

Rapport du Projet Application BI

Les étudiants :

Bilel GHOUDELBOURK
Mustapha GENOUIZ
Ayoub DJAGHLOUL

06/04/2025

**MASTER D'INFORMATIQUE
INTELLIGENCE ARTIFICIELLE (IA)**
ECUE AMS Application BI

Responsable
Pr. Remy KESSLER

Sommaire

Titre	1
Sommaire	2
I Introduction Générale	4
II Fouille de données	4
1 Introduction	4
2 Présentation des Données	4
2.1 Les fichiers sources	4
2.2 Enjeux et Particularités	4
3 Préparation et Nettoyage des Données	5
3.1 Préparation de la Table 1	5
Extraction automatique des tranches et labels	5
Mapping dynamique de RANGDEM	5
Filtrage spécifique	5
3.2 Préparation de la Table 2	5
Nettoyage et transformation des dates	5
Calcul des attributs temporels	5
Création des tranches et catégories	5
Filtrage et suppression des anomalies	5
Suppression de la colonne BPADH et harmonisation	5
Création d'un indicateur binaire	6
3.3 Fusion et Nettoyage Final	6
4 Choix des Features	6
5 Modélisation et Raffinement	6
5.1 Séparation des Données	6
5.2 Construction des Pipelines	7
5.3 Modèles et Raffinements	7
6 Résultats sur le Jeu de Validation	7
6.1 Visualisation des Matrices de Confusion	7
6.1.1 SVM (C=1.0)	7
6.1.2 Naïve Bayes	8
6.1.3 Random Forest	8
6.1.4 k-NN	9
7 Évaluation Finale sur le Jeu de Test	9
8 Analyse de l'Importance des Variables (Random Forest)	10
9 Conclusion	10

III Aide à la décision multicritère	11
1 Introduction	11
2 Rappel du jeu de données	11
2.1 Alternatives (véhicules)	11
2.2 Critères	11
2.3 Poids et paramètres de préférences	12
3 Méthodes PROMETHEE I et II	13
3.1 Rappels sur PROMETHEE	13
3.2 Fonction de préférence	13
3.3 Calculs et scénarios	13
3.3.1 Résultats pour le scénario A (PROMETHEE sans p_k)	14
3.3.2 Résultats pour le scénario B (PROMETHEE avec p_k)	14
3.4 Comparaison des deux classements	15
4 Méthode ELECTRE IV	15
4.1 Rappels sur ELECTRE IV	15
4.2 Résultats numériques	15
4.2.1 Relation de surclassement finale	16
5 Méthode ELECTRE IS	17
5.1 Rappels sur ELECTRE IS	17
5.2 Résultats pour nos données	17
6 Analyse comparative	18
7 Conclusion	19
IV Conclusion Générale	19

Première partie

Introduction Générale

Ce rapport présente deux projets réalisés dans le cadre du cours AMS Application BI. Le premier projet consiste à prédire la démission des clients d'un organisme bancaire. Pour ce faire, nous avons exploité, nettoyé et harmonisé deux jeux de données pour développer un modèle prédictif performant. Le second projet est une étude d'aide à la décision multicritère appliquée aux véhicules automobiles, utilisant des méthodes telles que PROMETHEE et ELECTRE pour guider le choix du véhicule le plus adapté.

Deuxième partie

Fouille de données

1 Introduction

Ce rapport porte sur la prédiction de la démission des clients d'un organisme bancaire. L'objectif principal est de développer un modèle prédictif permettant d'identifier, à partir de données clients, ceux susceptibles de quitter l'organisme. Ce travail s'inscrit dans le cadre de l'UE Business Intelligence – Systèmes décisionnels et repose sur l'exploitation de deux jeux de données distincts : l'un regroupant les clients ayant démissionné (Table 1) et l'autre, un échantillon comprenant également des clients actifs (Table 2).

Le présent document détaille l'ensemble du processus, depuis l'exploration et le nettoyage des données jusqu'à la modélisation, le raffinement et l'évaluation finale sur le jeu de test. Une attention particulière est portée à la préparation et à l'harmonisation des données, étape cruciale pour garantir la qualité du modèle final.

2 Présentation des Données

2.1 Les fichiers sources

- **Table 1 (table1.csv)** : Regroupe les données des clients démissionnaires (période 1999-2006). Les variables incluent l'identifiant, le code du sexe, le montant des revenus, le nombre d'enfants, la situation familiale, la date d'adhésion, des codes relatifs au statut, des indicateurs de démission (date, année, motif) ainsi que des variables dérivées (âge à l'adhésion, âge à la démission et tranches d'âge ou d'ancienneté).
- **Table 2 (table22.csv)** : Contient un échantillon comprenant des clients mixtes (actifs et démissionnaires), incluant la date de naissance (**DTNAIS**). La colonne **BPADH**, présente dans ce tableau, comporte majoritairement des valeurs vides et a été supprimée afin d'homogénéiser sa structure avec la Table 1.

2.2 Enjeux et Particularités

Les deux tableaux présentent des structures et contenus différents. Leur fusion nécessite une harmonisation des types de données, la création de nouvelles variables (âge, durée d'adhésion, tranches) et la suppression des anomalies et colonnes inutiles.

3 Préparation et Nettoyage des Données

Dans cette section, nous décrivons en détail les étapes de préparation appliquées séparément aux deux tableaux avant leur fusion.

3.1 Préparation de la Table 1

Extraction automatique des tranches et labels Une fonction personnalisée a été développée pour extraire dynamiquement les intervalles (*bins*) et les étiquettes (*labels*) à partir des valeurs textuelles présentes dans les colonnes de tranches (**RANGAGEAD**, **RANGAGEDEM**, **RANGADH**). Cette méthode, basée sur des expressions régulières, permet d'identifier les bornes et de générer les étiquettes correspondantes. Par exemple, la tranche "0 0-18" a été ajoutée pour regrouper les clients mineurs.

Mapping dynamique de RANGDEM Un mapping dynamique a été généré pour associer les années de démission à un rang. Ce mapping intègre :

- L'extraction des valeurs existantes dans la Table 1.
- L'ajout automatique des années présentes dans la Table 2 qui ne figurent pas dans la Table 1.
- L'attribution de la valeur "0 Actif" pour les clients n'ayant pas démissionné.

Filtrage spécifique Les lignes de la Table 1 où la durée d'adhésion (**ADH**) est égale à 0 ont été supprimées. En effet, une durée d'adhésion de moins d'une année suivie d'une démission ne permet pas de prédire la cause de la démission et introduit un biais potentiel dans le modèle.

3.2 Préparation de la Table 2

Nettoyage et transformation des dates Les colonnes **DTNAIS**, **DTADH** et **DTDEM** ont été converties au format `datetime` pour faciliter le calcul des âges et des durées d'adhésion.

