AVIGNON UNIVERSITÉ

Fouille de données et aide à la décision

Groupe 10
Fataï IDRISSOU
Paul Moïse GANGBADJA

6 avril 2025

Master 2 informatique
IA
UE Business intelligence & Systèmes décisionnels
ECUE Application Business Intelligence

Responsable
Remy Kessler

UFR
SCIENCES
TECHNOLOGIES
SANTÉ

AVIGNON UNIVERSITÉ
CERI

CENTRE
D'**ENSEIGNEMENT**
ET DE **RECHERCHE**
EN **INFORMATIQUE**
ceri.univ-avignon.fr

Sommaire

Titre 1

Sommaire 2

1 Présentation 4
- 1.1 Contexte . . . 4
- 1.2 Organisation . . . 5

2 Données 6
- 2.1 Données communes aux deux tables . . . 6
- 2.2 Spécificités de `table1.csv` . . . 6
- 2.3 Spécificité de `table2.csv` . . . 7
- 2.4 Nettoyage . . . 7
- 2.5 Analyse descriptive . . . 8

3 Analyse des comportements de démission 9
- 3.1 Facteurs influençant la démission . . . 9
- 3.2 Caractéristiques liées à la durée d'adhésion . . . 9
- 3.3 Évolution temporelle des adhésions et démissions . . . 9
- 3.4 Identification de profils types . . . 10
- 3.5 Évaluation . . . 10
  - Découpage des données. . . . 10
  - Équilibrage des classes. . . . 10
  - Justification du découpage temporel. . . . 10
  - * . . . 11
- 3.6 Implémentation et choix des paramètres . . . 11
  - Support Vector Machines (SVM) : . . . 11
  - k-Nearest Neighbors (kNN) : . . . 12
  - Random Forest : . . . 12
  - Naive Bayes . . . 12
  - Type utilisé : Gaussian Naive Bayes . . . 12

4 Résultats 12
- 4.1 Performances individuelles . . . 13
- 4.2 Comparaison et généralisation . . . 13

5 Conclusion générale 14

6 Présentation 14
- 6.1 Organisation . . . 14
  - 6.1.1 La composition du groupe . . . 14
  - 6.1.2 Répartition du travail . . . 14
  - 6.1.3 Organisation du travail dans le temps . . . 15
  - 6.1.4 Distribution des taches entre les membres du groupe . . . 15

7 Données 15
- 7.1 Caractéristiques . . . 15
- 7.2 Nettoyage . . . 17
- 7.3 Relations entre les attributs . . . 17

8 Méthodes **17**
8.1 Outils d'aide à la décision . . . 17
8.1.1 Algorithmes appliqués . . . 17
8.1.2 Choix des paramètres . . . 17
8.2 Evaluation . . . 19
8.2.1 Classement relatif des véhicules . . . 19
8.3 Implémentation . . . 20
8.3.1 Etapes détaillées de chaque méthode . . . 20

9 Résultats **22**
9.1 Graphes (Professionnels actifs . . . 22
9.2 Interpretation . . . 24
9.3 Classements comparés des méthodes . . . 25

10 Conclusion **25**
10.1 Critique du projet . . . 25
10.2 Propositions pour améliorer le projet . . . 25
10.3 Perspectives . . . 25

Bibliographie **26**

Sujet Fouille de données (Données Bancaires)

1 Présentation

Le domaine de la science des données est devenu incontournable dans de nombreux secteurs d'activité, offrant des solutions puissantes pour comprendre et optimiser des processus complexes. Ce projet a pour objectif de démontrer l'application de techniques de data science à deux domaines distincts mais complémentaires : l'analyse de données bancaires et l'aide à la décision dans le secteur automobile.

Dans un premier temps, ce projet s'intéresse à l'analyse de données clients issues d'une organisation mutualiste, avec pour objectif de prédire la fidélité des adhérents à travers l'étude de données démographiques, économiques et comportementales. L'accent est mis sur la préparation des données : nettoyage, transformation, et fusion de fichiers hétérogènes. Une fois cette étape accomplie, des analyses exploratoires et des modélisations prédictives sont envisagées pour mieux comprendre les facteurs influençant la rétention ou le départ des clients.

Dans un second temps, l'objectif est de développer un système d'aide à la décision pour les consommateurs dans le secteur automobile. Grâce à l'analyse de données relatives aux véhicules, telles que la consommation de carburant, les émissions de CO_2, la sécurité ou le coût, ce projet cherche à fournir des recommandations personnalisées adaptées aux préférences et besoins des utilisateurs. Des approches de modélisation multicritère et de scoring seront explorées pour offrir un outil performant permettant d'accompagner les décisions d'achat.

Ainsi, ce projet illustre la diversité des applications possibles de la science des données, allant de la gestion des relations clients dans le secteur bancaire à la création de systèmes d'assistance pour des choix éclairés dans le secteur automobile. Il repose sur une approche rigoureuse de collecte, de nettoyage et de transformation des données, afin de les rendre exploitables pour des analyses approfondies et des prises de décision efficaces.

1.1 Contexte

Ce projet s'inscrit dans le cadre de deux thématiques distinctes mais complémentaires en matière de data science appliquée : l'analyse de données bancaires d'une part, et l'aide à la décision dans le domaine automobile d'autre part.

La première partie du projet repose sur deux fichiers sources, `table1.csv` et `table2.csv`, contenant des données clients issues d'une organisation mutualiste. L'objectif principal est de préparer ces données à des fins d'analyse prédictive, notamment pour identifier les facteurs influençant la fidélité ou la démission des clients. Cela comprend une phase exploratoire avec visualisation des distributions, le calcul de statistiques descriptives, et un important travail de nettoyage : gestion des valeurs manquantes, traitement des valeurs aberrantes, et harmonisation des formats et codifications. Les deux tables sont fusionnées selon une clé commune, avec un recodage et un prétraitement adapté, en vue d'analyses avancées comme la segmentation ou la modélisation du churn.

La seconde partie du projet s'oriente vers l'aide à la décision pour les usagers automobiles. L'objectif est de concevoir un système de recommandation ou d'assistance basé sur des données liées aux caractéristiques des véhicules (consommation, sécurité, coût, émissions de CO_2, etc.) ainsi qu'aux préférences des utilisateurs. Dans ce contexte, le travail porte sur la structuration d'un jeu de données adapté à des analyses multicritères. Des méthodes d'analyse comparative (telles que l'analyse multicritère ou les systèmes de scoring) sont explorées pour fournir un outil d'aide au choix pertinent pour les consommateurs.

Ce projet mobilise ainsi des compétences variées en science des données : exploration,

nettoyage, transformation, fusion de sources hétérogènes, encodage de variables, mais aussi modélisation et formalisation de critères décisionnels. Il illustre la diversité des cas d'usage de la data science, entre optimisation de la relation client dans le secteur bancaire et accompagnement intelligent dans le domaine de la mobilité.

1.2 Organisation

Notre groupe est composé de deux personnes Paul - Fataï.

Répartition des tâches

La répartition des tâches a été faite de manière équilibrée, en tenant compte des compétences de chacun :

- Fatai s'est chargé de l'analyse de la qualité des données, de l'identification des variables clés, du choix des méthodes adaptées pour traiter les valeurs manquantes, ainsi que du nettoyage des deux tables disponibles en vue de leur fusion tout en préservant les informations essentielles. Il a ensuite fusionné les données, entraîné et évalué les modèles SVM et Naive Bayes.
- Paul a travaillé sur la description des variables et l'analyse des relations statistiques entre elles. Il a également entraîné les modèles Random Forest et KNN.

Bibliothèques utilisées

Nous avons utilisé plusieurs bibliothèques Python pour mener à bien ce projet :

- `pandas` : pour le traitement et la manipulation des données tabulaires.
- `numpy` : pour les opérations numériques efficaces.
- `matplotlib` et `seaborn` : pour la visualisation des données.
- `scikit-learn` : pour certaines tâches de prétraitement (imputation, normalisation) et l'analyse exploratoire.
- `scipy` : pour les calculs mathématiques et statistiques complémentaires.

Toutes ces bibliothèques ont été choisies pour leur robustesse et leur familiarité, certaines étant déjà utilisées en TP. Aucun outil exotique ou non vu en cours n'a été introduit sans justification claire.

