Projet Data Science

Analyse prédictive des prêts bancaires

Salma Echalih

Table des matières

I. Introduction

Dans ce projet, j'ai choisi de tenter de prédire si un crédit sera accepté par la banque ou non en fonction des caractéristiques des individus fournies. Le jeu de données utilisé pour cette tâche est obtenu sur Kaggle :

https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset .

Ce projet et ces données ont été choisis pour plusieurs raisons. Tout d'abord, il offre une opportunité d'explorer les tendances socio-économiques liées aux prêts et de développer des modèles prédictifs pour évaluer les risques de crédit .

En outre, ces données sont vastes et contiennent une grande quantité d'informations relatives aux individus et aux prêts , ce qui les rend précieuses pour les projets impliquant l'apprentissage automatique. Il est possible de déduire des modèles de recommandations à partir de ces données en utilisant des techniques d'apprentissage automatique.

À partir de ces données, nous pensons qu'il serait possible de construire des modèles qui peuvent être utilisés pour prédire si une demande de prêt sera approuvée ou refusée.

II. Tâche

La tâche principale de ce projet est de prédire si un crédit sera accepté ou non, à partir de ces données il sera alors possible de fournir à la banque des informations pertinentes pour prendre une décision éclairée quant à l'approbation ou le rejet d'une demande de crédit .

Afin de comprendre la structure des données et d'identifier les facteurs potentiellement pertinents pour notre tâche de prédiction, nous nettoyons d'abord les données, puis on a fait une analyse exploratoire pour bien comprendre les données on faisant des visualisations et on travaillons sur des caractéristique et des colonnes précises de notre base de donnée. Les données seront ensuite transformées et les caractéristiques pertinentes seront extraites dans le cadre du processus de pré-traitement pour rendre les données adaptées à l'apprentissage automatique.

Ensuite, nous allons entraîner différents modèles d'apprentissage automatique sur les données d'entraînement et les évaluer sur les données de test. Nous allons explorer différents types de modèles d'apprentissage automatique, tels que les arbres de décision, la régression logistique, KNN et identifier celui qui donne les meilleures performances pour notre tâche.

III. Description du jeu de données :

Le jeu de données est disposé d'un fichier test.csv contenant les informations de l'analyse de crédit pour évaluer la probabilité qu'une personne obtienne un prêt en fonction de diverses caractéristiques.

Le jeu de données contient un grand nombre de données catégorielles anonymisées et transformées en IDs, des données manquantes et des données quantitatives.

Le jeu de données que je vais aborder concerne des informations sur des prêts, comprenant les champs suivants :

Loan_ID : Identifiant unique associé à chaque demande de prêt.

Gender : Genre de l'emprunteur, pouvant prendre les valeurs "Male" (masculin), "Female" (féminin) .

Married : Statut matrimonial de l'emprunteur, indiqué par "Yes" (oui) ou "No" (non).

Dependents : Nombre de personnes à charge de l'emprunteur.

Education : Niveau d'éducation de l'emprunteur, avec les catégories "Graduate" (diplômé) et "Not Graduate" (non diplômé).

Self_Employed : Statut d'emploi indépendant de l'emprunteur, indiqué par "Yes" (oui) ou "No" (non).

ApplicantIncome : Revenu de l'emprunteur.

CoapplicantIncome : Revenu du co-emprunteur, s'il y en a un.

LoanAmount : Montant du prêt demandé.

Loan_Amount_Term : Durée du prêt en mois.

Credit_History : Historique de crédit de l'emprunteur, représenté par des valeurs booléennes "true" (vrai) ou "false" (faux), avec également des valeurs nulles ("[null]").

Loan_Status : Statut de l'approbation du prêt, indiqué par "Y" (oui, approuvé) ou "N" (non, non approuvé).

1. Preprocessing

Premièrement, nous devons prétraiter les données afin de pouvoir les utiliser dans les outils de machine Learning. Après le chargement de notre jeu de données à partir d'un fichier CSV ,j'ai dentifié les valeurs manquantes dans les colonnes pertinente , et puis pour les variables catégoriques j'ai remplacé les valeurs manquantes par les valeurs qui se répètent le plus ,pour la variable Loan_Status je l'ai transformée en une variable binaire pour être utilisée comme cible dans la modélisation,en utilisant abelEncoder de scikit-learn j'ai remplacé les valeurs catégoriques par des valeurs numériques 0, 1, 2,ect et J'ai également supprimé la variable loan_id parce qu'elle ne comportait pas assez d'informations.

Pour bien comprendre notre jeu de donnée j'ai fait une étude exploratoire .

Premièrement j'ai essaye de visualiser la variable cible en utilisant les bibliothèques matplotlib et seaborn pour générer un graphique et calculer le pourcentage des crédits accordés et non accordés dans cette variable.

