Segmentation des Clients dans le Secteur des Prêts Bancaires"

nom du dataset:
LOANS

MEMBRES DU GROUPE:

ARWA REBHI
FIRAS ABIDI
ZIED FADHLAOUI
YOUNES BOUALLEGUI

SOMMAIRE

01

COMPREHENSION DU METIER

OBJECTIF MÉTIER :

Notre objectif métier consiste à optimiser la gestion des clients bancaires en identifiant des segments homogènes au sein de notre base de données. Cette segmentation contribuera à l'élaboration de processus de **marketing prédictif**, renforçant ainsi la compréhension de la clientèle et de ses habitudes de consommation.

OBJECTIF ML:(PROBLÉMATIQUE)

- On va utiliser l'approche de segmentation pour regrouper les individus en fonction des données de crédit .
- Prédire à quel groupe appartient chaque individu donné .

ETAT DE L'ART:

Dans cette étude de l'art, nous avons examiné un article intitulé "A customer segmentation approach in commercial banks" rédigé par V. Mihova et V. Pavlov, publié dans les AIP Conference Proceedings en octobre 2018. L'article met en lumière l'importance cruciale de la classification appropriée des emprunteurs dans le secteur bancaire commercial, soulignant les tendances émergentes et la nécessité d'adopter des stratégies de marché modernes et des approches individualisées envers les clients.

L'étude se concentre particulièrement sur l'amélioration de la segmentation des clients, en utilisant la méthode de clustering K-means pour identifier trois segments distincts de clients fidèles : "platinum," "gold," et "silver." La segmentation est basée sur une base de données de 100 emprunteurs d'une succursale bancaire commerciale ayant contracté des prêts à la consommation garantis. Les clients sont définis comme fidèles en fonction de leur historique de crédit. L'article explore différentes variables de segmentation et formule des stratégies potentielles en fonction de ces variables.

AIP Publishing AIP Conference Proceedings

HOME BROWSE FOR AUTHORS FOR ORGANIZERS ABOUT

Volume 2025, Issue 1
25 October 2018

APPLICATION OF MATHEMATICS IN TECHNICAL AND NATURAL SCIENCES: 10th International Conference for Promoting the Application of Mathematics in Technical and Natural Sciences - AMITaNS'18
20–25 June 2018
Albena, Bulgaria

RESEARCH ARTICLE | OCTOBER 25 2018

A customer segmentation approach in commercial banks

V. Mihova, V. Pavlov

Check for updates

Author & Article Information

AIP Conf. Proc. 2025, 030003 (2018)

Share Tools

The results of various recent analyses of commercial banking trends show that proper classification of borrowers is fundamental for the development of a successful business. The increasing competition in both banking system and non-bank institutions requires the use of modern market strategies and individual customer approaches. One of the main priorities in the banking sector is to improve customer segmentation and take it into account in the design and distribution of new products. A common tool to improve competitiveness is the designing of a special range of products and services targeting loyal customers or offering them special discounts for existing products. This represents the so-called "loyalty program", which includes the issuance of various types of cards for such customers. Three clusters (segments) of loyal borrowers: "platinum," "gold," and "silver," are identified in the present work, using K-means clustering. A database of 100 borrowers from a commercial bank branch that took secured consumer loans is analyzed. The clients are defined as loyal based on their credit history (they have less than 3 missed payments for the last year). Three variables are used for their segmentation. Initially, the initial segmentation variables are taken as input data for the analysis, and further study on standardized segmentation variables is carried out. The potential segmentation strategies are formulated depending on the leading segmentation variable. A comparative analysis of the results of both methods examined (with initial and standardized segmentation variables) and of a two-step clustering (obtained in a previous study from one of the authors) is made within each of the strategies. It is specified which type of cluster analysis suits best to each of the strategies.

Topics
Data visualization, Transition metals

View Metrics

Citing Articles Via
Google Scholar
CrossRef (11)

Publish with us - Request a Quote!

Most Read Most Cited

Phytochemical analysis of bioactive compounds in ethanolic extract of Stercula quadrifida R.Br.

Inkjet- and textrail-printing of silicon polymer-based inks for local passivating contacts

The implementation of reflective assessment using Gibbs' reflective cycle in assessing students' writing skill

REFERENCES

1. A team of Investor.bg, "Can we also see interest rates of 3.5% on some retail products?" *Investor.bg*. [in Bulgarian]
2. V. Mihova and V. Pavlov (2017) An approach of estimating the probability of default for new borrowers, *Economy & Business Journal* 11.1, 200–208.
Google Scholar
3. V. Mihova and V. Pavlov, "Comparative analysis on the probability of being a good payer," in

02

COMPREHENSION DES DONNÉES

- Importation des bibliothèques requises

```
from sklearn.preprocessing import MinMaxScaler

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pylab as plt
%matplotlib inline
from sklearn.preprocessing import MinMaxScaler
from scipy.stats import norm
from scipy import stats
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram
from scipy.cluster.hierarchy import fcluster
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix
import itertools
from sklearn.metrics import silhouette_score
```