Calcul des attributs temporels

- **AGEAD** : Âge à l'adhésion, calculé comme la différence entre l'année d'adhésion et l'année de naissance.
- **AGEDEM** : Âge à la démission, calculé comme la différence entre l'année de démission et l'année de naissance. Pour les clients actifs, une valeur spécifique a été utilisée.
- **ADH** : Durée d'adhésion, calculée selon que le client est actif (utilisation d'une année de référence, 2007) ou démissionnaire.

Création des tranches et catégories Les *bins* et *labels* extraits de la Table 1 ont été appliqués pour créer les variables :

- **RANGAGEAD** (tranche d'âge à l'adhésion)
- **RANGAGEDEM** (tranche d'âge à la démission)
- **RANGADH** (tranche de durée d'adhésion), incluant une catégorie spécifique "0 0-0+" pour les clients récemment inscrits.

Filtrage et suppression des anomalies Plusieurs opérations de filtrage ont été appliquées :

- Suppression des lignes où **AGEAD** est manquant (dates de naissance erronées).
- Suppression des lignes où **RANGAGEDEM** est vide pour les clients inactifs (ligne erronée).
- Suppression des lignes où **RANGADH** vaut "0 0-0+" pour les clients inactifs (Démission sur une durée très courte ne permettant pas de dégager de tendances).

Suppression de la colonne BPADH et harmonisation La colonne **BPADH** a été supprimée car la majorité de ses valeurs est vide et elle n'existe pas dans la Table 1. Cette opération permet d'homogénéiser la structure du deuxième tableau avec celle du premier, facilitant ainsi leur fusion ultérieure.

Création d'un indicateur binaire Un nouvel attribut, `estClientActif`, a été créé pour distinguer les clients actifs (valeur 1) des démissionnaires (valeur 0).

3.3 Fusion et Nettoyage Final

- Les identifiants originaux ont été supprimés pour éviter les doublons, et un nouvel identifiant unique a été généré pour le jeu de données fusionné.
- Les deux tableaux ont été concaténés pour constituer un jeu de données complet.
- Dans le jeu final, les valeurs manquantes dans les colonnes `CDDEM`, `CDMOTDEM` et `RANGAGEDEM` ont été remplacées par la valeur "Actif".
- La conversion définitive des colonnes de dates en format `datetime` et la transformation explicite des variables catégorielles ont été réalisées.

Le fichier final nettoyé est sauvegardé sous le nom `df_final_cleaned.csv`.

4 Choix des Features

Afin de construire un modèle prédictif robuste, nous avons procédé à une sélection rigoureuse des variables explicatives. Nous avons choisi d'utiliser uniquement les features disponibles au moment de la prédiction, évitant ainsi toute fuite d'information (data leakage) liée aux variables qui contiennent des informations postérieures (comme les indicateurs de démission réelle).

Les variables sélectionnées pour l'entraînement des modèles sont les suivantes :

- **CDSEX** : Permet de capturer d'éventuelles disparités comportementales entre les différents sexes.
- **MTREV** : Montant des revenus, indicateur de la stabilité financière.
- **NBENF** : Nombre d'enfants, reflétant la situation familiale.
- **CDSITFAM** : Situation familiale du client.
- **CDTMT** : Statut du client (catégorisation liée au type de contrat).
- **CDCATCL** : Type de client, une information utile pour segmenter la clientèle.
- **AGEAD** : Âge à l'adhésion, un indicateur clé de la phase de vie du client lors de l'ouverture de son compte.
- **RANGAGEAD** : La tranche d'âge à l'adhésion, dérivée de **AGEAD**, permet de regrouper les clients par cohorte d'âge, facilitant ainsi l'analyse des tendances comportementales spécifiques à chaque groupe d'âge.
- **ADH** : Durée d'adhésion, qui reflète la stabilité de la relation client.
- **RANGADH** : Tranche de durée d'adhésion, indiquant différentes catégories de fidélité.

Les variables relatives à la démission effective (telles que `CDDEM`, `CDMOTDEM`, `AGEDEM`, `RANGAGEDEM` et `RANGDEM`) n'ont pas été utilisées comme features car elles contiennent des informations postérieures à l'adhésion et ne seraient pas disponibles au moment de la prédiction. Ce choix garantit la validité du modèle prédictif en évitant toute fuite d'information.

5 Modélisation et Raffinement

5.1 Séparation des Données

Le jeu de données final a été divisé de manière stratifiée en :

- 70% pour l'apprentissage (Train)
- 15% pour la validation (Validation)
- 15% pour le test final (Test)

5.2 Construction des Pipelines

Un pipeline de prétraitement a été construit pour standardiser les variables numériques et encoder les variables catégorielles. Ce pipeline a été intégré aux modèles de classification.

5.3 Modèles et Raffinements

Quatre modèles ont été entraînés et raffinés sur le jeu de validation :

- **SVM** : Test de différentes valeurs de C (0.1, 1.0, 10). Le modèle avec $C = 1.0$ a été retenu.
- **k-NN** : Test de différentes valeurs de k (5 et 10).
- **Naïve Bayes**
- **Random Forest**

6 Résultats sur le Jeu de Validation

Les performances des modèles ont été évaluées à l'aide de rapports de classification et de matrices de confusion. Le tableau ci-dessous récapitule les résultats :

Modèle	Accuracy	Recall (Macro)	F1-score (Macro)
SVM ($C=1.0$)	0.94	0.93	0.93
SVM ($C=0.1$)	0.93	0.91	0.92
SVM ($C=10$)	0.93	0.91	0.92
k-NN ($k=5$)	0.93	0.91	0.92
k-NN ($k=10$)	0.93	0.90	0.92
Naïve Bayes	0.91	0.90	0.90
Random Forest	0.93	0.92	0.93

6.1 Visualisation des Matrices de Confusion

6.1.1 SVM ($C=1.0$)

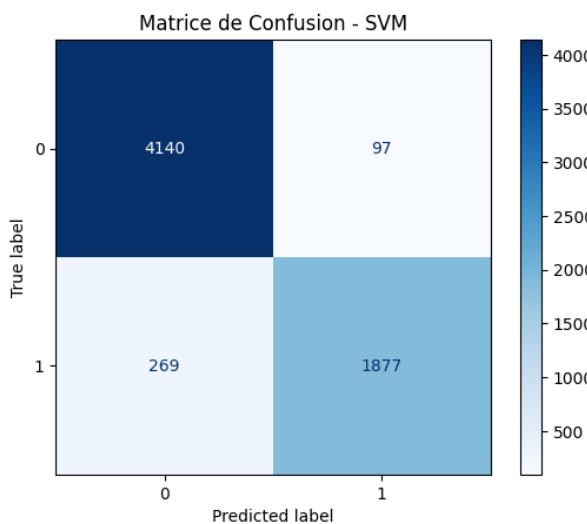


Figure 1. Matrice de confusion du modèle SVM sur Validation.

