Etude et analyse de données
Campagne marketing d'une institution bancaire

Julien Tissier
Cyril Maillot
Yishuo Lyu
Zunzun Wang

Sommaire

Introduction

L'informatisation des systèmes a conduit à une explosion des données stockées. Que ce soit des données clients, de marché, de gestion ou de performances, il est devenu difficile pour une entreprise de se passer des principaux avantages que procurent l'analyse d'information.

Avec le développement des outils Big Data, les outils permettant d'extraire une valeur dans ces données se sont multipliés. C'est dans cette optique que nous avons réalisé cette étude. Nous avons utilisé les principaux outils d'analyse et de fouille de données afin de permettre à une banque de déterminer quels sont les critères influents sur la réponse d'une personne à une campagne marketing téléphonique.

Nous avons dans un premier temps fait une analyse des données clients, puis nous avons regroupé les clients dans différentes catégories (clustering). Nous avons ensuite utilisé les outils du Machine Learning afin de prédire la réponse d'une personne (régression et arbre décisionnel). Enfin, nous proposons une évolution du système d'information de la banque afin de traiter leur données de manières plus efficace (Hadoop).

Présentation du jeu de données

Les données misent à notre disposition proviennent d'une banque Portugaise. Elles concernent une campagne marketing téléphonique réalisée afin de savoir si le client allait souscrire à un plan d'épargne. Chaque client appelé a été enregistré avec les informations le concernant, ainsi que sa réponse suite à l'appel de la banque.

Les données à propos d'un client sont :

- l'âge
- le type de travail (parmi 12 catégories)
- la situation familiale (célibataire, marié ou divorcé)
- le niveau d'éducation (primary, secondary ou tertiary)
- est-ce que le client a des dettes ?
- le montant actuel dans le compte du client
- est-ce que le client a un prêt immobilier ?
- est-ce que le client a un crédit à la consommation ?
- comment le client a été contacté (téléphone fixe ou mobile)
- le mois du dernier contact avec le client
- le jour du dernier contact avec le client
- la durée de l'appel au client
- le nombre de contacts lors de cette campagne
- le nombre de jours depuis le contact de la campagne précédente
- le nombre de contacts lors de la campagne précédente
- la réponse à la campagne marketing précédente
- la réponse de la campagne actuelle

Au total, il y a 17 paramètres sur chacun des clients. Cependant, certains champs ne sont pas connus (valeur 'unknown').

Le jeu de données possède 45 211 enregistrements, allant de Mai 2008 à Novembre 2010.

Objectifs

A travers cette étude, nous avons analysé les données à notre disposition afin de déterminer les catégories de personnes qui ont répondu de manière positive, ainsi que les raisons de ce choix. Nous avons aussi pu voir une certaine similarité entre les personnes qui ont répondu oui, ce qui nous a permis de prédire la réponse d'un client avant que l'appel soit effectué.

Analyse statistique

Nous avons réalisé des statistiques sur les données afin de voir quelle est la répartition des clients de la banque, ainsi que la fréquence de chacune des catégories. Cette première étape nous a permis d'avoir une vue générale des différents types de clients.

Clustering

Après avoir examiné chaque catégorie successivement, nous avons regroupé les clients ayant des attributs similaires. Cette étape nous a permis de connaître les tendances au sein de l'ensemble des clients.

Prédiction

Notre objectif dans cette étape est de prédire la réponse d'un client avant de l'appeler. Pour cela, deux approches ont été utilisées :

- une approche poussée, qui utilise l'ensemble des données afin de prédire de manière exacte la réponse d'un client
- une approche plus simple, ne se basant que sur les attributs ayant la plus forte influence. Ce modèle permet d'avoir une idée de la réponse du client en prenant en compte que 3 attributs.

Perspectives d'évolution

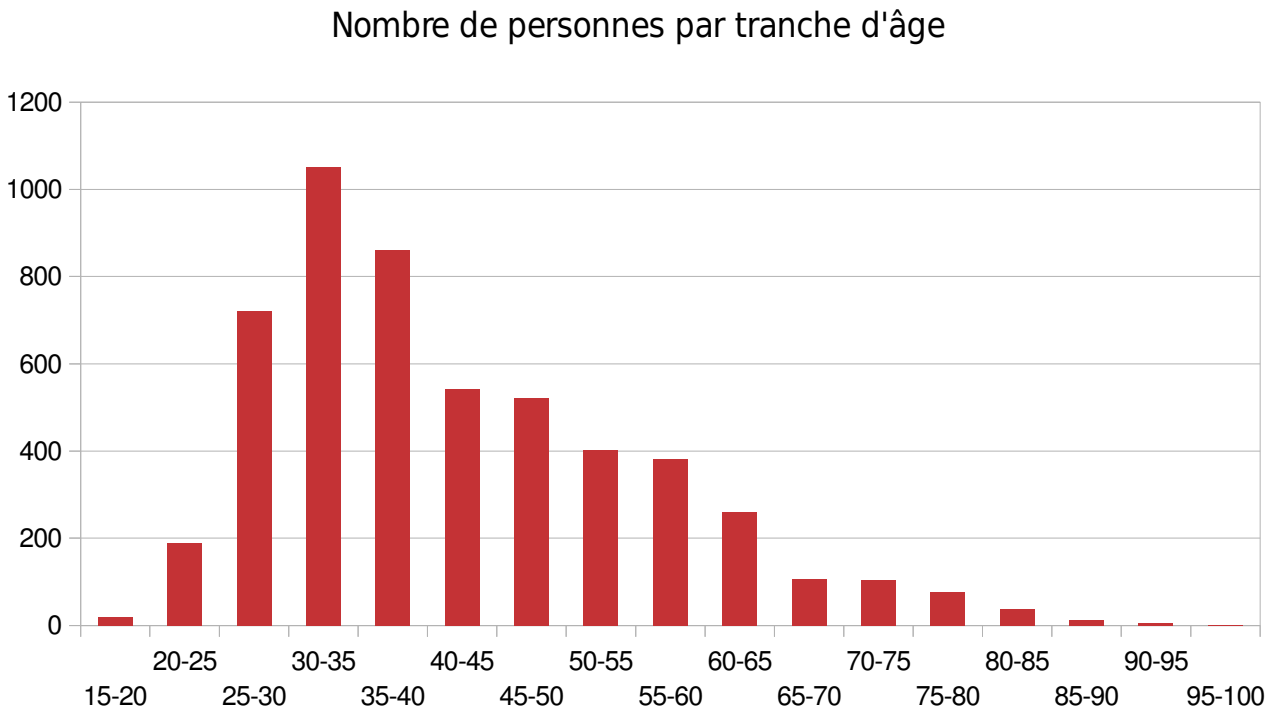
Afin de traiter les données clients de manières plus efficaces, la banque peut se tourner vers une solution Big Data. Nous détaillons dans cette étape la procédure à adopter, ainsi que les performances apportées par un tel système.