Nos données contiennent des détails sur le crédit des clients (emprunteurs).
Nous allons lire, comprendre nos données .Notre data set comprend 12 colonnes :

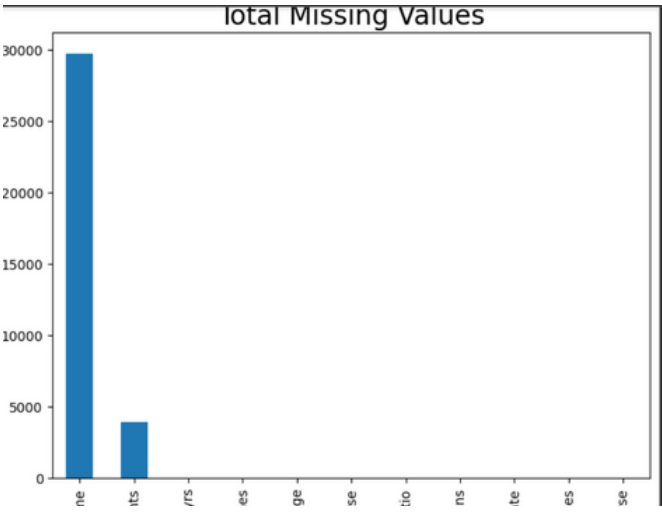
-Unammed: 0 : replique d'id
-SeriousDlqin2yrs : Indique si le client a eu un grave retard de paiement de plus de 90 jours au cours des 2 dernières années.
-RevolvingUtilizationOfUnsecuredLines : Taux d'utilisation renouvelable des lignes non garanties.
-age : Âge du client.
-NumberOfTime30-59DaysPastDueNotWorse : Nombre de fois où le client a été en retard de paiement de 30 à 59 jours mais pas plus grave au cours des 2 dernières années.
-DebtRatio : Ratio de la dette.
-MonthlyIncome : Revenu mensuel du client.
-NumberOfOpenCreditLinesAndLoans : Nombre total de prêts et de lignes de crédit ouverts par le client.
-NumberOfTimes90DaysLate : Nombre de fois où le client a été en retard de paiement de plus de 90 jours.
-NumberRealEstateLoansOrLines : Nombre de prêts immobiliers ou lignes de crédit.
-NumberOfTime60-89DaysPastDueNotWorse : Nombre de fois où le client a été en retard de paiement de 60 à 89 jours mais pas plus grave au cours des 2 dernières années.
-NumberOfDependents : Nombre de personnes à charge du client.

```
[ ]  cell_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150000 entries, 0 to 149999
Data columns (total 11 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   SeriousDlqin2yrs                      150000 non-null  int64
 1   RevolvingUtilizationOfUnsecuredLines  150000 non-null  float64
 2   age                                   150000 non-null  int64
 3   NumberOfTime30-59DaysPastDueNotWorse  150000 non-null  int64
 4   DebtRatio                             150000 non-null  float64
 5   MonthlyIncome                         120269 non-null  float64
 6   NumberOfOpenCreditLinesAndLoans       150000 non-null  int64
 7   NumberOfTimes90DaysLate               150000 non-null  int64
 8   NumberRealEstateLoansOrLines          150000 non-null  int64
 9   NumberOfTime60-89DaysPastDueNotWorse  150000 non-null  int64
 10  NumberOfDependents                    146076 non-null  float64
dtypes: float64(4), int64(7)
memory usage: 12.6 MB
```

D'après la sortie ci-dessus, nous avons 150000 entrées, numérotées de 0 à 149999. Nous avons un mélange de types de données numériques :int et float.

- On va chercher les valeurs NAN




```
[ ]  cell_df.isnull().sum()

SeriousDlqin2yrs                          0
RevolvingUtilizationOfUnsecuredLines      0
age                                       0
NumberOfTime30-59DaysPastDueNotWorse      0
DebtRatio                                 0
MonthlyIncome                         29731
NumberOfOpenCreditLinesAndLoans           0
NumberOfTimes90DaysLate                   0
NumberRealEstateLoansOrLines              0
NumberOfTime60-89DaysPastDueNotWorse      0
NumberOfDependents                     3924
dtype: int64
```

29731 valeurs indéfinies dans l'attribut 'MonthlyIncome'.
3924 valeurs indéfinies dans l'attribut 'NumberOfDependents'

