Rapport de Projet

MSPR 3 – Big Data & Analyse de Données

Membres du groupe

BADREDDINE KHALIL
REGUIEG ZAKARIA
KHEDIM SOFIANE
GHANMI AHMED
ISMAIL ALI

Encadrant

BOUFADEN HEDI

June 19, 2025

Contents

List of Figures

Introduction

Dans un contexte où les décisions politiques s'appuient de plus en plus sur des données concrètes, l'intelligence artificielle s'impose comme un levier stratégique pour anticiper les tendances électorales. Cette convergence entre politique et technologies de la donnée ouvre la voie à de nouvelles perspectives, tant pour les décideurs que pour les cabinets spécialisés en conseil politique.

Le projet que nous présentons dans ce rapport s'inscrit dans cette dynamique. Il a pour objectif de développer une preuve de concept (POC) capable de prédire les évolutions électorales au sein d'un territoire géographique restreint. Pour ce faire, nous mobilisons des techniques avancées d'analyse de données et de machine learning.

En collaboration avec notre client fictif, Jean-Edouard de la Motte Rouge, expert en stratégie électorale, nous cherchons à proposer une solution innovante permettant de mieux comprendre les dynamiques socio-économiques qui influencent le vote et d'en tirer un avantage stratégique.

Cette introduction pose ainsi les fondements de notre démarche : explorer, analyser et modéliser les données afin de construire un outil prédictif pertinent, à la croisée de la data science et de l'analyse politique.

I Présentation du cadre de projet

1 Présentation de l'entreprise et de son activité

Jean-Edouard de la Motte Rouge a créé une start-up spécialisée dans le conseil sur la thématique des campagnes électorales.
La start-up comprend un expert en analyse politique, un business développeur, et un assistant.
Il souhaite pouvoir prédire, grâce à l'intelligence artificielle, les tendances des élections à venir, en se basant sur un certain nombre d'indicateurs, comme la sécurité, l'emploi, la vie associative, la population, la vie économique (nombre d'entreprises), la pauvreté, etc.

2 Objectif

L'entreprise a fait appel à notre groupe pour établir une preuve de concept (POC).
L'objectif de cette preuve de concept est de démontrer qu'il est possible d'utiliser l'intelligence artificielle (IA) pour prédire les tendances des élections à venir dans un secteur géographique donné en se basant sur un ensemble d'indicateurs économiques, sociaux et démographiques.

En démontrant la faisabilité de ce type de modèle prédictif, il serait possible de créer un outil qui permettrait aux candidats ou partis politiques de mieux comprendre les attentes et les préférences des électeurs d'un secteur donné. Cela pourrait permettre aux campagnes électorales d'être plus ciblées et plus efficaces en s'adaptant aux caractéristiques socio-économiques des électeurs.

Cet outil pourrait également être utile pour les organisations qui travaillent dans le domaine de la démocratie et de la gouvernance, en leur permettant d'anticiper les résultats des élections et d'identifier les défis socio-économiques qui pourraient influencer les résultats.

II Environnement de travail

1 Choix de la zone géographique

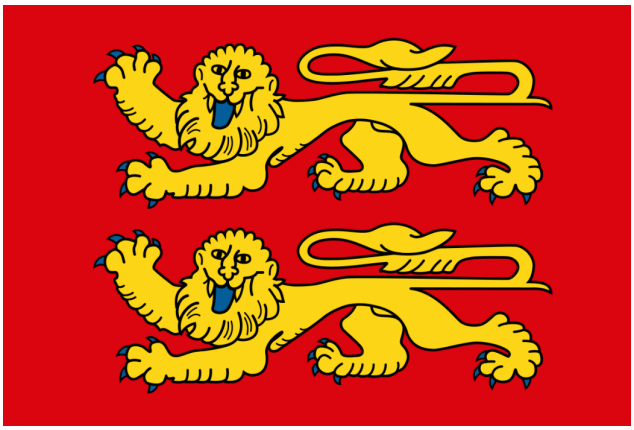

Figure 1: Drapeau de Normandie

Dans ce projet, nous avons décidé de choisir la région **Normandie** comme zone d'étude, car elle présente des caractéristiques particulièrement intéressantes pour notre analyse prédictive.

Située dans le nord-ouest de la France, la Normandie regroupe cinq départements : le Calvados, l'Eure, la Manche, l'Orne et la Seine-Maritime. Elle compte environ **3,3 millions d'habitants** et se distingue par une combinaison de territoires urbains (comme *Rouen*, *Caen*, *Le Havre*) et de zones rurales à forte identité locale. Cette diversité géographique, démographique et socio-économique en fait un terrain d'étude pertinent pour modéliser les dynamiques électorales.

Justification de notre choix

Lors du second tour de l'élection présidentielle de **2022**, les résultats dans la région Normandie étaient les suivants :

- **Emmanuel Macron** :
  - France : 58,55 %
  - Normandie : 55,84 %
- **Marine Le Pen** :
  - France : 41,45 %
  - Normandie : 44,16 %

Conséquence : Bien que la Normandie soit légèrement plus favorable à Marine Le Pen que la moyenne nationale, elle reste globalement alignée sur les tendances politiques du pays.

Ce profil en fait une région à la fois représentative des dynamiques nationales et riche en spécificités locales intéressantes à analyser.

Analyser les élections en **Normandie** permet donc de saisir à la fois les grandes tendances politiques nationales et les nuances propres à certains territoires, renforçant ainsi la pertinence de cette région pour notre projet d'analyse électorale.

2 Choix des critères

Dans le cadre de notre étude, nous avons mobilisé un modèle d'intelligence artificielle pour tenter de prédire les orientations électorales à l'échelle locale. Pour cela, nous nous sommes appuyés sur plusieurs indicateurs disponibles dans notre jeu de données. Ces variables ont été choisies en raison de leur lien probable avec les comportements de vote, qu'ils soient d'ordre économique, social ou démographique. Voici les raisons de notre sélection :

- **Taux de criminalité** : Le niveau de criminalité moyen par commune peut jouer un rôle non négligeable dans la manière dont les citoyens votent. Un sentiment d'insécurité accru pousse souvent une partie de l'électorat vers des partis promettant des réponses fermes, généralement situés à droite ou à l'extrême droite. À l'inverse, un contexte plus apaisé peut réduire l'importance de ces enjeux dans la décision de vote.

- **Création d'entreprises** : Le pourcentage d'entreprises créées dans une commune est un bon indicateur de dynamisme économique. Dans les zones où cette activité est forte, les habitants peuvent être sensibles aux discours libéraux ou pro-entrepreneuriat, souvent portés par des partis du centre ou de la droite.

- **Structure sociale** : La répartition entre classes populaires et classes supérieures nous semble essentielle pour comprendre certaines tendances électorales. Les zones à forte présence ouvrière ou précaire sont historiquement plus favorables aux partis de gauche, en raison de leurs propositions sociales. Les communes plus aisées, quant à elles, tendent à voter pour des partis valorisant l'initiative privée.

- **Abstention parmi les inscrits** : Même si notre analyse ne repose pas directement sur les résultats électoraux bruts, nous avons pris en compte le taux d'abstention. Un niveau élevé d'abstention peut indiquer une défiance envers les institutions, ce qui profite souvent aux formations politiques dites « antisystème ».

- **Part des étrangers** : Le poids des étrangers dans la population peut influencer le débat public local, notamment sur les questions d'intégration, d'identité ou de politique migratoire. Ce facteur peut alors renforcer le vote pour certains partis sensibles à ces thématiques.

- **Indice de jeunesse** : L'âge moyen de la population a un impact sur les attentes électorales. Une commune jeune peut faire émerger des priorités différentes : climat, emploi, éducation, numérique. Ce qui peut modifier le paysage électoral attendu.

- **Taille de la population** : Enfin, la population totale nous permet de distinguer des contextes urbains et ruraux. Ce critère est crucial car les enjeux ne sont pas perçus de

la même manière selon que l'on vive dans une grande ville ou un petit village. Le vote rural peut ainsi se différencier du vote urbain, tant dans la participation que dans les préférences politiques.

En combinant ces éléments, notre modèle vise à établir des liens entre les caractéristiques locales et les tendances électorales observées, afin d'anticiper au mieux les évolutions futures.

