21_GF.csv. 8, 9, 16, 27, 30

E-PRTR European Pollutant Release and Transfer Register. 7

EDA Exploratory Data Analysis. 8, 16, 30

f1_4_data F_1_4 Detailed releases at facility level with E-PRTR Sector and Annex I Activity detail into Air.csv. 8, 12, 17, 26, 29, 30

LCPs Large Combustion Plants. 7

LRTAP Convention Convention on Long-range Transboundary Air Pollution. 7–9

ML Machine Learning. 3, 4, 16

NEC National Emission Ceilings. 7, 8

Chapitre 1

Introduction

1.1 Formation de la pollution atmosphérique

Les phénomènes naturels (éruptions volcaniques, incendies de forêts...) mais surtout les activités humaines industrielles sont à l'origine d'émissions de polluants, sous forme de gaz ou de particules, dans l'atmosphère. Une fois émises dans l'air, ces substances sont transportées sous l'effet du vent, de la pluie, des gradients de températures dans l'atmosphère et cela parfois jusqu'à des milliers de kilomètres de la source d'émission. Elles peuvent également subir des transformations par réactions chimiques sous l'effet de certaines conditions météorologiques (chaleur, lumière, humidité...) et par réactions dans l'air entre ces substances. Il en résulte l'apparition d'autres polluants. [gouvernement.fr, 2022].

1.2 Effets de la pollution atmosphérique

Les niveaux de pollution de l'air restent dangereusement élevés dans de nombreuses parties du monde. Selon les données de l'Organisation mondiale de la Santé (OMS) 9 personnes sur 10 respirent un air contenant des niveaux élevés de polluants [OMS, 2018]. Les particules pénètrent profondément dans les poumons et dans le système cardiovasculaire, ce qui cause des affections comme les accidents vasculaires cérébraux, les

cardiopathies, les cancers du poumon, les bronchopneumopathies chroniques obstructives et les infections respiratoires, notamment la pneumonie. Les dernières estimations révèlent que 7 millions de personnes meurent chaque année, cependant, près de l'ensemble de la population mondiale (99%) respirent un air qui dépasse les limites fixées. Un nombre record de plus de 6 000 villes dans 117 pays surveillent désormais la qualité de l'air, toutefois les habitants de ces villes y respirent toujours des niveaux dangereux de particules fines et de dioxyde d'azote, les populations vivant dans des pays à revenu faible ou intermédiaire étant les plus exposées. [OMS, 2022].

FIGURE 1.1 – La pollution dans certaines villes peut atteindre des niveaux déplorables

1.3 Objectifs et hypothèses

Dans le cadre de ce projet, nous nous intéresserons aux données fournies par l'Agence européenne pour l'environnement (AEE). L'objectif principal du projet est de prédir l'évolution de la pollution atmosphérique dans les villes en utilisant ces données ainsi

que les différentes techniques d'apprentissage automatique (ML). Le deuxième (mais tout aussi important) objectif sera l'analyse des données obtenues, issue de la visualisation des datasets.

Sur la base des sources de données précédentes, nous pouvons faire des hypothèses concernant les polluants ainsi que les secteurs géographiques où les émissions sont les plus élevées. Le niveau de pollution de l'air est inclus dans le calcul du PIB, par conséquent, nous nous attendons à ce que les pays ayant un PIB moins élevé aient potentiellement un niveau de pollution plus élevé. Cependant, l'industrialisation joue un rôle plus clé, donc, nous pouvons tout à fait faire face à un pays développé avec des niveaux élevés de pollution, comme potentiellement l'Allemagne, connue pour l'industrie automobile et électrique. Mais finallement, quel que soit le résultat, nous le découvrirons plus loin dans le projet.

Chapitre 2

Méthodologie de travail

Dans le cadre de ce projet d'application nous souhaitons obtenir des predictions pour le niveau des émissions, cela implique l'utilisation d'un domaine scientifique tel que Machine Learning (ML) ou l'apprantissag automatique. Le ML est une forme d'Artificial Intelligence (AI) qui est axée sur la création de systèmes qui apprennent, ou améliorent leurs performances, en fonction des données qu'ils traitent [Oracle, 2022].

Le domaine de ML est principalement composé de deux types d'apprentissages : supervisées et non supervisées. La principale différence entre les deux types réside dans le fait que pour l'apprentissage non supervisé nous n'avons pas de résultats étiquetés. En revanche, l'apprentissage supervisé nécessite une connaissance préalable de ce que devraient être les valeurs de sortie de nos échantillons. Dans notre cas, c'est le deuxième type qui est applicable et qui sera utilisé dans les étapes du projet.

2.1 Les étapes à réaliser

Mener à bien un projet de ML consiste à réaliser six étapes consécutives [Eni, 2019], que nous allons également suivre :

1. **Définition du problème à résoudre** : la problématique à laquelle nous devons répondre consiste à predire le niveau de la pollution atmosphérique des villes.

2. **Acquisition des données d'apprentissages et de test** : les fichiers que nous avons téléchargés et copiés dans notre projet comportent les données nécessaires à la résolution de notre problème et proviennet du site de l'Agence européenne pour l'environnement (AEE).
3. **Analyse et l'exploration des données** : nous allons réaliser une lecture approfondie et visualisation de nos données afin de comprendre leur rôle et les impacts qu'elles peuvent avoir dans l'objectif de prédiction fixé.
4. **Préparation et nettoyage des données** : nous allons résoudre les problèmes dans les ensembles de données, améliorer la qualité de nos datasets avec la normalisation, l'amputation des données et les techniques de programmation.
5. **Choisir un modèle d'apprentissage** : nous allons appliquer plusieurs modèles pour en trouver la plus adaptée au problème.
6. **Visualiser les résultats, et ajuster ou modifier le modèle d'apprentissage** : une fois le modèle trouvé, nous allons dscuter des résultats et des améliorations possibles.

2.2 Machine Learning avec Python

Python est le langage de programmation préférable pour le traitement des données. Nous pouvons écrire notre algorithme d'apprentissage automatique en utilisant Python et cela fonctionnera bien. Cependant, python posséde déjà de nombreux modules implémentés et bibliothèques qui peuvent nous rendre la tâche beaucoup plus facile. Parmi eux, nous notons les bibliothèques suivantes que nous utiliserons dans le cadre de notre projet :

- `Numpy` : bibliothèque mathématique permettant travailler avec des tableaux à n dimensions en Python et faire des calculs de manière efficace et efficiente.
- `Matplotlib` : bibliothèque destinée à tracer et visualiser des données sous formes de graphiques 2 ou 3D.
- `Pandas` : bibliothèque de très haut niveau qui fournit des structures de données

hautes performances et faciles à utiliser, comprend de nombreuses fonctions pour l'importation, la manipulation et l'analyse des données. En particulier, propose des structures de données et des opérations pour manipuler des tableaux numériques et des séries chronologiques.

- `Scikit Learn` : bibliothèque d'apprentissage automatique, contient une collection d'algorithmes et d'outils pour l'apprentissage automatique. La plupart des tâches qui doivent être effectuées dans un pipeline d'apprentissage automatique sont déjà implémentées dans `Scikit Learn` notamment pre-processing des données, fitting de modèls, tuning des paramètres ainsi que la prediction.

2.3 Outils Pratiques

Nous avons mentionné l'utilisation du Python dans notre projet, le choix de l'environnement de développement pour ce langage est tombé sur PyCharm, car il a rassemblé plusieurs plate-formes et permis l'utilisation simple d'autres outils dont nous avons besoin tels que :

- `Jupyter Notebook` : application open-source permettant de créer et de partager des documents contenant du code, des équations, des visualisations, et du texte narratif.
- `GitHub` : plate-forme d'hébergement de code pour le contrôle de version et la collaboration.
- `Dash` : framework permettant de développer des applications web pour la visualisation des données.
- `Heroku` : PaaS permettant le déploiement d'applications web.

Finallement, si nous souhaitons résumer, nous allons étudier et corriger les données avec `Jupyter Notebook`, puis en utilisant `Dash` nous créerons une application qui sera déployée sur `Heroku`. Et pendant tout ce temps, `GitHub` assurera la sécurité du succès du développement et de la recherche, et rendra également possible le travail en équipe.

Chapitre 3

Solution proposée

3.1 Préparation du problème

La toute première étape de ce projet d'application consistait à analyser les sources de datasets fournies par l'AEE. Au total, quatre sections ont été étudiées :

- Les émissions nationales déclarées à la Convention sur la pollution atmosphérique transfrontière à longue distance (LRTAP Convention).
- L'Inventaire de la Directive sur les Plafonds d'Émission Nationaux (NEC) de l'UE.
- `Dash` : Le Portail Européen des Émissions Industrielles : regroupe les données de la Directive sur les Émissions Industrielles (IED) et du Registre Européen des Rejets et Transferts de Polluants (E-PRTR).
- Les données sur les grandes installations de combustion (LCPs).