Commentaire : La diagonale montre que la majorité des prédictions sont correctes, avec seulement 97 erreurs sur la classe 0 et 269 sur la classe 1.

6.1.2 Naive Bayes

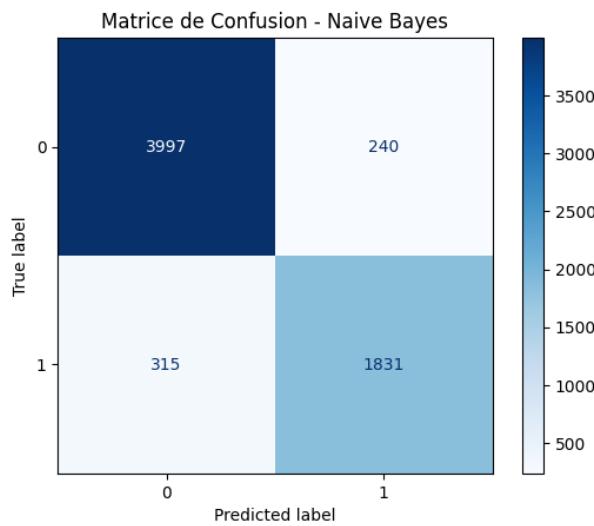


Figure 2. Matrice de confusion du modèle Naive Bayes.

Commentaire : 240 erreurs sur la classe 0, 315 sur la classe 1. Les performances sont légèrement inférieures à celles de SVM et k-NN.

6.1.3 Random Forest

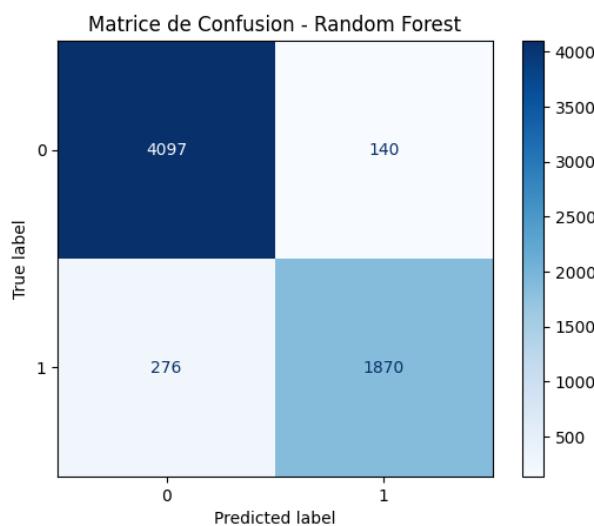


Figure 3. Matrice de confusion du modèle Random Forest.

Commentaire : 140 erreurs sur la classe 0, 276 sur la classe 1. Les résultats sont proches de ceux de SVM.

6.1.4 k-NN

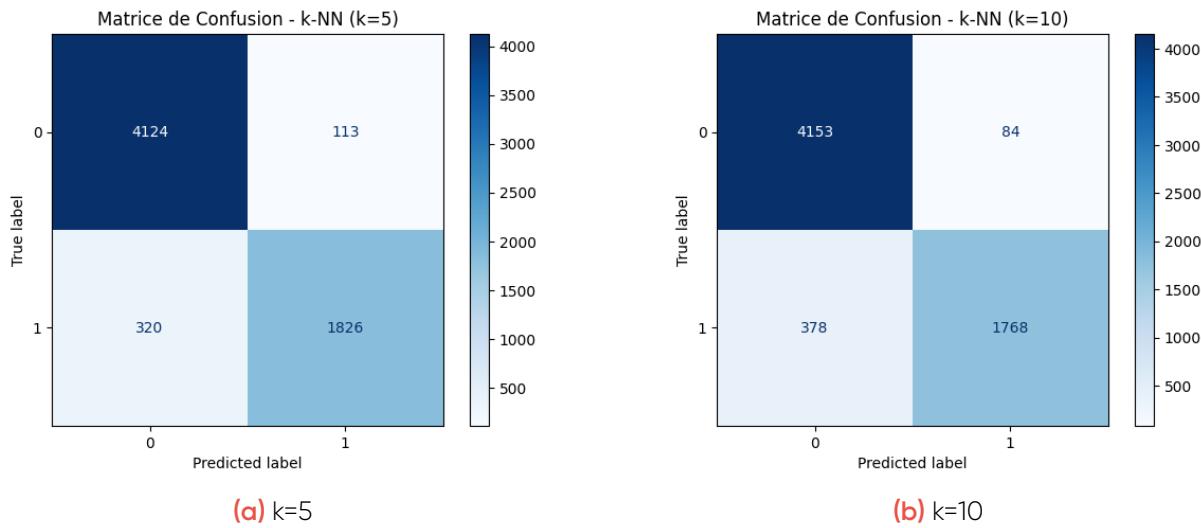


Figure 4. Matrices de confusion du modèle k-NN pour k=5 (gauche) et k=10 (droite).

Commentaire : Pour $k=5$, on note 113 erreurs sur la classe 0 et 320 sur la classe 1. Pour $k=10$, on observe 84 erreurs sur la classe 0 et 378 sur la classe 1. Les performances globales restent proches.

7 Évaluation Finale sur le Jeu de Test

Le modèle SVM (avec $C = 1.0$), qui a présenté les meilleures performances sur le jeu de validation, a été évalué sur le jeu de test. Les résultats obtenus sont les suivants :

Précision : 0.94 pour les clients actifs, 0.95 pour les démissionnaires.

Recall : 0.98 pour les actifs, 0.87 pour les démissionnaires.

F1-score : 0.96 pour les actifs, 0.91 pour les démissionnaires.

Accuracy globale : 0.94.

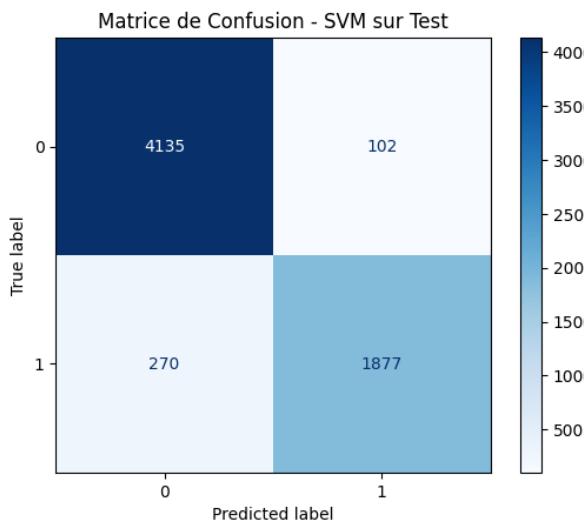


Figure 5. Matrice de confusion du modèle SVM sur le jeu de Test.

Figure 1: Commentaire : On observe une très bonne précision sur la classe 0 (4135 correctement prédits sur 4237) et un rappel légèrement plus faible sur la classe 1 (1877 sur 2147). Les performances globales restent excellentes.