3 La démarche suivie et les méthodes employées

3.1 workflow

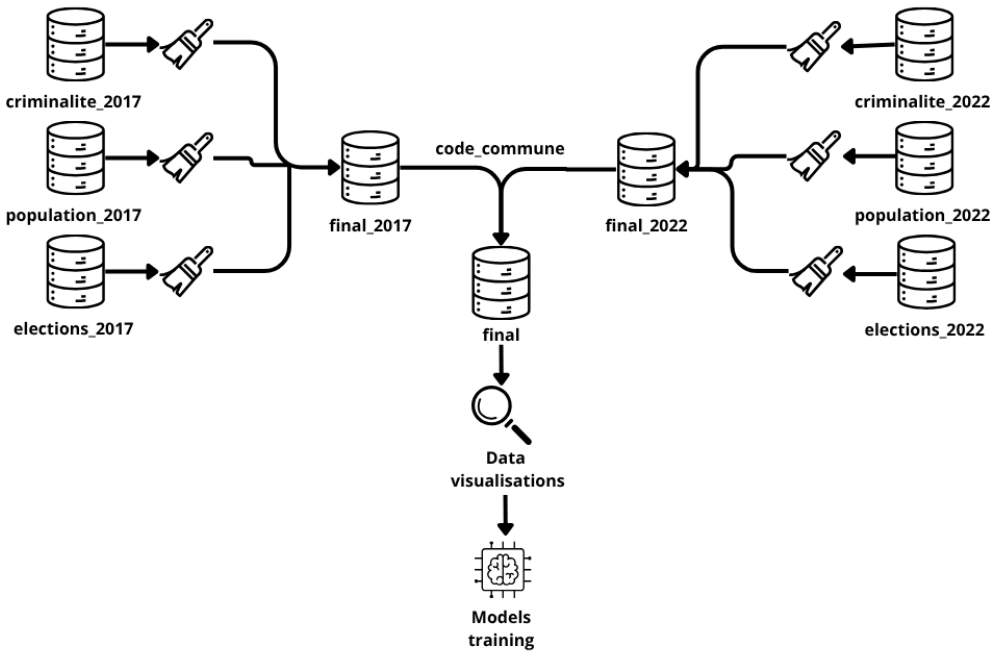

Figure 2: workflow

Le schéma ci-dessus illustre l'architecture de traitement des données mise en place pour construire le dataset final utilisé dans notre modèle de prédiction.

Dans un premier temps, trois sources de données distinctes sont mobilisées pour les années 2017 et 2022 : les données de population, les données de criminalité et les résultats des élections présidentielles. Chacune de ces sources fait l'objet d'un nettoyage préalable (traitement des valeurs manquantes, harmonisation des formats, extraction d'indicateurs pertinents).

Les données nettoyées de chaque année sont ensuite fusionnées entre elles sur la clé commune « `code_commune` » afin de constituer deux tables agrégées : `final_2017` et `final_2022`.

Ces deux tables sont ensuite jointes, toujours sur la base du code de la commune, pour constituer le dataset final regroupant les informations sur les deux périodes.

Ce dataset final est ensuite exploité dans deux objectifs principaux :

- La production de visualisations pour explorer les tendances, les corrélations et les variations temporelles,
- L'entraînement et l'évaluation de modèles de classification visant à prédire le parti politique dominant par commune.

Cette architecture permet une chaîne de traitement rigoureuse et reproductible, assurant une base de données fiable pour la modélisation.

3.2 Méthode de gestion de projet : Agile (Scrum)

Pour la conduite de ce projet, nous avons adopté une approche agile inspirée de la méthode **Scrum**, sans la structuration stricte en sprints. L'objectif était de rester **flexibles** tout en suivant une organisation **claire et collaborative**. Nous avons utilisé l'outil **Jira** pour planifier, répartir et suivre les tâches, comme illustré sur la capture d'écran ci-dessous.

Chaque membre de l'équipe s'est vu attribuer des responsabilités spécifiques en fonction de ses compétences : **collecte de données**, **nettoyage**, **modélisation**, **évaluation**, etc.

Phases du projet :

- **Initialisation du projet**
  - Définition de l'objectif : prédire les tendances électorales grâce à l'intelligence artificielle.
  - Identification des indicateurs clés : criminalité, population, entreprises, niveau d'études, part des jeunes, etc.
  - Répartition des rôles et des tâches dans Jira.
- **Collecte et préparation des données**
  - Récupération des données via **data.gouv.fr** et l'**INSEE**.
  - Nettoyage, fusion et normalisation des jeux de données (élections, criminalité, démographie etc.).
  - Outils utilisés : **Python sur Google Colab**, avec les bibliothèques **pandas**, **NumPy** et **scikit-learn**,**matplotlib**.
- **Modélisation**
  - Nous avons testé plusieurs modèles de classification supervisée :
    * XGBoost
    * Gradient Boosting
    * Random Forest Classifier

    - K-Nearest Neighbors (KNN)
    - Régression Logistique
  - Chaque modèle a été entraîné avec des données de 2017 et 2022, et comparé sur les mêmes jeux de données normalisés.

- **Évaluation**
  - Les performances des modèles ont été évaluées selon plusieurs métriques :
    - **Accuracy**
    - **Précision**
    - **Rappel (Recall)**
    - **F1 Score**
  - Nous avons privilégié ces indicateurs pour analyser l'efficacité des prédictions selon les classes politiques.