Analyse Statistique

Outils utilisés

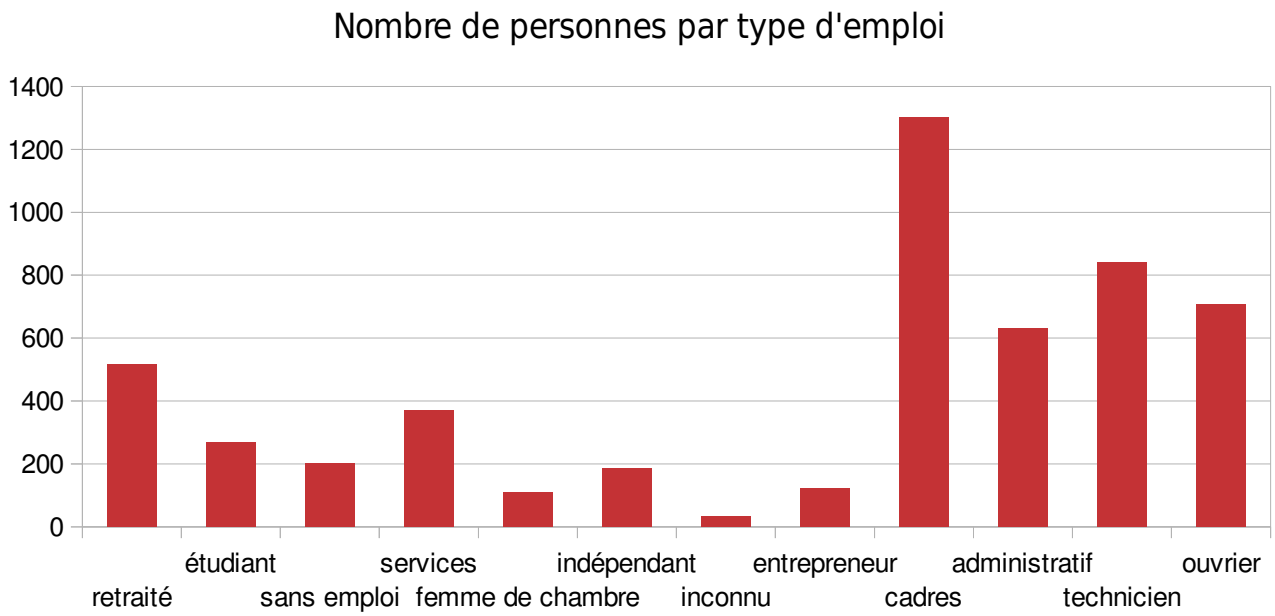
Dans cette partie, nous avons écrit un programme en Python3, qui calcule la fréquence de chacune des catégories. Nous avons d'abord séparé chaque attribut du fichier client, afin de les placer dans une structure de données qui nous permet d'effectuer des traitements. Nous avons ensuite compté le nombre d'occurrence de chacune des catégories.

Age



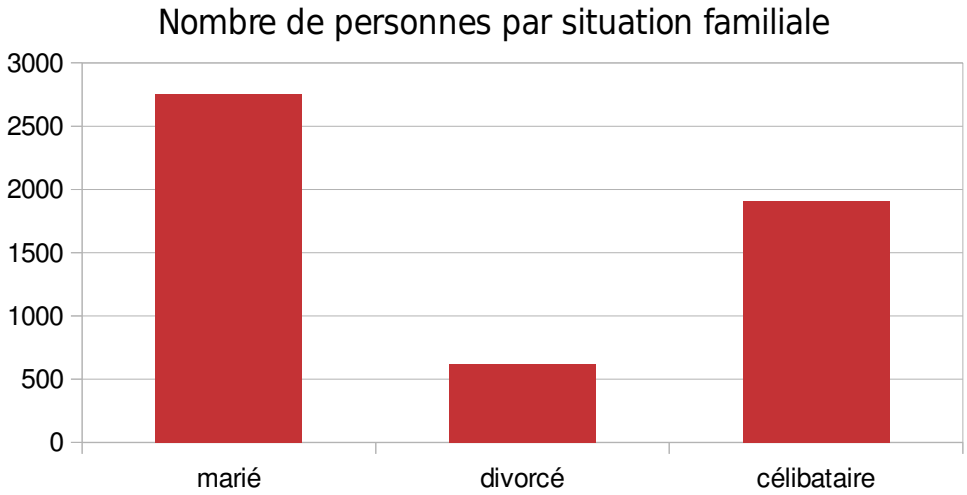
Parmi les clients de la banque qui ont répondu positivement, la catégorie la plus présente est celle des 30-35 ans. Ceci nous paraît logique, puisque c'est dans cette tranche d'âge en général que les personnes trouvent un emploi stable, et réalisent l'achat d'un bien immobilier. Ce sont deux facteurs qui favorisent la souscription d'un plan d'épargne. La moyenne d'âge de ces clients est de 41,7 ans.