8 Analyse de l'Importance des Variables (Random Forest)

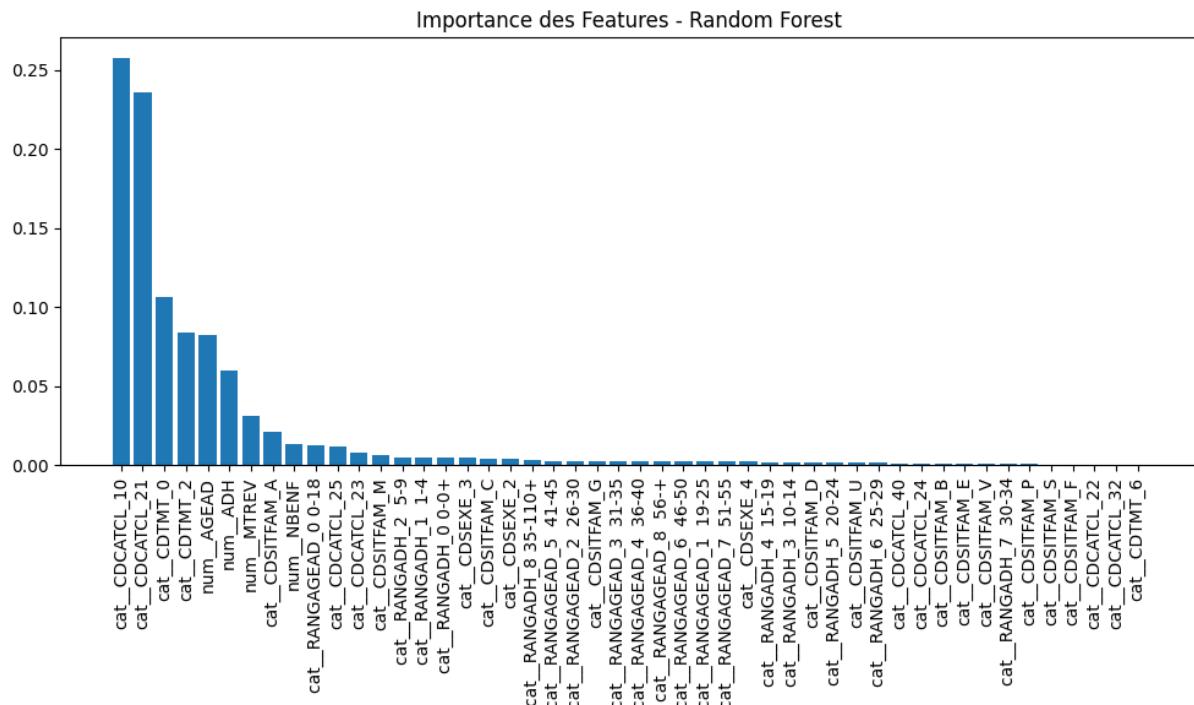


Figure 6. Importance des variables obtenue par Random Forest.

Figure 2 : Commentaire : Les variables *CDCATCL* et *CDTMT* apparaissent comme les plus discriminantes, suivies de *MTREV* et *AGEAD*. Les variables catégorielles relatives au type de client ou au statut sociétaire influencent donc fortement la séparation des classes.

9 Conclusion

Le projet a permis de développer un pipeline complet allant de l'exploration et du nettoyage des données à la modélisation prédictive. La préparation des données a été particulièrement soignée, avec :

- Extraction automatique des tranches (*bins* et *labels*) pour les variables d'âge et d'ancienneté.
 - Mapping dynamique de la variable **RANGDEM**.
 - Suppression de la colonne **BPADH** dans la Table 2 pour homogénéiser sa structure avec celle de la Table 1.
 - Suppression des lignes de la Table 1 avec une durée d'adhésion nulle (**ADH** = 0), car une durée d'adhésion inférieure à une année suivie d'une démission ne permet pas de prédire la cause et induit un biais potentiel.
 - Fusion des deux tableaux après harmonisation et création d'un identifiant unique.

Les modèles ont été raffinés en testant divers hyperparamètres. Le modèle SVM (avec $C = 1.0$) a été retenu comme le meilleur, affichant une accuracy de 94% et des scores élevés en recall et F1-score sur le jeu de test. Les matrices de confusion (Section précédente) et le graphique d'importance des variables (Section 8) viennent appuyer l'analyse des résultats.

Des pistes d'amélioration future pourraient inclure une optimisation plus fine des hyperparamètres et l'exploration d'autres techniques d'encodage pour améliorer encore la performance du modèle.

Troisième partie

Aide à la décision multicritère

1 Introduction

Dans ce chapitre, nous présentons une étude d'aide à la décision multicritère portant sur la comparaison de véhicules automobiles *turbo diesel* européens. L'objectif est d'illustrer l'usage de plusieurs méthodes d'aide à la décision, notamment **PROMETHEE** (dans ses variantes I et II) ainsi que **ELECTRE** (notamment ELECTRE IV et ELECTRE IS).

L'étude repose sur un jeu de données fictif, mais représentatif, comprenant dix véhicules et sept critères. Les critères incluent : le *prix*, la *vitesse maximale*, la *consommation moyenne*, la *distance de freinage*, le *confort*, le *volume de coffre* et l'*accélération sur 1000 mètres*.

Dans un premier temps, nous décrivons les données et la manière dont elles sont codées. Ensuite, nous revenons sur les méthodes PROMETHEE I et II, tout en comparant deux scénarios : l'un où nous n'utilisons pas de seuils de préférence ($p_k = 0$), l'autre où nous appliquons des seuils spécifiques ($p_k > 0$). Nous présentons et discutons enfin les résultats, avant de proposer un regard complémentaire grâce aux méthodes ELECTRE (IV et IS).

2 Rappel du jeu de données

2.1 Alternatives (véhicules)

L'exemple considère 10 véhicules turbo diesel (versions européennes), tirés et légèrement modifiés à des fins pédagogiques. Voici la liste des alternatives (avec leur code) :

1. Alfa Romeo 156 2.4 JTD (Alfa_156)
2. Audi A4 1.9 TDI (Audi_A4)
3. Citroën Xantia 2.1 TD (Cit_Xantia)
4. Peugeot 406 2.1 TD SV (Peugeot_406)
5. Saab 9.3 TID (Saab_TID)
6. Renault Laguna 2.2 dT RXT (Rnlt_Laguna)
7. Volkswagen Passat 1.9 TDI Sport (VW_Passat)
8. BMW 320d (BMW_320d)
9. Citroën Xsara 2.0 HDi (Cit_Xsara)
10. Renault Safrane 2.2 dT RXE (Rnlt_Safrane)

2.2 Critères

Sept critères sont retenus pour la comparaison :

- **Prix** (g_1), exprimé en euro. À minimiser.
- **Vitesse maximale** (g_2), en km/h. À maximiser.
- **Consommation moyenne** (g_3), en litres pour 100 km. À minimiser.
- **Distance de freinage** (g_4), en mètres pour un freinage à 90 km/h. À minimiser.
- **Confort** (g_5), critère qualitatif, rang issu d'un classement d'expert (1 = meilleur, 8 = moins bon). À minimiser.
- **Volume de coffre** (g_6), en dm³. À maximiser.
- **Accélération** (g_7), temps (s) pour parcourir 1000 m départ arrêté. À minimiser.