Le script implémente différentes étapes de traitement et de visualisation de données à l'aide de bibliothèques couramment utilisées en data science, notamment `pandas`, `numpy`, `matplotlib.pyplot`, et `seaborn`. Ces bibliothèques sont respectivement utilisées pour la manipulation des données tabulaires, les opérations mathématiques de base, la génération de graphiques, et l'amélioration esthétique des visualisations.

Le traitement commence par le chargement des données à partir de deux fichiers CSV via la fonction `read_csv()` de `pandas`. Ces données représentent des informations client et transactionnelles. Une phase d'exploration initiale est réalisée, incluant un affichage des premières lignes, le résumé statistique des variables numériques (via `describe()`) et l'analyse des types de variables avec `info()`. Cela permet d'identifier les variables pertinentes et de repérer d'éventuelles incohérences.

La visualisation est assurée grâce à des histogrammes pour analyser la distribution des variables numériques, des boxplots pour identifier les valeurs aberrantes, et des graphiques circulaires personnalisés pour observer la répartition des modalités des variables catégorielles. Une fonction spécifique est définie pour automatiser la création de ces graphiques circulaires, avec des paramètres dynamiques permettant de cibler n'importe quelle colonne du jeu de données.

Le nettoyage des données intervient ensuite : les valeurs manquantes sont identifiées avec `isnull()` et remplacées ou supprimées selon leur impact. Des valeurs aberrantes sont

détectées grâce à des méthodes statistiques comme l'écart interquartile (IQR) et traitées en conséquence. Certaines colonnes contenant des dates sont converties au bon format à l'aide de `pd.to_datetime()`, afin de garantir leur cohérence temporelle.

La fusion des deux jeux de données est effectuée à l'aide de la méthode `merge()` de `pandas`, en utilisant une clé primaire commune. Cette opération permet d'enrichir les informations disponibles pour chaque observation. Après cette étape, un recodage est appliqué sur les variables catégorielles à l'aide de l'encodage one-hot (via `get_dummies()`) afin de les rendre exploitables pour une analyse ou un modèle de machine learning. Les variables numériques sont ensuite normalisées pour éviter les biais dus aux différences d'échelles.

Enfin, tout au long du traitement, des visualisations intermédiaires sont générées afin de contrôler l'impact de chaque transformation et de valider la qualité des données finales. Ce processus progressif permet d'assurer la fiabilité et la reproductibilité de l'analyse.

2 Données

Le projet repose sur deux fichiers CSV appelés `table1.csv` et `table2.csv`, contenant des données relatives à des clients d'une organisation mutualiste. Ces données couvrent des aspects démographiques, économiques et comportementaux des clients, notamment leur adhésion et leur départ éventuel de l'organisation.

Les deux fichiers présentent une structure similaire avec des colonnes partiellement communes. Afin d'homogénéiser le traitement des données, un renommage explicite des colonnes a été appliqué à l'aide de dictionnaires de correspondance.

Les principaux attributs des données sont :

2.1 Données communes aux deux tables

- `ID` : identifiant unique du client (type texte ou entier).
- `Sexe` (`CDSEXE`) : variable catégorielle (1 = Homme, 2 = Femme).
- `RevenuMontant` (`MTREV`) : montant du revenu mensuel estimé, en euros.
- `NombreEnfants` (`NBENF`) : nombre d'enfants à charge du client (entier positif).
- `SituationFamiliale` (`CDSITFAM`) : code de la situation familiale (catégorielle, valeurs entières représentant marié, célibataire, etc.).
- `DateAdhesion` (`DTADH`) : date d'adhésion à l'organisation.
- `StatutSocietaireCode` (`CDTMT`) : code représentant le statut du client au sein de l'organisation.
- `MotifDemissionCode` (`CDMOTDEM`) : code catégoriel précisant la raison de départ du client.
- `ClientTypeCode` (`CDCATCL`) : typologie du client (catégorielle, ex : particulier, entreprise, etc.).
- `DateDemission` (`DTDEM`) : date à laquelle le client a quitté l'organisation ou `31/12/1900` si toujours adhérent.

2.2 Spécificités de `table1.csv`

- `DemissionCode` (`CDDEM`) et `AnneeDemission` (`ANNEEDEM`) : permettent d'identifier les clients démissionnaires et l'année de leur départ.
- `AgeAdhesion` (`AGEAD`) : âge lors de l'adhésion (en années).
- `TrancheAgeAdhesion` (`RANGAGEAD`) : tranche d'âge lors de l'adhésion (ex : 18-25, 26-35...).
- `AgeDemission` (`AGEDEM`) : âge au moment de la démission.
- `TrancheAgeDemission` (`RANGAGEDEM`) : tranche d'âge à la démission.
- `DureeAdhesion` (`ADH`) : durée en années de l'adhésion.
- `TrancheDureeAdhesion` (`RANGADH`) : tranche de durée d'adhésion.

- `DateDemissionCode` (`RANGDEM`) : codage ordinal de la date de démission.

2.3 Spécificité de `table2.csv`

- `DateNaissance` (`DTNAIS`) : utilisée pour le calcul de l'âge réel du client.
- `BPADH` : colonne inconnue, supposée contenir un code interne ou une information peu documentée. Sa signification reste à éclaircir ou à exclure selon les analyses.

Ces colonnes représentent un mélange de :

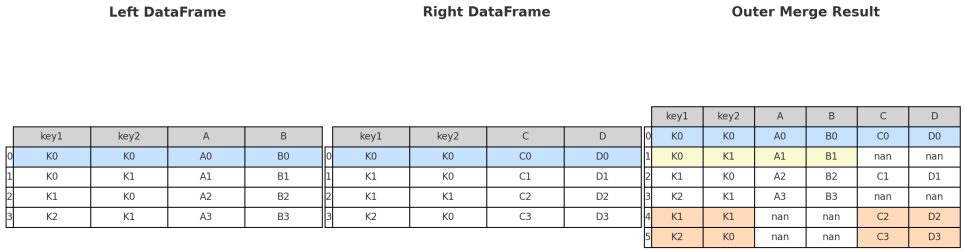
- **Variables numériques** : RevenuMontant, NombreEnfants, AgeAdhesion, DureeAdhesion.
- **Variables qualitatives** : Sexe, SituationFamiliale, ClientTypeCode.
- **Variables temporelles** : DateAdhesion, DateDemission, DateNaissance.
- **Variables ordinales** : Tranches d'âge et de durée.

Une exploration préliminaire des données a été réalisée via les fonctions `head()`, `describe()` et `info()` de la bibliothèque `pandas`. Cela a permis d'avoir un aperçu de la distribution des variables numériques, de détecter les types de données, ainsi que la présence éventuelle de valeurs manquantes ou incohérentes. La majorité des variables numériques sont exprimées en années (pour les âges, durées) ou en euros (pour les revenus). Les variables catégorielles sont représentées sous forme de codes, qui nécessitent un dictionnaire d'interprétation externe (non fourni dans les fichiers initiaux).

2.4 Nettoyage

Plusieurs types d'erreurs et d'incohérences ont été détectés et corrigés au cours de la phase de nettoyage :

- **Valeurs manquantes** : Certaines colonnes comme `DateDemission` contiennent des valeurs fictives (ex. `31/12/1900`) pour représenter une absence de démission. Ces valeurs ont été remplacées par `NaT` (Not a Time) pour faciliter le traitement temporel.
- **Colonnes redondantes ou inconnues** : La colonne `ColonneInconnue` (`BPADH`) dans `table2.csv` ne disposant pas de documentation claire, a été supprimée lors du nettoyage.
- **Types de données** : Les colonnes de type date (comme `DateAdhesion`, `DateDemission`, `DateNaissance`) ont été converties en format `datetime` via `pd.to_datetime()`.
- **Doublons** : La présence de doublons a été vérifiée via la méthode `duplicated()` et les entrées redondantes ont été supprimées.
- **Valeurs aberrantes** : Des valeurs aberrantes dans le revenu ou l'âge ont été détectées à l'aide de boîtes à moustaches (boxplots) et supprimées si elles dépassaient largement les bornes de l'écart interquartile (IQR).
- **Uniformisation des noms de colonnes** : Les colonnes des deux jeux de données ont été renommées de manière systématique à l'aide des dictionnaires `rename_dict_df1` et `rename_dict_df2`, afin d'harmoniser leur nomenclature et de simplifier le processus de fusion.
- **Fusion des jeux de données** : Une jointure a été réalisée sur les colonnes suivantes : *Sexe*, *SituationFamiliale*, *DateDemission*, *DateAdhesion*, *RevenuMontant*, *StatutSocietaireCode*, *MotifDemissionCode*, *Target*, *NombreEnfants*, et *ClientTypeCode*. Cette opération a permis de regrouper les informations provenant des deux fichiers en un seul jeu de données enrichi.