Le Portail Européen des Émissions Industrielles couvre plus de 60 000 sites industriels de 65 activités économiques à travers l'Europe. Il montre l'emplacement des sites et les données administratives, les transferts de déchets, ainsi que les rejets et les transferts de substances réglementées dans l'air, l'eau et le sol. Il met ainsi en disposition les données volumineuses, separées en plusieurs datasets dont LCPs faisait partie. Par la méthode d'élimination et d'observation, nous avons choisi celui qui offrait la version la

plus détaillée avec le plus de données sur l'air. Nous avons ensuite découvert que, de la même manière, les données de NEC représentent une version plus globalisée des données de LRTAP Convention, et nous avons également préféré un dataset plus étendu. Finallement, nous nous sommes retrouvés avec deux sets de données suivants :

- F_1_4 Detailed releases at facility level with E-PRTR Sector and Annex I Activity detail into Air.csv (`f1_4_data`)
- CLRTAP_NVFR14_V21_GF.csv (`clrtap_data`)

3.2 L'analyse exploratoire des données

L'analyse exploratoire des données (EDA) est un processus consistant à analyser et étudier les ensembles de données, puis résumer leurs principales caractéristiques, souvent en employant des méthodes de visualisation des données [IBM, 2020]. L'EDA peut permettre d'identifier les erreurs évidentes, mais aussi de mieux comprendre les modèles (patterns) au sein des données, de détecter les valeurs aberrantes ou les événements anormaux, de trouver des relations intéressantes entre les variables. Dans le cycle de vie d'un projet d'apprentissage automatique cette étape initiale est non seulement utile, mais primordiale. C'est pourquoi une fois que la phase d'acquisition des données est terminée, nous l'abordons immédiatement.

3.2.1 L'installation des bibliothèques

Pour préparer le site pour l'analyse exploratoire, nous devons choisir les bons outils pour l'extraction et la visulation des données :

- `pandas` : permet la manipulation et l'analyse des données.
- `matplotlib` : destinée à tracer et visualiser des données sous formes de graphiques 2D et 3D.
- `seaborn` : bibliothèque similaire à `matplotlib`, permet la visualisation plus avancée des données.

3.2.2 L'exploiration des donées

Une fois les bibliothèques sont correctement installées et importées, nous pouvons procéder à notre exploiration. Commençons par le premier dataset, celui de LRTAP Convention.

TABLE 3.1 – Une brève description de clrtap_data.

RangeIndex : 3794700 entries		
#	Column	Dtype
0	Country_Code	object
1	Country	object
2	Pollutant_name	object
3	Format_name	object
4	Sector_code	object
5	Year	int64
6	Emissions	float64
7	Unit	object
8	Notation	object
9	VersionId	int64
10	Parent_sector_code	object
11	Sector_name	object

Le dataset `clrtap_data` se compose de 12 colonnes, mais la plus importante est la colonne cible, c'est-à-dire la colonne d'émission pour laquelle nous avons constaté le manque de %72 de valeurs. Nous examinerons cela de plus près dans la partie suivante, mais compte tenu de la grande quantité de données, même cette perte n'empêchera pas l'exploration et l'entraînement des modèles.

Tout d'abord, nous voulons examiner l'évolution des niveaux d'émission et leur répartition dans les pays, pour cela nous utilisons `pandas` et modélisons les graphiques simples avec `matplotlib`. Les résultats sont présentés dans la Figure 3.1 et Figure 3.2. Nous constatons que les émissions ont tendance à diminuer, à l'exception du saut en 2011, qui peut s'expliquer soit par des inexactitudes de données, soit par un facteur déclencheur. Une réponse potentielle pourrait être la sécheresse qui a touché toute l'Europe [Wikipedia, 2011], ce qui pourrait mettre en évidence l'importance des événements majeures et leur impact sur l'évolution des niveaux d'émission.

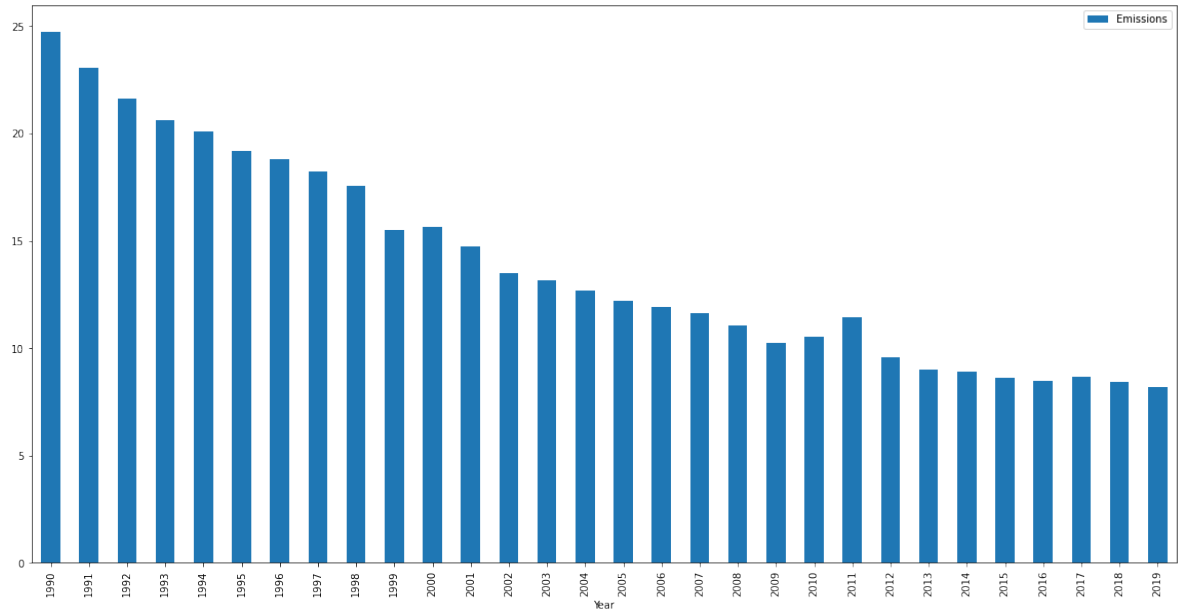

FIGURE 3.1 – L'évolution des émissions dans le temps.

Nous notons également que, d'après la Figure 3.2, le dataset ne contient pas seulement les pays, mais aussi les ensembles de pays (EEA33, EU28) qui peuvent entraîner une augmentation des émissions totales d'au moins deux fois. Pour cette raison, nous nous tournons vers une fonction telle que `pivot_table` de `pandas` permettant de présenter les données sous une forme facile à analyser. Plus tard, cette fonction sera indisponsable pour le modéle et son entraînement, mais dans cette partie son utilisation se temine par une simple visualisation. La matrice issue de cette application de tableau croisé et de la fonction `heatmap` de `seaborn` est presentée sur la Figure 3.3.

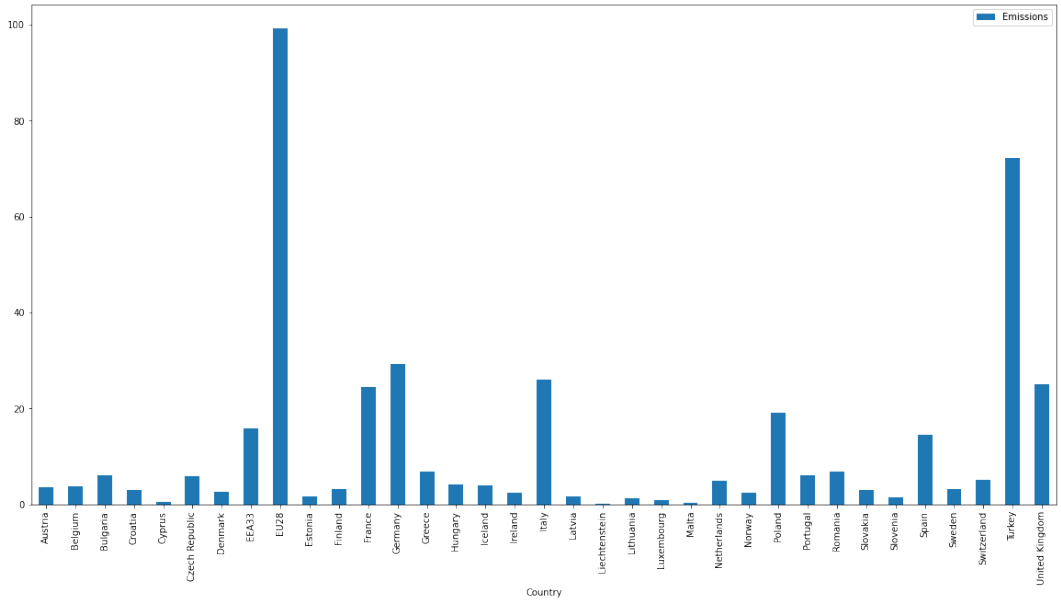

FIGURE 3.2 – La répartion des émissions par pays/ensemble des pays.