3.3 Organisation de l'équipe

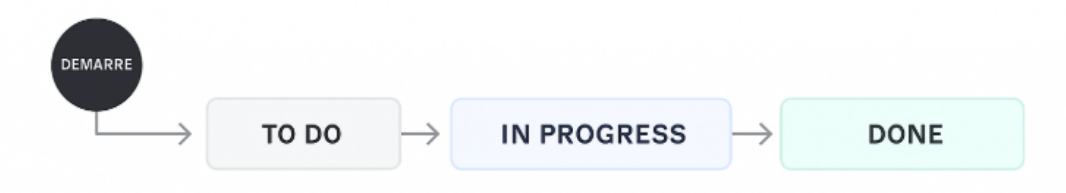

Figure 3: L'organisation du projet

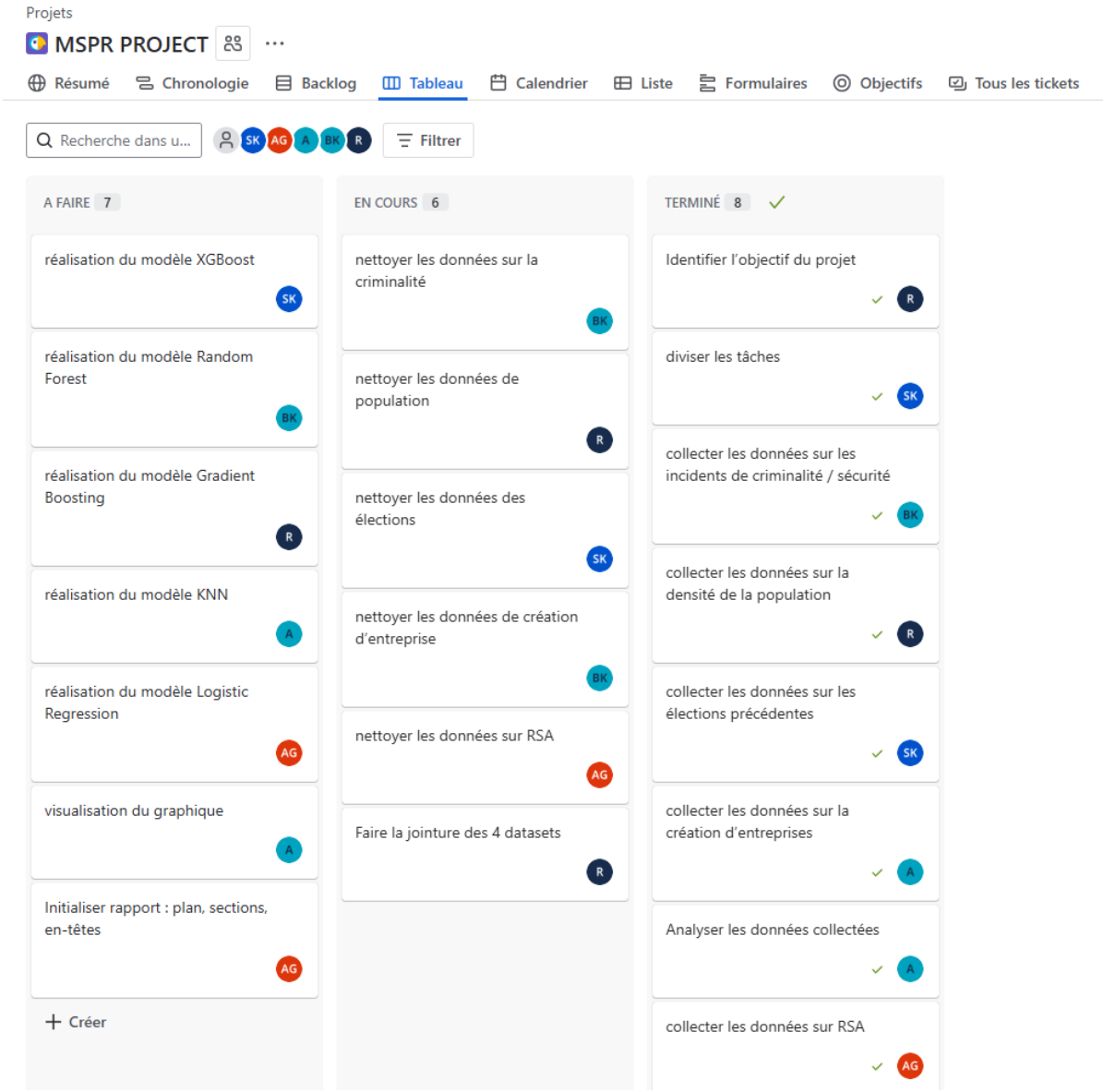

Figure 4: Suivi de projet sur Jira

Grâce à **Jira**, nous avons pu suivre efficacement l'avancement des tâches, de la collecte des données à l'entraînement des modèles. Le tableau ci-dessus montre l'organisation réelle du projet avec les tâches **à faire**, **en cours**, et **terminées**.

Chaque membre de l'équipe a contribué activement à une ou plusieurs étapes. La communication s'est faite régulièrement par messages et réunions informelles pour synchroniser les avancées et résoudre les éventuels blocages.

4 Collaboration et communication

Afin d'assurer un suivi rigoureux de notre travail, nous avons mis en place une organisation collaborative efficace. Des réunions régulières ont été tenues, aussi bien en ligne (via notre

groupe Discord) qu'en présentiel. Ces rencontres physiques ont eu lieu principalement au sein de notre école, l'EPSI, ou à la bibliothèque, afin de faire le point sur l'avancement des tâches de chacun et garantir une coordination optimale entre les membres de l'équipe.

5 Modélisation conceptuelle des données

La modélisation conceptuelle des données consiste à représenter de manière abstraite les entités, leurs attributs et les relations qui les unissent au sein d'un domaine d'application donné. Cette étape essentielle dans la conception d'un système d'information constitue la fondation d'une base de données cohérente et bien structurée. Elle facilite également la communication entre les concepteurs et les utilisateurs, en traduisant les besoins métiers en structures compréhensibles et exploitables.

Dans le cadre de notre projet, nous avons réalisé cette modélisation à l'aide de l'outil **StarUML**, qui nous a permis de construire un diagramme de classes représentant les principales entités analysées : *élections, population, criminalité* et *création d'entreprise*. Vous trouverez ci-dessous une illustration de notre diagramme conceptuel :

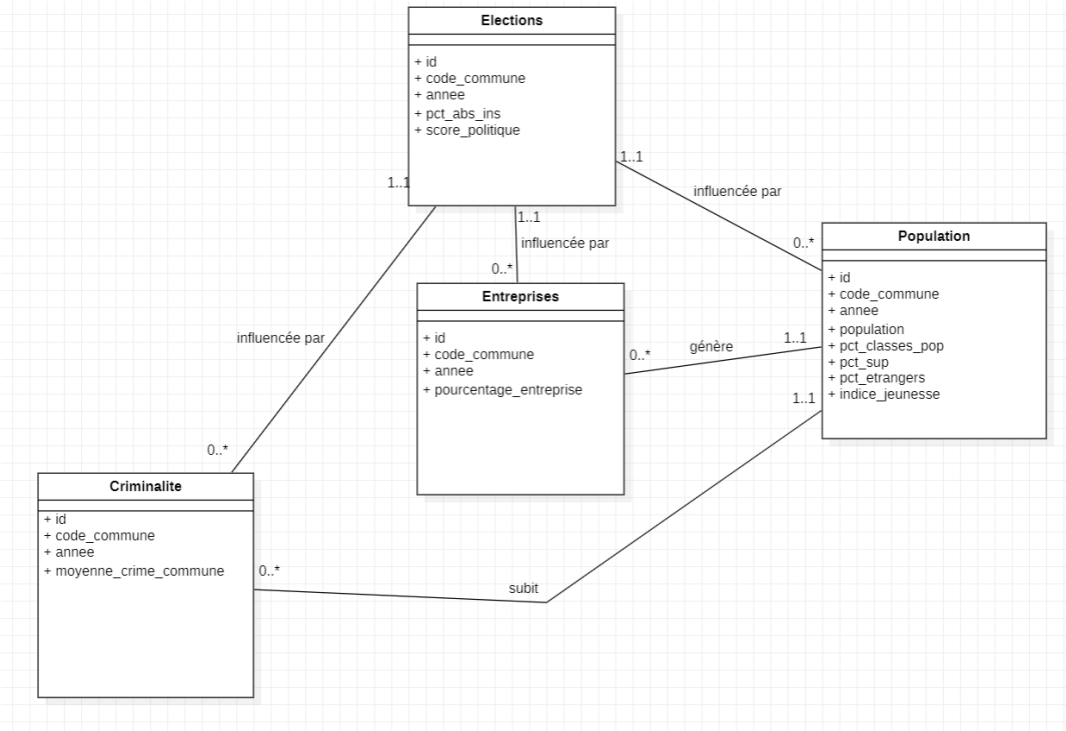

Figure 5: Modèle conceptuel des entités et relations

5.1 Entités

Les entités représentent les concepts fondamentaux observés. Dans notre modèle, nous distinguons les entités suivantes :

- **Population** : informations démographiques d'une commune donnée (taille de population, classes sociales, jeunesse, etc.).
- **Entreprises** : indicateur de l'activité économique à travers le pourcentage d'entreprises dans la commune.
- **Élections** : résultats électoraux, taux d'abstention, scores politiques, etc.
- **Criminalité** : niveau moyen de criminalité dans une commune donnée.

5.2 Attributs

Chaque entité possède plusieurs attributs qui permettent de caractériser précisément les données qu'elle représente. À titre d'exemple :

- **Population** : *population*, *pct_classes_pop*, *pct_sup*, *pct_etrangers*, *indice_jeunesse*.
- **Entreprises** : *pourcentage_entreprise*.
- **Élections** : *pct_abs_ins*, *score_politique*.
- **Criminalité** : *moyenne_crime_commune*.

5.3 Relations

Les relations entre les entités expriment les influences et interactions observées :

- La **Population** influence les résultats des **Élections**.
- La **Criminalité** peut impacter les **Élections** et l'activité des **Entreprises**.
- Les **Entreprises**, générées par la **Population**, influencent également les **Élections**.

5.4 Cardinalités

Les cardinalités définissent le nombre minimum et maximum d'occurrences qu'une entité peut avoir dans une relation donnée. Voici les cardinalités spécifiques de notre modèle :

- **Population – Élections** : une population (1..1) est toujours associée à une seule commune lors d'une élection, mais une commune peut avoir plusieurs élections successives (0..*). Cela traduit l'évolution des résultats au fil du temps dans une même zone géographique.
- **Criminalité – Élections** : une donnée de criminalité (1..1) est associée à plusieurs élections (0..*), traduisant une influence potentielle du climat sécuritaire sur les choix politiques.
- **Entreprises – Élections** : la présence d'entreprises (1..1) peut avoir une influence sur plusieurs résultats électoraux (0..*), reflétant l'impact du tissu économique local sur les opinions politiques.

- **Population – Entreprises** : une population (1..1) génère potentiellement plusieurs entreprises dans une commune (0..*), illustrant la corrélation entre densité démographique et activité économique.
- **Population – Criminalité** : une population (1..1) peut être exposée à plusieurs formes ou niveaux de criminalité (0..*), mais chaque niveau de criminalité est associé à une seule commune.

Ces cardinalités traduisent les dépendances logiques observées dans les données et permettent d'assurer l'intégrité du modèle conceptuel, tout en garantissant une représentation fidèle de la réalité.