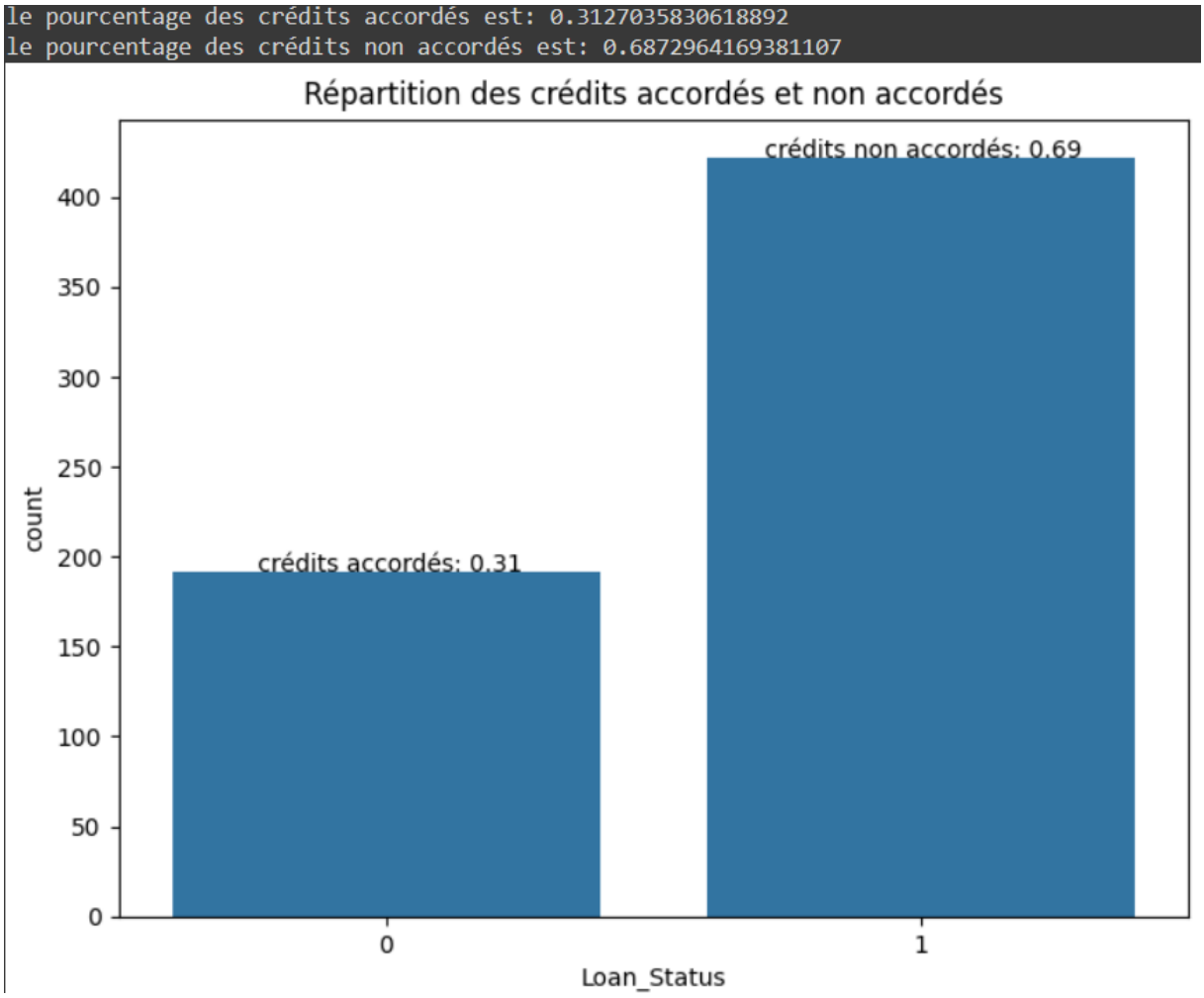

Figure 1 :Répartition des crédits

Et maintenant je vais voir l'impact de chaque variable sur le fait d'accorder un crédit ou pas on commence par la variable