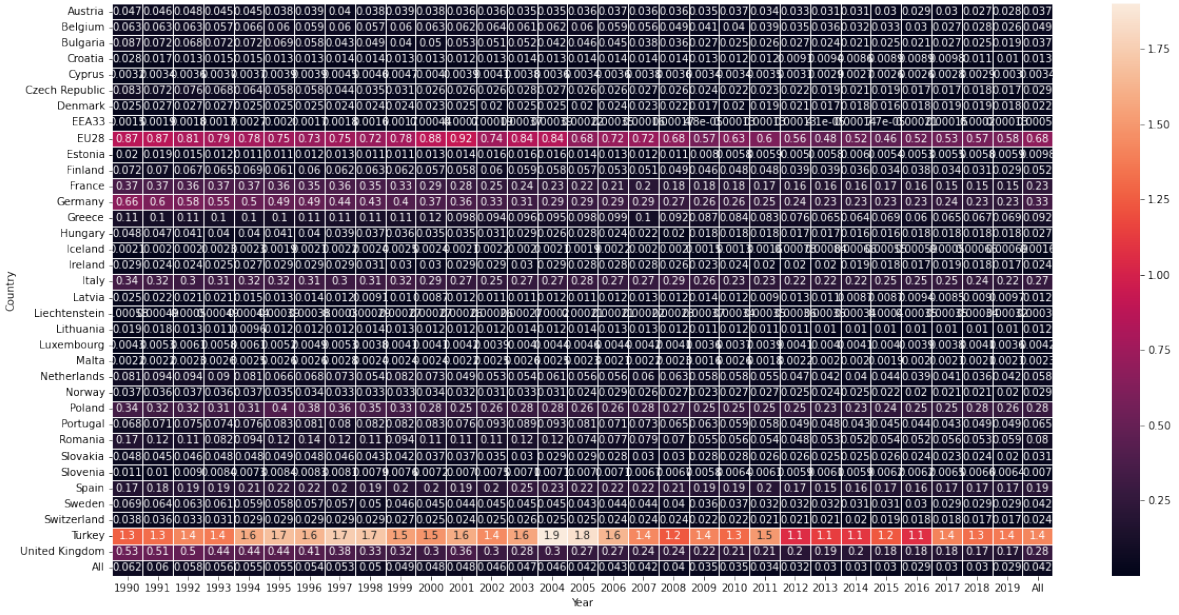

FIGURE 3.3 – La matrice d'évolution des emissions en fonction de pays et l'année.

Nous constatons que les émissions provenant de la Turquie ont un impact important sur les données globales, car celles des pays européens ont tendance à diminuer.

Cependant, pour les pays comme la France, l'Allemagne, l'Italie ou encore la Pologne le niveau d'émissions est supérieur à la moyenne, ce qui pourrait s'expliquer par des installations industrielles ou énergétiques, les données des quelles seront presentée dans le dataset suivant.

Le second dataset, `f1_4_data`, contient des données sur la quantité d'émissions dans l'air des différentes installations industrielles européenne, ainsi que des informations sur celle-ci.

TABLE 3.2 – Une brève description de f1_4_data.

RangeIndex : 288566 entries		
#	Column	Dtype
0	countryName	object
1	EPRTRSectorCode	object
2	eprtrSectorName	object
3	EPRTRAnnexIMainActivityCode	object
4	EPRTRAnnexIMainActivityLabel	object
5	FacilityInspireID	object
6	facilityName	object
7	Longitude	float64
8	Latitude	float64
9	City	object
10	targetRelease	object
11	pollutant	object
12	emissions	float64
13	reportingYear	int64

Le dataset contient 14 colonnes, dont la colonne emissions qui est la colonne qui nous interesse le plus. Parmis ces colones nous des informations sur la position géographique de l'installations, pays, ville, latitude et longitude, des informations sur l'installation elle-même, type d'installation, le nom, et enfin le polluant emis. Certaines données manques dans ce dataset, 224 villes, 19 nom d'installations, et plus important 95 données d'émissions.

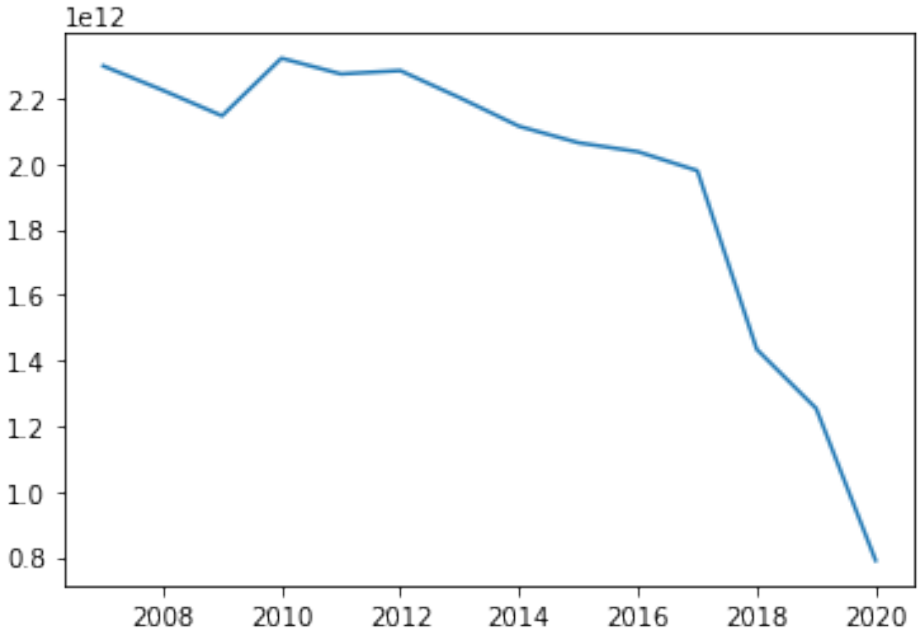

FIGURE 3.4 – Evolution des émissions en Europe entre 2007 et 2020

Pour commencer on s'intéresse à l'ensemble des émissions émise en Europe entre 2007 et 2020, sur la Figure 3.4 on remarque une tendance à la baisse des émissions entre 2007 et 2020, particulièrement entre 2017 et 2018, ainsi que 2019 et 2020, cette dernière peut s'expliquer avec la pandémie de Covid-19 durant laquelle il y a eu une diminutions d'activité dans tout les secteurs.

Sur la Figure 3.5 on remarque que le type principale de polluants en Europe est le CO2, à tel point que les autres polluants n'apparaissent presque pas sur le graphique. Cela risque de poser problème lorsque l'on effectue une prédiction sur l'ensemble des données puisque le CO2 aura une une influance beaucoup plus importantes que les autres polluants sur le résultat.