Afin de pouvoir utiliser les méthodes PROMETHEE et ELECTRE, nous avons converti systématiquement les critères à *maximiser*. Autrement dit, pour chaque critère à minimiser, nous avons multiplié la colonne par -1. Ainsi, du point de vue des calculs, tout est ramené à une

maximisation. Bien sûr, nous gardons à l'esprit que Prix et Conso_moy, par exemple, sont en réalité à minimiser.

2.3 Poids et paramètres de préférences

Les méthodes multicritères exigent le plus souvent des *pondérations* (poids) et parfois des *seuils* (p_k) ou *veto* selon la méthode. Dans notre exemple, nous prenons la grille de pondérations ci-dessous (leur somme est égale à 1,0) :

Critère	Poids
Prix	0,20
Vitesse_Max	0,15
Conso_moy	0,15
Dis_Freinage	0,15
Confort	0,15
Vol_Coffre	0,10
Acceleration	0,10

Pour certains scénarios, nous considérerons des seuils $\{p_k\}$:

- $p(\text{Prix}) = 1600$,
- $p(\text{Vitesse_Max}) = 8$,
- $p(\text{Conso_moy}) = 1$,
- $p(\text{Dis_Freinage}) = 1$,
- $p(\text{Confort}) = 1$,
- $p(\text{Vol_Coffre}) = 30$,
- $p(\text{Acceleration}) = 1$.

En complément des pondérations et des seuils p_k , certaines méthodes multicritères comme **ELECTRE** nécessitent la fixation d'un *seuil de concordance global* (s) ainsi que de *seuils de veto* pour certains critères.

Le **seuil de concordance** représente la proportion minimale du poids total des critères nécessaires pour qu'une alternative soit considérée comme surclassant une autre. Dans notre exemple, nous fixons :

$$s = 0,75$$

Ce qui signifie qu'au moins 75 % de l'importance cumulée des critères doivent être en faveur d'une alternative pour qu'un surclassement soit établi (en l'absence de veto).

Le **veto** permet d'empêcher un surclassement global lorsqu'un critère présente une différence trop défavorable, jugée inacceptable. Voici les seuils de veto utilisés pour chaque critère :

Critère	Seuil de veto
Prix	2500
Vitesse_Max	10
Conso_moy	1
Dis_Freinage	2
Confort	1
Vol_Coffre	40
Acceleration	1

Ces seuils sont choisis pour représenter environ 25 à 33 % de l'étendue typique des valeurs de chaque critère. Ils modélisent une règle stricte : une alternative ne peut pas être considérée comme globalement meilleure si elle est fortement moins performante sur un critère déterminant, même si elle est meilleure sur d'autres.

Ce choix est arbitraire et vise à modéliser une zone de préférence graduelle (pour PROMETHEE) ou un seuil de veto (pour ELECTRE).

3 Méthodes PROMETHEE I et II

La famille de méthodes **PROMETHEE** (**P**reference **R**anking **O**rganization **M**ethod for **E**nrichment **E**valuations) propose un classement *partiel* ou *complet* des alternatives.

3.1 Rappels sur PROMETHEE

- **PROMETHEE I** fournit pour chaque alternative a deux flux de surclassement :
 - $\phi^+(a)$ (flux sortant) : mesure à quel point a surclasse globalement les autres alternatives ;
 - $\phi^-(a)$ (flux entrant) : mesure à quel point a est surclassé par les autres alternatives. Cela aboutit souvent à un *classement partiel*, car il peut y avoir des incomparables.
- **PROMETHEE II** agrège ces flux en un flux *net* :

$$\phi(a) = \phi^+(a) - \phi^-(a),$$

et permet ainsi de produire un *classement total*.

3.2 Fonction de préférence

Pour calculer un degré de préférence de a sur b (au niveau d'un critère k), PROMETHEE utilise une *fonction de préférence* P_k appliquée à la différence $\Delta_k = f_k(a) - f_k(b)$. Lorsque Δ_k est positive (mieux vaut a), alors la préférence est plus ou moins grande.

- **Fonction usuelle** ($p_k = 0$) :

$$P_k(\Delta_k) = \begin{cases} 1 & \Delta_k > 0, \\ 0 & \Delta_k \leq 0. \end{cases}$$

- **Fonction linéaire** (avec un seuil $p_k > 0$) :

$$P_k(\Delta_k) = \begin{cases} 0 & \Delta_k \leq 0, \\ 1 & \Delta_k \geq p_k, \\ \frac{\Delta_k}{p_k} & 0 < \Delta_k < p_k. \end{cases}$$

Dans ce cas, pour un $\Delta_k > 0$ inférieur à p_k , la préférence n'est pas maximale, mais croît linéairement.

3.3 Calculs et scénarios

Nous avons mis en place deux scénarios :

1. **Scénario A** : sans seuils de préférence, c'est-à-dire $p_k = 0$ pour tous les critères. L'implémentation retient la fonction usuelle.
2. **Scénario B** : avec des seuils de préférence p_k (cf. tableau ci-dessus). On applique alors une fonction linéaire pour chaque critère.

Les deux scénarios génèrent des *flux sortants* ϕ^+ , *flux entrants* ϕ^- et un *flux net* ϕ . Les résultats sont reportés ci-après.

3.3.1 Résultats pour le scénario A (PROMETHEE sans p_k)

Le tableau 1 montre à la fois la matrice de préférences $\pi(a,b)$ pour chaque couple d'alternatives et, en fin de ligne, les flux entrants $\phi^+(a)$. En bas de colonne, on trouve les flux sortants $\phi^-(a)$. La dernière ligne indique le flux net $\phi(a) = \phi^+(a) - \phi^-(a)$. Le Tableau 2 illustre le $\phi(a)$ net et le classement (PROMETHEE II).