6 DataSet final

code_commune	annee	pct_etrangers	pct_sup	pct_classes_pop	pourcentage_entreprise	population	pct_abs_ins	indice_jeunesse	moyenne_crime_commune	score_politique
1001	2017	1.0468	12.3694	37.0606	2.44	776	15.38	1.4569	63.102	4
1001	2022	1.1674	9.3001	33.4344	80.67	832	16.74	1.251	56.0485	2
1002	2017	0.823	7.6923	25.641	53.85	248	11.96	1.5	63.102	4
1002	2022	0.3776	9.9054	20.3533	73.91	267	17.84	1.6466	56.0485	2
1004	2017	8.5304	6.9361	33.7881	4.96	1154.75	22.85	1.6045	63.102	4
1004	2022	9.7887	6.9959	33.986	76.19	1154.75	23.71	1.4172	56.0485	4
1005	2017	1.3644	7.899	35.1817	3.61	1689	18.34	1.6907	63.102	4
1005	2022	1.8469	8.1076	34.4699	81.01	1897	18.25	1.4662	56.0485	4
1006	2017	6.3636	5.5556	33.3333	100	111	20.2	0.5769	63.102	1
1006	2022	3.5088	10.5263	57.8947	63.64	113	22.33	0.4375	56.0485	2
1007	2017	1.9998	9.6992	33.2606	7.69	2726	14.26	1.6903	63.102	4
1007	2022	1.9736	9.5611	33.1487	78.3	2833	16.64	1.6089	56.0485	4
1008	2017	2.3841	8.0645	35.4839	4.55	752	15.66	1.7745	63.102	4
1008	2022	2.3619	7.9335	35.4088	87.05	768	14.44	1.7007	56.0485	4
1009	2017	3.0864	1.8182	23.6364	23.81	330	16.14	0.809	63.102	2
1009	2022	2.9782	1.6694	23.4331	90.48	322	17.2	0.7187	56.0485	2
1010	2017	2.9596	5.2632	42.6901	53.19	1115	18.43	1.8079	63.102	4
1010	2022	3.7428	1.5972	32.6924	66.08	1125	21.21	1.6675	56.0485	4
1011	2017	1.8617	12.069	15.5172	45.95	376	15.64	1.6462	63.102	4
1011	2022	2.4117	16.2605	23.3915	83.19	383	15.33	1.4846	56.0485	4
1012	2017	1.5337	1.8868	32.0755	100	326	19.06	0.9176	63.102	1
1012	2022	1.4524	5.1691	28.8573	70	327	21.91	1.0247	56.0485	1
1013	2017	3.4241	8.6957	43.4783	100	144	16.54	1.2414	63.102	4
1013	2022	1.4493	7.4074	14.8148	81.82	141	23.2	0.5526	56.0485	4
1014	2017	13.5135	3.4296	32.852	0.43	3367	23.88	1.2426	63.102	2
1014	2022	15.5662	3.9554	34.7954	83.47	3409	29.07	1.0678	56.0485	1
1015	2017	1.5456	14.6789	25.6881	33.33	644	14.77	0.8466	63.102	2
1015	2022	1.5628	14.7464	25.4891	73.33	667	16.19	0.8185	56.0485	2
1016	2017	1.3151	6.0295	28.59	33.33	463	20.5	1.5958	63.102	4
1016	2022	2.4159	4.1797	34.9916	66.91	459	17.85	1.2693	56.0485	4

Figure 6: Drapeau de Normandie

7 Visualisation

Code	Partie politicue
0	Extrème gauche
1	Gauche
2	Centre
3	Droite
4	Extrème droite

Figure 7: codage des parties politiques

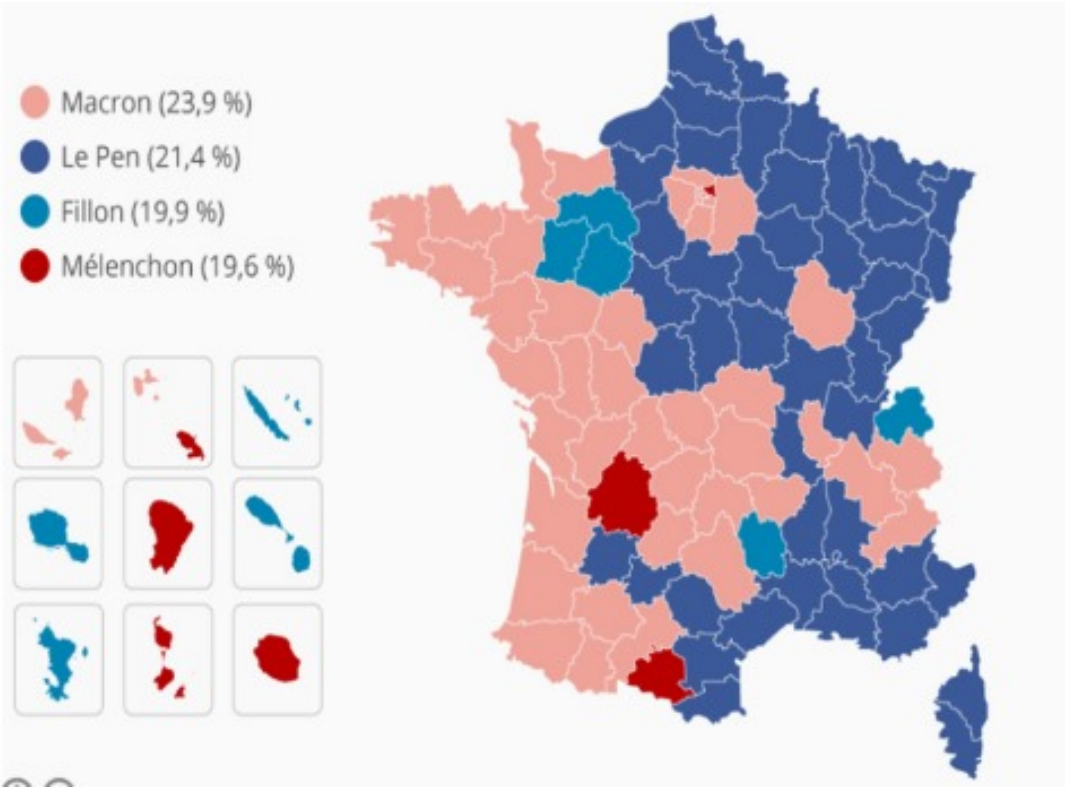

Figure 8: Résultats des élections présidentielles 2017 1er tour :

Cette visualisation compare le taux de criminalité par département en Normandie entre 2017 et 2022. En 2017, la **Seine-Maritime** enregistrait de loin le taux le plus élevé, représentant environ **35 %** de la criminalité régionale. En revanche, en 2022, la situation évolue : **l'Eure** devient le département le plus concerné, tandis que la criminalité diminue nettement en Seine-Maritime.

Cette évolution met en évidence un *basculement géographique* des actes criminels au sein de la région, suggérant une amélioration sécuritaire dans certains territoires et une possible dégradation dans d'autres.

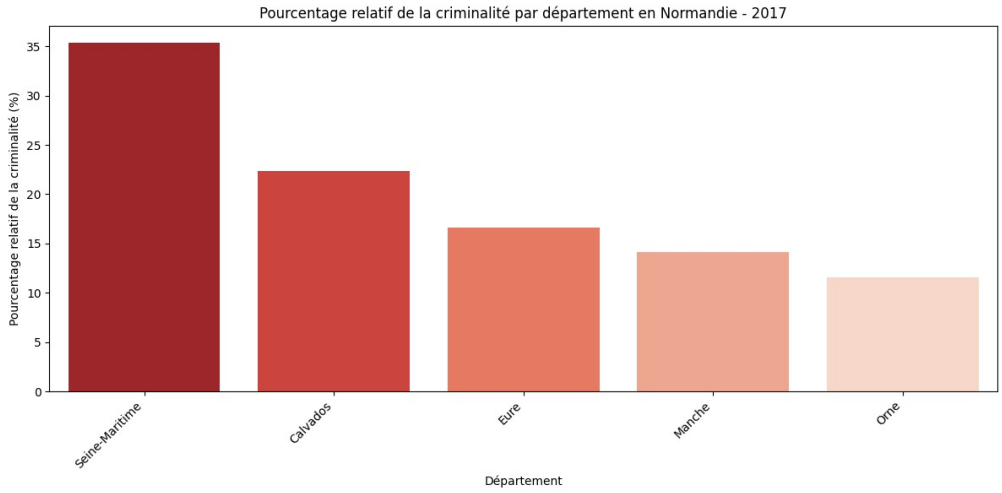

Figure 9: Pourcentage relatif de la criminalité par département en Normandie -2017