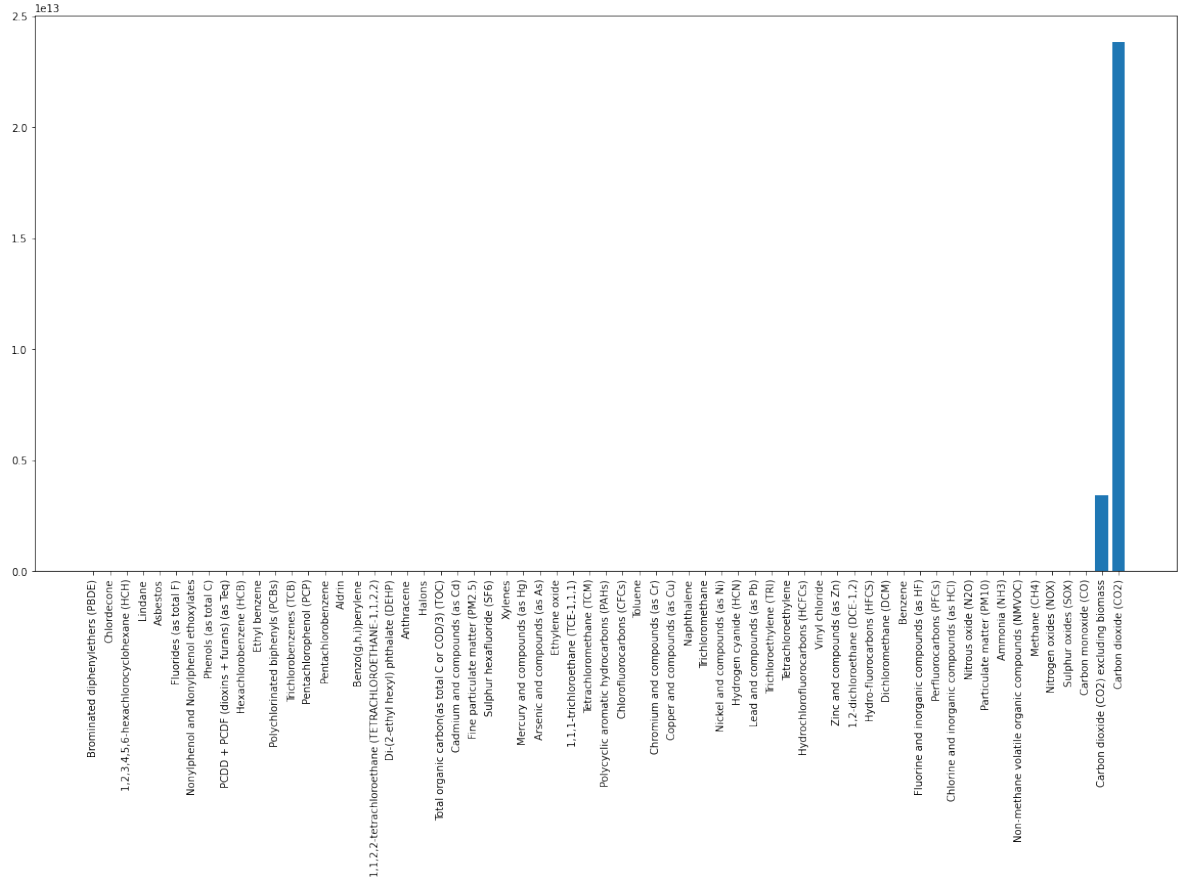

FIGURE 3.5 – Emissions par polluants en Europe

3.3 Préparation des données

La phase de préparation des données , également appelée «pré-traitement», est une étape indispensable pour valider, nettoyer et enrichir correctement les données brutes afin de pouvoir en tirer des enseignements clairs et pertinents. La validité et l'efficacité de notre solution dépendent très fortement de la qualité de préparation des données effectuée au tout début, pour cette raison, la majeure partie du temps du projet a été consacrée à cette étape que nous pouvons diviser en quatre sous parties : data amputation, data cleaning, l'encodage et la fenêtre glissante.

3.3.1 Data Amputation

Pendant la partie exploratoire, nous avons remarqué que le dataset contenait des valeurs manquantes concernant les émissions. Pour mieux le comprendre nous avons utlisé la bibliothèque open-source `missingno`. Elle fournit un ensemble d'outils de visualisations qui permettent d'obtenir un résumé visuel rapide de l'exhaustivité de l'ensemble de données.

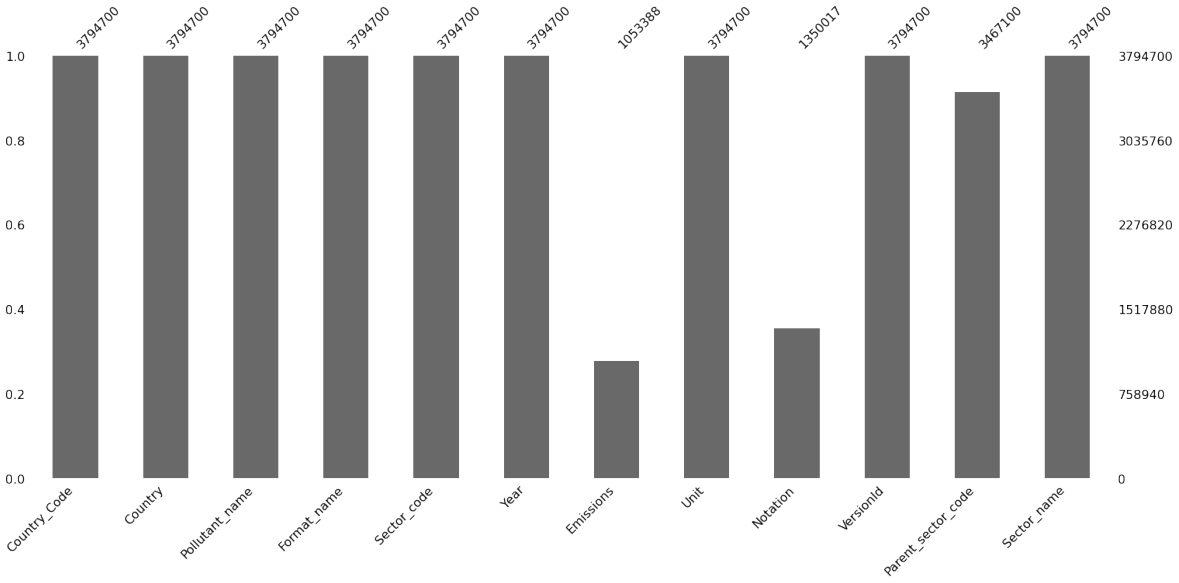

FIGURE 3.6 – La visualisation de la nullité par colonne

Lors de la visualisation les données à l'aide de la fonction `bar` de `missingno` 3.6, nous détectons trois colonnes avec des valeurs manquantes : «Emissions», «Notation» et «Parent_sector_code». Une fois les colonnes manquantes sont détectées, nous souhaitons comprendre leur corrélation, pour cela on utilise la fonction de dendrogramme dont les résultats sont présentés sur la Figure 3.7.

Le dendrogramme utilise un algorithme de clustering hiérarchique, l'interpretation de ce graphique nécessite donc une lecture de haut en bas. Les feuilles de cluster qui se lient prédisent l'inter-présence (ou l'absence) des variables, ainsi on peut constater que les colonnes «Emissions» et «Notation» sont interdépendantes et que les colonnes «Parent_sector_code» et «VirsionId» y dérivent. Ainsi, nous pouvons conclure que

l'absence de variable d'émission peut signifier la même absence dans les trois autres colonnes. Nous devons donc amputer les lignes contenant des valeurs vides pour la colonne d'émissions.

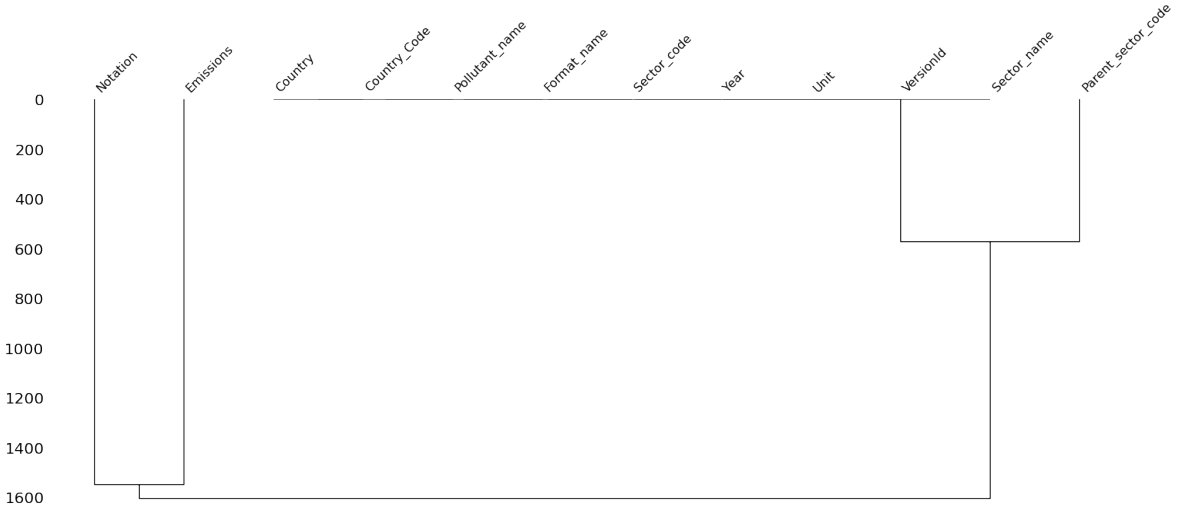

FIGURE 3.7 – Le dendrogramme du regroupement hiérarchique des colonnes

3.3.2 Data Cleaning

Le Data Cleaning ou nettoyage de données est une étape indispensable en ML. Nous devons résoudre les problèmes dans nos ensembles de données, afin de pouvoir les exploiter par la suite. L'EDA nous a permis de détecter plusieurs points faibles et sensibles dans nos datasets, que nous devons résoudre.

