Rapport d'Analyse Statistique

*Un problème de Scoring

Réalisé par Joachim Bryan

Projet N°8 : Un modèle d'apprentissage automatique secteur de la Banque.

Informations : Simplifying decision trees (1987) J. R. Quinlan, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 545, Technology Square, Cambridge, MA 02139, U.S.A.

Résumé du contexte

Dans le secteur bancaire, la distribution de crédit et la gestion de moyens de paiements sont l'une des principales activités. L'attribution d'une carte de crédit est un élément essentiel au développement des activités économiques d'une société, d'un particulier. Quelles sont les problèmes rencontré à l'attribution d'une carte de crédit ? Comment le secteur bancaire les combattent-ils ? Quels sont les principaux enjeux ?

[1]Dès la première mensualité de crédit impayée, la banque déclenche le service de recouvrement à l'amiable. Un mail, un appel par le conseillé vous informera que le prélèvement n'as pu être effectué. Lorsque la phase de recouvrement a échoué, la banque se tourne vers le tribunal qui lui délivre une ordonnance « d'injonction de payer », puis charge un huissier de vous la délivrer.
Pour pallier à ces problèmes diverses et variés, le secteur bancaire utilise des algorithmes de Scoring pour évaluer en amont si un un client peut ou non poser problème à l'avenir.

Dans ce rapport d'analyse statistique, l'enjeu est de vous montrer une manière de procéder. Les données utilisés provienne d'une banque installé en Chine qui répond au problème cité plus haut. Nous avons eu accès aux données via la plateforme « [2]Kaggle ». Nous allons tenter de répondre à la problématique suivante :

« Comment créer un modèle performant, qui puisse prédire le comportement futurs d'un prospect, dans le but de pouvoir apporter une réponse clair et précise à l'obtention d'une carte de crédits et d 'éviter au maximum les problèmes liés aux impayés ? »

[1] Consulter : *https://www.emprunter-malin.com/renforcement-delai-credit-impaye*

[2] Pour en savoir plus consulter : *https://www.kaggle.com/rikdifos/credit-card-approval-prediction/tasks?taskId=1416*

Table des matières

Introduction du rapport

Les cartes de crédits sont une méthode de contrôle des risques courants dans le secteur financier. Elles utilisent les informations et les données personnelles soumises par les demandeurs de carte de crédit pour prédire la probabilité de futurs défauts de paiements et d'emprunts par carte de crédit. La banque est en mesure de décider si elle émet une carte de crédit au demandeur. Les scores de crédits permettent de quantifier objectivement l'ampleur du risque.

En général, les cartes de crédits sont basées sur des données historiques. Lorsqu'elles sont confrontées à de grandes fluctuations économiques, les modèles antérieurs peuvent perdre leur pouvoir prédictif initial. Le modèle logistique est une méthode courante de notation de crédit. Parce qu'il convient aux tâches de classification binaire et peut calculer les coefficients de chaque caractéristique.

Actuellement, avec le développement des algorithmes d'apprentissage automatique, des méthodes plus prédictives telles que le [3]Boosting, Random Forest et Support Vector Machine ont été introduite dans le scoring des cartes de crédit. Cependant, ces méthodes manquent souvent de transparence. Il peut être difficile de fournir aux clients et aux régulateurs la raison du rejet ou de l'acceptation.

Commençons l'analyse en détaillant les informations présentent dans le jeu de donnée de travail.

[3] Domaine de l'apprentissage automatique. C'est un principe qui regroupe de nombreux algorithmes qui s'appuient sur des ensembles de classifieurs binaires : le boosting optimise leurs performances.

A. Modèle, prédiction

I. Jeu de donnée de travail

Pour des raisons de sécurité, certaine variables ont été enlevé pour garantir l'anonymat des informations des individus ciblé lors de cette étude. Le noms, prénoms, compte bancaire etc… ne feront pas partis des caractéristiques étudié.

Chaque individus est notifié par un numéro d'identification (noté : ID) permettant de pouvoir le retrouver pour d'éventuelle analyse ciblé en cas de rejet ou d'acceptation d'une carte de crédit.

○ Les variables

Le genre d'un individu est précisé à l'aide d'un code binaire stipulant :
0 : genre masculin
1 : genre féminin
On appelle cette variable **CODE_GENDER**.

On connait également le nombre d'enfant de l'individus grâce à la variable **CNT_CHILDREN**.
Soit 1 enfant => **CNT_CHILDREN** = 1 etc…

Le revenu annuel de chaque individus est spécifié sous l'appellation **AMT_INCOME_TOTAL**. Cette valeur est en Dollar mais peut être converti en de nombreuse monnaie tel que l'Euro.

Un client appartient à une catégorie de revenue. Chaque catégorie est étudié et retranscrit sous l'appellation **NAME_INCOME_TYPE** selon le modèle suivant :
0 : Working (Travailleur)
1 : Commercial Associate (Collaborateur, attaché ou associé commerciale)
2 : Pensioner (Retraité)
3 : State servante (Fonctionnaire)
4 : Student (Étudiant)

Le niveau d'éducation est spécifié via la variable **NAME_EDUCATION_TYPE** selon le modèle :
0 : Higher éducation (Enseignement supérieur)
1 : Secondary / secondaire spécial (Secondaire)
2 : Incomplet higher (Incomplète supérieur)
3 : Lower secondaire (Secondaire inférieur)
4 : Academic dégrée (Diplôme académique)

La situation familial d'un individus est spécifié par la variable **NAME_FAMILY_STATUS** selon le modèle :
0 : Civil mariage (Mariage civile)
1 : Married (Marié)
2 : Single / not married (Célibataire)
3 : Separated (Séparé)
4 : Widow (Veuf, veuve)