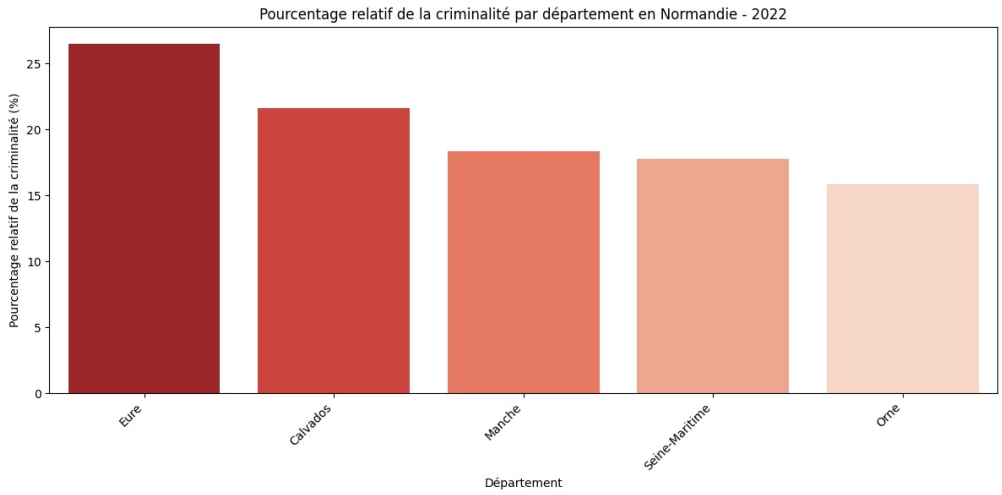

Figure 10: Pourcentage relatif de la criminalité par département en Normandie - 2022

Cette visualisation compare la répartition des **classes populaires** et des **classes supérieures** par département en Normandie entre 2017 et 2022.

On observe que les classes populaires restent **largement majoritaires** dans tous les départements, bien que leur part ait **légèrement diminué** dans certains cas comme le **Calvados** ou l'**Orne**.

Parallèlement, une **progression modeste mais notable** des classes supérieures apparaît, notamment en **Seine-Maritime** et dans le **Calvados**, suggérant une *évolution socio-économique progressive* dans certaines zones de la région.

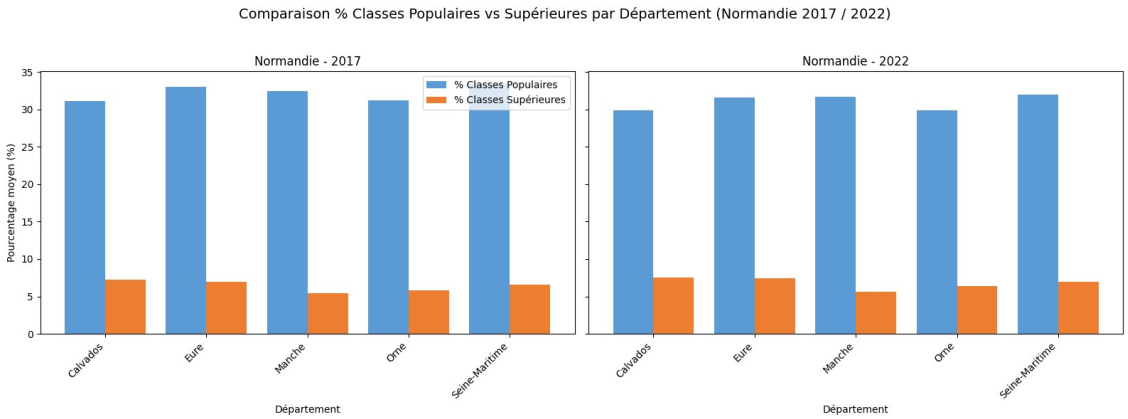

Figure 11: Comparaison du pourcentage de classes populaires et de classes supérieures par département en Normandie 2017 - 2022

Cette visualisation montre l'évolution de l'**indice de jeunesse moyen** selon le **score politique** en Normandie entre 2017 et 2022. En 2017 comme en 2022, on observe que les **scores politiques les plus élevés** (notamment le *score 4*) sont associés à un **indice de jeunesse plus important**.

Cela signifie que les territoires où la population jeune est plus présente tendent à obtenir des scores politiques plus élevés. Cette corrélation suggère que la **jeunesse** pourrait jouer un rôle significatif dans les *dynamiques électorales régionales*, en influençant les tendances

politiques locales.

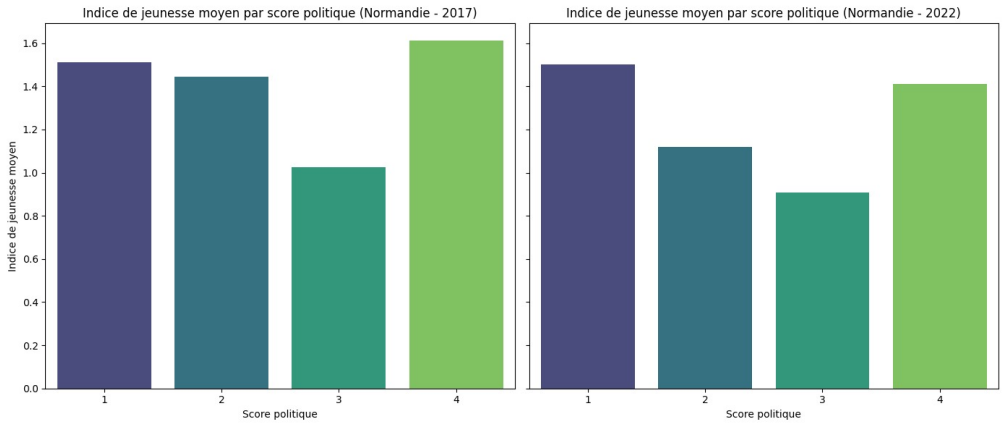

Figure 12: Indice de jeunesse moyen par score politique 2017 - 2022

Cette visualisation montre l'évolution du **pourcentage moyen d'étrangers** par département en Normandie entre 2017 et 2022. On constate une **légère hausse** dans tous les départements sur la période.

L'Orne et **la Manche** restent les départements affichant les **taux les plus élevés** de population étrangère, renforçant leur position en tête du classement régional.

Cette tendance suggère une *dynamique migratoire modérée mais continue* dans la région.

