Année 2022/2023

Université Paris-Nanterre

UFR SEGMI

**Statisitiques Big Data Avancé**

M2 GDA

YANG Lulu

TIAN Minghao

**1 Introduction**

Aujourd'hui, les données sont devenues l'actif le plus précieux du secteur bancaire. L'analyse d'informations pertinentes grâce à de grandes quantités de données aide non seulement les banques à comprendre les besoins des clients et à prendre des décisions plus efficaces fondées sur les données, mais elle renforce également les capacités commerciales, accroît l'efficacité opérationnelle, améliore les services existants et génère davantage de revenus grâce à toutes ces actions.

Ce projet démontre comment de multiples méthodes d'apprentissage automatique peuvent être appliquées pour prédire le degré de risque des clients d'une banque. Plus précisément, pour répondre à l'exigence du cours de statistiques avancées sur le Big Data - appliquer des méthodes d'apprentissage automatique telles que les forêts tournantes pour prédire certaines mesures liées au secteur bancaire - nous avons appliqué un programme Python pour analyser les caractéristiques historiques des clients des prêts à la consommation et avons finalement appliqué les méthodes de l'arbre de décision, de la forêt aléatoire et de la forêt tournante respectivement pour prédire quelles les clients sont susceptibles d'être confrontés à des défauts de paiement des prêts et ont fait des comparaisons.

**1.1 Introduction à la méthode**

**1.1.1 Arbre de décision**

L'arbre de décision [1] est un algorithme classique de classification et de régression en apprentissage automatique. Il s'agit d'un algorithme de classification supervisée couramment utilisé qui analyse diverses situations (valeurs de caractéristiques) en construisant une structure de décision arborescente basée sur un ensemble connu de circonstances.

Les arbres de décision présentent l'avantage d'être faciles à comprendre et à interpréter, d'extraire facilement des règles et de pouvoir traiter aussi bien des données nominales que numériques. Dans ces conditions, le modèle d'arbre de décision peut traiter des échantillons avec des attributs manquants ou des échantillons contenant des caractéristiques non corrélées, en plus des données générales. D'autre part, l'arbre de décision s'exécute relativement rapidement lors des tests et son efficacité est considérable.

**1.1.2 Forêt aléatoire**

Une forêt aléatoire [2] est un classificateur qui contient plusieurs arbres de décision, et la classe de sa sortie est déterminée par la pluralité des classes des sorties des arbres individuels. En d'autres termes, plusieurs arbres de décision se classent eux-mêmes et les résultats de chacun d'entre eux sont rassemblés, le vote le plus élevé étant la classification effectuée par l'ensemble de la "forêt".

Une forêt aléatoire peut déterminer l'importance des caractéristiques et la façon dont les différentes caractéristiques s'influencent mutuellement. Pour les ensembles de données non équilibrés, il peut équilibrer l'erreur et maintenir la précision même si une grande partie des caractéristiques sont manquantes. Cependant, il a été démontré que les forêts aléatoires s'adaptent trop à certains problèmes de classification ou de régression bruyants, et pour les données comportant des attributs de valeurs différentes, les attributs dont les valeurs sont plus divisées auront un impact plus important sur la forêt aléatoire, de sorte que les poids des attributs produits par la forêt aléatoire sur de telles données ne sont pas crédibles. Cependant, à l'exception des quelques cas mentionnés ci-dessus, les forêts aléatoires sont garanties d'avoir une performance supérieure dans la plupart des cas grâce au mécanisme de vote.

**1.1.3 Forêt de rotation**

La forêt de rotation [3] est une méthode permettant de générer une collection de classificateurs basée sur l'extraction de caractéristiques, qui surpasse les méthodes d'arbre de décision et de forêt de rotation dans plusieurs scénarios. Contrairement aux deux premières méthodes qui nécessitent l'intégration d'un grand nombre d'arbres pour obtenir de bons résultats, la méthode Rotation Forest peut utiliser des arbres plus petits pour obtenir des résultats identiques, voire meilleurs. La forêt de rotation conserve le mécanisme de vote de la méthode de la forêt aléatoire, mais ajoute des étapes supplémentaires pour améliorer encore les performances du modèle. Dans l'algorithme de la forêt de rotation, les attributs sont divisés en K sous-ensembles non chevauchants de taille égale, puis combinés avec l'ACP, la matrice de rotation, pour achever la construction du modèle.

Comme il n'y a pas de code existant pour la forêt tournante qui puisse être utilisé directement, nous avons terminé la construction du code de la forêt tournante en nous basant sur l'algorithme décrit dans l'article [3] et d'autres sources. La logique algorithmique de la forêt tournante est la suivante.