- Vérifions les valeurs doubles

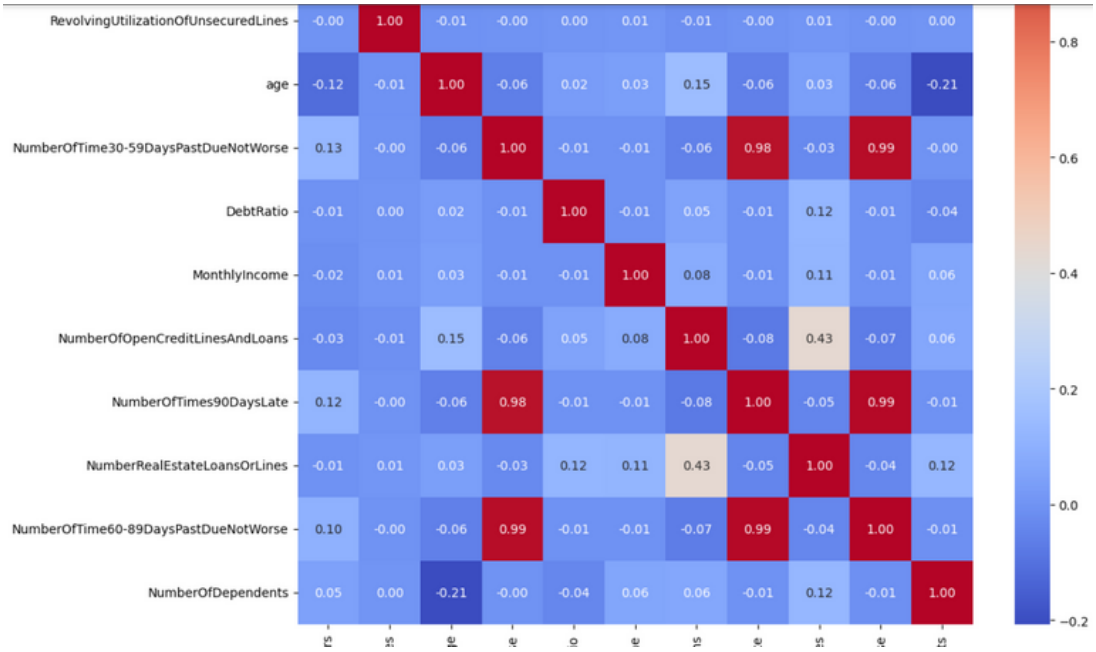
```
[31] duplicates = cell_df.duplicated()
     print("Number of duplicate rows:", duplicates.sum())

     # Display the duplicated rows (if any)
     if duplicates.any():
         duplicated_rows = cell_df[duplicates]

     Number of duplicate rows: 609
```

Nous avons 609 lignes en double

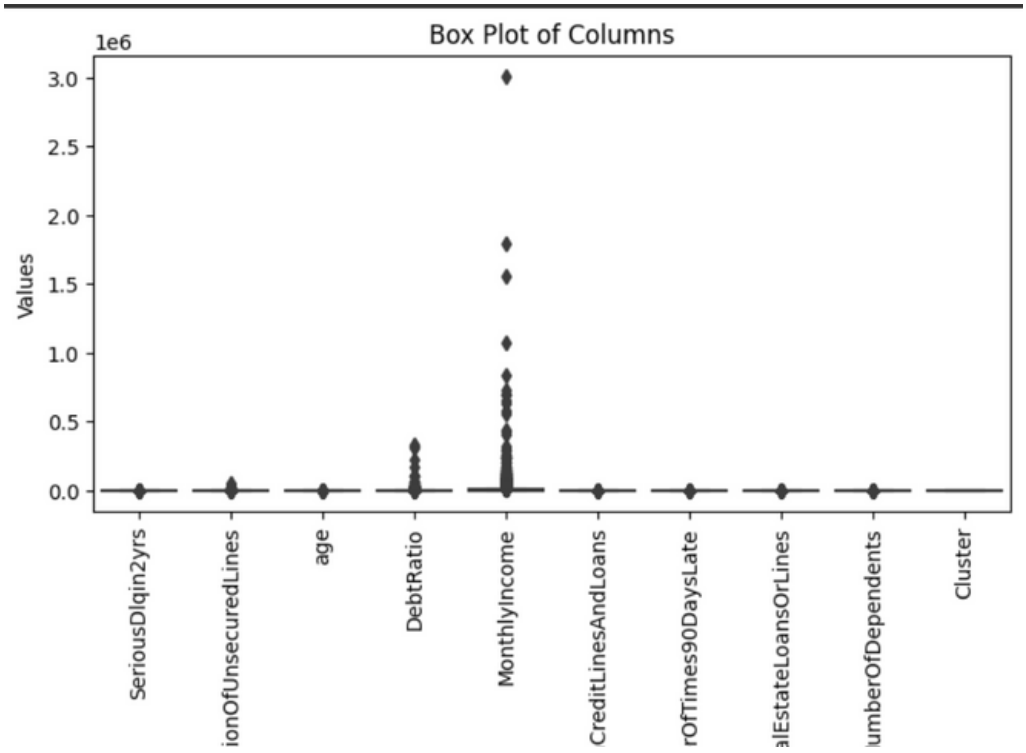
- A La recherche de corrélation



Ce heatmap permet de visualiser les corrélations entre les colonnes

Nous avons trois colonnes avec une valeur de corrélation supérieure à 0,05. Cela peut entraîner une redondance, affecter les modèles et potentiellement ses performances.