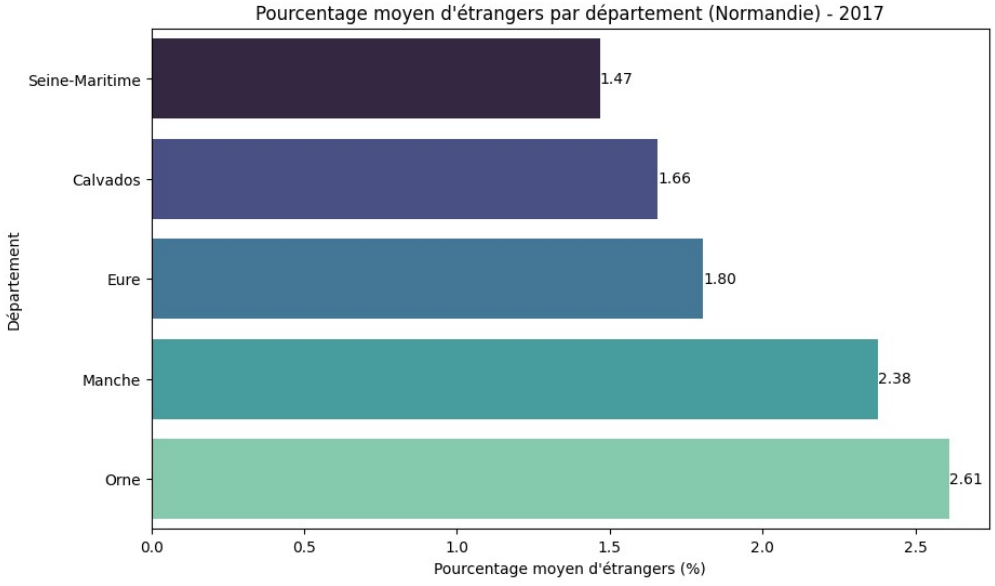

Figure 13: Pourcentage moyen d'étrangers par département en Normandie 2017

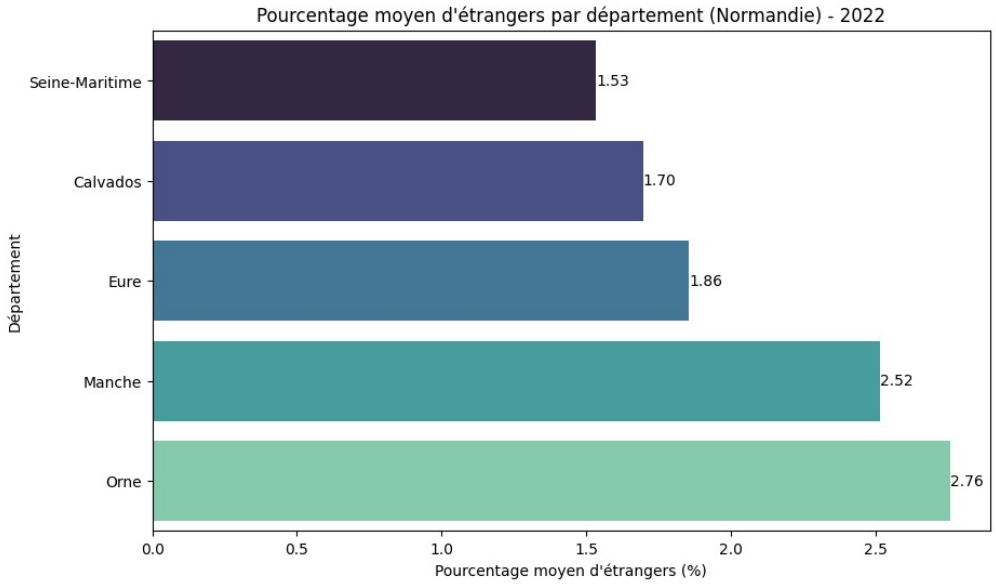

Figure 14: Pourcentage moyen d'étrangers par département en Normandie 2022

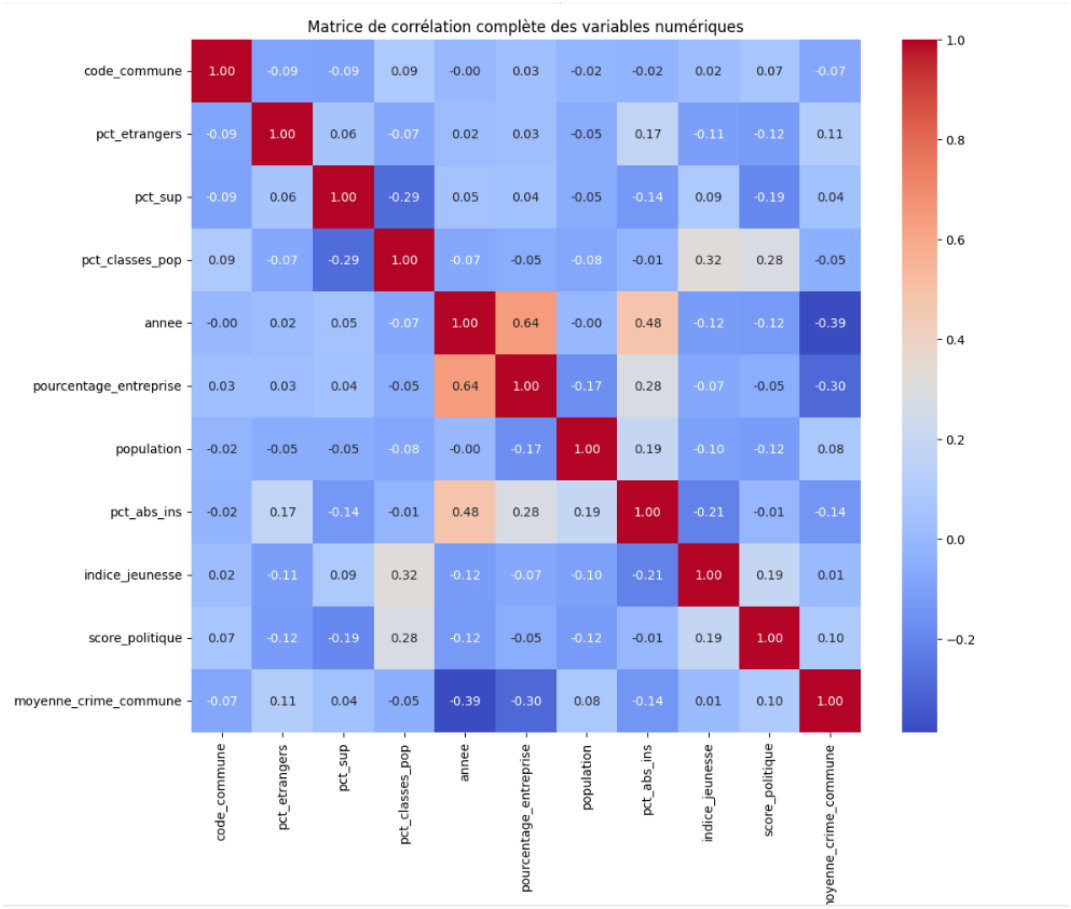

Figure 15: la matrice de corrélation

L'analyse de la matrice de corrélation complète des variables numériques révèle plusieurs relations notables entre les indicateurs socio-démographiques et économiques des communes.

- On remarque d'abord une **forte corrélation positive entre l'année et le pourcentage d'entreprises** (0,64), ce qui peut traduire un développement économique progressif au fil des années, avec une augmentation de la création d'entreprises.
- De même, **l'indice de jeunesse est modérément corrélé avec le pourcentage de classes populaires** (0,32), suggérant que les communes ayant une population plus jeune tendent à avoir une part plus importante de catégories sociales populaires.
- Par ailleurs, **ce même indice est négativement corrélé au taux d'abstention** (-0,21), ce qui peut laisser penser que les jeunes sont potentiellement plus mobilisés politiquement dans certaines zones.
- Une **corrélation modérée est aussi visible entre l'année et le taux d'abstention** (0,48), ce qui pourrait indiquer une tendance à la hausse de l'abstention au fil du temps.
- En revanche, **le score politique est faiblement lié aux autres variables**, ce qui peut signifier que les préférences politiques locales sont moins déterminées par les facteurs socio-économiques mesurés ici.
- Enfin, **on note une corrélation négative entre la moyenne de crimes par commune et l'année** (-0,39), ce qui pourrait refléter une baisse des crimes déclarés au cours du temps, potentiellement due à des politiques locales de sécurité ou des changements démographiques.

8 Démarche suivie

Nous avons initié notre travail par une analyse approfondie des différentes tables disponibles. La première phase a consisté en un **nettoyage rigoureux**, avec l'élimination des variables jugées non pertinentes ou redondantes. Cette étape a été suivie d'**agrégations ciblées**, permettant de consolider l'information et d'en améliorer la lisibilité.

Ces agrégations ont structuré les informations de manière **cohérente** et fidèle aux réalités de chaque territoire. La table finale obtenue offre une base de travail **solide**, **homogène** et **exploitable** pour l'entraînement d'un **modèle d'apprentissage supervisé**.

Certains jeux de données, comme ceux relatifs à la **criminalité**, n'étaient disponibles qu'au niveau **départemental**. Nous avons donc **calculé des moyennes** pour projeter ces données à l'échelle communale. Bien que d'autres sources aient fourni des informations directement par commune, cette **hétérogénéité** nous a conduits à construire une **base harmonisée**, **cohérente à l'échelle locale**.

Enfin, nous avons appliqué une répartition classique des données : **80 % pour l'apprentissage** et **20 % pour les tests**, afin d'évaluer rigoureusement la **capacité de généralisation** du

modèle.