Algorithme 1.1 Pseudo-code pour les forêts tournantes

|  |
| --- |
| Entrée : k, le nombre d'arbres, f, le nombre de caractéristiques, p, la proportion de l'échantillon  1 : Soit F =< F1 . . . Fk > sont les arbres C4.5 de la forêt.  2 : pour i ← 1 à k, faire  3 : Partitionner aléatoirement les caractéristiques originales en r sous-ensembles, chacun avec f caractéristiques (r =  m/f), désignés par < S1 . . . Sr >.  4 : Soit Di l'ensemble de formation pour l'arbre i, initialisé aux données originales, Di ← D.  5 : pour j ← 1 à r, faire  6 : Sélectionnez un sous-ensemble non vide de classes et extrayez uniquement les cas avec ces étiquettes de classe.  Chaque classe a une probabilité d'inclusion de 0,5.  7 : Tirage d'une proportion p de cas (sans remplacement) parmi ceux ayant la  valeur de classe sélectionnée  8 : Effectuez une analyse en composantes principales (ACP) sur les caractéristiques de Sj sur ce sous-ensemble de données.  sous-ensemble de données  9 : Appliquer la transformée PCA construite sur ce sous-ensemble aux caractéristiques dans Sj de l'ensemble de  ensemble du train  10 : Remplacez les caractéristiques Sj dans Di par les caractéristiques PCA.  11 : Construire le classificateur C4.5 Fi sur les données transformées Di. |

**1.2 Introduction à l'ensemble des données**

Nous avons obtenu cet ensemble de données à partir de données accessibles au public [4]. Les données ont été auto-segmentées en un ensemble de formation et un ensemble de test contenant respectivement 250 000 et 28 000 lignes de données. La quantité de données est énorme et répond aux exigences du projet. Le tableau suivant présente, entre autres, les caractéristiques des colonnes de l'ensemble de données ainsi que leur description et leur type.

Tableau 1.1 Description de l'ensemble des données

|  |  |  |
| --- | --- | --- |
| Column | Description | Type |
| income | Income of the user | int |
| age | Age of the user | int |
| experience | Professional experience of the user in years | int |
| profession | Profession | string |
| married | Whether married or single | string |
| house\_ownership | Owned or rented or neither | string |
| car\_ownership | Does the person own a car | string |
| risk\_flag | Defaulted on a loan | string |
| current\_job\_years | Years of experience in the current job | int |
| current\_house\_years | Number of years in the current residence | int |
| city | City of residence | string |
| state | State of residence | string |

D'après le tableau, nous savons que les données contiennent un grand nombre de caractéristiques de clients et que beaucoup de ces caractéristiques sont de type chaîne, ce qui rend plus difficile leur modélisation ultérieure.

**2 Développement**

**2.1 Description des données**

Les données originales consistaient en un ensemble de données d'apprentissage de 252 000 lignes et 13 colonnes et un ensemble de données de test de 28 000 lignes. Une partie du jeu de données d'entraînement est présentée dans la figure suivante.

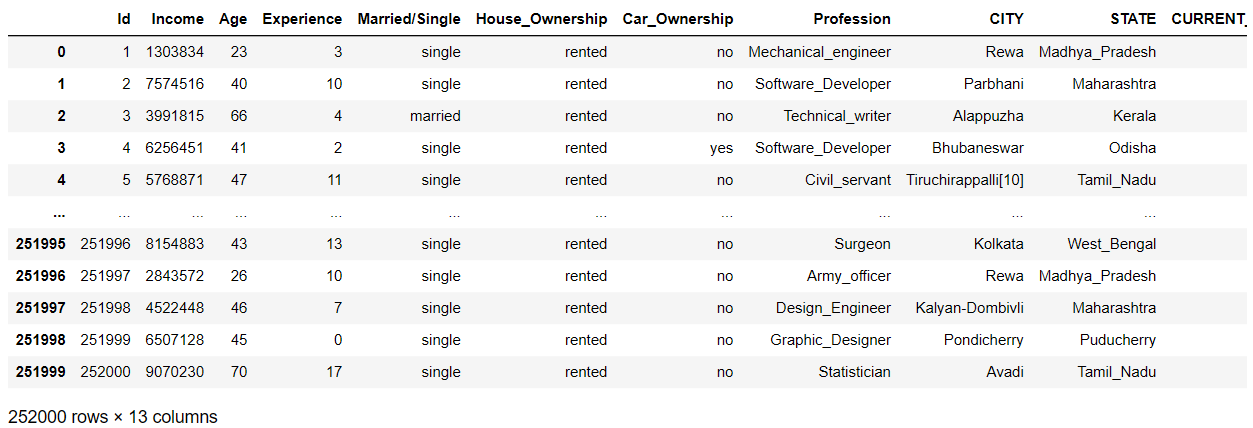


Figure 2.1 Présentation partielle des données de l'ensemble d'entraînement

Pour aller plus loin, nous examinons les détails de chaque colonne à travers le code correspondant, comme le montre la figure 2.2.