Emploi



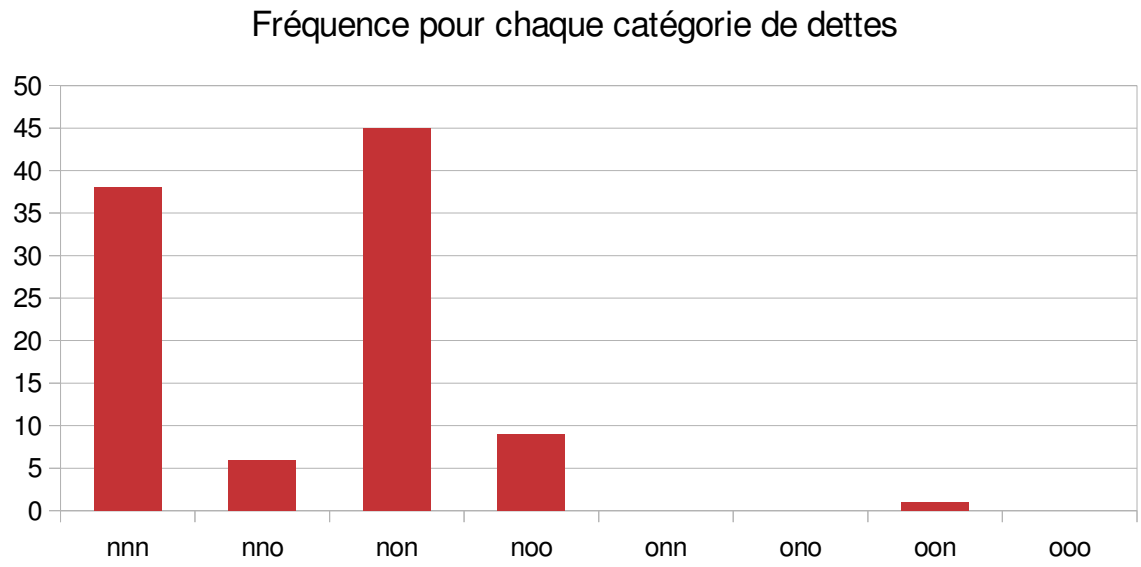
Parmi les clients ayant répondu positivement, la catégorie la plus présente est celle des cadres (25%). Cela est compréhensible car ce sont souvent ces personnes qui ont les salaires les plus importants. Les catégories technicien (16%), ouvrier (13%) et administratif (12%) sont aussi très présentes. De manière générale, cette étude nous montre que les personnes ayant un plan d'épargne sont celles qui ont une situation professionnelle stable, à l'inverse des entrepreneurs et des indépendants.

Situation familiale



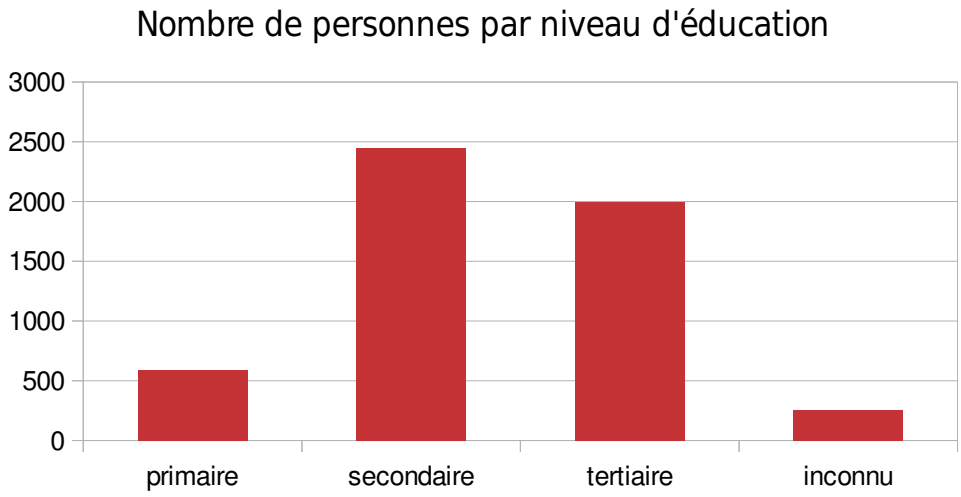
Parmi les clients ayant répondu postivement, les personnes mariées sont les plus présentes. Cela semble logique puisque les ressources budgétaires sont couplées à celles de son conjoint, ce qui facilite la souscription à un plan d'épargne.

Dettes



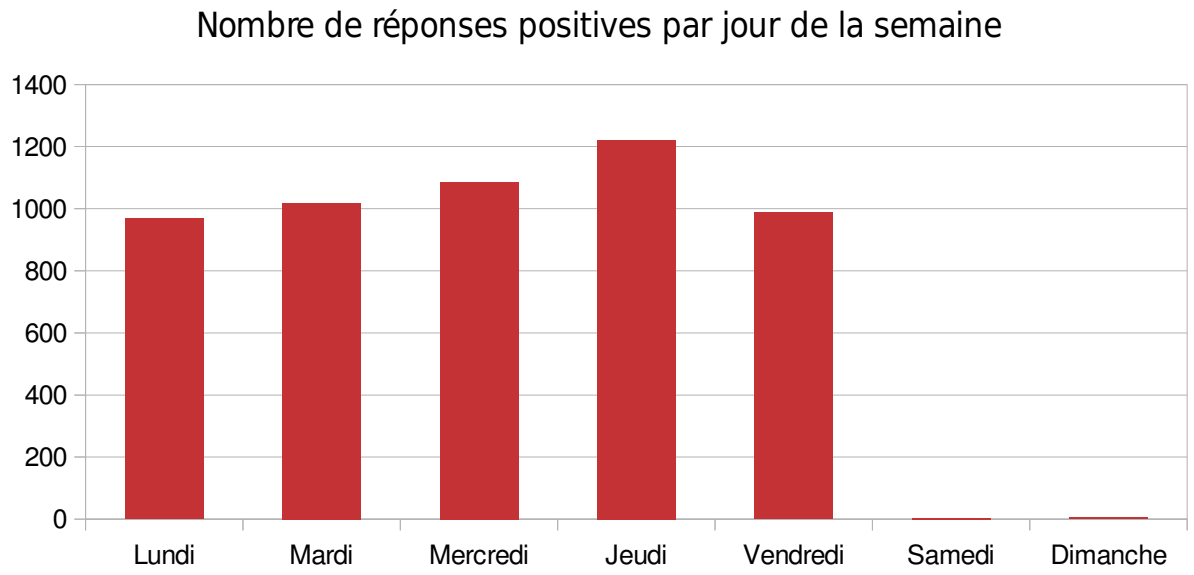
Nous avons jugé pertinent de regrouper les attributs Dettes, Prêt Immobilier et Crédit à la consommation. Chaque catégorie dans le diagramme est un mot de 3 lettres. La première lettre indique si le client a une dette (**o**ui ou **n**on), la seconde s'il a un prêt immobilier et la troisième s'il a un crédit à la consommation. Parmi les clients ayant répondu oui, ceux n'ayant ni de dette, ni de crédit à la consommation, représentent 83%. On remarque aussi que parmi les personnes ayant des dettes (les quatre barres sur la droite du diagramme), seule 1% a souscrit à un plan d'épargne. Cela semble logique, puisque les personnes avec des soucis d'argent ne sont pas celles qui épargnent.