- Recherche des valeurs aberrantes



Nous avons quelques valeurs aberrantes dans 'MonthlyIncome', cela pourrait avoir un impact négatif sur notre modèle.

03

PREPARATION DES DONNÉES:

Nous allons prendre soin des valeurs NaN et les remplacer par la MOYENNE de leur attribut. MEAN est une mesure statistique qui représente la valeur moyenne d'un ensemble de nombres. Nous ne voulons pas supprimer les lignes car il est important de conserver les données des autres attributs.

```
cell_df.fillna(cell_df.mean(),inplace=True)
```

Nous avons 609 lignes en double. On va les supprimer

```
cell_df.drop_duplicates
```

Au niveau de corrélation, nous allons conserver 'NumberOfTimes90DaysLate' car elle a une période plus longue en termes de jours et nous supprimerons les deux autres.

```
cell_df.drop("NumberOfTime30-59DaysPastDueNotWorse", axis=1,inplace=True)
cell_df.drop("NumberOfTime60-89DaysPastDueNotWorse", axis=1,inplace=True)
```

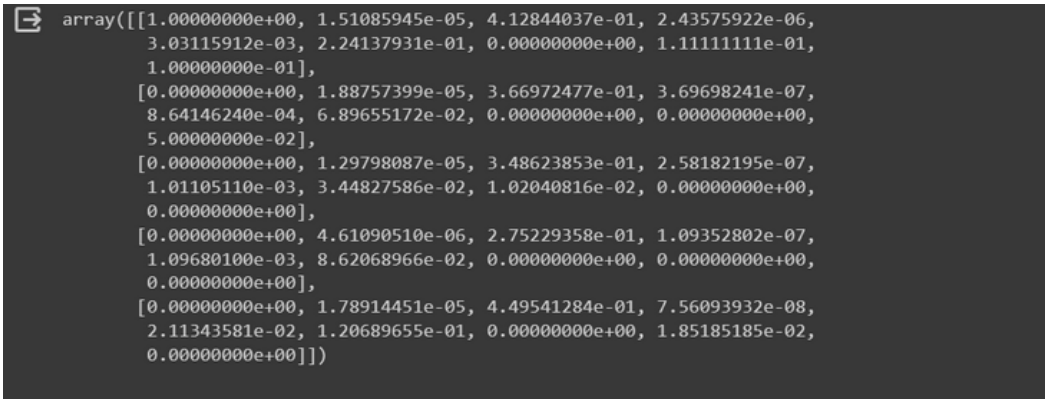
Pour les valeurs aberrantes, nous allons trier toutes nos valeurs de 'MonthlyIncome' et ne sélectionner que les quatre dernières, et les supprimer.

```
cell_df.sort_values(by = 'MonthlyIncome', ascending = False)[:4]
cell_df.drop(cell_df.index[[73764,137141,111366,50641]],inplace=True)
```

- Normalisation

Nous pouvons maintenant normaliser l'ensemble de données. Cela transforme les valeurs des attributs en mettant à l'échelle chaque valeur dans une plage donnée. Par défaut, cette plage est (0, 1).

Nous avons utilisé MinMaxScaler car il est mieux adapté à notre modèle (segmentation)



```
array([[1.00000000e+00, 1.51085945e-05, 4.12844037e-01, 2.43575922e-06,
        3.03115912e-03, 2.24137931e-01, 0.00000000e+00, 1.11111111e-01,
        1.00000000e-01],
       [0.00000000e+00, 1.88757399e-05, 3.66972477e-01, 3.69698241e-07,
        8.64146240e-04, 6.89655172e-02, 0.00000000e+00, 0.00000000e+00,
        5.00000000e-02],
       [0.00000000e+00, 1.29798087e-05, 3.48623853e-01, 2.58182195e-07,
        1.01105110e-03, 3.44827586e-02, 1.02040816e-02, 0.00000000e+00,
        0.00000000e+00],
       [0.00000000e+00, 4.61090510e-06, 2.75229358e-01, 1.09352802e-07,
        1.09680100e-03, 8.62068966e-02, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00],
       [0.00000000e+00, 1.78914451e-05, 4.49541284e-01, 7.56093932e-08,
        2.11343581e-02, 1.20689655e-01, 0.00000000e+00, 1.85185185e-02,
        0.00000000e+00]])
```

04

MODÉLISATION

Après avoir préparé et nettoyé nos données, il est maintenant temps de mettre en œuvre nos MODÈLES.