Credit history

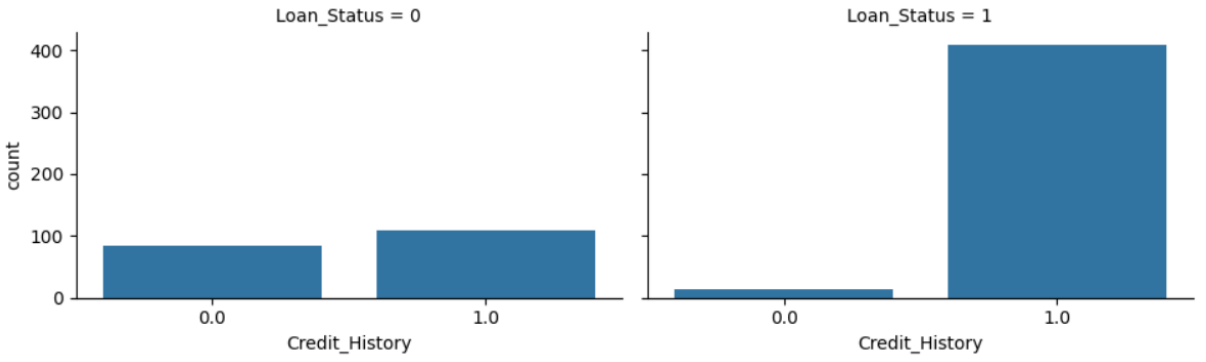

Figure 2 : Répartition des crédits selon variable crédit history

Ce qu'on remarque c'est qu'on accorde les crédits justes pour les personnes qui ont déjà un historique de crédit ,donc cette variable est importante pour accorder un crédit .

Gender

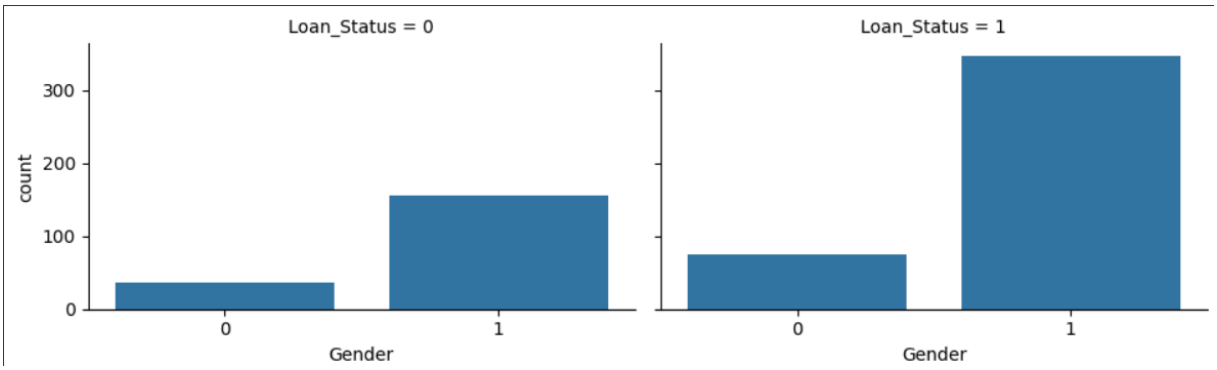

Figure 3 : Répartition des crédits selon variable gender

On remarque que les hommes qui ont demande beaucoup les crédits ,c'est eux qui ont les crédits acceptées .

Married

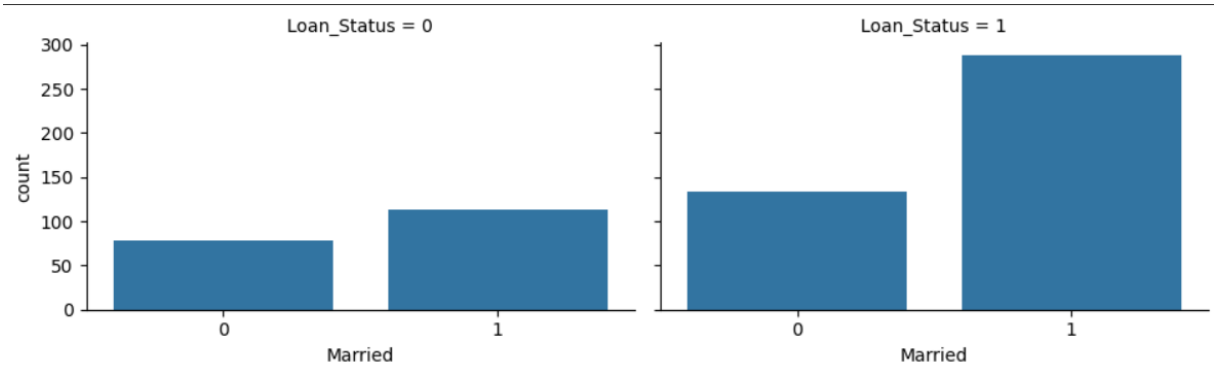

Figure 4 : Répartition des crédits selon variable Married

On observe que la variable Married n'a pas un grand impact sur l'accord d'un crédit .