Niveau d'éducation



Les personnes ayant un niveau d'éducation primaire sont les moins présentes (11%). On peut supposer que ces personnes ont un salaire moins important que les autres catégories, ce qui les fait renoncer à un plan d'épargne.

Jour de la semaine



Le jeu de données ne possède pas le jour de la semaine ni l'année où l'enregistrement a été fait. En revanche, nous savons que le premier enregistrement date de Mai 2008, et le dernier de Novembre 2010. Nous avons ainsi pu déterminer dans notre programme la date de chacun des enregistrements (puisque les enregistrements sont classés par date, on sait que l'on change d'année lorsque le mois passe de Décembre à Janvier). Cela nous a donc permis de savoir le jour de la semaine de l'enregistrement. Nous avons comptabilisé le nombre de fois où une réponse positive se produisait le Lundi, puis le Mardi...

Dans le diagramme ci-dessus, nous pouvons voir qu'il n'y a pas de différences notables entre les jours, tous sont environ égaux. En revanche, nous avons trouvé quelques réponses positives le Samedi le Dimanche, alors que les banques sont fermées ces jours-là. Nous supposons qu'il s'agit de données erronées.

Conclusion

Cette première étude statistique des données nous a permis de dresser un profil type du client susceptible de souscrire à un plan d'épargne :

- la personne a entre 30 et 40 ans
- la personne est cadre
- la personne a un niveau d'étude au moins supérieur au secondaire
- la personne est mariée
- la personne n'a pas de dettes, mais a éventuellement un crédit immobilier

Nous avons constaté que le taux de succès actuel est de 11.7 %. Nous conseillons à la banque de cibler ses prochaines campagnes marketing sur les personnes dont le profil se rapproche de celui-ci. La banque peut aussi attirer de nouveaux clients qui possèdent ces caractéristiques.

A présent, nous allons regrouper les clients qui partagent certains traits communs dans des groupes, afin de dégager des tendances.

Clustering

Outils utilisés

Nous avons utilisé des bibliothèques de calculs scientifiques, reposant sur le langage Python. La bibliothèque principale, celle contenant les algorithmes de Clustering, est scikit-learn (version 0.16 dans notre étude). Elle utilise aussi les bibliothèques numpy (version 1.8.2), et scipy (version 0.14).

Formatage des données

Nous avons dû modifier les données afin de les rendre exploitables. La première étape dans la transformation des données était de déterminer l'année et le jour de la semaine de chacun des enregistrements. Cette méthode est similaire à celle de la partie précédente (lors de l'analyse statistique). Nous avons ajouté ces infomations à chaque ligne du fichier bank-full.csv. Ici, nous pouvons constater la présence des attributs “2008” et “mon” en fin de ligne.

```
58;"management";"married";"tertiary";"no";2143;"yes";"no";"unknown";5;"may";261;1;-1;0;"unknown";"no";"2008";"mon"
```

Puis, nous avons dû formater les données. Pour chacun des attributs, nous lui avons assigné une valeur selon les tables de correspondance suivantes :

```
jobs = { "admin.":1, "blue-collar":2, "entrepreneur":3,
         "housemaid":4, "management":5, "retired":6,
         "self-employed":7, "services":8, "student":9,
         "technician":10, "unemployed":11, "unknown":12,
       }
status = { "divorced":1, "married":2, "single":3, "unknown":4 }
levels = { "primary":1, "secondary":2, "tertiary":3, "unknown":4 }
answers = { "no":1, "yes":2 }
contacts = { "unknown":1, "cellular":2, "telephone":3 }
outcomes = { "failure":1, "success":2, "other":3, "unknown":4 }
days = { 'mon':1, 'tue':2, 'wed':3, 'thu':4, 'fri':5, 'sat':6,
         'sun':7 }
```

Nous avons converti les champs ayant une valeur numérique en entier. Au final, chacun des enregistrements est devenu un vecteur d'un espace à 17 dimensions (les 17 attributs de départ, auxquels les attributs 'jour' et 'mois' ont été remplacé par 'année' et 'jour de la semaine'). Ci-dessous, un exemple d'enregistrement avant et après le formatage.

```
58;"management";"married";"tertiary";"no";2143;"yes";"no";"unknown";5;"may";261;1;-1;0;"unknown";"no";"2008";"mon"
```

```
58 5 2 3 1 2 1 2143 1 261 1 -1 0 4 1 2008 1
```

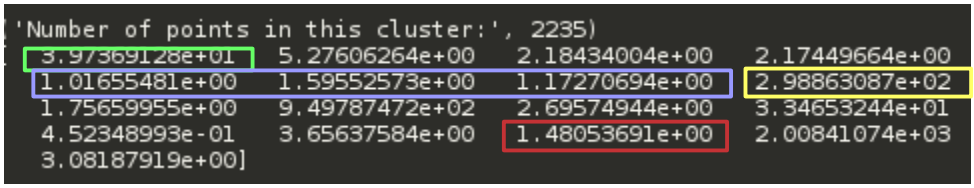
A présent, nous pouvons appliquer les algorithmes de Clustering de scikit-learn. L'algorithme utilisé est celui du KMean. Le fichier sur lequel nous avons travaillé est celui de 45 211 enregistrements.