Commençons par le dataset `clrtap_data`. Nous avons vu précedemment qu'il contenait la colonne «Unit», ce qui indique la hétérogénéité de nos données d'émissions. Nous distinguons notamment quatre unités de masse : le gigagramme (Gg), le kilogramme (kg), le gramme (g) et le milligramme (Mg). Ces valeurs seront transformées par programme en unités de base (kg) et remplacées en combinant des outils de `numpy` et `pandas`, la colonne «Unit» peut donc être supprimée.
Ensuite, nous avons vu que le dataset contient une colonne «Sector_code» composée de plus de cent valeurs uniques. En cas de prediction, une division aussi importante

provoquera un manque de données pour l'apprentissage. Par conséquent, en se référant au rapport fourni par l' AEE, nous les combinerons en sections suivantes :

- Energy production and distribution
- Energy use in industry
- Non-road transport
- Road transport
- Commercial, institutional and households
- Industrial processes and product use
- National total for the entire territory
- Agriculture
- Waste
- Other

Pour les mêmes raisons, nous mettrons en évidence des polluants présent dans tous les pays et dont le nombre est supérieur à la moyenne globale : CO, NH3, NMVOC, NOx, PM10, PM2.5, SOx, TSP. Cette fusion sera non seulement importante pour la prediction mais permettra également la création des paramètres pour la visualisation des résultats dans notre application dash.

Le dataset `f1_4_data` contient 14 colonnes et une quantité importante de données. Nous souhaitons donc supprimer les élements qui ne sont pas pertinants ou nécessaires pour la prédiction. Après avoir etudié les données, nous avons conclu que nous pouvions supprimer les colonnes suivantes :

- «Air» : car le dataset ne contient pas de données en dehors de l'air
- «EPRTRSectorCode» : la version technique de «EPRTRSectorName»
- «EPRTRAnnexIMainActivityCode» : la version technique de «EPRTRAnnexIMainActivityLabel»
- «FacilityInspireID» : contient des informations redondantes
- «facilityName» : contient des informations redondantes

- «EPRTRSectorCode» : répresente des informations plus détaillés de «EPRTR-SectorCode»

On va maintenant s'intéresser au données géographique du dataset, à savoir «countryName», «Longitude», «Latitude» et «City». Etant donnée que ce projet a pour but la prédiction du niveau de pollution dans une cité, la colonne «City» semble être essentielle, néanmoins, la latitude et la longitude sont plus intéressentes, car grace à ces données, on peut évaluer la proximité d'une ville avec ces installations industrielles et supprimer ainsi la colonne «facilityName».

TABLE 3.3 – Dataset f1_4_data après la suppression des colonnes

RangeIndex : 288566 entries		
#	Column	Dtype
0	countryName	object
1	EPRTRAnnexIMainActivityCode	object
2	Longitude	float64
3	Latitude	float64
4	pollutant	object
5	emissions	float64
6	reportingYear	int64

3.3.3 Fenêtre Glissante

En premier lieu pour la prédiction, nous avons pris l'ensemble des données au complet et les avons séparé en données d'entraînement et de test aléatoirement. Nous nous sommes vite rendus compte que la méthode classique qu'on utilisait n'était pas efficace car les modèles n'apprennaient pas et faisait une prédiction fausse.

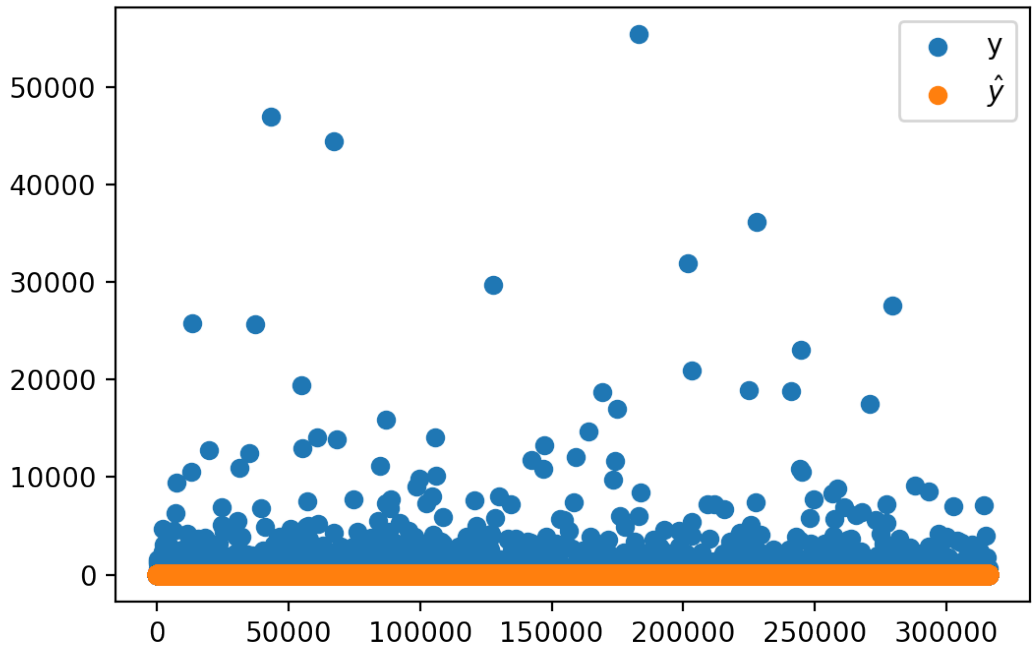

FIGURE 3.8 – Comparaison entre la prédiction et les vraies valeurs

Dans notre cas, nous ne pouvons pas choisir une séparation aléatoire des données car ce n'est pas possible d'utiliser les données du futur pour prédire celles du passé. Il faut donc les séparer en fonction de l'âge de celles-ci en faisant une fenêtre glissante.

La fênetre glissante est une méthode de prédiction scientifique basée sur des données historiques horodatées. Cela implique de construire des modèles à travers une analyse historique et de les utiliser pour faire des observations et guider la prise de décision stratégique future [tableau.com, 2022]. Pour cela, nous allons itérer les entraînements et les prédictions en prenant les données des années précédentes pour prédire une année donnée.

- Données d'entraînement : [2007] Données test : [2008]
- Données d'entraînement : [2007, 2008] Données test : [2009]
- Données d'entraînement : [2007, 2008, 2009] Données test : [2010]
- Données d'entraînement : [n-1, ..., n] Données test : [n + 1]

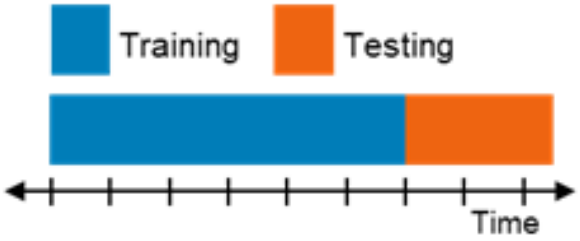

FIGURE 3.9 – Données d'entraînement et de test sur une base temporelle

Pour prédire l'année 2010 par exemple, nous aurions comme données d'entraînement celles de 2009 jusqu'aux plus vieilles données existantes et comme données tests, celles de 2010.

3.3.4 Encodage

Nos datasets contiennent des variables qui sont dites catégorielles. Une variable catégorielle est une variable qui prend pour valeur des modalités, des catégories ou bien des niveaux, par opposition aux variables quantitatives qui mesurent sur chaque individu une quantité [Wikipedia, 2021b]. La présence de ces variables dans les données complique généralement l'apprentissage. En effet, la plupart des algorithmes d'apprentissage automatique prennent des valeurs numériques en entrée. Ainsi, il faut trouver une façon de transformer nos modalités en données numériques. Pour cela nous allons passer par une étape d'encodage.

Nos variables étant toutes catégorielles (à l'exception de l'émission et de l'année), nous ne pouvions pas nous permettre de les supprimer. Nous avons donc encodé les données, numériser les variables catégorielles en remplaçant chaque valeur de catégorie unique par un entier. Pour cela, nous avons utiliser la classe `OrdinalEncoder()` de la bibliothèque `sklearn`.

Il existe un autre moyen d'encoder les données qui est le One Hot Encoding. Le one

country
Austria
Belgium
Bulgaria
Croatia

(a)

country
0
1
2
3

(b)

TABLE 3.4 – La colonne «country» avant (a) et après encodage (b) avec OrdinalEncoder

hot encoding crée de nouvelles colonnes indiquant la présence de chaque valeur possible dans les données d'origine. Nous n'avons pas utiliser cette technique en raison du trop grand nombre de valeurs uniques et l'absence de lien entre les variables.

color
red
red
yellow
green
yellow

(a)

red	yellow	green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

(b)

TABLE 3.5 – La colonne «color» avant (a) et après (b) avec One Hot Coder

3.4 Modèles de prédiction

Réaliser un apprentissage supervisé consiste à fournir à la machine des données étiquetées et propices à l'apprentissage. C'est-à-dire que nous allons analyser et préparer

les données et leur donner une signification. C'est à partir de cette signification que la machine va réaliser son apprentissage. Dans notre cas l'objectif est de prédire une valeur, c'est un problème de regression. L'apprentissage supervissé propose plusieurs algorithmes pour la régression, mais nous allons utiliser les trois suivants :

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor

Pour effectuer une prédiction, nous devons sélectionner un modèle puis l'entraîner en utilisant les données de test et de train de la fenêtre glissante qui contient des émissions, les années ainsi que les pays. Faisons une prédiction des émissions pour 2022 pour le CO2 dans le but de trouver le modèle le plus adapté.