	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K1	C2	D2
3	K2	K0	C3	D3

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	nan	nan
2	K1	K0	A2	B2	C1	D1
3	K2	K1	A3	B3	nan	nan
4	K1	K1	nan	nan	C2	D2
5	K2	K0	nan	nan	C3	D3

Figure 1. Type de jointure utilisée

- **Recodage des variables catégorielles** : Les colonnes contenant des codes (ex : sexe, statut sociétaire, motif de démission) ont été converties en catégories via `astype("category")` pour optimiser l'espace mémoire et préparer l'analyse.

Ces différentes étapes de nettoyage ont permis d'obtenir un jeu de données cohérent, interprétable, et prêt pour les analyses statistiques et les visualisations ultérieures.

2.5 Analyse descriptive

L'analyse descriptive post-nettoyage a été menée sur le jeu de données fusionné, en se concentrant d'abord sur les attributs pris individuellement, puis sur les relations entre certaines paires de variables, notamment celles ayant un lien potentiel avec la démission des clients.

2.3.1 Analyse univariée Variables numériques :

- `RevenuMontant` présente une distribution asymétrique avec une majorité des clients ayant des revenus inférieurs à 3000€. Une poignée d'individus ont des revenus très élevés, ce qui introduit des valeurs extrêmes (identifiées et traitées à l'étape de nettoyage).
- `NombreEnfants` suit une distribution centrée autour de 1 à 2 enfants, avec très peu de clients ayant plus de 4 enfants.
- `AgeAdhesion` et `AgeDemission` présentent des distributions modales autour de 25-35 ans et 35-45 ans respectivement.
- `DureeAdhesion` est très variable, avec une majorité de clients ayant entre 2 et 8 années d'adhésion.

Variables catégorielles :

- `Sexe` est globalement équilibré entre hommes et femmes, avec une légère surreprésentation des femmes.
- `SituationFamiliale` montre que les clients célibataires et mariés représentent la grande majorité.
- `ClientTypeCode` est dominé par les particuliers ; les entreprises ou autres types sont minoritaires.
- `StatutSocietaireCode` et `MotifDemissionCode` présentent une grande diversité de codes, reflétant une segmentation complexe.

Variables temporelles :

- `DateAdhesion` est concentrée entre 2010 et 2018, avec une légère augmentation des adhésions sur cette période.
- `DateDemission` montre que les départs sont plus fréquents après 2015, possiblement en lien avec des changements structurels ou économiques.
- `DateNaissance`, convertie en âge réel, permet de corroborer les données d'âge issues de `table1.csv`.

Variables ordinales :

- Les tranches d'âges (`TrancheAgeAdhesion`, `TrancheAgeDemission`) montrent une forte concentration dans les classes "26-35" et "36-45".
- Les tranches de durée d'adhésion révèlent une adhésion moyenne de 5 à 10 ans.

2.3.2 Analyse bivariée Lien avec la démission (variable cible implicite) :

- Les tranches d'âge au moment de l'adhésion et de la démission influencent significativement le taux de départ. Les jeunes adultes (18–25 ans) présentent un taux de démission plus élevé que les autres tranches.
- Le revenu mensuel est un facteur discriminant : les clients à revenus faibles (inférieurs à 1500€) sont davantage susceptibles de quitter l'organisation.
- Le sexe ne montre pas de lien fort avec la probabilité de démission.
- Le statut sociétaire joue un rôle important : certains statuts sont beaucoup plus associés aux départs.
- La durée d'adhésion est inversement corrélée à la probabilité de démission récente : les clients très fidèles quittent moins.

Autres associations intéressantes :

- `RevenuMontant` est positivement corrélé à l'âge d'adhésion, ce qui est cohérent avec la montée en revenus au fil de la carrière.
- `NombreEnfants` varie faiblement selon le sexe, mais montre une légère corrélation avec la situation familiale.
- Les tranches d'âges sont corrélées avec la durée d'adhésion : les plus âgés ont tendance à rester plus longtemps.

Des visualisations complémentaires (diagrammes circulaires, histogrammes, boxplots, heatmaps de corrélation) ont été produites pour illustrer ces tendances. Par exemple, une heatmap des corrélations numériques a permis de confirmer les liens entre durée d'adhésion, âge et revenu.

3 Analyse des comportements de démission

Ce travail exploratoire s'est articulé autour de plusieurs axes d'analyse visant à mieux comprendre les facteurs influençant la démission des clients dans un contexte bancaire.

3.1 Facteurs influençant la démission

Une analyse croisée entre la variable `Target` (démission) et les caractéristiques socio-démographiques a été menée (âge, revenu, statut, etc.). Les principaux enseignements sont :

- Les clients jeunes (18–25 ans) et à revenu inférieur à 1500 euros démissionnent plus fréquemment.
- Le statut sociétaire « PS » est associé à un risque plus élevé.
- Le sexe n'a pas d'influence significative.

3.2 Caractéristiques liées à la durée d'adhésion

L'étude de la variable `DureeAdhesion` révèle :

- Les clients âgés de 36 à 55 ans restent plus longtemps.
- Un revenu plus élevé est corrélé à une plus grande fidélité.
- Les statuts stables (sociétaire ou employé) sont associés à des durées plus longues.

3.3 Évolution temporelle des adhésions et démissions

L'analyse des tendances annuelles montre :

- Une croissance des adhésions entre 2010 et 2018.
- Une hausse notable des démissions à partir de 2015, possiblement liée à des facteurs externes (concurrence, saturation du marché).

3.4 Identification de profils types

Une segmentation client a été réalisée à l'aide de groupements simples et d'ACP suivie de clustering. Trois profils dominants émergent :

- Jeunes célibataires à faible revenu→risque élevé de démission.
- Trentenaires mariés avec enfants→fidélité modérée.
- Seniors aisés→fidélité forte.

Recommandations

Sur la base des observations précédentes, plusieurs actions sont à envisager :

- Développer des offres spécifiques pour les jeunes à revenus modestes.
- Fidéliser les sociétaires à forte ancienneté avec des programmes d'engagement.
- Surveiller les périodes de démission et adapter la stratégie commerciale.
- Exploiter les segments identifiés pour personnaliser les campagnes marketing.

Cette exploration statistique permet non seulement d'enrichir la compréhension du jeu de données, mais aussi de guider les décisions lors du choix et de la pondération des critères dans les algorithmes multicritères.

3.5 Évaluation

Méthode expérimentale

Découpage des données. Les données ont été segmentées en deux ensembles distincts pour simuler un scénario réaliste de prédiction à partir de données historiques :

- **Jeu d'entraînement :** Sociétaires ayant démissionné **avant 2006** (soit jusqu'en 2005 inclus), combinés avec un échantillon de non-démissionnaires encore actifs à cette période.
- **Jeu de test :** Sociétaires ayant démissionné **en 2006**, et non-démissionnaires toujours présents **fin 2007**.

Équilibrage des classes. Pour chaque ensemble, nous avons appliqué un sous-échantillonnage des non-démissionnaires afin d'obtenir une proportion cible de :

- **70 % de démissionnaires** et **30 % de non-démissionnaires**.

Après le découpage, les proportions observées sont :

Table 1. Répartition des classes après découpage

Jeu	Taille	Démissionnaires (%)	Non-démissionnaires (%)
Entraînement	35 955	70.0 %	30.0 %
Test	7 115	70.0 %	30.0 %

Justification du découpage temporel. L'utilisation de données **postérieures à celles de l'entraînement** (ici, les démissions de 2006) pour constituer le jeu de test permet d'évaluer la **capacité de généralisation du modèle dans un contexte futur**. En effet, tester sur des données jamais vues par le modèle (non utilisées durant l'apprentissage) garantit une évaluation impartiale et reflète une situation réelle où le modèle serait utilisé pour prédire des démissions à venir. Cela permet également de détecter tout *surapprentissage* éventuel, c'est-à-dire un modèle trop spécifique aux données passées et incapable de s'adapter à de nouveaux cas.

Mesures de performance

Nous avons utilisé les métriques suivantes pour évaluer les performances des modèles :

- **Précision (Accuracy) :** La précision mesure le pourcentage de prédictions correctes parmi l'ensemble des prédictions. Elle est calculée comme suit :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

où :

  - TP : Vrais positifs (transactions avec remise correctement identifiées).
  - TN : Vrais négatifs (transactions sans remise correctement identifiées).
  - FP : Faux positifs (transactions sans remise mal classées comme ayant une remise).
  - FN : Faux négatifs (transactions avec remise mal classées comme n'en ayant pas).