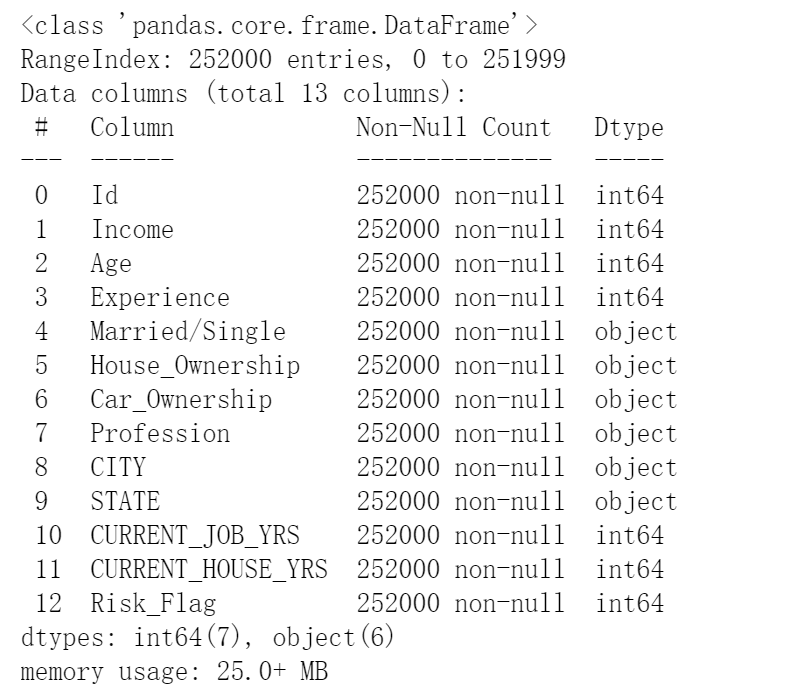


Figure 2.2 Informations de base sur les données de l'ensemble de formation

Ce graphique nous apprend qu'à part la dernière colonne d'étiquettes, les 12 autres colonnes contiennent des caractéristiques des clients de la banque telles que le revenu, l'âge, le lieu de résidence, etc. De plus, avec les informations de base présentées ci-dessus, nous constatons qu'il n'existe aucune donnée manquante dans cet ensemble de données.

Les caractéristiques numériques de l'ensemble de données peuvent facilement être introduites dans le modèle, mais les caractéristiques non numériques nécessitent un traitement supplémentaire. La figure 2.4 montre les informations de base pour toutes les caractéristiques non numériques.

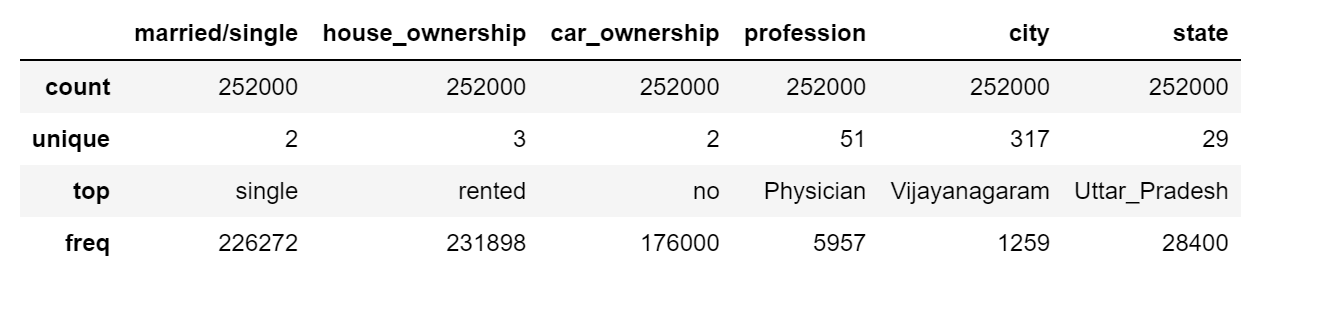


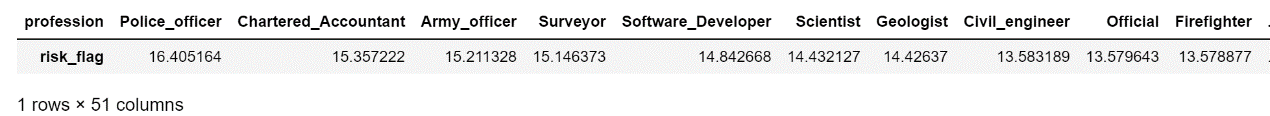
Figure 2.3 Présentation des indicateurs pour les caractéristiques non numériques