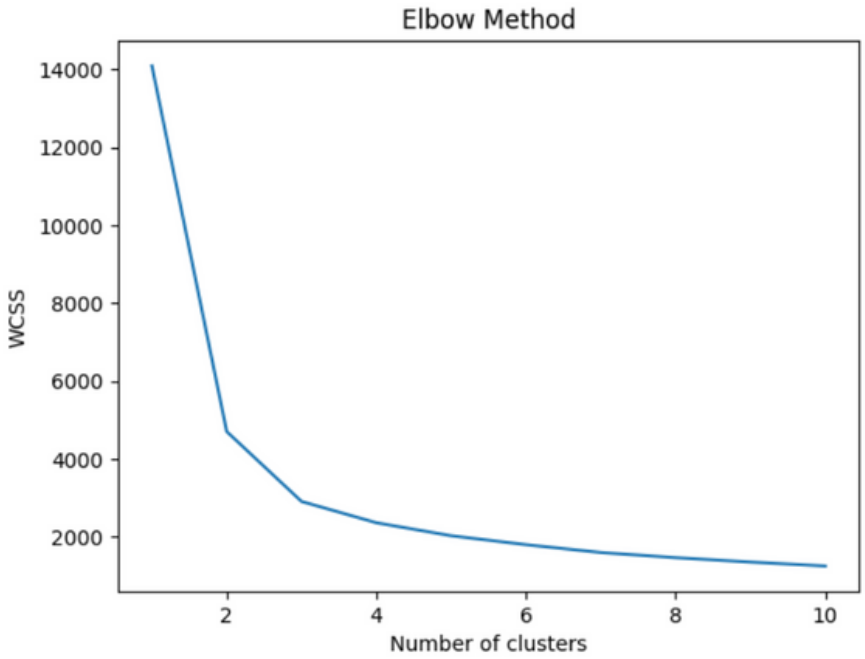
Nous allons appliquer deux :

1- K-Means Clustering

2- Agglomerative Hierarchical Clustering(CAH)

- **K-means**

K-means est un algorithme de regroupement qui divise les données en K clusters en attribuant itérativement des points au centroïde de cluster le plus proche et en mettant à jour les centroïdes en fonction de la moyenne des points attribués. Il poursuit ces étapes jusqu'à ce que les centres de cluster ne changent plus(convergence)
Pour déterminer le nombre optimal de clusters, nous allons utiliser la méthode du coude :



La méthode du coude est efficace et elle indique que nous avons besoin de 3 clusters.

- **application de Kmeans**

```
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10, random_state=0)
kmeans.fit(scaled_df)
```

Cluster
1
2
2
2
2
...
0

Nous avons divisé nos données en 3 clusters et ajouté une colonne nommée 'Cluster'.
La colonne 'Cluster' prendra les valeurs 0, 1 ou 2.

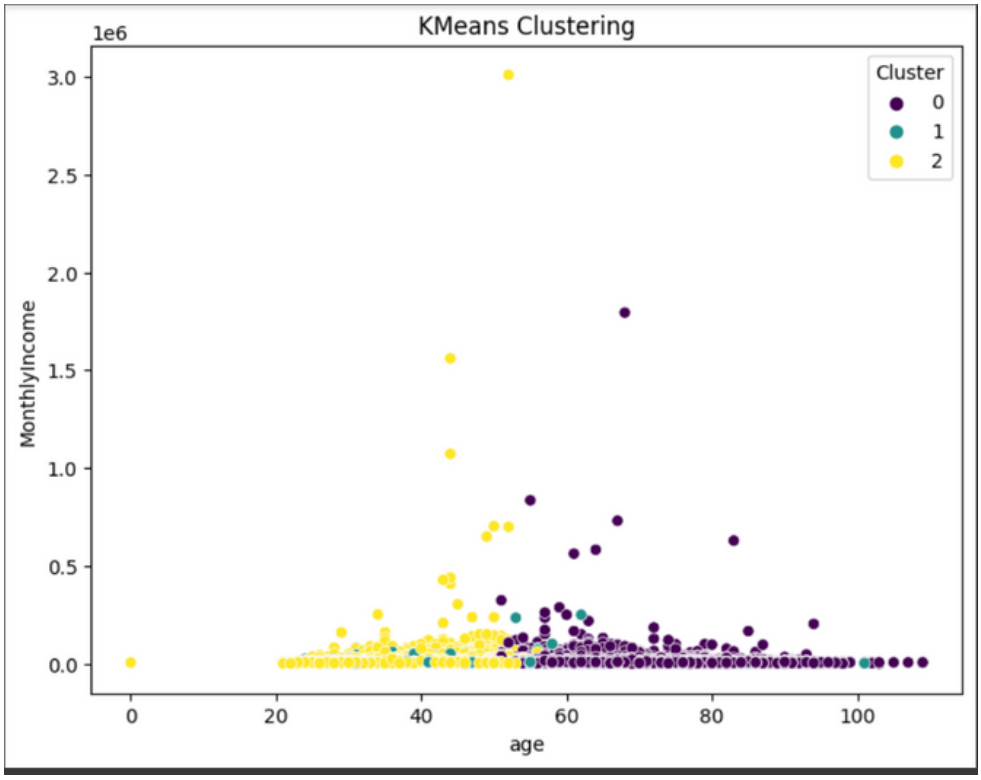
- **application du moyenne de chaque cluster**

```
cell_df.groupby('Cluster').mean()
```