- **F1-Score :** Le F1-Score combine la précision et le rappel, ce qui le rend particulièrement utile en cas de classes déséquilibrées. Il est défini comme :

$$\text{F1-Score} = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

avec :

$$\text{Précision} = \frac{TP}{TP + FP}, \quad \text{Rappel} = \frac{TP}{TP + FN}.$$

* Validation croisée

Afin d'assurer une évaluation robuste des modèles, nous avons utilisé la **validation croisée** (cross-validation) en 5 plis. Cette technique consiste à :

- Diviser l'ensemble d'entraînement en 5 sous-ensembles (*folds*) tout en conservant la proportion de classes.
- Entraîner le modèle sur 4 folds et le valider sur le 5^{e}, en répétant l'opération 5 fois en changeant à chaque fois le fold de validation.
- Moyennant les scores obtenus, on obtient une estimation plus fiable des performances attendues sur des données non vues.

Avantage : Cela permet de limiter les risques liés au découpage arbitraire des données et d'éviter une évaluation trop optimiste ou pessimiste liée à un seul split. Cette méthode est particulièrement pertinente dans notre cas où l'équilibre entre classes est important à respecter.

3.6 Implémentation et choix des paramètres

L'implémentation des modèles de classification a nécessité des choix de paramètres optimisés pour garantir des performances élevées. Voici une explication détaillée de ces choix :

Support Vector Machines (SVM) : L'objectif du SVM est de trouver une hyperplane qui sépare les classes (`Discount Availed : Oui/Non`) avec la marge la plus large possible. La fonction de décision pour un noyau linéaire est donnée par :

$$f(x) = \text{sign}(w^T x + b)$$

où :

- w est le vecteur de poids définissant l'orientation de l'hyperplane,
- b est le biais, définissant son décalage par rapport à l'origine,
- x est le vecteur d'entrée.

k-Nearest Neighbors (kNN) : Le kNN classe un point en fonction de la majorité des catégories parmi ses k plus proches voisins. La distance utilisée est la distance Euclidienne, définie comme :

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

où :

- x, y sont les vecteurs des caractéristiques des transactions,
- n est le nombre de caractéristiques.

Raison du choix de la plage [1, 21] : Cette plage couvre une transition entre des décisions très spécifiques (k faible) et des décisions généralisées (k élevé). Elle permet de tester la sensibilité du modèle à la granularité des données tout en restant dans des limites raisonnables pour des données de taille modérée. Un intervalle plus large aurait augmenté le temps de calcul sans apporter de gains significatifs pour ce projet.

Random Forest : Random Forest construit plusieurs arbres de décision à partir de sous-échantillons aléatoires des données. Chaque arbre effectue une prédiction, et le résultat final est déterminé par un vote majoritaire, rendant le modèle robuste face au surapprentissage :

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_n(x)\}$$

où $T_i(x)$ est la prédiction de l'arbre i pour un point x.

La valeur **n_estimators=100** a été choisie car elle garantit une précision élevée tout en maintenant un temps de calcul raisonnable. Ce nombre d'arbres permet au modèle d'être stable et performant. De plus, les gains en performance deviennent généralement négligeables au-delà de 100 arbres, rendant cette configuration largement utilisée.

Naive Bayes Le modèle **Naive Bayes** repose sur l'hypothèse "naïve" que toutes les variables explicatives $(X_1, X_2, \ldots, X_n)$ sont indépendantes conditionnellement à la classe cible Y.

Le théorème de Bayes est donné par :

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$

où :

Dans Naive Bayes, $P(X|Y)$ est décomposé en une multiplication de probabilités indépendantes pour chaque caractéristique :

$$P(X|Y) = \prod_{i=1}^{n} P(X_i|Y).$$

Type utilisé : Gaussian Naive Bayes Le modèle utilisé ici, **Gaussian Naive Bayes**, suppose que les caractéristiques X_i suivent une distribution normale (gaussienne). La vraisemblance $P(X_i|Y)$ est donnée par :

$$P(X_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(X_i - \mu_Y)^2}{2\sigma_Y^2}\right)$$

où :

- μ_Y : Moyenne de la caractéristique X_i pour la classe Y,
- σ_Y^2 : Variance de la caractéristique X_i pour la classe Y.

4 Résultats

Dans cette section, nous analysons les performances des différents algorithmes appliqués à nos données.

4.1 Performances individuelles

Table 2. Performances des modèles sur le jeu de validation

Modèle	Accuracy	Recall	Precision	F1-Score
SVM	0.950	0.925	0.955	0.938
k-NN	0.938	0.913	0.939	0.924
Naive Bayes	0.828	0.823	0.795	0.805
Random Forest	0.888	0.882	0.862	0.870

Support Vector Machines (SVM)

Le modèle SVM, avec des paramètres optimaux `C=10` et un noyau `rbf`, a présenté les meilleures performances globales. Il a atteint une précision de 0.950, un rappel de 0.925, une précision de 0.955 et un F1-Score de 0.938. Ces résultats témoignent de la capacité du SVM à modéliser efficacement les frontières de décision, même dans des cas où les classes ne sont pas strictement linéaires.

Interprétation : Les performances du SVM indiquent que les caractéristiques utilisées permettent une bonne séparation des classes, et que le choix d'un noyau non linéaire améliore la généralisation.

k-Nearest Neighbors (kNN)

Le modèle kNN, avec `k=7` et pondération par la distance, a également obtenu de bons résultats, avec un F1-Score de 0.924.

Interprétation : Le kNN a bien performé, bien que légèrement en retrait par rapport au SVM. Sa nature locale le rend plus sensible à la distribution des données dans l'espace des caractéristiques.

Naive Bayes

Le modèle Naive Bayes, sans réglages particuliers, a obtenu des résultats plus modestes, avec un F1-Score de 0.805.

Interprétation : L'hypothèse forte d'indépendance conditionnelle entre les variables limite la capacité du modèle à capturer des relations complexes.

Random Forest

Le modèle Random Forest, avec `max_depth=10` et `n_estimators=100`, a bien performé avec un F1-Score de 0.870.

Interprétation : Grâce à sa capacité d'agrégation, Random Forest est robuste et réduit les risques de surapprentissage. Il reste cependant légèrement moins performant que SVM et kNN dans ce cas.

4.2 Comparaison et généralisation

Les résultats présentés dans le tableau 2 montrent que le SVM surpasse les autres modèles en termes de performance globale, suivi de près par kNN. Random Forest reste une alternative robuste, tandis que Naive Bayes est plus limité pour des jeux de données aux dépendances complexes.

5 Conclusion générale

Ce travail pratique a permis de déployer une chaîne complète de traitement de données pour la prédiction des démissions de sociétaires. À partir de deux sources brutes, nous avons effectué un nettoyage rigoureux des données, leur fusion, ainsi qu'un ensemble de transformations (imputations, recodages, normalisations) afin de les rendre exploitables par des algorithmes d'apprentissage automatique.

L'un des aspects fondamentaux de ce projet a été la construction réfléchie des jeux de données d'entraînement et de test. Nous avons volontairement utilisé uniquement les données **antérieures à 2006** pour entraîner les modèles, et réservé les **données de 2006**, jamais vues pendant l'apprentissage, pour constituer le jeu de test. Cette séparation temporelle stricte permet de simuler un véritable cas d'usage prédictif, où l'on souhaite anticiper les départs futurs à partir de l'historique passé. Elle est essentielle pour garantir une **évaluation honnête** et **réaliste** de la performance des modèles, en évitant tout biais dû à la fuite d'information.

Parmi les modèles testés (SVM, k-NN, Naive Bayes, Random Forest), le SVM s'est illustré par ses performances supérieures, atteignant un F1-Score de 0.938, preuve de sa capacité à discriminer efficacement les démissionnaires des non-démissionnaires.

Ce TP a souligné l'importance de la qualité des données, de la rigueur méthodologique, et du choix des bonnes pratiques en machine learning. Les résultats obtenus ouvrent la voie à des optimisations futures : enrichissement des données, tuning plus fin des modèles ou encore expérimentation de techniques plus avancées telles que les modèles de gradient boosting ou les réseaux neuronaux.