3.4.1 Linear Regression

L'ensemble d'entrées et les sorties correspondantes sont examinés et quantifiés pour montrer une relation, nottament comment le changement d'une variable affecte une autre.

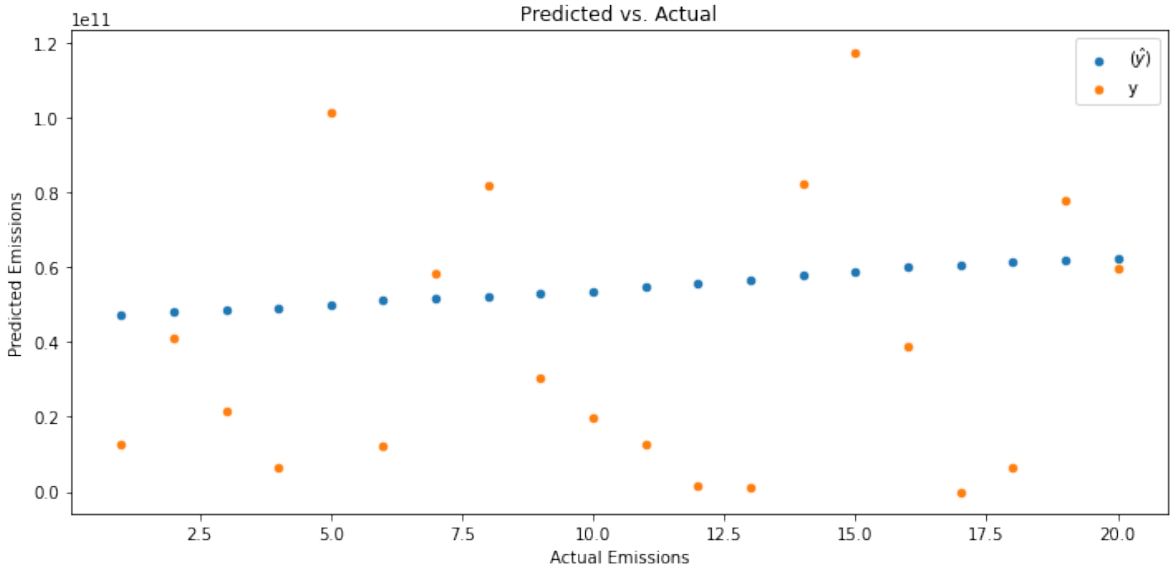

FIGURE 3.10 – Linear Regression

Nous obtenons un score R^2 de -0.1704 et RMSE de 38597224241. Dans notre cas, il n'y a pas de relation entre les variables, donc ce modèle n'est pas approprié pour résoudre notre problème.

3.4.2 Decision Tree Regressor

L'objectif est de créer un modèle qui prédit les valeurs de la variable cible, en se basant sur un ensemble de séquences de règles de décision déduites à partir des données d'apprentissage. Nous choisissons comme paramètre `random_state=1` pour notre modèle dont le résultat de prédiction est visualisé sur la Figure 3.11 ci-dessous.

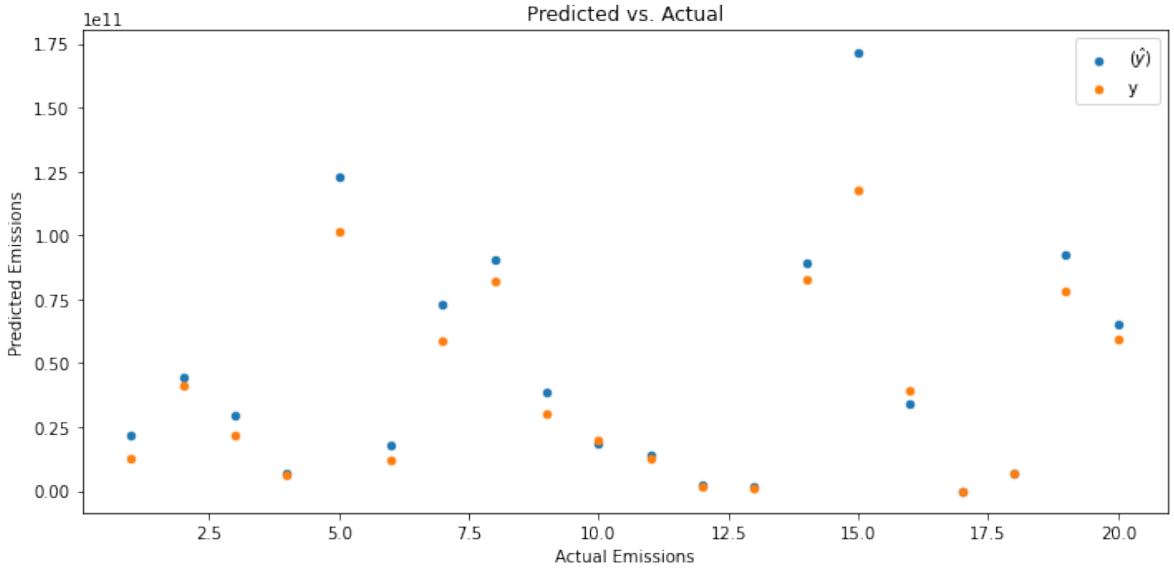

FIGURE 3.11 – Decision Tree Regressor

Nous obtenons un score R^2 de 0.8338 et RMSE de 14540912642. De plus, nous voyons avec scatter que la variable cible prédite (ŷ) est souvent proche et parfois parfaitement identique à la variable réelle (y), ce qui indique de bonnes performances du modèle.

3.4.3 Random Forest Regressor

Les forêts d'arbres décisionnels sont une technique d'apprentissage ensembliste qui s'appuie sur des arbres de décision. Le modèle implique la création d'arbres décisionnels multiples en utilisant ensembles de données fractionnés à partir des données d'origine et en sélectionnant aléatoirement un sous-ensemble de variables à chaque étape de l'arbre décisionnel. Le modèle sélectionne ensuite le mode de toutes les prédictions de chaque arbre décisionnel.

Après avoir effectué des tests sur plusieures hyperparamètres de `GridSearchCV`, nous avons utilisé le plus optimal pour notre modèle : `n_estimators=1000, n_jobs=-1, random_state=1`.

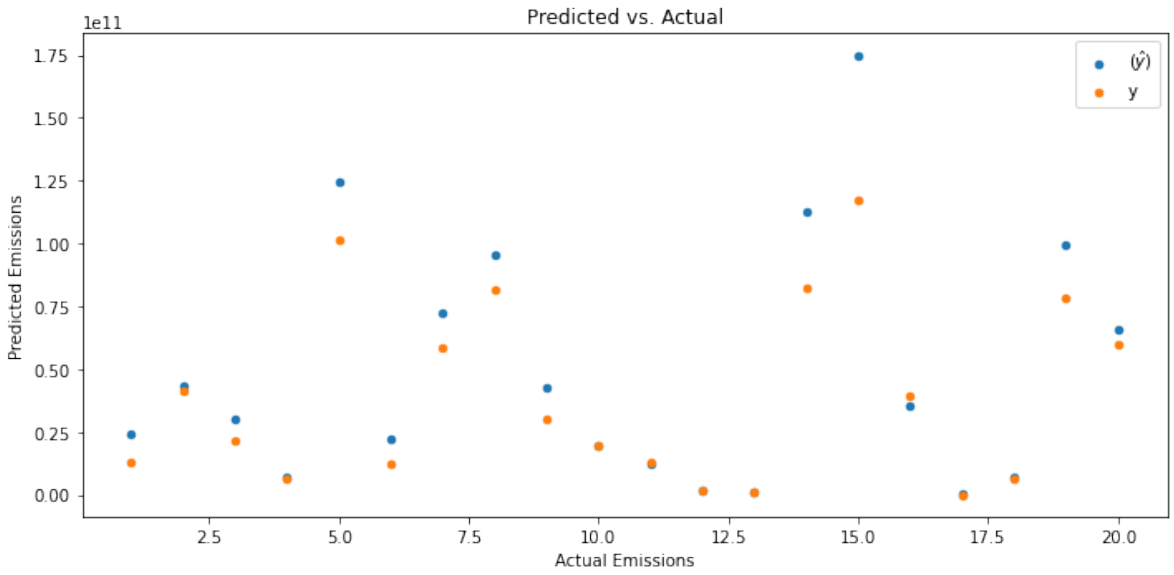

FIGURE 3.12 – Random Forest Regressor

Nous pouvons observer les résultats sur la Figure 3.12 ci-dessus sur la base de laquelle le modèle semble être réussi puisque les données cibles correspondent approximativement à la réalité, cependant, les scores pour ce modèle sont moins élevées que pour celui de l'arbre de decision : nous obtenons 0.7613 pour R^2 et 17429321368 de RMSE.