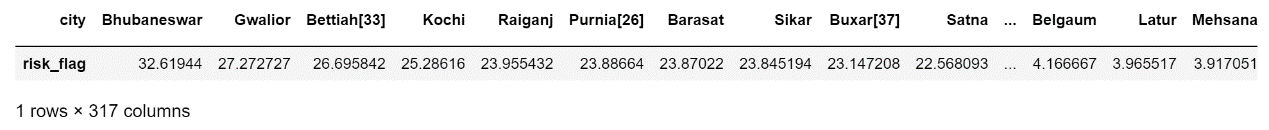
Dans le graphique ci-dessus, nous constatons que certaines des caractéristiques non numériques contiennent un petit nombre de variables, comme le fait d'être marié, d'avoir des biens, etc. Ce type d'information peut facilement être remplacé par les chiffres 0, 1, 2, etc. Cependant, d'autres informations, comme les villes, contiennent un très grand nombre de variables et les transformer directement en nombres incrémentaux fait perdre la relation entre la ville elle-même et le résultat final. Par conséquent, nous devons ensuite nous concentrer sur cette partie des caractéristiques non numériques.

En comptant les étiquettes, nous avons constaté que les données de formation totales comportaient 30 996 étiquettes positives (à risque) et 22 1004 étiquettes négatives (sans risque). Le grave déséquilibre entre les étiquettes positives et négatives pose un problème important pour l'entraînement ultérieur du modèle. En effet, même si le modèle présente toutes les entrées comme étant sans risque, le taux de précision atteint 87 %, alors qu'un tel modèle est pratiquement dénué de sens malgré son excellent taux de précision. Par conséquent, nous devons encore traiter le problème de l'inégalité des étiquettes positives et négatives.

**2.2 Numérisation d'éléments non numériques**

Pour les caractéristiques non numériques comportant un petit nombre de variables, nous leur attribuons directement une valeur de 0, 1, 2, etc. Pour les trois caractéristiques que sont la profession, la ville et l'état, nous calculons la proportion des résultats qui sont à risque sous une caractéristique spécifique.





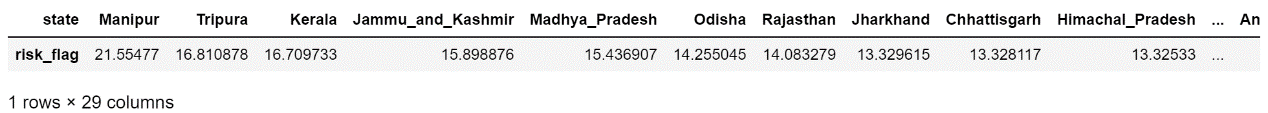
****

Figure 2.4 Résultats partiels du calcul de la probabilité de risque pour chaque étiquette non numérique

Ensuite, nous attribuons l'information de probabilité correspondante directement à la variable correspondante en arrondissant. De cette manière, ces caractéristiques non numériques sont également numérisées avec succès. En outre, dans cette transformation, on pense que les variables ayant des nombres plus élevés sont plus susceptibles de correspondre à des clients risqués, c'est-à-dire qu'elles fournissent au modèle des informations supplémentaires sur les préférences. En d'autres termes, nous avons numérisé les caractéristiques non numériques tout en conservant leurs relations relatives. Les données traitées sont présentées à la figure 2.5.

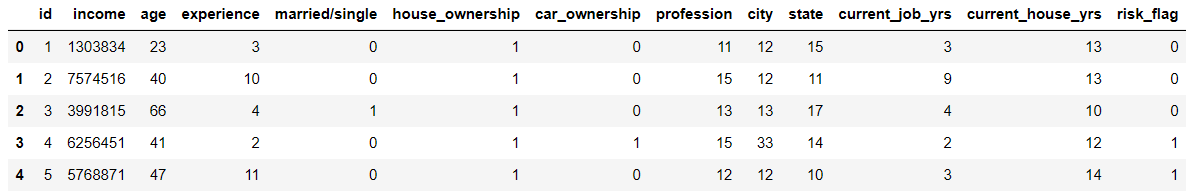


Figure 2.5 Données de l'ensemble d'entraînement après le traitement initial

Nous avons traité l'ensemble de test de la même manière que l'ensemble d'entraînement et avons vérifié qu'il n'y avait pas de caractéristiques non numériques dans l'ensemble de test qui n'apparaissaient pas dans l'ensemble d'entraînement.