Revenu du demandeur

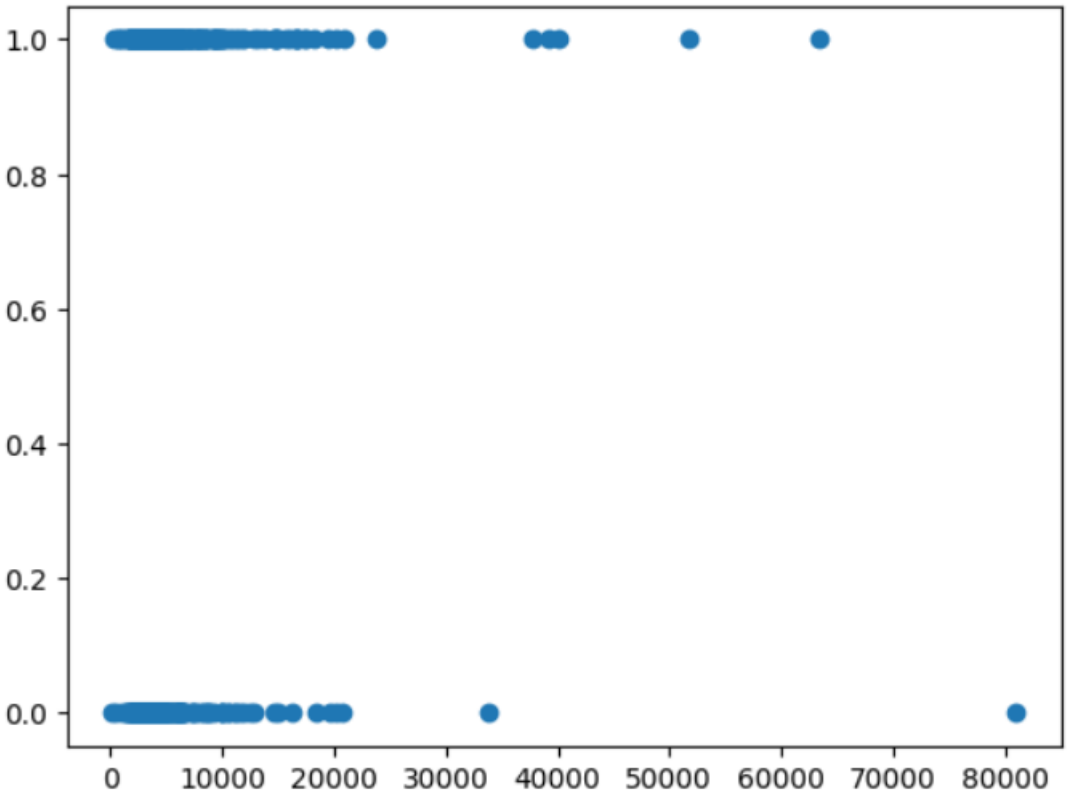

Figure 5 : Répartition des crédits selon variable ApplicantIncome

On observe que la variable revenu de demandeur n'impacte pas trop l'accord ou le refus du crédit .

Revenu du conjoint

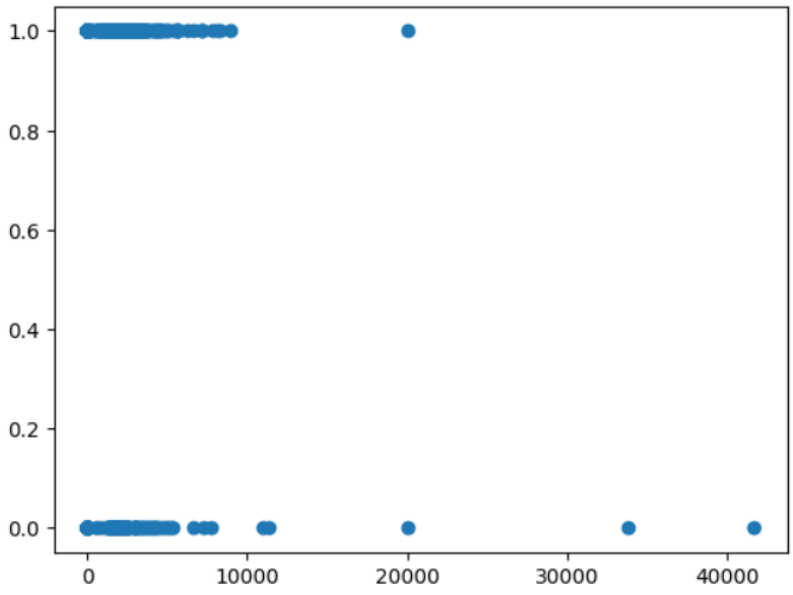

Figure 6 : Répartition des crédits selon variable CoapplicantIncome

On observe que le crédit est accepté si le revenu du conjoint est élevée sinon sera pas accepte alors la variable revenu de conjoint est importante pour accorder un crédit .