Afin de mieux nous rendre compte des résultats obtenus, nous avons tracer l'écart entre les valeurs prédites et réelles en prenant comme exemple, les données de la ville de Hambourg en Allemagne de 2007 à 2017.

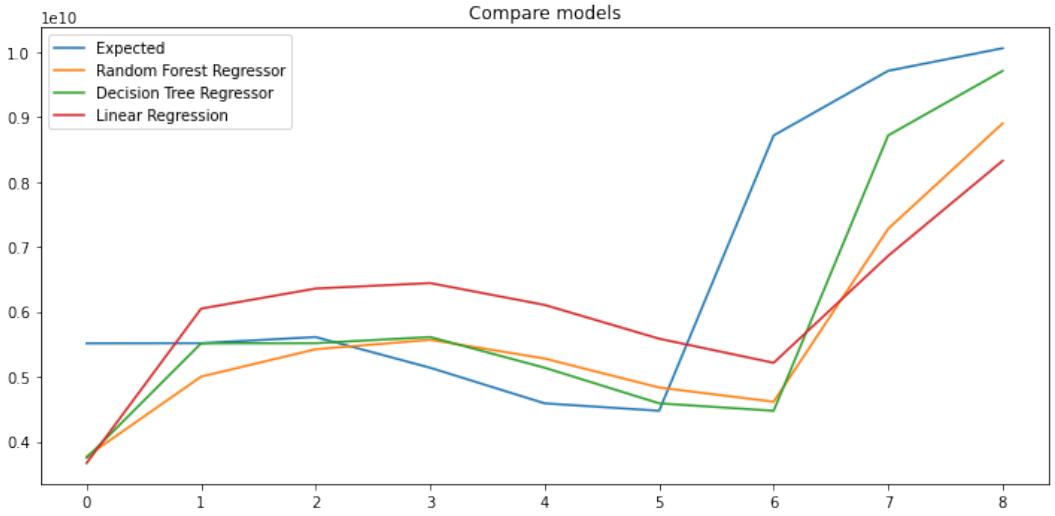

FIGURE 3.13 – comparaison des trois modèles avec le résultat réel

On s'apperçoit rapidement que la Régression Linéaire 3.13 est très inéfficace. D'autre part, on peut voir que les prédictions du Decision Tree Regressor sont très proches aux données réelles et representent même une similitudé avec un décalage léger sur le temps. On peut donc conclure que c'est le modèle de Decision Tree Regressor qui correspond le mieux à notre problème de prédiction de la pollution atmosphérique.

3.5 Dashboard

À l'issue de ce projet, l'une de nos solutions est l'application Dash, qui a été mise à jour à chaque nouvelle observation et modification dans nos données. Nous pouvons diviser notre application en deux parties, une carte et un graphique, représentés respectivement sur la Figure 3.14 et la Figure 3.15.

La carte est le résultat du travail sur le dataset `f1_4_data`, qui, à l'aide de l'outil Mapbox, visualise nos données sous formes des points avec des informations supplémentaires comme :

- ville de l'installation
- le nombre d'émissions
- le secteur d'activité

Toutes ces options varient selon l'année sélectionnée dans le moteur sous la carte, et la barre latérale vous permet d'ajouter un filtre supplémentaire pour la ville et le secteur.

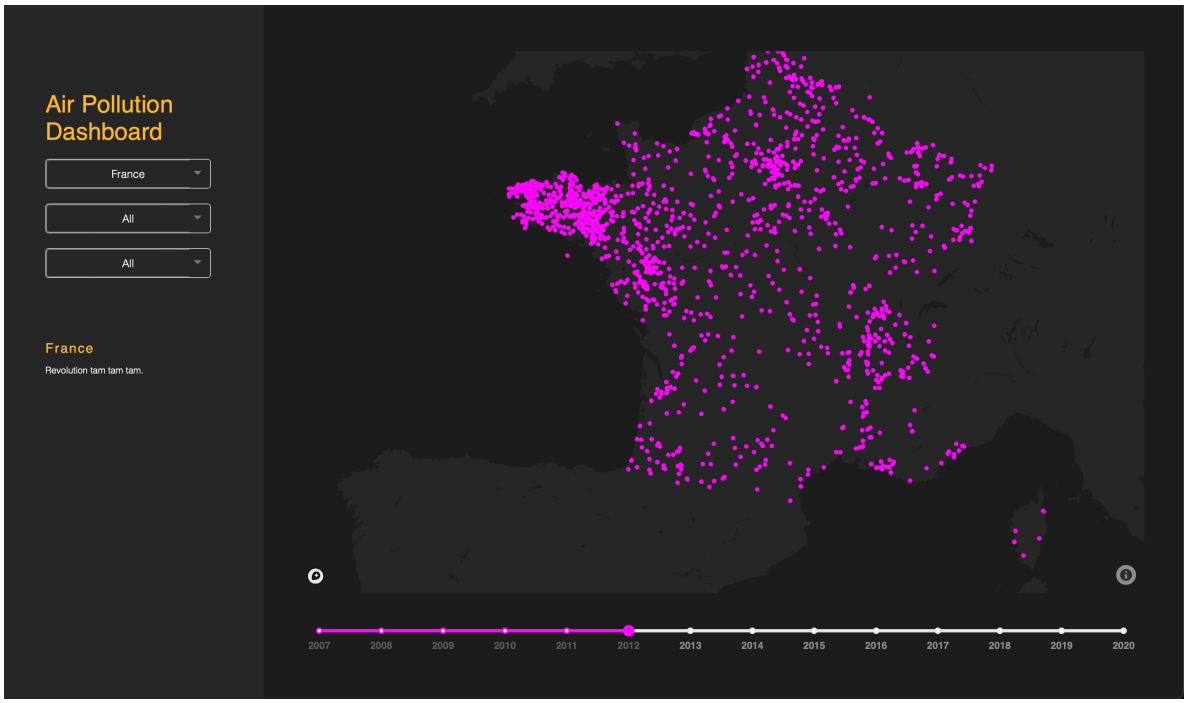

FIGURE 3.14 – Map avec données de `f1_4_data` dataset

Le graphique, tout comme la carte, permet la visualisation de données filtrées de dataset `clrtap_data`. Nous pouvons ainsi afficher l'évolution du niveau de la pollution en fonction des polluants et des secteurs définis dans la partie de pré-traitement de notre projet.

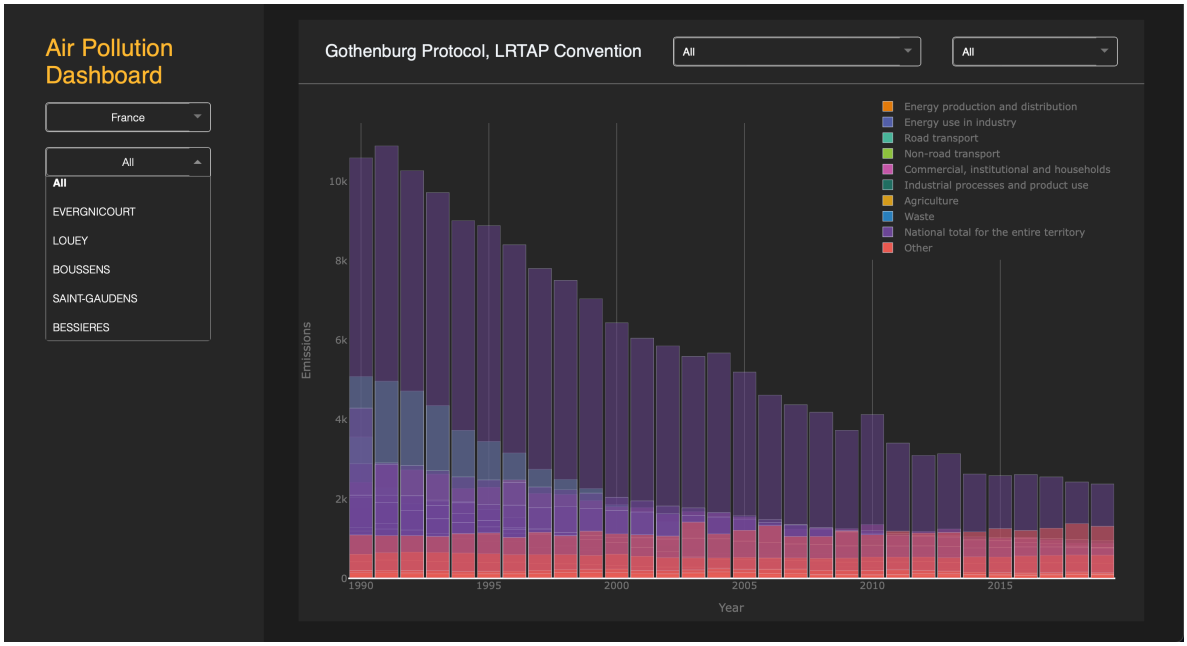

FIGURE 3.15 – Graph dynamique avec données de `clrtap_data` dataset

L'application a été successivement déployée sur PaaS `Heroku` et disponible sur le lien suivant : `air_pollution_prediction_app`.