Sujet Aide à la Décision (Voiture)

6 Présentation

Dans cette section, nous abordons différentes problématiques relatives au choix optimal d'un véhicule à partir de critères techniques et de préférences pondérées. Chaque sous-section détaille une question d'aide à la décision, les méthodes d'analyse employées ainsi que les principaux résultats obtenus.

6.1 Organisation

6.1.1 La composition du groupe

Notre groupe composé de deux étudiants, (Fatai IDRISSOU(IA) - Paul GANGBADJA(IA)), a travaillé sur plusieurs méthodes d'analyse multicritère pour comparer les alternatives de voitures. Les tâches ont été réparties pour que chaque membre se concentre sur une méthode spécifique, tout en travaillant ensemble pour intégrer les résultats de façon cohérente.

6.1.2 Répartition du travail

Notre groupe a divisé les tâches en fonction des compétences et préférences de chaque membre, tout en gardant une approche collaborative. Chaque membre a été responsable de l'implémentation d'une méthode d'analyse multicritère.

6.1.3 Organisation du travail dans le temps

Nous avons disposé de 9 heures pour réaliser ce projet, ce qui a nécessité une planification rigoureuse et une répartition précise des tâches. Le temps a été organisé en plusieurs étapes : une première phase de compréhension des méthodes, suivie de l'implémentation de chaque méthode par un membre du groupe, et enfin une phase de consolidation des résultats pour assurer une intégration cohérente des méthodes Promethee (avec et sans seuil de préférence) et Electre (IV et IS).

6.1.4 Distribution des taches entre les membres du groupe

Méthode Promethee I et II - Fatai

Fatai a commencé par étudier les principes de la méthode Promethee avec seuil de préférence pour bien comprendre son fonctionnement avant de débuter l'implémentation. Cette étape lui a permis de maîtriser les concepts nécessaires, comme le calcul des flux de préférence et leur interprétation. Une fois cette compréhension acquise, elle a programmé l'algorithme, en calculant les flux pour chaque alternative et ensuite intégrer la prise en compte du seuil. Enfin, elle a procédé à des vérifications manuelles pour s'assurer de l'exactitude des résultats.

Méthodes Electre IV et Electre IS - Paul

Paul a d'abord étudié les principes des méthodes Electre IV et Electre IS, en se concentrant sur les concepts clés comme les niveaux de concordance et de veto nécessaires pour évaluer les alternatives. Elle a ensuite programmé la méthode Electre IV, en intégrant les critères de concordance et de veto afin d'éliminer les alternatives moins performantes selon des seuils stricts. Parallèlement, elle a développé l'algorithme Electre IS en ajustant les niveaux de concordance et en appliquant des critères de préférence pour chaque alternative, Enfin, elle a comparer les résultats obtenus avec les méthodes Electre à ceux des méthodes Promethee.

7 Données

7.1 Caractéristiques

Les données utilisées dans ce projet proviennent d'un fichier CSV nommé `donnees.csv`. Ce fichier contient des informations sur les caractéristiques de différentes alternatives évaluées à l'aide des méthodes multicritères Promethee et Electre. Les colonnes et leur contenu sont décrits ci-dessous :

- **Nom des voitures** : Identifiant unique de chaque voiture(le nom d'u modèle).
- **Prix** : Valeur numérique exprimée en francs français, représentant le coût total de la voiture. Les valeurs varies entre 152900 et 191000
- **Vitesse maximale** : Valeur numérique exprimée en km/h, indiquant la vitesse maximale atteignable par la voiture. Les valeurs se situent généralement entre 182 km/h et 209 km/h.
- **Consommation** : Valeur numérique exprimée en litres/100 km, représentant la consommation moyenne de carburant.
- **Distance de freinage** : Valeur numérique exprimée en mètres, indiquant la distance nécessaire pour un arrêt complet. Les valeurs sont comprises entre 34 et 40 mètres.
- **Confort** : Score catégoriel représentant une évaluation qualitative du confort, où 1 est minimal et 8 est maximal.
- **Volume du coffre** : Valeur numérique exprimée en décimètres cubes, représentant la capacité de stockage du coffre. Les valeurs varient entre 300 et 600 décimètres cubes.

- **Accélération** : Valeur numérique exprimée en secondes, indiquant le temps nécessaire pour parcourir 1000 mètres départ arreté.

Indice	Nom de la voiture	Prix (€)	Vitesse	Conso	Freinage	Confort	Coffre	Accél.
Voiture 0	Alfa 156	23817	201	8.0	39.6	6	378	31.2
Voiture 1	Audi A4	25771	195	5.7	35.8	7	440	33.0
Voiture 2	Citroën Xantia	25496	195	7.9	37.0	2	480	34.0
Voiture 3	Peugeot 406	25649	191	8.3	34.4	2	430	34.6
Voiture 4	Saab TID	26183	199	7.8	35.7	5	494	32.0
Voiture 5	Renault Laguna	23664	194	7.7	37.4	4	452	33.8
Voiture 6	VW Passat	23344	195	7.6	34.4	3	475	33.6
Voiture 7	BMW 320d	26260	209	6.6	36.6	4	440	30.9
Voiture 8	Citroën Xsara	19084	182	6.4	40.6	8	408	33.5
Voiture 9	Renault Safrane	29160	203	7.5	34.5	1	520	32.0

Table 3. Matrice des performances des voitures (valeurs normalisées en euros et unités SI)

Les valeurs utilisées pour tester sont comme suit : **Seuil de concordance global** : **0.6**

Critère	Poids	Direction	Seuil de veto
Prix	0.2	Min	5000
Vitesse Max	0.2	Max	10
Consommation Moyenne	0.1	Min	1.5
Distance de Freinage	0.1	Min	3.0
Confort	0.2	Min	2
Volume du Coffre	0.1	Max	50
Accélération	0.1	Min	3.0

Table 4. Critères d'évaluation des voitures avec leur poids, direction et seuil de veto

7.2 Nettoyage

Les données étaient complètes et sans erreurs.

7.3 Relations entre les attributs

Nous avons remarqué que les voitures capables d'atteindre des vitesses élevées sont généralement plus chères, ce qui peut s'expliquer par la nécessité de moteurs performants et de matériaux de haute qualité. De plus, nous avons observé que les voitures avec un grand volume de coffre affichent souvent une accélération légèrement inférieure, reflétant une conception orientée vers le confort et la capacité de chargement.

8 Méthodes

8.1 Outils d'aide à la décision

Cette section présente les algorithmes utilisés ainsi que les paramètres sélectionnés pour leur mise en œuvre

8.1.1 Algorithmes appliqués

Méthode	Description	Justification du choix
Promethee sans seuil de préférence	Classement basé sur la comparaison directe des critères sans seuils, utilisant les flux globaux (ϕ^+ et ϕ^-).	Simple et efficace pour obtenir un classement initial des voitures.
Promethee avec seuil de préférence	Intègre des seuils pour moduler l'impact des petites différences et valoriser les grandes variations.	Prend en compte les marges d'erreur et affine l'analyse des préférences.
Electre IV	S'appuie sur des seuils stricts de concordance et de veto pour exclure les voitures non conformes.	Idéal pour éliminer les voitures qui ne respectent pas les seuils critiques.
Electre IS	Intègre des préférences partielles pour une évaluation plus flexible et nuancée.	Permet d'évaluer des alternatives partiellement satisfaisantes, offrant plus de souplesse.

Table 5. Description des méthodes d'analyse multicritère

8.1.2 Choix des paramètres

Dans ce projet, nous avons choisi de diviser les utilisateurs en quatre classes pour refléter les différents profils de besoins : professionnels actifs, étudiants, pères de famille, et retraités. Ces classes couvrent les cas les plus fréquents, allant des jeunes au budget limité (étudiants) aux personnes cherchant confort et sécurité (retraités).

Choix des poids des critères pour chaque classe :

Les poids des critères ont été utilisés pour moduler l'importance relative de chaque critère. Ils varient selon les priorités spécifiques de chaque classe d'utilisateurs.

Dans la méthode Promethee, les poids déterminent la contribution relative de chaque critère dans les flux de préférence (ϕ^+ et ϕ^-).

Dans la méthode Electre IS, les poids influencent les matrices de concordance et de discordance, qui sont utilisées pour évaluer la dominance entre alternatives.