	Alfa_156	Audi_A4	Cit_Xantia	Peugeot_406	Saab_TID	Rnlt_Laguna	VW_Passat	BMW_320d	Cit_Xsara	Rnlt_Safrane	ϕ^+
Alfa_156	0.00	0.60	0.45	0.60	0.45	0.25	0.25	0.20	0.55	0.30	3.65
Audi_A4	0.40	0.00	0.40	0.50	0.35	0.55	0.25	0.50	0.80	0.35	4.10
Cit_Xantia	0.55	0.45	0.00	0.70	0.35	0.55	0.25	0.45	0.55	0.20	4.05
Peugeot_406	0.40	0.50	0.15	0.00	0.50	0.30	0.15	0.50	0.55	0.35	3.40
Saab_TID	0.55	0.65	0.65	0.50	0.00	0.50	0.35	0.45	0.65	0.20	4.50
Rnlt_Laguna	0.75	0.45	0.45	0.70	0.50	0.00	0.00	0.30	0.55	0.20	3.90
VW_Passat	0.75	0.60	0.60	0.70	0.65	1.00	0.00	0.60	0.55	0.35	5.80
BMW_320d	0.80	0.40	0.55	0.50	0.55	0.55	0.40	0.00	0.65	0.60	5.00
Cit_Xsara	0.45	0.20	0.45	0.45	0.35	0.45	0.45	0.35	0.00	0.35	3.50
Rnlt_Safrane	0.70	0.65	0.80	0.65	0.70	0.80	0.65	0.40	0.65	0.00	6.00
ϕ^-	5.35	4.50	4.50	5.30	4.40	4.95	2.75	3.75	5.50	2.90	
ϕ_{net}	-1.70	-0.40	-0.45	-1.90	0.10	-1.05	3.05	1.25	-2.00	3.10	

Table 1. Matrice de préférences $\pi(a,b)$ (PROMETHEE I), scénario A, avec flux ϕ^+ , ϕ^- et ϕ_{net} .

Rang	Alternative	ϕ_{net}
1	Rnlt_Safrane	3.100
2	VW_Passat	3.050
3	BMW_320d	1.250
4	Saab_TID	0.100
5	Audi_A4	-0.400
6	Cit_Xantia	-0.450
7	Rnlt_Laguna	-1.050
8	Alfa_156	-1.700
9	Peugeot_406	-1.900
10	Cit_Xsara	-2.000

Table 2. Flux nets ϕ_{net} et classement PROMETHEE II, scénario A.

Les deux meilleures voitures selon le flux net sont la Renault Safrane et la VW Passat, nettement détachées par rapport aux autres. En bas de classement, la Citroën Xsara, $\phi(a) = -2.0$, se retrouve dominée par la plupart des modèles.

3.3.2 Résultats pour le scénario B (PROMETHEE avec p_k)

Lorsqu'on introduit des seuils ($p_k > 0$), la fonction de préférence devient linéaire : les Δ_k trop petits ne génèrent pas directement une préférence maximale. Ceci a pour effet de « tempérer » l'inclinaison de ϕ^+ et ϕ^- .

Au Tableau 3 sont indiqués les flux $\phi^+(a)$, $\phi^-(a)$ ET le flux net ϕ . au Tableau 4 le flux net ϕ et le classement (PROMETHEE II).

	Alfa_156	Audi_A4	Cit_Xantia	Peugeot_406	Saab_TID	Rnlt_Laguna	VW_Passat	BMW_320d	Cit_Xsara	Rnlt_Safrane	ϕ^+
Alfa_156	0.0000	0.5625	0.4125	0.4950	0.3175	0.2313	0.2125	0.2000	0.5500	0.2800	3.2613
Audi_A4	0.4000	0.0000	0.4000	0.3583	0.2015	0.3988	0.2100	0.3161	0.7050	0.3500	3.3397
Cit_Xantia	0.4150	0.2844	0.0000	0.3141	0.2359	0.3221	0.1667	0.3455	0.5500	0.2000	2.8336
Peugeot_406	0.4000	0.3153	0.1500	0.0000	0.3668	0.3000	0.1500	0.3764	0.5233	0.2150	2.7967
Saab_TID	0.4300	0.4400	0.3867	0.4250	0.0000	0.4438	0.2383	0.2446	0.6500	0.2000	3.4584
Rnlt_Laguna	0.4641	0.3900	0.2500	0.4996	0.3650	0.0000	0.0000	0.2400	0.5500	0.2000	2.9587
VW_Passat	0.5191	0.6000	0.4350	0.5800	0.5300	0.4704	0.0000	0.6000	0.5500	0.2150	4.4995
BMW_320d	0.7300	0.4000	0.4600	0.4333	0.5500	0.5200	0.4000	0.0000	0.6500	0.5475	4.6908
Cit_Xsara	0.4500	0.2000	0.4000	0.4500	0.3500	0.3800	0.3600	0.2300	0.0000	0.3500	3.1700
Rnlt_Safrane	0.5125	0.6500	0.7100	0.6200	0.5067	0.6800	0.5150	0.4000	0.6500	0.0000	5.2442
ϕ^-	4.321	3.842	3.604	4.175	3.423	3.746	2.252	2.953	5.378	2.558	
ϕ_{net}	-1.0597	-0.5023	-0.7705	-1.3787	0.0354	-0.7875	2.2475	1.7379	-2.2083	2.6867	

Table 3. Matrice de préférences $\pi(a,b)$ (PROMETHEE I), scénario B, avec flux ϕ^+ , ϕ^- et ϕ_{net} .

Rang	Alternative	ϕ_{net}
1	Rnlt_Safrane	2.687
2	VW_Passat	2.247
3	BMW_320d	1.738
4	Saab_TID	0.035
5	Audi_A4	-0.502
6	Cit_Xantia	-0.771
7	Rnlt_Laguna	-0.788
8	Alfa_156	-1.059
9	Peugeot_406	-1.379
10	Cit_Xsara	-2.208

Table 4. Flux nets ϕ_{net} et classement PROMETHEE II, scénario B.

On retrouve dans l'ensemble le même ordre de préférence qu'au scénario A, bien que certains flux positifs et négatifs soient moins extrêmes. La *Renault Safrane* et la *VW Passat* restent les deux premières, mais l'écart vis-à-vis de la *BMW 320d* est un peu plus marqué qu'au scénario sans seuils.

3.4 Comparaison des deux classements

Le rang de la plupart des véhicules ne change pas entre scénario A et B, ou très peu (éventuels inversions entre 7e et 8e place). On constate que l'introduction de p_k rend la méthode moins sensible aux petites différences de performances. Les flux $\phi^+(a)$ sont en moyenne plus faibles, car il faut dépasser le seuil avant de « gagner » la préférence. Les véhicules très performants sur un ou deux critères, mais de façon « marginale », pourront ne plus bénéficier d'une préférence maximale sur ce critère.

Ainsi, PROMETHEE II s'avère relativement cohérent d'un scénario à l'autre, mais les valeurs de flux changent. Les différences d'ordres (inversions) peuvent être plus importantes dans d'autres contextes (poids distincts, seuils plus grands, etc.).