Résultats

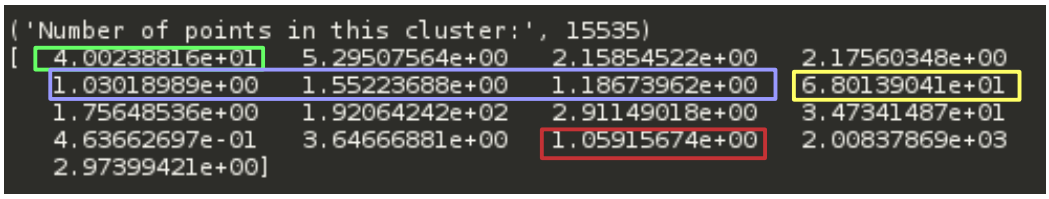
Le Clustering possède deux modes de fonctionnement : le premier consiste à indiquer le nombre de groupes (clusters) désirés. Le second consiste à laisser la machine décider de manière autonome le nombre de clusters.

Avec la méthode autonome, la machine trouve 10 clusters. Cependant, en analysant les résultats, nous nous apercevons que ce découpage n'est pas optimal, puisque la plupart des centres des clusters ont des valeurs similaires. Le clustering n'est pas pertinent dans ce cas, puisqu'il fait un travail de moyenne plutôt que de séparation.

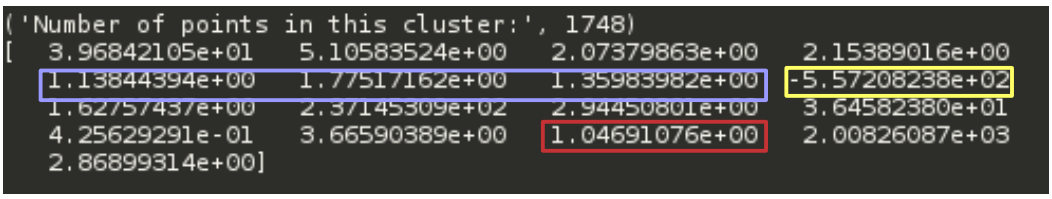
Nous avons ensuite indiqué nous-même le nombre de clusters. Avec 5, 10 et 15 clusters, les résultats ne sont pas interprétables. Lorsque nous passons le nombre de clusters à 20, des groupes au comportement similaire se dégagent. Voici les principaux clusters.



Ce cluster contient 2235 personnes. En ayant assigné la valeur 1 à 'non', et 2 à 'oui', il est aisé de voir que le taux de succès à l'intérieur de ce cluster est de 48 % (cadre rouge ci-dessus). Nous avons donc un groupe dont la moyenne de réponses positives est largement supérieure à la moyenne de l'ensemble des données (11.7 %). Parmi ces personnes, la moyenne d'âge est de 39.7 ans (cadre vert), ce qui est plus jeune que l'ensemble des clients. Avec la même logique, on peut constater que dans ce groupe, seul 1 % a des dettes, 59 % a un prêt immobilier, et 17 % a un crédit à la consommation (cadre bleu). De plus, ces personnes ont en moyenne 298€ (cadre jaune) dans leur compte en banque. Ces données sont en accord avec celles de l'analyse statistique. On constate que ce groupe, dont le taux de succès est de 48 %, a un profil qui se rapproche du profil type élaboré dans la partie précédente.



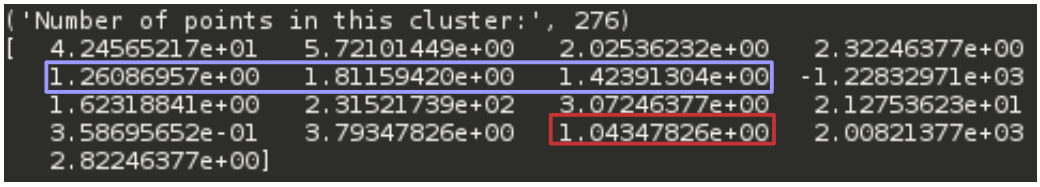
Ce cluster contient 15 535 personnes. Le taux de succès n'est que de 5.9 %, ce qui est inférieur à la moyenne du groupe (11.7 %). La moyenne d' âge est de 40 ans. Dans ce groupe, 3 % ont des dettes, 55 % ont un prêt immobilier et 18 % ont un crédit à la consommation. Nous constatons que ce cluster est similaire au précédent. Cependant, ces personnes ont en moyenne 68€ dans leur compte en banque, ce qui est inférieur aux 298€ du précédent cluster. Nous supposons que les personnes avec un faible montant en banque ne souscrivent pas à un plan d'épargne.



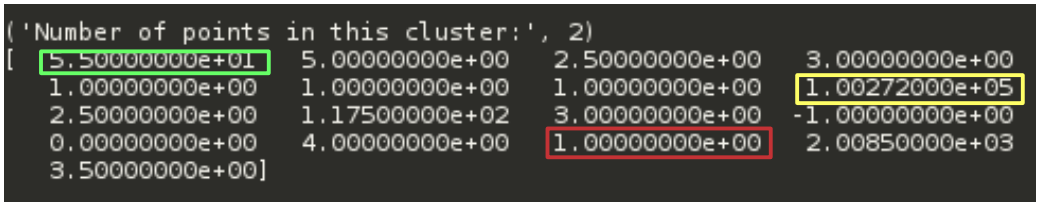
Le cluster ci-dessus confirme cette hypothèse. Ici, le taux de succès n'est que de 4.6 % (c'est le moins élevé parmi tous les clusters), et les personnes de ce groupe ont en moyenne -557€ dans leur compte en banque (ils sont à découvert). De plus, 13 % des personnes ont des dettes (alors que les clusters précédents n'avaient que 2 ou 3 %). Nous en concluons que les personnes en difficulté financières ne souscrivent pas à un plan d''épargne.