2. Séparation des données train / test

Dans le cadre de notre analyse de données, nous avons utilisé la méthode de séparation des données appelée Stratified Shuffle Split pour diviser notre base de données en ensembles d'entraînement et de test de manière stratifiée. Cette approche garantit que la répartition des classes au sein de nos données est maintenue dans les ensembles d'entraînement et de test, ce qui est crucial lorsque les classes ne sont pas équilibrées. En utilisant la classe StratifiedShuffleSplit de la bibliothèque scikit-learn, nous avons spécifié que 20 % de nos données seront utilisées comme données de test, avec une graine aléatoire fixée à 42 pour assurer la reproductibilité des résultats. Après la séparation des données, nous avons extrait les caractéristiques et les étiquettes des ensembles d'entraînement (X_train et y_train) ainsi que des ensembles de test (X_test et y_test). Cette approche nous permet de disposer de données d'entraînement et de test bien équilibrées, essentielles pour développer et évaluer nos modèles de manière fiable.

IV. Méthodologie

Tout d'abord, j'ai divisé les données en deux types : les données catégoriques (cat_data) et les données numériques (num_data). Les valeurs manquantes dans les données catégoriques sont remplies avec les valeurs les plus fréquentes de chaque colonne, tandis que les valeurs manquantes dans les données numériques sont remplacées par les valeurs précédentes de la même colonne. Ensuite, la colonne cible "Loan_Status" est transformée en valeurs numériques (1 pour "Y" et 0 pour "N"). Les autres variables catégoriques sont également transformées en valeurs numériques à l'aide de l'encodeur LabelEncoder. La colonne "Loan_ID" est supprimée car elle ne contribue pas à l'analyse. Ensuite, les données catégoriques et numériques sont concaténées pour former un seul ensemble de données (X) avec la colonne cible (y). J'ai utilisé des graphiques pour visualiser la répartition des crédits accordés et non accordés. Ainsi j'ai créé des graphiques supplémentaires pour explorer la relation entre le statut du prêt et d'autres variables telles que l'historique de crédit, le sexe, le statut matrimonial, l'éducation, le revenu de l'emprunteur et du co-emprunteur. Enfin, la base de données est divisée en ensembles d'entraînement (80 %) et de test (20 %) en utilisant StratifiedShuffleSplit pour garantir une répartition équilibrée des classes dans les deux ensembles. Ce prétraitement complet des données prépare efficacement l'ensemble de données pour l'analyse exploratoire et la construction de modèles prédictifs.

1. Algorithmes utilisés

J'ai testé trois algorithmes afin de trouver celui qui vérifie nos attentes pourrait nous donner les meilleurs résultats.

Régression logistique (Logistic Regression) : Ce modèle est souvent utilisé pour les tâches de classification binaire, comme celle que nous avons avec les prêts (accordé ou non accordé). Il est robuste, facile à interpréter et efficace lorsque la relation entre les caractéristiques et la cible est linéaire ou quasi-linéaire.

K-plus proches voisins (KNN) : KNN est un modèle non paramétrique qui ne fait pas d'hypothèses sur la distribution des données. Il fonctionne bien pour des ensembles de données de taille moyenne à petite, où les frontières de décision entre les classes ne sont pas linéaires. En utilisant la similarité des points, il classe de nouveaux exemples en se basant sur les exemples étiquetés les plus proches.

Arbre de décision (Decision Tree) : Les arbres de décision sont des modèles très interprétables qui fonctionnent bien pour des données complexes avec des interactions non linéaires entre les variables. En spécifiant la profondeur maximale de l'arbre (comme dans notre cas), on évite le surapprentissage et on obtient un modèle plus généralisable.

Le choix de ces trois algorithmes repose sur la diversité de leurs approches et de leurs capacités à modéliser différents types de relations entre les caractéristiques et la cible. La régression logistique est choisie pour sa simplicité et son efficacité dans les tâches de classification binaire. Le KNN est sélectionné pour sa flexibilité et sa capacité à capturer des structures non linéaires dans les données. Enfin, l'arbre de décision est retenu pour sa capacité à gérer des ensembles de données complexes et hétérogènes tout en permettant une interprétation aisée des résultats. En les combinant, on bénéficie d'une évaluation comparative des performances et de perspectives diverses sur la relation entre les caractéristiques des emprunteurs et l'approbation de prêts.