4 Méthode ELECTRE IV

4.1 Rappels sur ELECTRE IV

La méthode **ELECTRE IV** cherche à déterminer les *relations de surclassement* dans le cas où tous les critères sont *qualitatifs ou imprécis*, ou plus généralement, lorsque l'on ne souhaite pas utiliser de poids au sens fort. Néanmoins, dans notre implémentation illustrative, nous avons utilisé des *poids* et un *seuil de concordance global* $s = 0.75$.

Le principe consiste à :

1. Calculer la *mesure de concordance globale* pour chaque couple (a,b) . Si la somme des poids des critères où a surpasse ou égale b dépasse un seuil s , alors a surclasse (au sens de concordance) b .
2. Vérifier qu'il n'y a pas de *veto* (*discordance*) : si sur au moins un critère, la différence en faveur de b est trop grande, cela annule le surclassement éventuel de a sur b .

4.2 Résultats numériques

Avec nos poids (somme = 1,0) et un seuil global $s = 0.75$, la *matrice de concordance* (M_1) et la *matrice de non-discordance* (M_2) (binaire) sont illustrées ci-dessous. La Table 5 présente la matrice M_1 . Nous y plaçons la valeur de concordance (entre 0 et 1). La Table 6 montre le résultat binaire de non-discordance (0 ou 1).

	Alfa	Audi	Xantia	406	Saab	Laguna	Passat	BMW	Xsara	Safrane
Alfa_156	0.00	0.60	0.45	0.60	0.45	0.25	0.25	0.20	0.55	0.30
Audi_A4	0.40	0.00	0.55	0.50	0.35	0.55	0.40	0.60	0.80	0.35
Xantia	0.55	0.60	0.00	0.85	0.35	0.55	0.40	0.45	0.55	0.20
406	0.40	0.50	0.30	0.00	0.50	0.30	0.30	0.50	0.55	0.35
Saab_TID	0.55	0.65	0.65	0.50	0.00	0.50	0.35	0.45	0.65	0.30
Laguna	0.75	0.45	0.45	0.70	0.50	0.00	0.00	0.45	0.55	0.20
Passat	0.75	0.75	0.75	0.85	0.65	1.00	0.00	0.60	0.55	0.35
BMW_320d	0.80	0.50	0.55	0.50	0.55	0.70	0.40	0.00	0.65	0.60
Xsara	0.45	0.20	0.45	0.45	0.35	0.45	0.45	0.35	0.00	0.35
Safrane	0.70	0.65	0.80	0.65	0.80	0.80	0.65	0.40	0.65	0.00

Table 5. Matrice M_1 de concordance globale (seuil = 0.75). Les valeurs supérieures ou égales à 0.75 sont en gras.

	Alfa	Audi	Xantia	406	Saab	Laguna	Passat	BMW	Xsara	Safrane
Alfa_156	0	0	0	0	1	0	0	0	0	0
Audi_A4	0	0	0	0	0	0	0	0	0	0
Xantia	0	0	0	0	0	1	0	0	0	0
406	0	0	0	0	0	1	0	0	0	0
Saab_TID	0	0	0	0	0	0	0	0	0	0
Laguna	0	0	0	0	0	0	0	0	0	0
Passat	0	0	0	0	0	1	0	0	0	0
BMW_320d	1	1	0	0	0	0	0	0	0	0
Xsara	0	0	0	0	0	0	0	0	0	0
Safrane	0	0	0	0	0	0	0	0	0	0

Table 6. Matrice M_2 de non-discordance (0/1).

On constate que certains couples (a,b) n'ont aucun veto (1 dans M_2) alors que la majorité sont à 0. Dans la mesure où le volume de veto sur certains critères reste modéré, peu de surclassements sont complètement annulés.

4.2.1 Relation de surclassement finale

Nous considérons qu'il y a surclassement $a \mathcal{S} b$ si :

$$(M_1[a,b] \geq s) \text{ ET } (M_2[a,b] = 1).$$

Avec $s = 0.75$, la matrice finale (binaire) est donnée au Tableau 7.

	Alfa	Audi	Xantia	406	Saab	Laguna	Passat	BMW	Xsara	Safrane
Alfa_156	0	0	0	0	0	0	0	0	0	0
Audi_A4	0	0	0	0	0	0	0	0	0	0
Cit_Xantia	0	0	0	0	0	0	0	0	0	0
Peugeot_406	0	0	0	0	0	0	0	0	0	0
Saab_TID	0	0	0	0	0	0	0	0	0	0
Rnlt_Laguna	0	0	0	0	0	0	0	0	0	0
VW_Passat	0	0	0	0	0	1	0	0	0	0
BMW_320d	1	0	0	0	0	0	0	0	0	0
Cit_Xsara	0	0	0	0	0	0	0	0	0	0
Rnlt_Safrane	0	0	0	0	0	0	0	0	0	0

Table 7. Matrice finale de surclassement binaire ($a \mathcal{S} b$) pour ELECTRE IV.

À la lecture de cette table, on voit que BMW_320d surclasse Alfa_156 (car la valeur vaut 1), et VW_Passat surclasse Rnlt_Laguna. Tous les autres couples sont à 0, ce qui signifie qu'aucune autre relation de surclassement ne répond à la condition concordance ≥ 0.75 et non-discordance = 1.

Pour proposer un score net, on peut soustraire le nombre de surclassements subis. On obtient (Table 8) :

Alternative	Score net
VW_Passat	+1
BMW_320d	+1
Audi_A4	0
Saab_TID	0
Cit_Xantia	0
Peugeot_406	0
Cit_Xsara	0
Rnlt_Safrane	0
Alfa_156	-1
Rnlt_Laguna	-1

Table 8. Scores nets pour ELECTRE IV et classement.

De fait, on obtient un (mini-)classement final où VW_Passat et BMW_320d se détachent légèrement. Les autres sont non-discriminés à l'exception de Alfa_156 et Rnlt_Laguna qui ont un score négatif. Cela traduit un effet de *faible discrimination* par la méthode, avec ces paramètres ($s = 0.75$ et veto modéré).

5 Méthode ELECTRE IS

5.1 Rappels sur ELECTRE IS

La méthode **ELECTRE IS** est une variante focalisée sur les situations où la distinction entre préférence et indifférence se fait par une *fonction linéaire* avec seuil. L'idée est de :

- Calculer une *concordance locale* pour chaque critère via une fonction linéaire (dépendant de p_k) :

$$P_k(\Delta_k) = \begin{cases} 1 & \text{si } \Delta_k \geq 0, \\ 0 & \text{si } \Delta_k \leq -p_k, \\ 1 + \frac{\Delta_k}{p_k} & \text{si } -p_k < \Delta_k < 0. \end{cases}$$

où $\Delta_k = a_k - b_k$ est la différence de performance entre deux alternatives a et b sur le critère k , et $p_k > 0$ est le seuil d'indifférence.

- agréger en *concordance globale* (pondérée);
- introduire un veto éventuel si la différence Δ_k sur un critère dépasse un certain niveau.