**2.3 Calcul de la moyenne des étiquettes positives et négatives**

Dans l'analyse précédente, nous avons constaté que les étiquettes positives et négatives étaient gravement déséquilibrées. Nous avons donc pris les étiquettes négatives qui correspondaient au nombre d'étiquettes positives et les avons mélangées pour obtenir des données d'entraînement équilibrées (61992 lignes).

**3. Résultats**

Nous avons d'abord importé les paquets arbre de décision et forêt aléatoire de sklearn et appliqué les deux modèles pour former et prédire quels clients sont des clients à risque. Ensuite, nous avons implémenté l'algorithme de la forêt tournante en nous basant sur les articles et les sources pertinentes, et nous avons fait des prédictions après avoir importé les données d'entraînement dans le modèle. Les résultats des trois algorithmes sont les suivants.

Tableau 3.1 Résultats de la prédiction du modèle et temps passé

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Méthodes | MSE | Précision | Consommation de temps (s) | |
| Arbre de décision | 0.236 | 0.763 | | **0.209** |
| Forêt aléatoire | 0.211 | 0.789 | | 0.693 |
| Forêt de rotation | **0.209** | **0.791** | | 7.397 |

Le "MSE" dans le tableau est la valeur attendue de la différence au carré entre l'estimation du paramètre et la valeur réelle du paramètre, les valeurs plus petites indiquant des résultats supérieurs. D'après le tableau, nous pouvons constater que les trois méthodes d'arborescence ont donné de bons résultats, avec un taux de précision de plus de 75 %, ce qui signifie que le modèle peut être utilisé pour obtenir un jugement plus précis du client. La forêt tournante obtient les meilleures performances en termes d'erreur quadratique moyenne et de précision, mais au prix d'une consommation de temps plus élevée.

**4 Conclusion**

Pour analyser la façon dont les banques accordent des prêts à la consommation en fonction de divers paramètres, nous avons puisé les données utilisées pour ce projet dans l'ensemble de données publiques de Kaggle - plus de 250 000 lignes de données sur le comportement des utilisateurs et l'octroi de prêts.

Comme les données contiennent un grand nombre de caractéristiques non numériques, nous avons d'abord transformé toutes les caractéristiques non numériques en caractéristiques numériques. Il convient de noter qu'en raison du grand nombre de caractéristiques telles que la ville, nous avons compté la probabilité que l'indicateur correspondant corresponde à un prêt accordé et remplacé le texte original par cette valeur. Cela permet d'atteindre l'objectif numérique tout en fournissant la relation relative entre les différentes caractéristiques. D'autre part, les données de formation initiales comportaient des étiquettes positives et négatives extrêmement mal assorties, ce qui aurait sérieusement affecté la formation ultérieure du modèle, nous avons donc complété la distribution moyenne des étiquettes sur cette base.

Ensuite, nous avons appliqué l'algorithme de l'arbre de décision, l'algorithme de la forêt aléatoire et l'algorithme de la forêt rotative pour entraîner les données et prédire les résultats sur l'ensemble de test, respectivement. Puisqu'il n'existe pas de code existant pour la forêt tournante, nous avons construit plusieurs fonctions pour mettre en œuvre la fonctionnalité de la forêt tournante.

Les résultats finaux montrent que l'algorithme de forêt aléatoire et l'algorithme de forêt rotative donnent des résultats supérieurs, tandis que la forêt rotative peut obtenir de meilleures performances sur les métriques élevées de la forêt aléatoire. Les résultats montrent non seulement que nous avons rempli les exigences du cours, mais aussi que les méthodes d'apprentissage automatique peuvent être efficaces pour aider à la prise de décision dans le secteur bancaire.

**5 Référence**

[1] Myles, Anthony J., et al. "An introduction to decision tree modeling." Journal of Chemometrics: A Journal of the Chemometrics Society 18.6 (2004): 275-285.

[2] Biau, Gérard, and Erwan Scornet. "A random forest guided tour." Test 25 (2016): 197-227.

[3] Rodriguez, Juan José, Ludmila I. Kuncheva, and Carlos J. Alonso. "Rotation forest: A new classifier ensemble method." IEEE transactions on pattern analysis and machine intelligence 28.10 (2006): 1619-1630.

[4] Kaggle: Loan Prediction Based on Customer Behavior. <https://www.kaggle.com/datasets/subhamjain/loan-prediction-based-on-customer-behavior>.