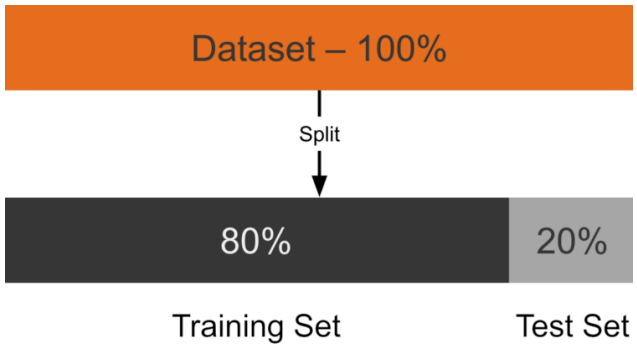

Figure 16: Répartition des données : **80 %** pour l'apprentissage et **20 %** pour le test.

9 Modèles évalués

Dans le cadre de notre démarche d'évaluation, plusieurs algorithmes de classification reconnus pour leur performance ont été testés. Parmi ceux-ci figurent :

- **Random Forest** : un modèle qui utilise plusieurs arbres de décision et combine leurs résultats pour obtenir une meilleure précision.
- **Régression Logistique** : un modèle qui prédit la probabilité qu'un élément appartienne à une catégorie donnée.
- **Gradient Boosting** : un modèle qui construit plusieurs petits modèles les uns après les autres, chacun corrigeant les erreurs du précédent.
- **XGBoost** : une version améliorée du Gradient Boosting, plus rapide et souvent plus précise.
- **K-Nearest Neighbors (KNN)** : un modèle qui classe un élément en fonction des catégories des éléments les plus proches de lui.

Pour chaque modèle, après entraînement sur l'ensemble d'apprentissage, nous avons procédé à leur évaluation sur l'ensemble de test. La précision a été calculée pour chacun, offrant ainsi une comparaison directe de leur capacité à prédire correctement les classes.

Ces résultats fournissent des indications précieuses pour orienter le choix du modèle le

plus adapté à notre problème de classification.

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# Chargement des données
df = pd.read_csv("dataset_final_vinf.csv")

# Extraction du département (2 premiers chiffres)
df["departement"] = df["code_commune"].astype(str).str[:2]

# Filtrer les départements Normandie
departements_normandie = ["14", "27", "50", "61", "76"]
df = df[df["departement"].isin(departements_normandie)]

# Séparation features / target
X = df.drop(columns=["score_politique", "code_commune", "annee", "departement"])
y = df["score_politique"]

# Encoder les labels avec LabelEncoder
le = LabelEncoder()
y_enc = le.fit_transform(y)  # classes 1,2,3,4 deviennent 0,1,2,3

# Standardisation des features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split train/test stratifié sur y_enc
try:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y_enc, test_size=0.2, random_state=42, stratify=y_enc
    )
except ValueError:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y_enc, test_size=0.2, random_state=42
    )

# Modèle XGBoost
model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', random_state=42)
model.fit(X_train, y_train)

# Prédictions
y_pred = model.predict(X_test)

# Décoder les labels pour revenir à l'original (1,2,3,4)
y_test_orig = le.inverse_transform(y_test)
y_pred_orig = le.inverse_transform(y_pred)

# Évaluation
print("📊 Résultats XGBoost")
print(f"Accuracy : {accuracy_score(y_test_orig, y_pred_orig):.4f}")
print(f"Precision: {precision_score(y_test_orig, y_pred_orig, average='macro', zero_division=0):.4f}")
print(f"Recall   : {recall_score(y_test_orig, y_pred_orig, average='macro', zero_division=0):.4f}")
print(f"F1 Score : {f1_score(y_test_orig, y_pred_orig, average='macro', zero_division=0):.4f}")

print("\nRapport détaillé :")
print(classification_report(y_test_orig, y_pred_orig, zero_division=0))
```

Figure 17: Entraînement d'un modèle XGBoost

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# Chargement des données
df = pd.read_csv("dataset_final_vinf.csv")

# Extraction département (2 premiers chiffres)
df["departement"] = df["code_commune"].astype(str).str[:2]

# Filtrer départements Normandie
departements_normandie = ["14", "27", "50", "61", "76"]
df = df[df["departement"].isin(departements_normandie)]

# Séparation features / target
X = df.drop(columns=["score_politique", "code_commune", "annee", "departement"])
y = df["score_politique"]

# Standardisation
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split train/test
try:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42, stratify=y)
except ValueError:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42)

# Modèle Gradient Boosting
model = GradientBoostingClassifier(random_state=42)
model.fit(X_train, y_train)

# Prédiction
y_pred = model.predict(X_test)

# Évaluation
print("📊 Résultats Gradient Boosting")
print(f"Accuracy : {accuracy_score(y_test, y_pred):.4f}")
print(f"Precision: {precision_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"Recall   : {recall_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"F1 Score : {f1_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print("\nRapport détaillé :")
print(classification_report(y_test, y_pred, zero_division=0))
```

Figure 18: Entraînement d'un modèle Gradient Boosting

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# Chargement des données
df = pd.read_csv("dataset_final.csv")

# Extraction du département (les 2 premiers chiffres du code commune)
df["departement"] = df["code_commune"].astype(str).str[:2]

# Filtrer uniquement les départements de la région Normandie
departements_normandie = ["14", "27", "50", "61", "76"]
df = df[df["departement"].isin(departements_normandie)]

# Séparation features / target
X = df.drop(columns=["score_politique", "code_commune", "annee", "departement"])
y = df["score_politique"]

# Standardisation des features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split train / test avec stratification si possible
try:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42, stratify=y
    )
except ValueError:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42
    )
# Modèle Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42, class_weight='balanced')
# Entraînement
model.fit(X_train, y_train)

# Prédictions
y_pred = model.predict(X_test)

# Évaluation
print("📊 Résultats Random Forest")
print(f"Accuracy : {accuracy_score(y_test, y_pred):.4f}")
print(f"Precision: {precision_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"Recall   : {recall_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"F1 Score : {f1_score(y_test, y_pred, average='macro', zero_division=0):.4f}")

# Rapport de classification
print("\nRapport de classification détaillé :")
print(classification_report(y_test, y_pred, zero_division=0))
```

Figure 19: Entraînement d'un modèle Random Forest

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# Chargement des données
df = pd.read_csv("dataset_final_vinf.csv")

# Extraction département (2 premiers chiffres)
df["departement"] = df["code_commune"].astype(str).str[:2]

# Filtrer départements Normandie
departements_normandie = ["14", "27", "50", "61", "76"]
df = df[df["departement"].isin(departements_normandie)]

# Séparation features / target
X = df.drop(columns=["score_politique", "code_commune", "annee", "departement"])
y = df["score_politique"]

# Standardisation
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split train/test
try:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42, stratify=y)
except ValueError:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42)

# Modèle Régression Logistique
model = LogisticRegression(max_iter=1000, random_state=42, multi_class='ovr', class_weight='balanced')
model.fit(X_train, y_train)

# Prédiction
y_pred = model.predict(X_test)

# Évaluation
print("📊 Résultats Régression Logistique")
print(f"Accuracy : {accuracy_score(y_test, y_pred):.4f}")
print(f"Precision: {precision_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"Recall   : {recall_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"F1 Score : {f1_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print("\nRapport détaillé :")
print(classification_report(y_test, y_pred, zero_division=0))
```

Figure 20: Entraînement d'un modèle Régression logistique

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# Chargement des données
df = pd.read_csv("dataset_final_vinf.csv")

# Extraction département (2 premiers chiffres)
df["departement"] = df["code_commune"].astype(str).str[:2]

# Filtrer départements Normandie
departements_normandie = ["14", "27", "50", "61", "76"]
df = df[df["departement"].isin(departements_normandie)]

# Séparation features / target
X = df.drop(columns=["score_politique", "code_commune", "annee", "departement"])
y = df["score_politique"]

# Standardisation
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split train/test
try:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42, stratify=y)
except ValueError:
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42)