Critère	Professionnels actifs	Étudiants	Pères de famille	Retraités
Prix (€)	0.20	0.30	0.20	0.25
Vitesse maximale (km/h)	0.20	0.10	0.10	0.10
Consommation (L/100km)	0.10	0.25	0.20	0.20
Distance freinage (m)	0.10	0.10	0.10	0.10
Confort (note 1–8)	0.2	0.10	0.20	0.15
Volume coffre (L)	0.10	0.05	0.15	0.10
Accélération (sec/1000m)	0.10	0.10	0.05	0.10

Table 6. Poids des critères par classe normalisées

Justification du choix des poids

- **Professionnels actifs** : équilibre prix/perf (vitesse, confort), mobilité importante.
- **Étudiants** : priorité au prix et à la consommation, faible poids sur le confort ou coffre.
- **Pères de famille** : Valorisent le confort, la capacité (volume du coffre) et la consommation pour des raisons pratiques et familiales.
- **Retraités** : Privilégient la consommation et le confort, tout en conservant une certaine attention à la sécurité.

Types de critères

Les types de critères (minimisation ou maximisation) définissent comment chaque critère est évalué :

Critère	Professionnels actifs	Étudiants	Pères de famille	Retraités
Prix	Min	Min	Min	Min
Vitesse maximale	Max	Max	Min	Min
Consommation	Min	Min	Min	Min
Distance freinage	Min	Min	Min	Min
Confort	Min	Max	Max	Max
Volume coffre	Max	Min	Max	Max
Accélération	Min	Min	Min	Min

Table 7. Types de critères par classe

Tous les utilisateurs cherchent à minimiser le prix, tandis que le confort et le volume du coffre sont des critères à maximiser, ces derniers étant particulièrement importants pour les pères de famille et les retraités. En revanche, la vitesse maximale est valorisée par les professionnels actifs et les étudiants, mais elle est minimisée pour les pères de famille et les retraités, qui accordent une priorité plus élevée à la sécurité.

Seuils de préférence

Les seuils de préférence utilisés dans Promethee avec seuils et Electre IS suivent les mêmes logiques que celles des poids, reflétant les priorités spécifiques de chaque classe d'utilisateurs. Ces seuils permettent d'ajuster les comparaisons en tenant compte des marges acceptables définies par les besoins et les exigences de chaque groupe.

Critère	Professionnels actifs	Étudiants	Pères de famille	Retraités
Prix (€)	2000	1500	1800	1800
Vitesse maximale (km/h)	6	6	5	5
Consommation (L/100km)	0.4	0.4	0.6	0.5
Distance freinage (m)	1	1.0	1.5	1.0
Confort (note 1–8)	1	1	1	1
Volume coffre (L)	30	20	40	35
Accélération (sec/1000m)	0.8	0.8	1.2	1.0

Table 8. Seuils de préférence (valeurs brutes) par classe

Seuils de veto

Les seuils de veto suivent la même logique que celle des poids, en reflétant les priorités et exigences spécifiques de chaque classe d'utilisateurs. Ils permettent d'exclure les alternatives qui ne respectent pas des valeurs critiques définies en fonction des besoins

Critère	Professionnels actifs	Étudiants	Pères de famille	Retraités
Prix (€)	5000	3000	4000	3500
Vitesse maximale (km/h)	10	10	8	8
Consommation (L/100km)	1.5	1.2	1.5	1.0
Distance freinage (m)	3.0	2.5	3.0	2.5
Confort (note 1–8)	2	2	2	2
Volume coffre (L)	50	30	60	55
Accélération (sec/1000m)	3	2.0	3.0	2.5

Table 9. Seuils de veto (valeurs brutes) par classe

Seuil de concordance

Le seuil de concordance, fixé à **0.6** suivant ce qui est écrit dans le sujet, est un paramètre clé commun à toutes les classes d'utilisateurs dans les méthodes Electre IV et Electre IS. Ce seuil représente le pourcentage minimum de critères pour lesquels une alternative doit être considérée comme satisfaisante, par rapport à une autre, afin d'être jugée dominante.

8.2 Evaluation

8.2.1 Classement relatif des véhicules

Pour les méthodes Promethee

Dans le cadre de notre projet, nous avons appliqué les méthodes Promethee pour classer les véhicules en fonction des flux nets ($\phi_{\text{net}} = \phi^+ - \phi^-$), qui mesurent le degré global de domination ou de faiblesse d'un véhicule par rapport aux autres. Pour obtenir ces flux, nous avons suivi les étapes suivantes :

$$\phi^+ = \textbf{Somme des valeurs de chaque ligne de la matrice de préférence},$$

$$\phi^- = \textbf{Somme des valeurs de chaque colonne de la matrice de préférence}.$$

Le classement final des véhicules est établi en triant les flux nets (ϕ_{net}) dans l'ordre décroissant. Les véhicules ayant les flux nets les plus élevés sont considérés comme les meilleures options.

Pour les méthodes Electre IV et Electre IS

Pour les méthodes Electre IV et Electre IS, nous avons construit un graphe de surclassement basé sur les matrices de concordance et de discordance. Ces matrices sont calculées pour évaluer les relations entre les véhicules, en suivant ces étapes :

- **Calcul de la matrice de concordance ($C(i,j)$)** : Nous avons mesuré le degré d'accord pour chaque paire de véhicules en additionnant les poids des critères satisfaisant les conditions minimales. Une relation de surclassement est établie lorsque $C(i,j)$ dépasse un seuil défini.
$$C(i,j) \geq \textbf{seuil de concordance}$$
- **Calcul de la matrice de non-discordance ($D(i,j)$)** : Nous avons identifié les écarts critiques entre véhicules pour vérifier qu'ils ne dépassent pas les seuils de veto. Une relation de surclassement est établie si $D(i,j) = 1$.
- **Construction du graphe de surclassement** : En utilisant les matrices de concordance et de non-discordance, nous avons ajouté des arcs entre les véhicules satisfaisant les deux conditions suivantes :
$$C(i,j) \geq \textbf{seuil de concordance} \quad \textbf{et} \quad D(i,j) = 1.$$
- **Extraction du noyau du graphe** : À partir du graphe, nous avons déterminé le noyau en identifiant les véhicules non dominés. Ces véhicules sont considérés comme les meilleures options, car ils ne sont surclassés par aucun autre.

8.3 Implémentation

8.3.1 Etapes détaillées de chaque méthode

Dans cette partie on détaille les étapes de chaque méthode utilisée :

Promethee sans seuils de préférence

Nous avons suivi les étapes suivantes pour appliquer cette méthode :

1. **Normalisation des données** : Nous avons ramené toutes les données à une échelle entre 0 et 1.
2. **Calcul de la matrice de préférence** :
Pour chaque paire de voitures, leurs performances ont été comparées pour chaque critère en tenant compte de son type (`min` pour minimiser ou `max` pour maximiser). La préférence a été calculée selon la règle suivante :
$$P(i,j) = \begin{cases} w, & \text{si } d > 0, \\ 0, & \text{si } d \leq 0, \end{cases}$$
où :

- d est la différence entre les performances des deux voitures ($d = a - b$ si le critère est à maximiser, ou $d = b - a$ si le critère est à minimiser),
- w est le poids du critère.

3. **Calcul des flux de préférence (ϕ^+ et ϕ^-)** : À partir de la matrice de préférence, nous avons calculé les flux sortants (ϕ^+) et entrants (ϕ^-) pour chaque voiture.
4. **Classement final** : Le flux net (ϕ_{net}) a été calculé pour chaque voiture, et celles-ci ont été classées en fonction de leurs flux nets.

Promethee avec seuils de préférence

Voici les étapes suivies :

1. **Définition des seuils de préférence** :Le seuil de préférence a permis de marquer clairement les différences significatives entre les performances.
2. **Calcul de la matrice de préférence avec intégration des seuils** : Pour chaque paire de voitures, nous avons évalué leurs performances en tenant compte des seuils de préférence définis. La préférence a été calculée selon la règle suivante :

$$P(i,j) = \begin{cases} 0, & \text{si la différence } d \leq 0, \\ w, & \text{si } d \geq \text{seuil de préférence}, \\ \frac{d}{\text{seuil}} \cdot w, & \text{si } 0 < d < \text{seuil de préférence}. \end{cases}$$

où :

- d est la différence entre les performances des deux voitures ($d = a - b$ si le critère est à maximiser, ou $d = b - a$ si le critère est à minimiser),
- w est le poids du critère,
- seuil de préférence est la valeur au-delà de laquelle une différence est pleinement valorisée.