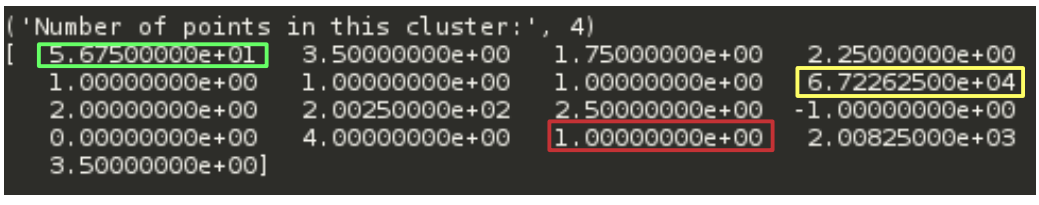
Nous constatons aussi que 35 % des personnes de ce groupe ont un crédit à la consommation, alors que dans les autres clusters, ce chiffre est compris entre 5 % et 20 %. Cette différence de valeur explique aussi le faible taux de succès de ce cluster.

Nous avons ensuite augmenté le nombre de clusters à 30, afin d'avoir un découpage plus précis des clients de la banque.



Ce cluster composé de 276 personnes confirme ce qui a été dit avant. En effet, dans ce cluster, le taux de succès n'est que de 4.3 %, et ces personnes ont en moyenne -1228€ dans leur compte en banque. Le taux de personnes endettées est de 26 %, et 42 % des personnes ont un crédit à la consommation. Ce cluster montre que certains clients de la banque ont d'importantes difficultés financières. De plus, avec 81 % des personnes ayant un prêt immobilier, nous conseillons à la banque de ne pas proposer un plan d'épargne à ce type de client, mais plutôt de les accompagner dans le remboursement de leurs crédits.





Les deux clusters ci-dessus nous indiquent qu'il existe une catégorie de personnes qui ne souscrivent pas à un plan d'épargne ; dans les deux cas, le taux de succès est de 0 %. Ces

personnes sont agées (55 ans en moyenne), mais possèdent un fort patrimoine économique, avec au moins 60 000€ sur leur compte en banque. Le faible nombre de personnes dans ces clusters nous empêche de savoir s'il s'agit d'une tendance ou de cas isolés. Un jeu de données plus important nous permettrait de résoudre ce problème.

Conclusion

Cette étude sur le Clustering nous a permis de regrouper les personnes à caractères similaires :

- les personnes dont le taux de succès est élevé sont celles qui n'ont pas de dettes, et ont un solde en banque environ égal à 250€.
- les personnes en défaut de paiement, avec des dettes et un compte à découvert, ne souscrivent pas à un plan d'épargne.
- les personnes âgées avec un compte en banque élevé ne souscrivent pas à un plan d'épargne. Ceci n'est qu'une hypothèse, qui peut être confirmée avec davantage de données.

Nous conseillons à la banque de :

- accentuer leur campagne téléphonique sur les personnes en situation régulière, et avec une somme d'argent modeste (de l'ordre de quelques centaines d'euros sur leur compte en banque).
- ne pas démarcher les personnes en défaut de paiement, mais plutôt de les accompagner à rembourser leur crédit.
- ne pas démarcher les personnes très aisées.

Prédiction

Régression

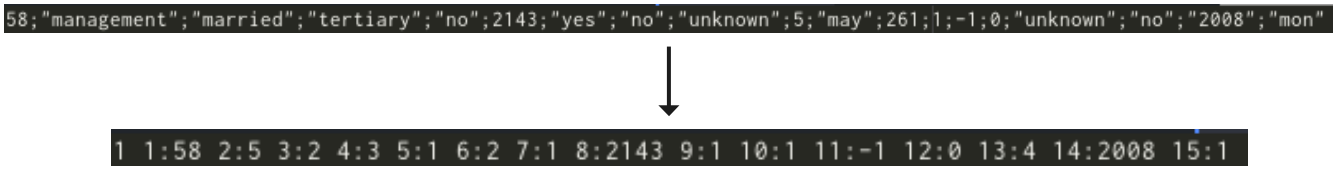
Outils utilisés

Nous avons utilisé la bibliothèque libsvm (version 3.20) pour faire la régression ainsi que le langage Python pour formater les données.

Formatage des données

A l'aide d'un script Python, nous avons utilisé les algorithmes précédents pour déterminer l'année et le jour de la semaine de chacun des enregistrements, ainsi que pour donner une valeur numérique à chacun des attributs. Nous avons ensuite supprimé l'attribut 'duration'. En effet, si nous l'avions laissé, le résultat aurait été faussé, puisque que lorsque la durée de l'appel est de 0, celui-ci est un échec.

Puis nous avons formaté la ligne pour qu'elle soit exploitable par libsvm. Nous avons placé le label en début de ligne (la réponse du client), et pour chacun des autres attributs, nous l'avons précédé d'un index. Ce qui nous donne la transformation suivante :



Notre vecteur comporte 15 attributs. Nous avons utilisé le fichier avec 45 211 enregistrements. Nous avons séparé ce fichier en deux sous fichiers. Le premier fichier nous sert à faire l'apprentissage. Nous avons sélectionné un vecteur sur 5, ce qui fait 9043 vecteurs. Le deuxième est le fichier de test. C'est sur celui-ci que nous évaluons le modèle créée lors de l'apprentissage. Il contient l'ensemble des vecteurs qui ne sont pas dans le fichier d'apprentissage, ce qui représente 36 168 vecteurs.

Nous avons ensuite normalisé chacun des vecteurs, afin que les attributs soient compris entre -1 et 1. L'outil svm-scale de la bibliothèque libsvm nous a permis de réaliser cette tâche. Après normalisation, nos vecteurs sont semblables à celui-ci:

```
1 1:0.0540541 2:-0.272727 4:0.333333 5:-1 6:1 7:-1 8:-0.876095 9:-1 10:-1 11:-1 12:-1 13:1 14:-1 15:-1
```

A présent nous pouvons utiliser les outils de libsvm afin d'établir un modèle et de prédire les réponses des clients.