# Modèle KNN
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)

# Prédiction
y_pred = model.predict(X_test)

# Évaluation
print("📊 Résultats KNN")
print(f"Accuracy : {accuracy_score(y_test, y_pred):.4f}")
print(f"Precision: {precision_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"Recall   : {recall_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print(f"F1 Score : {f1_score(y_test, y_pred, average='macro', zero_division=0):.4f}")
print("\nRapport détaillé :")
print(classification_report(y_test, y_pred, zero_division=0))
```

Figure 21: Entraînement d'un modèle K-Nearest Neighbors (KNN)

10 Résultats des modèles

Les métriques suivantes ont été utilisées pour évaluer les performances du modèle. Chacune apporte une information spécifique sur sa capacité à bien classer les données :

- **Accuracy (Exactitude)** : représente la proportion totale de prédictions correctes sur l'ensemble du jeu de test. Par exemple, une accuracy de 0,72 signifie que le modèle a correctement prédit la classe dans 72 % des cas. *Plus la valeur est proche de 1, meilleure est la performance globale.*

- **Précision (Precision)** : indique, pour une classe donnée, la proportion de prédictions correctes parmi toutes les prédictions effectuées pour cette classe. Par exemple, une précision de 0,64 signifie que sur 100 prédictions en faveur de cette classe, 64 étaient exactes. *Une précision élevée signifie que le modèle commet peu de fausses alertes (faux positifs).*

- **Rappel (Recall)** : mesure, pour chaque classe, la capacité du modèle à retrouver tous les exemples pertinents. Par exemple, un rappel de 0,60 signifie que le modèle a détecté 60 % des cas réels de cette classe. *Un bon rappel reflète une faible proportion de cas oubliés (faux négatifs).*

- **F1-score** : correspond à la moyenne harmonique entre la précision et le rappel. Cette métrique permet d'avoir une mesure globale de la performance d'un modèle, notamment lorsque les classes sont déséquilibrées. *Plus le F1-score est élevé, plus le modèle est à la fois précis et complet.*

- **Support** : indique le nombre d'exemples appartenant à chaque classe dans le jeu de test. Cette information ne reflète pas une performance, mais elle permet d'apprécier l'importance relative de chaque classe dans les résultats.

- **Macro moyenne (macro avg)** : calcule la moyenne des métriques (précision, rappel, F1-score) pour chaque classe, sans pondération selon leur taille. *Elle fournit une vue équilibrée des performances, indépendamment de la distribution des classes.*

L'analyse conjointe de ces métriques permet d'avoir une vision complète de la performance du modèle, à la fois sur l'ensemble des classes et sur chacune d'elles individuellement. Elle est particulièrement utile dans les contextes où certaines classes sont sous-représentées.

📊 Résultats XGBoost
Accuracy : 0.7226
Precision: 0.6427
Recall   : 0.5351
F1 Score : 0.5698

Rapport détaillé :

	precision	recall	f1-score	support
1	0.62	0.27	0.37	30
2	0.64	0.60	0.62	310
3	0.54	0.43	0.48	116
4	0.78	0.84	0.81	723
accuracy			0.72	1179
macro avg	0.64	0.54	0.57	1179

Figure 22: Résultat modèle XGBoost

📊 Résultats Gradient Boosting
Accuracy : 0.7337
Precision: 0.6794
Recall   : 0.5320
F1 Score : 0.5730

Rapport détaillé :

	precision	recall	f1-score	support
1	0.70	0.23	0.35	30
2	0.68	0.57	0.62	310
3	0.56	0.46	0.50	116
4	0.77	0.87	0.82	723
accuracy			0.73	1179
macro avg	0.68	0.53	0.57	1179

Figure 23: Résultat modèle Gradient Boosting

📊 Résultats Random Forest
Accuracy : 0.7142
Precision: 0.7458
Recall   : 0.4521
F1 Score : 0.4920

Rapport détaillé :

	precision	recall	f1-score	support
1	1.00	0.10	0.18	30
2	0.69	0.50	0.58	310
3	0.56	0.31	0.40	116
4	0.73	0.89	0.80	723
accuracy			0.71	1179
macro avg	0.75	0.45	0.49	1179

Figure 24: Résultat modèle Random Forest

📊 Résultats Régression Logistique
Accuracy : 0.5649
Precision: 0.4180
Recall   : 0.4807
F1 Score : 0.4253

Rapport détaillé :

	precision	recall	f1-score	support
1	0.07	0.23	0.11	30
2	0.48	0.53	0.50	310
3	0.31	0.57	0.40	116
4	0.81	0.59	0.68	723
accuracy			0.56	1179
macro avg	0.42	0.48	0.43	1179

Figure 25: Résultat modèle Régression logistique

📊 Résultats KNN
Accuracy : 0.6361
Precision: 0.4788
Recall   : 0.4238
F1 Score : 0.4431

Rapport détaillé :

	precision	recall	f1-score	support
1	0.33	0.20	0.25	30
2	0.48	0.46	0.47	310
3	0.37	0.24	0.29	116
4	0.73	0.79	0.76	723
accuracy			0.64	1179
macro avg	0.48	0.42	0.44	1179

Figure 26: Résultat modèle K-Nearest Neighbors (KNN)

11 Outils et Technologies Utilisés

Figure 27: Outils et Technologies Utilisés

- **Matplotlib** : bibliothèque Python utilisée pour créer des visualisations graphiques comme des courbes, histogrammes ou nuages de points.
  Elle est largement utilisée en Data Science pour l'exploration visuelle des données.
- **Greenleaf** : symbole souvent associé aux outils ou bibliothèques orientés vers l'analyse environnementale ou durable — à préciser selon le contexte exact.
  Utile dans les projets à visée écologique ou d'analyse environnementale.
- **Microsoft Excel** : logiciel de tableur permettant de manipuler, analyser et visualiser des données sous forme de feuilles de calcul.
  Il est souvent utilisé pour des tâches simples de prétraitement ou d'analyse rapide.
- **Jupyter Notebook** : environnement interactif qui permet d'écrire et d'exécuter du code Python tout en documentant avec du texte, des graphiques et des équations.
  Idéal pour prototyper rapidement des analyses de données reproductibles.
- **Google Colab** : plateforme cloud gratuite de Google qui permet d'exécuter des notebooks Jupyter avec support GPU sans configuration locale.
  Elle facilite la collaboration et permet d'exécuter des notebooks même sans machine puissante.
- **Jira** : outil de gestion de projet et de suivi de tâches largement utilisé dans le développement logiciel agile.
  Il permet de structurer efficacement le travail en équipe avec des sprints, des tickets et des backlogs.

- **Python** : langage de programmation polyvalent, populaire en data science, développement web, automatisation et intelligence artificielle.
  Sa richesse en bibliothèques (Pandas, Scikit-learn, TensorFlow...) en fait un outil central en analyse de données.

12 Conclusion

Cette recherche met en lumière l'efficacité des techniques d'intelligence artificielle pour anticiper les comportements électoraux. En s'appuyant sur des données socio-économiques, nos modèles ont réussi à prévoir de manière fiable les résultats des scrutins présidentiels, ce qui confirme le rôle clé des contextes économiques et sociaux dans les décisions de vote. Ces informations peuvent ainsi guider les stratégies électorales en permettant aux acteurs politiques d'adapter leur discours aux attentes réelles des électeurs.

Néanmoins, il convient de garder à l'esprit que le paysage politique reste soumis à de nombreuses influences imprévisibles et complexes. Les outils d'IA doivent donc être intégrés avec discernement, en complément d'autres analyses qualitatives. En poursuivant le perfectionnement des modèles et en testant leur robustesse sur divers territoires, cette approche ouvre un champ prometteur pour mieux comprendre les évolutions politiques et soutenir des choix éclairés dans la prise de décision publique.

13 Sources et Références

- **Résultats électoraux officiels – Ministère de l'Intérieur :**
  https://www.resultats-elections.interieur.gouv.fr
- **Données démographiques et économiques – INSEE :**
  https://www.insee.fr/fr/statistiques/2011101?geo=REG-84
- **Plateforme ouverte des données publiques – Data.gouv.fr :**
  https://www.data.gouv.fr
- **Données électorales – Data.gouv.fr :**
  https://www.data.gouv.fr/fr/pages/donnees-des-elections/
- **Données sur la sécurité (police, gendarmerie) – Data.gouv.fr :**
  https://www.data.gouv.fr/fr/pages/donnees-securite/
- **Données sur l'emploi – Data.gouv.fr :**
  https://www.data.gouv.fr/fr/pages/donnees-emploi/
- **Jeux de données INSEE – Organisation officielle :**
  https://www.data.gouv.fr/fr/organizations/institut-national-de-la-statistique-et-des-etudeseconomiq
  insee/?datasets$_page = 7organization - datasets$
- **Crimes et délits enregistrés par la police et la gendarmerie (depuis 2012) :**
  https://www.data.gouv.fr/fr/datasets/crimes-et-delits-enregistres-par-les-services-de-gendarmerie-et-de-police-depuis-2012/