Cluster	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	DebtRatio	MonthlyIncc
0	0.0	4.748628	65.190274	421.743421	7000.5579
1	1.0	4.367702	45.927880	295.150390	5803.9996
2	0.0	7.507818	41.025828	296.280610	6479.4814

les individus plus âgés peuvent avoir des revenus plus élevés mais ont également accumulé des dettes plus importantes au fil du temps. D'autre part, les jeunes adultes pourraient avoir des revenus plus bas mais gérer leur dette de manière plus prudente

- scatter plot avec 'âge' et 'revenu mensuel'.



	Cluster	SeriousDlqin2yrs	NumberOfOpenCreditLinesAndLoans
0	0	0.0	7.564708
1	1	1.0	7.882306
2	2	0.0	9.478103

Les clusters 0 et 2 representent des clients relativement stables sur le plan financier bien que le cluster 0 a un portefeuille financier plus diversifié que le cluster 2.

Le cluster 1 representent des clients à risque élevé à cause de leur historique de défaut de paiement

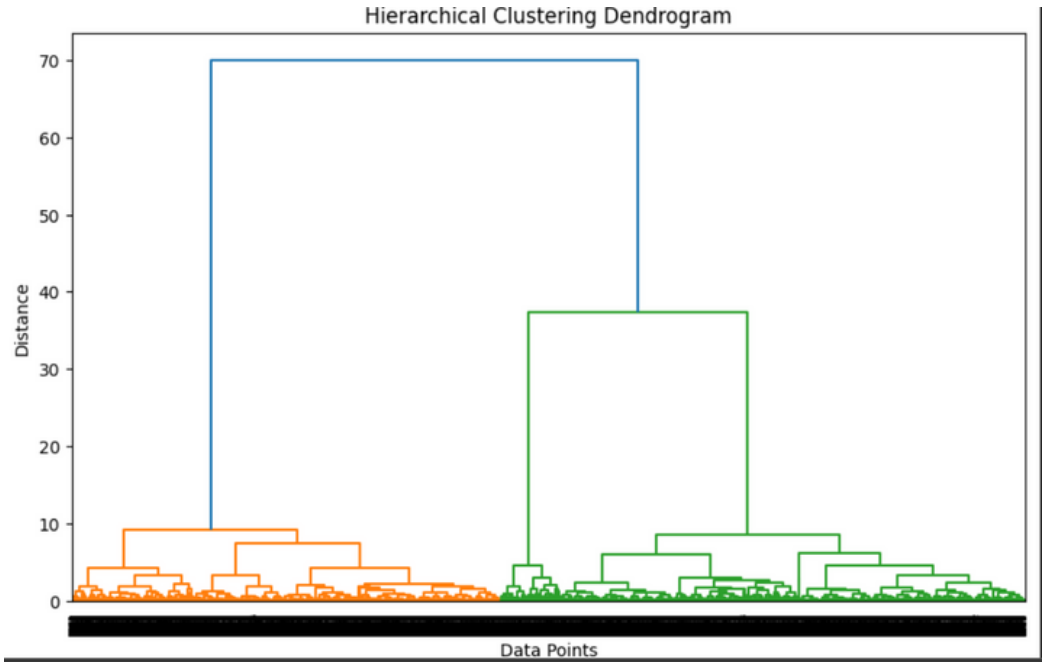
- # Agglomerative Hierarchical Clustering

CAH fusionne progressivement des points de données ou des clusters similaires étape par étape, formant une hiérarchie de clusters. Elle ne nécessite pas un nombre prédéfini de clusters, itérant jusqu'à ce qu'un critère d'arrêt soit atteint.

Nous allons travailler sur les variables : 'cell' au lieu de 'cell_df' et 'scaled_cell' au lieu de 'scaled_df' pour comparer les deux .

Dendrogramme :

hc = linkage(scaled_cell, method='ward')



D'après le dendrogramme, nous allons couper à un seuil de 10.