2.Les hyperparamètres

Dans le cadre de notre analyse pour déterminer les meilleurs modèles de prédiction de l'approbation de prêts, nous avons utilisé trois algorithmes : régression logistique, K plus proches voisins (KNN) et arbre de décision. Notre approche a impliqué une exploration approfondie des hyperparamètres de ces algorithmes pour trouver les configurations optimales qui maximisent la précision des prédictions. Pour chaque algorithme, nous avons défini des grilles de paramètres contenant différentes valeurs pour des hyperparamètres spécifiques. Pour la régression logistique, nous avons ajusté le paramètre de régularisation C. Pour KNN, nous avons varié le nombre de voisins. Enfin, pour l'arbre de décision, nous avons examiné la profondeur maximale de l'arbre et le critère de division (gini ou entropie). Nous avons utilisé la méthode de recherche sur grille (GridSearchCV) avec une validation croisée à 5 plis pour évaluer chaque combinaison de paramètres et sélectionner celle qui produit les meilleures performances. Les résultats de cette exploration ont été intégrés dans notre processus d'évaluation des modèles, où nous avons calculé la précision de chaque modèle optimisé sur notre ensemble de test.

Nous nous attendons à obtenir un modèle avec une précision élevée et une faible erreur de classification. Nous évaluerons notre modèle en utilisant des métriques telles que la précision .

V. Résultats :

Comme le montre tableau 1 la différence de performance entre les modèles, avec la régression logistique obtenant la précision la plus élevée, suivie de l'arbre de décision, puis du modèle KNN qui présente la précision la plus basse parmi les trois**.**

Modèle	Précision
Régression Logistique	0.854
KNeighborsClassifier	0.650
DecisionTreeClassifier	0.846

Tableau 1 :précisions des modèles

Pour rendre compte de nos résultats nous avons calculé les métriques Accuracy et on a fait la courbe roc .