3. **Recalcul des flux et classement final** : Les flux sortants (ϕ^+), entrants (ϕ^-), et nets (ϕ_{net}) ont été recalculés en intégrant les seuils. Ces flux ont ensuite été utilisés pour établir un classement final des voitures.

Electre IV et Electre IS

Les méthodes Electre IV et Electre IS partagent certaines étapes, mais elles diffèrent principalement dans la manière de calculer la matrice de concordance et d'intégrer les préférences.

Étapes communes :

- **Matrice de concordance** : Les deux méthodes calculent une matrice qui mesure le degré d'accord entre les alternatives. Cette matrice est utilisée pour identifier si une alternative surclasse une autre.
- **Matrice de discordance** : Les deux méthodes incluent une matrice de discordance pour vérifier si des écarts critiques dépassent les seuils de veto.
- **Construction du graphe de surclassement** : En utilisant les matrices de concordance et de discordance, un graphe dirigé est construit pour représenter les relations de dominance entre les alternatives.
- **Extraction du noyau** : Le noyau du graphe est déterminé en identifiant les alternatives non dominées par les autres. Ces alternatives constituent les meilleures options selon la méthode.

Différences :

Matrice de concordance dans Electre IV : [1] Dans Electre IV, la concordance est stricte. Une pondération complète est attribuée uniquement si les conditions minimales sont respectées. La formule est donnée par :

$$C(i,j) = \begin{cases} w, & \text{si } a \geq b \text{ (critère à maximiser) ou } a \leq b \text{ (critère à minimiser),} \\ 0, & \text{sinon.} \end{cases}$$

Matrice de concordance dans Electre IS : Electre IS introduit des préférences partielles, ce qui rend la méthode plus flexible. Une pondération proportionnelle est attribuée si la différence entre a et b est inférieure au seuil de préférence. La formule devient :

$$C(i,j) = \begin{cases} w, & \text{si } a \geq b \text{ (critère à maximiser) ou } a \leq b \text{ (critère à minimiser),} \\ w \cdot \left(1 - \frac{|a-b|}{\text{seuil préf.}}\right), & \text{si } |a-b| < \text{seuil préf.,} \\ 0, & \text{si } |a-b| \geq \text{seuil préf..} \end{cases}$$

9 Résultats

9.1 Graphes (Professionnels actifs

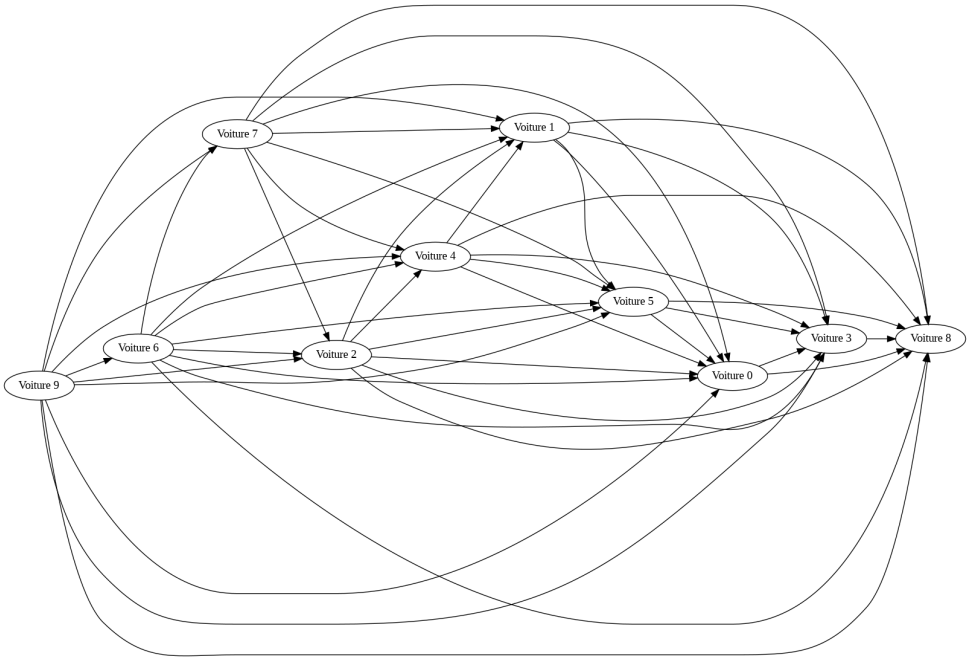


Figure 2. Graphe de surclassement Promethee I entre les voitures

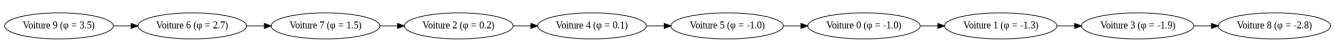


Figure 3. Graphe de surclassement Promethee II entre les voitures

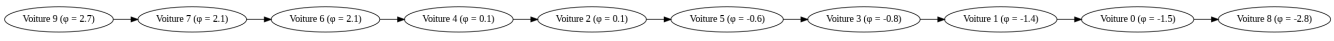


Figure 4. Graphe de surclassement Promethee II entre les voitures avec seuil de preference