5.2 Résultats pour nos données

Le paramétrage correspond ici à celui de la section PROMETHEE B (mêmes p_k et mêmes veto que ELECTRE IV). Nous fixons un seuil de concordance globale $s = 0.75$. Les matrices finales (concordance, non-discordance, surclassement) sont listées Table 9 à 11.

	Alfa	Audi	Xantia	406	Saab	Laguna	Passat	BMW	Xsara	Safrane
Alfa_156	0.00	0.60	0.58	0.60	0.57	0.54	0.48	0.27	0.55	0.49
Audi_A4	0.44	0.00	0.72	0.68	0.56	0.61	0.40	0.60	0.80	0.35
Cit_Xantia	0.59	0.60	0.00	0.85	0.61	0.68	0.57	0.54	0.60	0.29
Peugeot_406	0.51	0.64	0.69	0.00	0.58	0.50	0.42	0.57	0.55	0.38
Saab_TID	0.68	0.80	0.76	0.63	0.00	0.64	0.47	0.45	0.65	0.49
Rnlt_Laguna	0.77	0.60	0.68	0.70	0.56	0.00	0.53	0.48	0.62	0.32
VW_Passat	0.79	0.79	0.83	0.85	0.76	1.00	0.00	0.60	0.64	0.49
BMW_320d	0.80	0.68	0.65	0.62	0.76	0.76	0.40	0.00	0.77	0.60
Cit_Xsara	0.45	0.30	0.45	0.48	0.35	0.45	0.45	0.35	0.00	0.35
Rnlt_Safrane	0.72	0.65	0.80	0.79	0.80	0.80	0.65	0.45	0.65	0.00

Table 9. Table de concordance globale (ELECTRE IS).

	Alfa	Audi	Xantia	406	Saab	Laguna	Passat	BMW	Xsara	Safrane
Alfa_156	0	0	0	0	1	0	0	0	0	0
Audi_A4	0	0	0	0	0	0	0	0	0	0
Cit_Xantia	0	0	0	0	0	1	0	0	0	0
Peugeot_406	0	0	0	0	0	1	0	0	0	0
Saab_TID	0	0	0	0	0	0	0	0	0	0
Rnlt_Laguna	0	0	0	0	0	0	0	0	0	0
VW_Passat	0	0	0	0	0	1	0	0	0	0
BMW_320d	1	1	0	0	0	0	0	0	0	0
Cit_Xsara	0	0	0	0	0	0	0	0	0	0
Rnlt_Safrane	0	0	0	0	0	0	0	0	0	0

Table 10. Table de non-discordance (ELECTRE IS).

	Alfa	Audi	Xantia	406	Saab	Laguna	Passat	BMW	Xsara	Safrane
Alfa_156	0	0	0	0	0	0	0	0	0	0
Audi_A4	0	0	0	0	0	0	0	0	0	0
Cit_Xantia	0	0	0	0	0	0	0	0	0	0
Peugeot_406	0	0	0	0	0	0	0	0	0	0
Saab_TID	0	0	0	0	0	0	0	0	0	0
Rnlt_Laguna	0	0	0	0	0	0	0	0	0	0
VW_Passat	0	0	0	0	0	1	0	0	0	0
BMW_320d	1	1	0	0	0	0	0	0	0	0
Cit_Xsara	0	0	0	0	0	0	0	0	0	0
Rnlt_Safrane	0	0	0	0	0	0	0	0	0	0

Table 11. Matrice de surclassement finale (ELECTRE IS), (*aSb*) binaire.

Le résultat final est similaire à l'ELECTRE IV précédent (passat surclasse laguna, bmw surclasse alfa, etc.), sauf que les valeurs de concordance locale sont un peu différentes. Le classement net (score +1 ou -1) aboutit au même constat de faible discrimination.

6 Analyse comparative

Voici un résumé succinct de l'ordre (PROMETHEE II) vs. l'ordre (ELECTRE) :

- **PROMETHEE II** (scénario A ou B) nous donne un *classement complet*, avec Rnlt_Safrane et VW_Passat aux premières places, tandis que Cit_Xsara ou Peugeot_406 se trouvent

en queue.

- **ELECTRE IV / IS** (avec $s = 0.75$) nous signale que seule la *BMW_320d* domine *Alfa_156* et la *VW_Passat* domine *Rnlt_Laguna*. Toutes les autres paires ne dépassent pas le niveau de concordance (ou sont bloquées par veto). On obtient donc un *classement partiel* très peu discriminant.

Le fait qu'ELECTRE n'aboutisse pas aux mêmes discriminations que PROMETHEE est normal :

1. Les *seuils de veto* sont assez stricts, ou mal adaptés aux écarts réels ;
2. Le *seuil de concordance global* 0,75 peut être jugé exigeant.

En pratique, il est possible d'ajuster les $\{p_k\}$, les veto et le seuil s pour affiner la discrimination.

7 Conclusion

Nous avons illustré l'usage de différentes méthodes multicritères (**PROMETHEE I/II, ELECTRE IV, ELECTRE IS**) sur un même jeu de données. Les résultats montrent que :

1. **PROMETHEE II** permet un *classement total*, mais suppose l'existence d'une échelle de préférences (fonctions usuelle ou linéaire). Les poids et les seuils (p_k) influencent fortement les flux ϕ^+ , ϕ^- et ϕ . Cependant, l'ordre final reste relativement stable dans notre exemple.
2. **ELECTRE IV et IS** permettent de modéliser un *seuil de concordance global* et un *veto* par critère. Dans notre cas, le choix $s = 0.75$ et quelques vetos parfois activés aboutissent à une *faible discrimination*. La conclusion est qu'il faut affiner les seuils si l'on veut obtenir plus de distinctions entre les alternatives.

En final, on retrouve un **noyau dur** des alternatives dominantes : *Renault Safrane*, *VW Passat*, *BMW 320d*, qui sortent souvent en tête. À l'opposé, *Citroën Xsara* et *Peugeot 406* se retrouvent fréquemment moins bien classées, conséquence probable d'un ratio performances / prix défavorable.

Ainsi, pour un **décideur**, la robustesse des résultats est à examiner, notamment en testant plusieurs profils de poids et de seuils. Dans une démarche *d'aide à la décision*, nous pourrions recommander la *VW Passat* ou la *BMW 320d* comme bons compromis de performances globales, à condition que leur prix (relativement élevé) soit acceptable, ou la *Renault Safrane* si l'on valorise particulièrement le confort.

Quatrième partie Conclusion Générale

Ces deux projets illustrent l'importance d'une préparation minutieuse des données et de l'utilisation de méthodes analytiques adaptées pour soutenir la prise de décision. Le modèle prédictif développé pour la démission bancaire (notamment par SVM) affiche d'excellentes performances, tandis que l'étude multicritère sur les véhicules offre une comparaison claire des alternatives. Ces travaux fournissent des perspectives intéressantes pour de futures applications en Business Intelligence.