- **application du moyenne de chaque cluster**

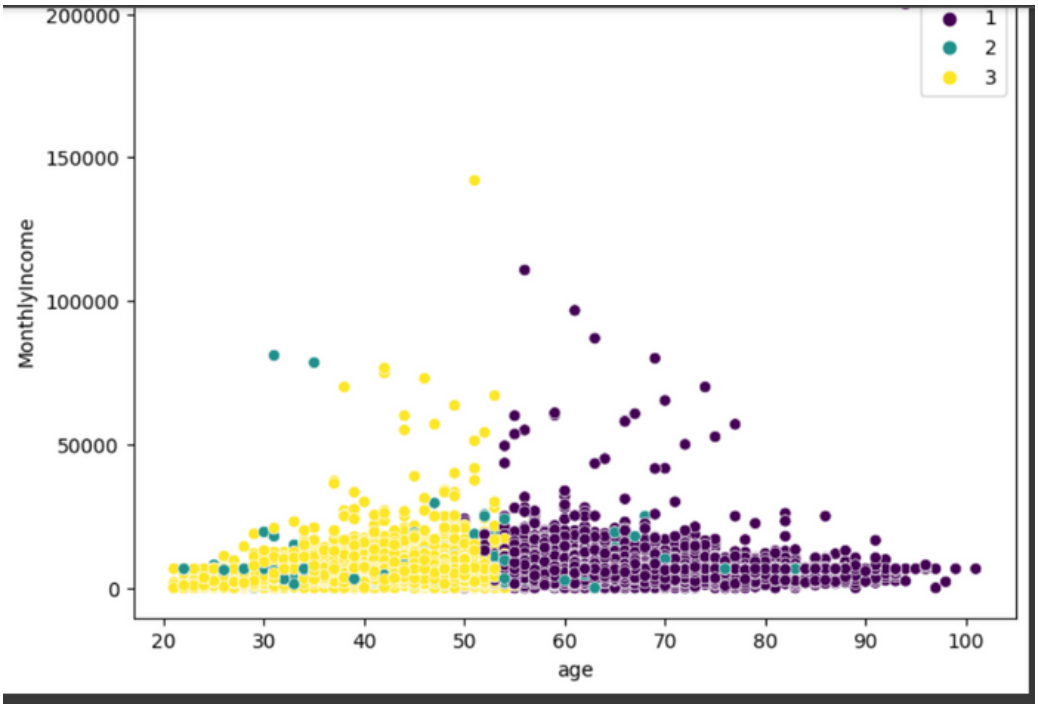
cell.groupby('Cluster').mean()

luster	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	DebtRatio	MonthlyIncom
0	0.0	3.705891	65.168075	435.086366	7005.34808
1	1.0	13.463222	45.834375	323.606336	5832.07220
2	0.0	4.941690	40.967092	275.698469	6365.12419

CAH nous donné aussi 3 clusters
Comme K-means, il semble y avoir un motif dans les colonnes 'age' et 'monthlyincome'

- Les adultes âgés ont un revenu mensuel élevé et un ratio de dette élevé.
- Les jeunes adultes ont un revenu mensuel plus faible et un ratio de dette plus faible.

--> deux clusters avec deux caracteristiques differèntes .



05

ÉVALUATION

Silhouette Score : (SI)
Utilisé en apprentissage non supervisé et qui évalue la qualité des clusters en mesurant la similarité des points de données à leur propre cluster par rapport aux autres clusters.
Plus le score est plus proche de 1 plus qui il est meilleur

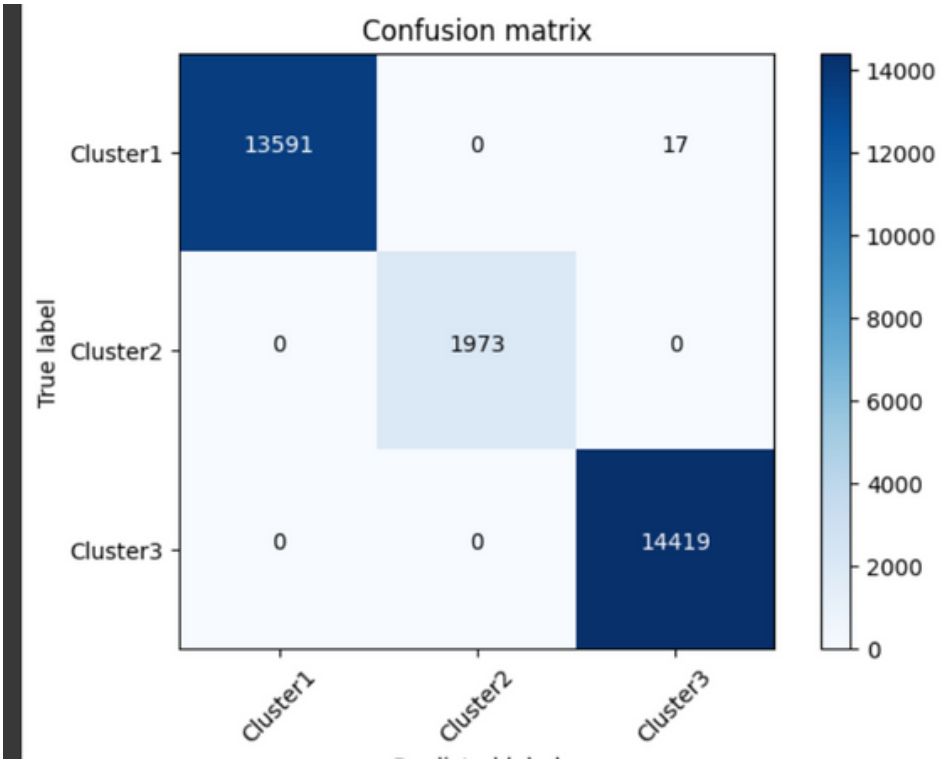
K-means	CAH
Silhouette Score: 0.39817425205159207	Silhouette Score: 0.7871846030206465