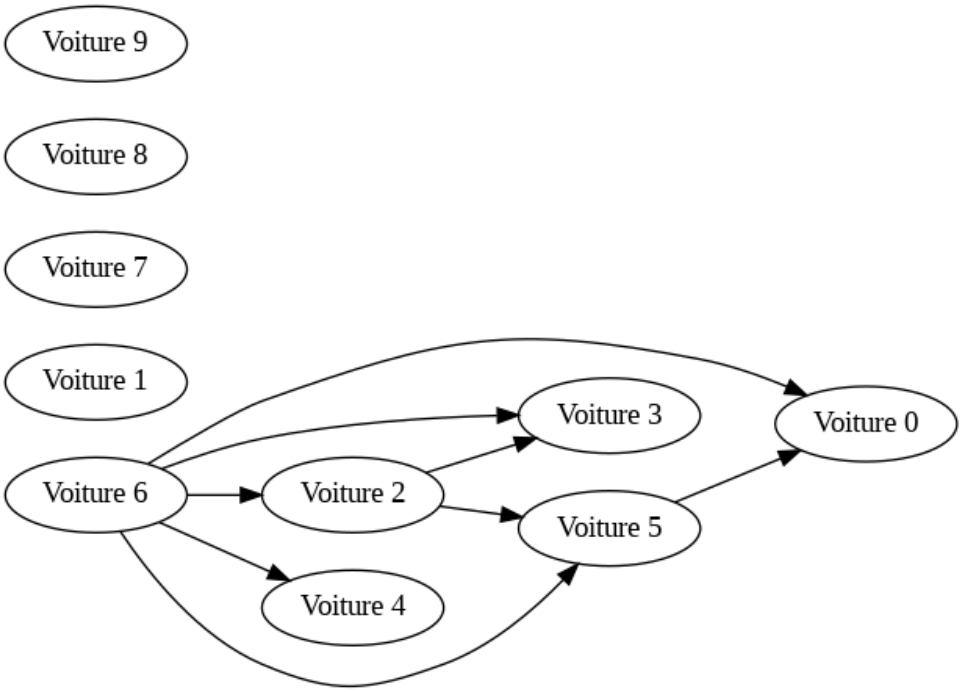


Figure 5. Graphe de surclassement Electre IV entre les voitures

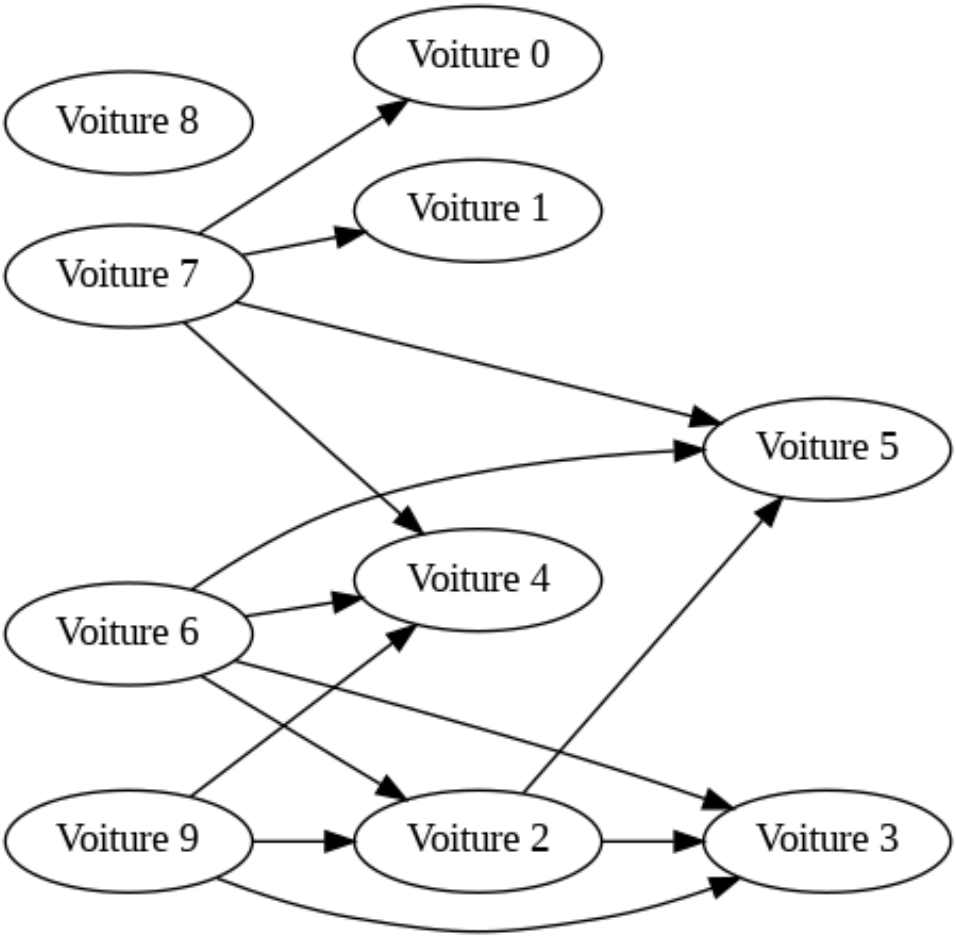


Figure 6. Graphe de surclassement Electre Is entre les voitures

9.2 Interpretation

PROMETHEE I

Le graphe de surclassement issu de PROMETHEE I permet de visualiser les préférences partielles entre les voitures. Il révèle les relations de domination directe sans imposer un ordre total. Certaines alternatives peuvent être incomparables (absence de relation directe entre deux nœuds).

Exemples d'arcs visibles : Voiture 7 (BMW_320d) surclasse Voiture 1 (Audi_A4), Voiture 5 (Rnlt_Laguna) et Voiture 4 (Saab_TID), mais reste incomparée à Voiture 9 (Rnlt_Safrane). Cela met en lumière la complexité des préférences dans un contexte multicritère.

Interprétation : PROMETHEE I est utile pour visualiser les dominances claires et les conflits potentiels, mais il ne permet pas de tirer un classement global.

PROMETHEE II sans seuil

Cette méthode fournit un **ordre complet** des alternatives, basé sur les flux nets ϕ. Chaque voiture est comparée à toutes les autres, ce qui permet de construire une hiérarchie.

Ordre obtenu (du meilleur au moins bon) : Voiture 9 (Rnlt_Safrane) > Voiture 7 (BMW_320d) > Voiture 6 (VW_Passat) > Voiture 4 (Saab_TID) > Voiture 2 (Cit_Xantia) > Voiture 5 (Rnlt_Laguna) > Voiture 3 (Peugeot_406) > Voiture 1 (Audi_A4) > Voiture 0 (Alfa_156) > Voiture 8 (Cit_Xsara)

Interprétation : Cette méthode classe Voiture 9 (Rnlt_Safrane) comme la meilleure, malgré sa faible présence dans les noyaux d'Electre, ce qui montre qu'elle est bien perçue sur l'ensemble des critères pondérés.

PROMETHEE II avec seuils de préférence

Grâce à l'introduction de seuils, cette méthode affine la prise en compte des préférences partielles. Elle conserve un **ordre total**, mais atténue l'impact des petites différences non significatives.

Ordre obtenu (du meilleur au moins bon) : Voiture 9 (Rnlt_Safrane) > Voiture 6 (VW_Passat) > Voiture 7 (BMW_320d) > Voiture 2 (Cit_Xantia) > Voiture 4 (Saab_TID) > Voiture 5 (Rnlt_Laguna) > Voiture 0 (Alfa_156) > Voiture 1 (Audi_A4) > Voiture 3 (Peugeot_406) > Voiture 8 (Cit_Xsara)

Noyau observé (arcs sortants uniquement, non dominés) : [Voiture 9 (Rnlt_Safrane)]

Interprétation : Avec les seuils, le classement devient plus robuste : seules les différences jugées significatives influencent l'ordre. Voiture 9 (Rnlt_Safrane) reste en tête, mais les écarts sont atténués et Voiture 6 (VW_Passat) et Voiture 7 (BMW_320d) montrent une bonne stabilité. Audi_A4 descend dans le classement, car ses différences favorables sont jugées marginales selon les seuils fixés.

9.3 Classements comparés des méthodes

Méthode	Ordre du meilleur au moins bon
PROMETHEE I	Aucun ordre total. Relations de surclassement partiel observées. Certaines alternatives sont incomparables entre elles.
PROMETHEE II (sans seuil)	Voiture 9 (Rnlt_Safrane) > Voiture 7 (BMW_320d) > Voiture 6 (VW_Passat) > Voiture 4 (Saab_TID) > Voiture 2 (Cit_Xantia) > Voiture 5 (Rnlt_Laguna) > Voiture 3 (Peugeot_406) > Voiture 1 (Audi_A4) > Voiture 0 (Alfa_156) > Voiture 8 (Cit_Xsara)
PROMETHEE II (avec seuils)	Voiture 9 (Rnlt_Safrane) > Voiture 6 (VW_Passat) > Voiture 7 (BMW_320d) > Voiture 2 (Cit_Xantia) > Voiture 4 (Saab_TID) > Voiture 5 (Rnlt_Laguna) > Voiture 0 (Alfa_156) > Voiture 1 (Audi_A4) > Voiture 3 (Peugeot_406) > Voiture 8 (Cit_Xsara)
ELECTRE IV	Pas de classement global. Voitures les plus présentes dans les noyaux : Voiture 1 (Audi_A4), Voiture 6 (VW_Passat), Voiture 7 (BMW_320d), Voiture 8 (Cit_Xsara), Voiture 9 (Rnlt_Safrane)
ELECTRE IS	Pas de classement global. Voitures les plus présentes dans les noyaux : Voiture 6 (VW_Passat), Voiture 7 (BMW_320d), Voiture 8 (Cit_Xsara), Voiture 9 (Rnlt_Safrane)

Table 10. Classement des alternatives selon les différentes méthodes

10 Conclusion

10.1 Critique du projet

Points positifs :

- L'approche méthodique pour appliquer et comparer différentes méthodes a renforcé notre compréhension des outils d'aide à la décision.
- L'utilisation de bibliothèques comme Pandas, NumPy, Matplotlib et Graphiz a simplifié les calculs et rendu les résultats visuellement accessibles.
- Le projet a permis de collaborer efficacement et de répartir les tâches en fonction des compétences individuelles.

Points négatifs :

- Les seuils de préférence et de veto n'ont pas toujours été calibrés avec précision pour refléter parfaitement les besoins des classes d'utilisateurs.
- Certaines méthodes, notamment Electre IV, nécessitent des paramètres stricts qui peuvent exclure des alternatives potentiellement intéressantes.

10.2 Propositions pour améliorer le projet

Pour remédier aux limitations identifiées, nous suggérons les actions suivantes :

- **Affiner les paramètres :** Réaliser une analyse plus fine des seuils de préférence et de veto, soit par simulation, soit par consultation d'experts du domaine.
- **Élargir les critères :** Ajouter des dimensions comme l'impact environnemental, les coûts d'entretien ou les notes des utilisateurs pour une analyse plus complète.
- **Tester sur des cas réels :** Appliquer les méthodes sur des données réelles ou élargir l'échantillon pour valider la généralisation des résultats.

10.3 Perspectives

Ce projet pourrait être étendu pour inclure :

- L'intégration d'algorithmes d'apprentissage automatique pour affiner les paramètres automatiquement.
- Une extension à d'autres types de décisions multicritères.
- La mise en place d'une plateforme web interactive où les utilisateurs peuvent définir leurs propres critères et priorités pour obtenir des recommandations personnalisées.

Références

[1] Daniel Gourien. *Cours d'aide à la décision*. Université d'Avignon, disponible sur Moodle. Ce cours fournit les bases théoriques sur les méthodes Promethee et Electre, appliquées dans ce projet. 2024.