1.accuracy

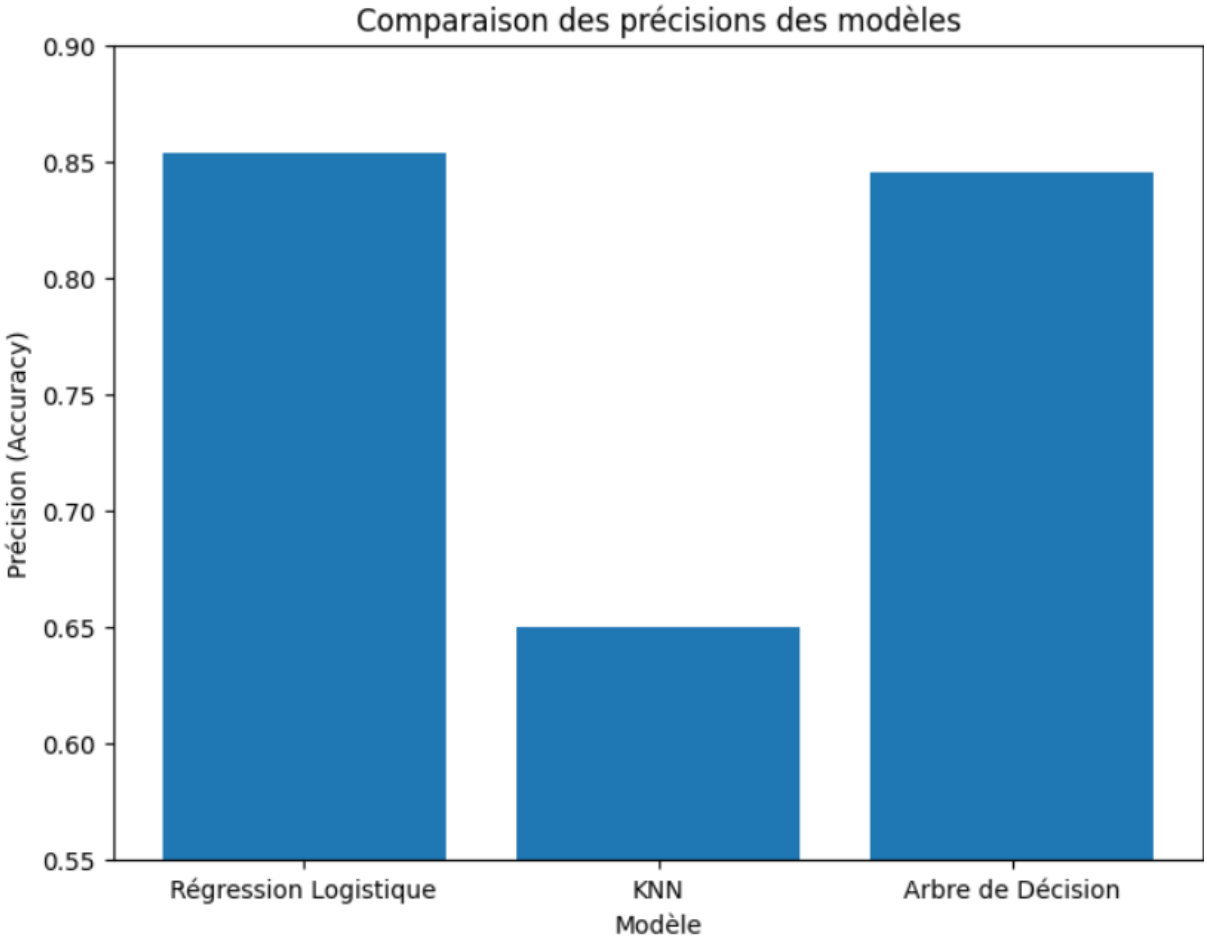

Figure 7 :comparaison des précisions des modèles

Ces résultats(voir figure 7) mettent en lumière les performances différenciées des modèles que nous avons entraînés pour prédire l'approbation des prêts. Tout d'abord, la régression logistique a atteint une précision de 0.85, ce qui signifie qu'elle a correctement classé environ 85% des échantillons de test dans la bonne catégorie de prêt accordé ou non accordé. Cette précision élevée indique que la régression logistique a bien généralisé à de nouvelles données et qu'elle est capable de prendre des décisions précises. En ce qui concerne l'arbre de décision, sa précision de 0.84 montre également une performance robuste, bien que légèrement inférieure à celle de la régression logistique. Cela suggère que l'arbre de décision a réussi à capturer les schémas importants dans les données pour prédire les prêts avec une bonne précision. Le modèle KNN, en revanche, a une précision plus basse de 0.650, ce qui signifie qu'il a eu plus de difficulté à classifier les échantillons correctement par rapport à la régression logistique.