Chapitre 4

Analyse, interprétation et discussion des résultats

Nous avons fait la sélection de notre modèle dans la partie précédente en fonction des métriques de regression, les prédictions étant numériques, et l'analyse de graphiques. Cependant, les données que nous avons étudié pour nos trois modèles ne comprenaient pas un tel facteur important qu'un événement majeur de la pandémie, dont les conséquences pourraient être suffisament fortes pour causer des perturbations dans l'ensemble des données et donc aurait un impact sur l'entrainement et la prédiction.

Revenons à notre exemple avec le CO2 et observons les scores de prédiction avec la fenêtre glissante pour deux modèles avec la Figure 4.1. Nous arrivons à la conclusion que la qualité de prédiction de l'un n'est pas inférieure à l'autre, même si le Random Forest nécessitait un temps supérieur pour l'apprentissage initial. Cependant, les choses changent quand il s'agit de prévisions pour l'année de 2020. Les deux modèles abandonnent les positions, cependant, la capacité de prédiction de l'arbre de décision est moins fortement endommagée. Pourquoi ? La réponse potentielle pourrait être la différence même entre les algorithmes de ces deux modèles.
Les arbres de décision sont très rapides et fonctionnent facilement sur de grands ensembles de données par rapport à la forêt aléatoire. Un arbre de décision combine certaines décisions, alors qu'une forêt aléatoire combine plusieurs arbres de décision.

C'est donc un processus sûr, mais lent et qui nécessite une formation rigoureuse. C'est donc la raison pour laquelle nous pouvons observer un retard dans l'apprentissage du modèle de Random Forest Regressor sur la Figure 4.1.

```
Results for RandomForestRegressor
R² for train: [2007] test: [2008] : 0.826
R² for train: [2007, 2008] test: [2009] : 0.918
R² for train: [2007, 2008, 2009] test: [2010] : 0.952
R² for train: [2007, 2008, 2009, 2010] test: [2011] : 0.990
R² for train: [2007, 2008, 2009, 2010, 2011] test: [2012] : 0.990
R² for train: [2007, 2008, 2009, 2010, 2011, 2012] test: [2013] : 0.995
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013] test: [2014] : 0.987
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014] test: [2015] : 0.991
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015] test: [2016] : 0.987
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016] test: [2017] : 0.985
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017] test: [2018] : 0.987
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018] test: [2019] : 0.921
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019] test: [2020] : 0.761

Results for DecisionTreeRegressor
R² for train: [2007] test: [2008] : 0.997
R² for train: [2007, 2008] test: [2009] : 0.966
R² for train: [2007, 2008, 2009] test: [2010] : 0.983
R² for train: [2007, 2008, 2009, 2010] test: [2011] : 0.997
R² for train: [2007, 2008, 2009, 2010, 2011] test: [2012] : 0.995
R² for train: [2007, 2008, 2009, 2010, 2011, 2012] test: [2013] : 0.994
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013] test: [2014] : 0.989
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014] test: [2015] : 0.995
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015] test: [2016] : 0.992
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016] test: [2017] : 0.987
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017] test: [2018] : 0.988
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018] test: [2019] : 0.930
R² for train: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019] test: [2020] : 0.834
```

FIGURE 4.1 – Map avec données de `f1_4_data` dataset

Et c'est là que réside la supériorité du modèle des arbres dans le cadre de la résolution de notre problème. Ils s'adaptent rapidement à la quantité importante de nouvelles données et prennent moins de temps à apprendre. En outre, si nous nous souvenons de la phase d'amputation, nous avons vu un lien entre le secteur parent et les valeurs d'émissions, ce qui suggère également un modèle d'arbres.

Chapitre 5

Conclusion et perspectives

La problématique de ce projet consistait à prédire le niveau de pollution dans des villes, à l'aide notamment de données météorologiques et macroéconomiques, dans le contexte de l'industrialisation et l'urbanisation pouvant être dangereuse pour la santé. Pour résoudre ce problème nous avons utilisé des données de deux différents dataset, `clrtap_data` et `f1_4_data`, tous deux provenant de AEE.

Pour répondre à ce problème, nous avons utilisé des outils de programmation et des algorithmes d'apprentissage automatique qui nécessitaient un nettoyage minutieux des datasets. La manipulation des données nécessitait une claire compréhention de tous ses éléments, ainsi que la relation entre les valeurs de l'ensembe de données. C'est pourquoi la phase la plus longue et la plus difficile était celle de l'exploitation et du nettoyage des données (EDA). Dans la méthodologie de travail, une étape suit l'autre, mais dans notre cas réel, cela nécessitait de nombreux sauts d'une phase à l'autre, ce qui éloignait la phase d'apprentissage du modèle en raison de situations imprévues.
Ce projet nous a également poussés à aller au-delà des lections académiques et à commencer une confrontation avec des données et des outils réels. De cette façon, nous avons appris une nouvelle compétence comme la fenêtre glissante, sans laquelle la prédiction serait tout simplement impossible, étant donné que nous travaillons avec `TimeSeries`, c'est-à-dire des données qui suivent la chronologie précise. Cette fenêtre nous a en-

suite permis d'effectuer des prédictions avec trois modèles différent : Linear Regression, RandomForest Regressor et Decision Tree Regressor. Grâce aux outils de visualisation de données et de notre propre observation, nous avons pu identifier le dernier modèle comme le plus approprié et possédant les meilleurs performances. Cependant, ce modèle peut potentiellement être amélioré, notamment en utilisant des outils dédiés au remplissage de données manquantes au lieu d'amputation. Un autre moyen efficace d'augmenter la quantité des données serait d'obtenir des sources supplementaires avec une préference pour les données mensuelles, et performer ainsi la précision des prédictions.

Pour améliorer nos données nous pouvons également se référer au processus de l'inférence causale permettant l'établissement d'une relation de causalité entre un élément et ses effets, et rajouter ainsi les nouvelles colonnes aux datasets, tel que le milieu, la taille de la ville ou encore la densité de population.

En introduction nous avons mentionné que certains polluants atmosphériques peuvent fusionner avec d'autres et en créer d'autres, qui sont les plus dangereux pour l'homme. Une option intéressante serait de faire avancer le projet dans cette direction en combinant l'adoption de nouvelles données et l'utilisation de la statistique bayésienne [Wikipedia, 2021a] pour prévoir de nouveaux polluants et évaluer leurs risques pour la santé.

Finallement, la conclusion logique serait de donner accès à l'analyse de ces données et de leurs prédictions constamment mises à jour via l'application, qu'il s'agisse de Dash ou d'une solution plus performante.

Bibliographie

[Eni, 2019] Eni, E. (2019). Intelligence artificielle vulgarisée - le machine learning et le deep learning par la pratique. *Les étapes à réaliser pour mener à bien un projet de Machine Learning :* Source link.

[gouvernement.fr, 2022] gouvernement.fr (2022). S'informer et agir efficacement contre la pollution de l'air. *Article :* Source link.

[IBM, 2020] IBM (2020). Analyse exploratoire des données. *Qu'est-ce que l'analyse exploratoire des données ? :* Source link.

[OMS, 2018] OMS (2018). Neuf personnes sur 10 respirent un air pollué dans le monde. *Article :* Source link.

[OMS, 2022] OMS (2022). Des milliards de personnes respirent toujours un air pollué : nouvelles données de l'oms. *Article :* Source link.

[Oracle, 2022] Oracle (2022). Qu'est-ce que le machine learning ? *Article :*Source link.

[tableau.com, 2022] tableau.com (2022). Fenêtre glissante. *Définition :* Source link.

[Wikipedia, 2011] Wikipedia (2011). Sécheresse de 2011 en europe. *Impact :* Source link.

[Wikipedia, 2021a] Wikipedia (2021a). Statistique bayésienne. *Définition :* Source link.

[Wikipedia, 2021b] Wikipedia (2021b). Variable catégorielles. *Définition :* Source link.