Résultats

Nous avons effectué de multiples tests afin de déterminer les paramètres optimaux à utiliser pour définir le modèle. Le meilleur modèle nous permet d'obtenir un taux de prédiction correct de 89.1 %.

```
~/master/SVM » ./libsvm-3.20/svm-train -t 2 -c 1 -g 0.75 -q client.scale.tr client.model
~/master/SVM » ./libsvm-3.20/svm-predict client.scale.t client.model result
Accuracy = 89.1313% (32237/36168) (classification)
```

Conclusion

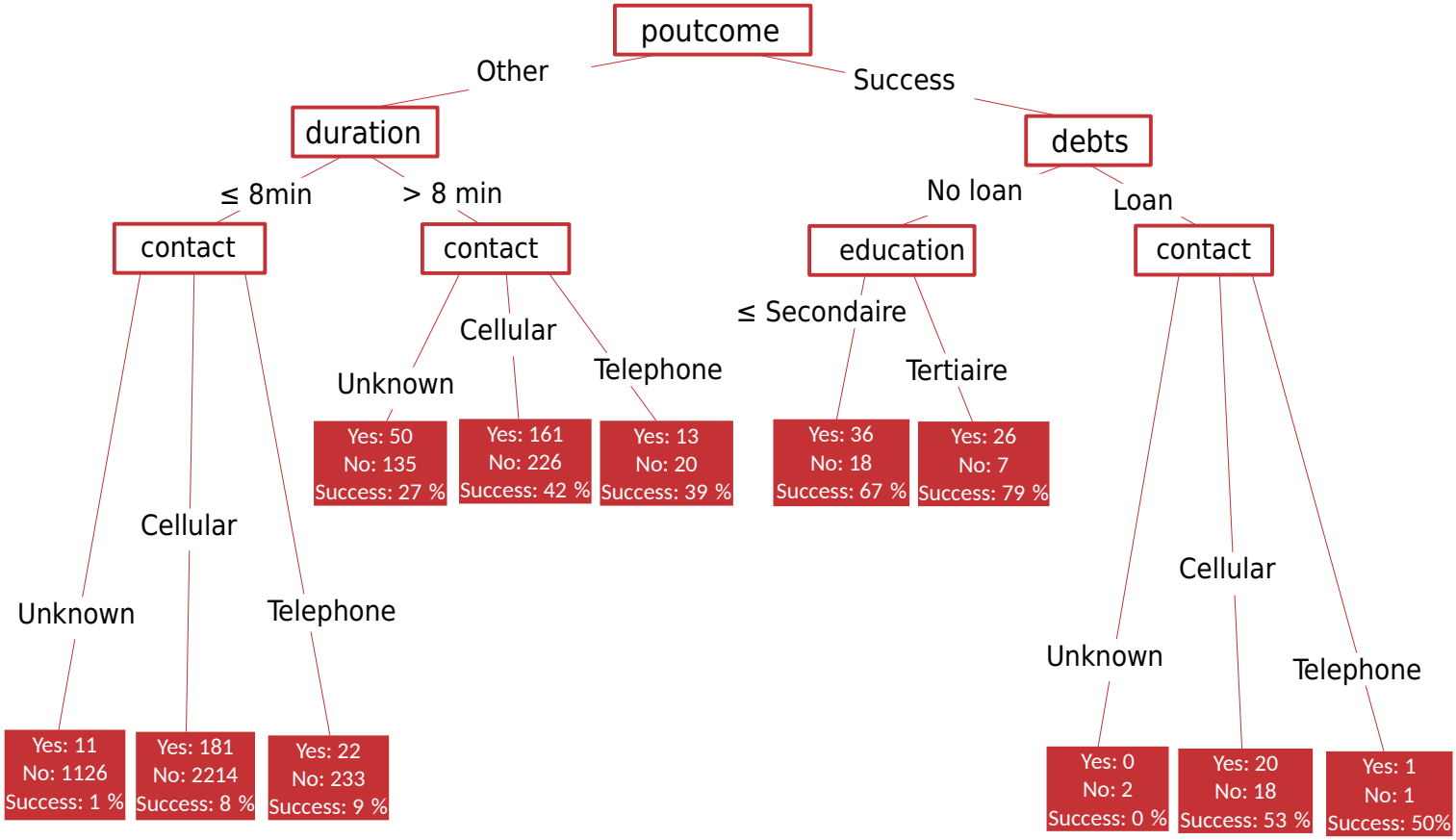
Avec ces outils, la banque peut prédire dans près de neuf cas sur dix la réponse d'un client avant de l'avoir appelé. Cette méthode de prédiction est précise, et permet de ne cibler que les personnes qui sont susceptibles de souscrire à un plan d'épargne.

Arbre décisionnel

Outils utilisés

Pour réaliser l'arbre décisionnel, nous avons implémenté l'algorithme dans le langage Python3.

Résultats



Avec l'arbre décisionnel, la banque peut avoir une estimation de la réponse du client en ne se basant que sur quelques attributs. Nous constatons que :

- les personnes ayant déjà souscrit à un plan d'épargne lors de la campagne précédente ont un taux de succès plus élevé. Et dans le cas où ces personnes n'ont pas de dettes, le taux de succès est d'environ 70 %.
- les personnes qui n'ont pas souscrit à un plan d'épargne lors de la campagne précédente ont un taux de succès plus élevé lorsque la durée de l'appel est supérieure à 8min.

Conclusion

L'arbre décisionnel est un moyen rapide d'estimer la réponse du client puisqu'il suffit de ne connaître que quelques attributs, mais cela reste peu précis, comparé à la prédiction par régression.

Perspectives d'évolution

La banque disposant de plusieurs agences, l'utilisation d'une architecture distribuée et évolutive augmenterait l'efficacité de l'exploitation des données.