2.Courbe ROC

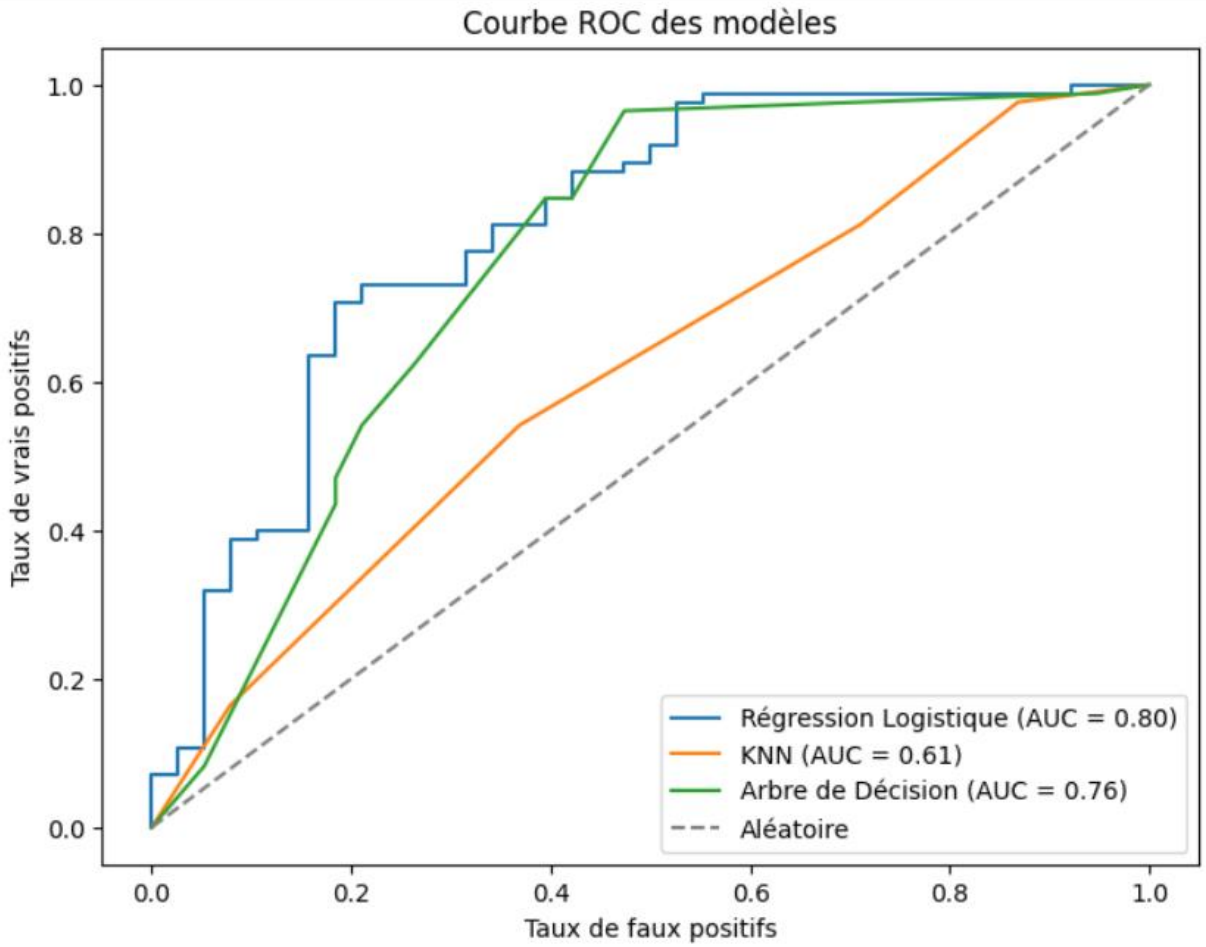

Figure 8 :Courbe ROC

Les résultats obtenus(figure 8) pour les aires sous la courbe ROC (AUC) fournissent des indications importantes sur la capacité de chaque modèle à discriminer entre les classes de prêt accordé et non accordé. Tout d'abord, la régression logistique a obtenu un AUC de 0.80, ce qui indique une capacité assez solide à séparer les deux classes. Cela signifie que le modèle de régression logistique a pu discriminer de manière efficace entre les prêts accordés et non accordés, avec une marge de certitude significative. En revanche, le modèle KNN a affiché un AUC de 0.61, ce qui est relativement bas. Cela suggère que le modèle KNN a eu plus de difficulté à bien discriminer les deux classes et a eu des performances moins satisfaisantes en comparaison avec la régression logistique. Quant à l'arbre de décision, il a obtenu un AUC de 0.76, montrant une capacité intermédiaire à séparer les classes par rapport aux deux autres modèles. Bien que l'AUC de l'arbre de décision soit inférieur à celui de la régression logistique, il reste dans une plage acceptable et indique une capacité de discrimination raisonnable. En conclusion, ces résultats mettent en évidence que la régression logistique a la meilleure capacité à discriminer entre les prêts accordés et non accordés, suivie de près par l'arbre de décision. Le modèle KNN semble avoir des difficultés à généraliser et à bien distinguer les deux classes dans ce contexte particulier de prédiction des prêts.

VI. Conclusion :

Après avoir analysé et évalué les performances des meilleurs modèles pour la prédiction de l'approbation des prêts, plusieurs observations et conclusions peuvent être tirées. Tout d'abord, la régression logistique a émergé comme le modèle le plus performant parmi ceux testés, avec une précision de 0.85 et un AUC de 0.80. Ces résultats indiquent une capacité solide à généraliser aux données de test et à discriminer efficacement entre les prêts accordés et non accordés. En revanche, le modèle KNN a affiché une performance inférieure avec une précision de 0.65 et un AUC de 0.61, suggérant des difficultés à généraliser et à bien distinguer les classes. L'arbre de décision a obtenu des résultats intermédiaires avec une précision de 0.85 et un AUC de 0.76, montrant une capacité raisonnable à discriminer entre les classes. Au-delà des performances numériques, des améliorations potentielles pourraient être envisagées pour renforcer les modèles. Par exemple, une exploration plus approfondie des hyperparamètres, notamment pour le modèle KNN, pourrait aider à optimiser la performance et à améliorer la capacité de généralisation. De plus, l'inclusion de nouvelles variables pertinentes dans l'ensemble de données pourrait fournir des informations supplémentaires pour améliorer la prédiction des prêts. En rétrospective, une analyse plus approfondie de l'impact de chaque variable sur la prédiction aurait pu être bénéfique, ainsi qu'une évaluation de la robustesse des modèles face à des données déséquilibrées. De plus, une validation croisée plus rigoureuse aurait pu être réalisée pour évaluer la stabilité des performances des modèles sur différents ensembles de données.

Dans le cadre de ce projet, une analyse plus approfondie utilisant la matrice de confusion aurait été bénéfique pour évaluer la performance des modèles sous différents angles. Malheureusement, en raison de contraintes de temps, cette étape n'a pas été incluse dans l'analyse initiale. En rétrospective, l'ajout de la matrice de confusion aurait permis une compréhension plus détaillée des résultats, en mettant en évidence les vrais positifs, les faux positifs, les vrais négatifs et les faux négatifs pour chaque modèle. Cela aurait fourni des informations précieuses sur les erreurs de classification spécifiques de chaque modèle et sur leur capacité à bien généraliser aux données de test.

En conclusion, tout en reconnaissant les performances satisfaisantes de la régression logistique et de l'arbre de décision, il reste des opportunités d'amélioration et d'exploration plus approfondie pour renforcer la capacité prédictive des modèles et garantir leur généralisable dans des contextes réels de décision de prêt.