<u>---> Il semble que le modèle (CAH) a un meilleur score, ce qui signifie que les lignes appartiennent mieux à leur cluster que dans le cas de K-means.</u>

Le score bas de K-means est notable. Pour vérifier la précision de ses clusters, il est important de déterminer si K-means nous donne les clusters de manière cohérente à chaque utilisation.
Nous allons diviser nos données en TraintSet et TestSet, respectivement à 80 % et 20 %.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)

Puis, on va appliquer les différents fonctions du métrique comme f1 score et la matrcie de confusion



	precision	recall	f1-score	support
0	1.00	1.00	1.00	13608
1	1.00	1.00	1.00	1973
2	1.00	1.00	1.00	14419
accuracy			1.00	30000
macro avg	1.00	1.00	1.00	30000
weighted avg	1.00	1.00	1.00	30000

--> La matrice de confusion et le différents métriques montrent des bon résultats

- **Application de ACCURACY**

metrics.accuracy_score(y_train, kmeans.predict(X_train))
metrics.accuracy_score(y_test, yhat)

Train set Accuracy: 0.9995083169438981
Test set Accuracy: 0.9994333333333333

-->Une grande précision indique que le modèle K-means utilise les mêmes centroïdes en fin d'itération, ce qui donne le même regroupement à chaque utilisation.

Les deux modèles ont divisé nos données en 3 clusters, mais le modèle CAH est plus efficace en termes de proximité entre chaque donnée et son cluster.

06

DISCUSSION

Dans le cadre de notre étude, nous avons exploré deux modèles d'apprentissage non supervisé, à savoir l'Analyse Hiérarchique Ascendante (CAH) et l'algorithme de clustering (K-means). Ces approches de clustering sont des outils puissants permettant de découvrir des structures non explicitement définies dans des ensembles de données variés. Leur utilisation s'étend à différents domaines, notamment :

- **La segmentation de la clientèle :** on regroupe les clients afin de mieux adapterles produits et les offres .
- **Le regroupement de textes, de documents ou de résultats de recherche :** regroupement pour trouver des sujets dans un texte .
- **Le regroupement d'images ou compression d'images :** regroupe les images ou les couleurs similaires

Notre travail se situe dans le domaine de la segmentation de la clientèle:

- Dans notre cas , nous avons appliqué les deux modèles de Clustering pour segmenter les clients bancaires de notre "dataset": "Loans". L'utilisation de ces modèles de clustering vise à mieux comprendre les comportements et les caractéristiques des clients, ouvrant ainsi la voie à des stratégies plus personnalisées et efficaces.

- Dans le cadre du déploiement, le modèle de clustering est spécifiquement intégré à la gestion des risques.
- Cette approche est d'autant plus cruciale dans notre cas, où notre "dataset" comporte plusieurs colonnes indicatrices spécifiquement dédiées à la gestion des risques liés aux prêts.

➡ En exploitant ces informations, nous cherchons à affiner notre segmentation pour une gestion proactive des risques, permettant une allocation judicieuse des ressources et une identification rapide des potentiels risques émergents dans le domaine des prêts bancaires. En résumé, notre déploiement avec le modèle de clustering se concentre sur l'amélioration de la gestion des risques pour une meilleure prise de décision dans le domaine des prêts bancaires.

Exemple de déploiement de gestion des risques:

On a calculé certaines statistiques agrégées pour chaque cluster, telles que la moyenne de 'SeriousDlqin2yrs' (probabilité de défaut de paiement) et la moyenne de 'NumberOfOpenCreditLinesAndLoans' (nombre de lignes de crédit ouvertes et de prêts)

```
cluster_profile = cell_df.groupby('Cluster').agg({
    'SeriousDlqin2yrs': 'mean',
    'NumberOfOpenCreditLinesAndLoans': 'mean',
    # Ajoutez d'autres caractéristiques pertinentes
}).reset_index()

print(cluster_profile)
```

Interprétation:

	Cluster	SeriousDlqin2yrs	NumberOfOpenCreditLinesAndLoans
0	0	0.0	7.564708
1	1	1.0	7.882306
2	2	0.0	9.478103

Les clusters 0 et 2 representent des clients relativement stables sur le plan financier bien que le cluster 0 a un portefeuille financier plus diversifié que le cluster 2.

Le cluster 1 representent des clients à risque élevé à cause de leur hisrorique de défaut de paiement

Cette interprétation fournit une base solide pour la prise de décision en matière de gestion des risques. Les informations spécifiques aux clusters permettent une allocation de ressources plus précise et des stratégies de gestion des risques mieux adaptées à chaque segment client.