Une des solutions actuelles est Hadoop. Elle fournit les outils nécessaires afin de mettre en place une application distribuée, et permet le stockage d'un très grand nombre de données.

En pratique, des nœuds de données sont installés dans les agences et un nœud principal dans l'une des agences. Les nœuds de données stockent les données, tandis que le nœud principal gère l'ensemble des autres nœuds. Ce dernier indexe les données et permet de voir en temps réel l'évolution des informations au sein des agences.

Nous avons expérimenté cette architecture distribuée sur un ordinateur avec la base de données HBase. Nous avons simulé une seule agence se connectant sur le nœud principal pour accéder à la base de données.

Afin de mettre en place la solution Hadoop, les outils suivants ont été installés :

- **Hadoop Distributed File System (HDFS) :** système de fichiers distribué stockant d'importants volumes de données sur plusieurs machines. D'un point de vue utilisateur, les données sont vues comme étant sur un seul disque.
- **Hadoop YARN :** gère la planification des tâches et les ressources des divers clusters.
- **Hadoop Common :** permet le déploiement de modules Hadoop.
- **Zookeeper :** permet la coordination des services pour les applications distribuées.
- **Hbase :** base de données distribuée et évolutive orientée dans la sauvegarde de données à grande échelle.

Nous avons configuré ces outils (voir annexes), afin qu'ils interagissent ensemble. Les différents services qui tournent sur notre ordinateur de test sont les suivants (les numéros sont les PID de chacun des services) :

```
tsuna@tsuna-VirtualBox ~/Hadoop/hadoop-2.7.1 $ jps
6349 HMaster
3119 SecondaryNameNode
5861 HQuorumPeer
3604 NodeManager
6443 HRegionServer
3727 QuorumPeerMain
11519 Jps
7615 ThriftServer
3305 ResourceManager
2942 DataNode
2805 NameNode
```

Le noeud principal (*NameNode*) ainsi que le noeud de données (*DataNode*) sont en service, ainsi que les outils permettant de gérer les nœuds (*NodeManager*) et les ressources (*ResourceManager*).

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
localhost:50010 (127.0.0.1:50010)	1	In Service	25.47 GB	27.09 MB	15.61 GB	9.83 GB	20	27.09 MB (0.1%)	0	2.7.1

A l'aide d'un script Python, nous avons mesuré le temps de lecture des données depuis la base de données et depuis le fichier texte.

```
tsuna@tsuna-VirtualBox ~/Hadoop/Projet Data Mining/bank $ python readDataTest.py
Temps de lecture a partir du fichier : 0:00:00.034447
Temps de lecture a partir de HBase   : 0:00:10.155973
```

La lecture depuis le fichier texte met moins de 0.04 secondes, tandis que depuis la base de données, le temps d'accès est de 10 secondes. Cette comparaison défavorable à Hadoop est à relativiser. Hadoop délivre ses meilleures performances uniquement sur une importante base de données et une architecture distribuée. Nous avons testé cette solution sur une machine virtuelle, ce qui peut aussi expliquer le long temps d'accès aux données.

Hadoop permet aussi d'intégrer des modules d'analyses de données. Ces modules utilisent les algorithmes du Machine Learning et d'optimisation afin de traiter les données en temps réel. Mahout est un de ces modules. Il permet le data mining et l'apprentissage automatique des données. Mahout peut être utilisé sans Hadoop, mais il délivre des meilleures performances lorsqu'il est couplé à une architecture Hadoop.

L'architecture distribuée réplique les données sur plusieurs nœuds afin d'avoir une redondance des données. Cela lui confère de nombreux avantages par rapport à une architecture centralisée :

- le temps d'accès aux données est réduit
- il n'y a pas de latence en cas d'accès simultané aux données
- le traitement est plus rapide puisque les calculs peuvent être faits en parallèle

Annexes :

```
<configuration>
....<property>
........<name>fs.defaultFS</name>
........<value>hdfs://localhost:54310</value>
....</property>
</configuration>
```

Configuration de HDFS

```
<configuration>
....<property>
........<name>dfs.replication</name>
........<value>1</value>
....</property>
....<property>
........<name>dfs.datanode.address</name>
........<value>0.0.0.0:50010</value>
....</property>
</configuration>
```

Configuration du DataNode

```
tickTime=2000
dataDir=/home/tsuna/Hadoop/zookeeper-3.4.6/dataDir
clientPort=2181
```

Configuration de Zookeeper

```
<configuration>
....<property>
........<name>hbase.rootdir</name>
........<value>hdfs://localhost:54310/hbase</value>
....</property>
....
....<property>
........<name>hbase.zookeeper.property.dataDir</name>
........<value>/home/tsuna/Hadoop/zookeeper-3.4.6/dataDir</value>
....</property>
....<property>
......<name>hbase.zookeeper.property.clientPort</name>
......<value>2181</value>
......<description>Property from ZooKeeper's config zoo.cfg.
......The port at which the clients will connect.
......</description>
....</property>
....<property>
......<name>hbase.zookeeper.quorum</name>
......<value>localhost</value>
......<description>Comma separated list of servers in the ZooKeeper Quorum.
......</description>
....</property>
....
....<property>
........<name>zookeeper.znode.parent</name>
........<value>/hbase-unsecure</value>
....</property>
....
....<property>
........<name>hbase.cluster.distributed</name>
........<value>true</value>
....</property>
</configuration>
```

Configuration de Hbase en mode distribué

Conclusion générale

A travers cette étude, nous avons utilisé différentes méthodes de Machine Learning sur des données clients. Ces méthodes nous ont permis de trouver des similitudes dans les comportements des clients, de savoir quels sont les groupes majoritaires au sein de ces clients, et de prédire leur réponse à une future campagne marketing.

Nous pouvons proposer à la banque les services suivants :

- Analyse des clients (composition, habitudes...)
- Ciblage plus précis des clients (pour n'appeler que ceux qui seraient interessés)
- Prédiction de réponse de client (de manière précise, ou simplifiée)
- Conception d'une architecture distribuée pour améliorer leur traitement des données