Le mode de vie est une variable appeler **NAME_HOUSING_TYPE** spécifiant le logement selon le modèle suivant :
0 : Rented apartment (Appartement loué)
1 : House / apartment (Maison / appartement)
2 : Municipal apartment (Appartment municipal)
3 : With parents (Chez les parents)
4 : Co-op apartment (Collocation)
5 : Office apartment (Appartement de fonction)

Le type de profession(**OCCUPATION_TYPE**) est également spécifié par les 19 codes suivants :
0 : Non renseigné
1 : Security staff (Personnel de sécurité)
2 : Sales staff (Personnel de vente)
3 : Accountants (Comptable)
4 : Laborers (Ouvrié)
5 : Managers (Manager)
6 : Drivers (Chauffeur)
7 : Core staff (Personnel permanent)
8 : High skill tech staff (Personnel technique)
9 : Cleaning staff (Personnel de nettoyage)
10 : Private service staff (Personnel de service privé)
11 : Cooking staff (Personnel de cuisine)
12 : Low-skill Laborers (Ouvrié peu qualifié)
13 : Medicine staff (Personnel de santé)
14 : Secretaries (Secrétaire)
15 : Waiters/barmen staff (Serveur/ barmans)
16 : HR staff (Ressource Humaine)
17 : Realty agents (Agent immobilier)
18 : IT staff (Personnel informatique)

Le nombre de personne dans la famille de l'individus est quantifié par la variable **CNT_FAM_MENMBERS**.

L’âge des clients est donné par la variable **DAYS_BIRTH** et le nombre d’année travaillé est quantifié dans la variable **DAYS_EMPLOYED** selon le modèle :
Valeur inférieur à 0 : année de chômage
Valeur supérieur à 0 : année de travail

Pour les variables binaires qui suivent :
0 : Non
1 : Oui

Chaque individus possèdent ou non une voiture. Ceci est spécifié à l’aide d’un code binaire appelé **FLAG_OWN_CAR**.

Pour savoir si un individus possèdent ou non un bien immobilier, on utilise la variable noté **FLAG_OWN_REALTY**.

Une personne possèdent ou non un téléphone professionnelle, mobile ou fixe qualifié par un code binaire appelé respectivement par, **FLAG_WORK_PHONE**, **FLAG_MOBIL**, **FLAG_PHONE**.

Y a-t’il un email ou pas notifié par **FLAG_EMAIL**.

Grace au jeu de donnée ci-dessus, nous allons mettre en place un modèle de Machin Learning.
Le Machin Learning est une branche de ce que l’on appel aujourd’hui l’Intelligence artificiliel ou IA. Cela permet d’entrainer des données dans un modèle prédéfini à l’avance et de faire de la prédiction.
Dans la suite, nous mettons en place un modèle dans le but de trouver le plus performant selon les objectifs cités dans l’introduction (ci-dessus).

II. Le Modèle (XGB)

Ici l'objectif ne sera pas d'expliquer dans les détails l'algorithme en back du modèles utilisés. Une explication brève et compréhensible pour tous permet de clarifié les décisions prise lors de ce chapitre. Pour rappel l'objectif est de créer un modèle permettant de pouvoir prédire si un client aura un bon ou mauvais comportement si l'autorisation d'une carte de crédit est accepté. Dans le système bancaire, l'objet du rejet ou de l'acceptation est une partie importante de l'explication et de prise de décision. C'est pourquoi l'objectif sous-jacent est de pouvoir analyser la décision sur une instance (un individu) en évaluant les caractéristiques qui contribue ou non à la décision du modèle.
Pour pouvoir répondre de manière très significative au problème posé, on utilise le modèle [4]XGB (extrême gradient boosting). Ce modèle est un modèle qui utilise en back les arbres de décisions. Il est utilisé pour les problèmes de classifications comme c'est le cas ici et les problèmes de régressions. C'est un modèle ensembliste qui est optimisé pour améliorer les algorithmes dits faible comme les arbres de décisions.

« Le Boosting de Gradient est un algorithme d'apprentissage supervisé dont le principe et de combiner les résultats d'un ensemble de modèles plus simple et plus faibles afin de fournir une meilleur prédiction.
On parle d'ailleurs de méthode d'agrégation de modèles. L'idée est donc simple : au lieu d'utiliser un seul modèle, l'algorithme va en utiliser plusieurs qui serons ensuite combinés pour obtenir un seul résultat. »

Dans le secteur bancaire, on veut pouvoir prédire le comportement d'un individu en pouvant lui donner une explication du rejet ou de l'acceptation.

○ Le modèle

Dans notre jeu de donnée, on connait le comportement d'un total de 9 709 sur 90 085 clients (soit 10%) présent dans la base de donnée. On utilise donc ces 9 709 clients pour construire notre modèle d'apprentissage automatique.

Le modèle est très performant avec une précision de 97% de bonne valeurs prédite. Dans les valeurs prédite, il y a les vrais négatifs et les vrais positifs. Mais on trouve également les faux négatifs (les clients qui sont prédit comme ayant un mauvais comportement mais qui en réalité on un bon comportement) et les faux positifs (les clients qui sont prédit comme ayant un bon comportement mais qui en réalité on un mauvais comportement).

Dans ce secteur, l'idée est de minimiser le nombre de faux positifs car les frais engendré par les impayé peuvent coûter chère à la banque.

[4] XGBoost est une bibliothèque logicielle open source permettant de mettre en œuvre des méthodes de Gradient Boosting en R, python et Julia (langage de programmation).

Voici les résultats du modèle :

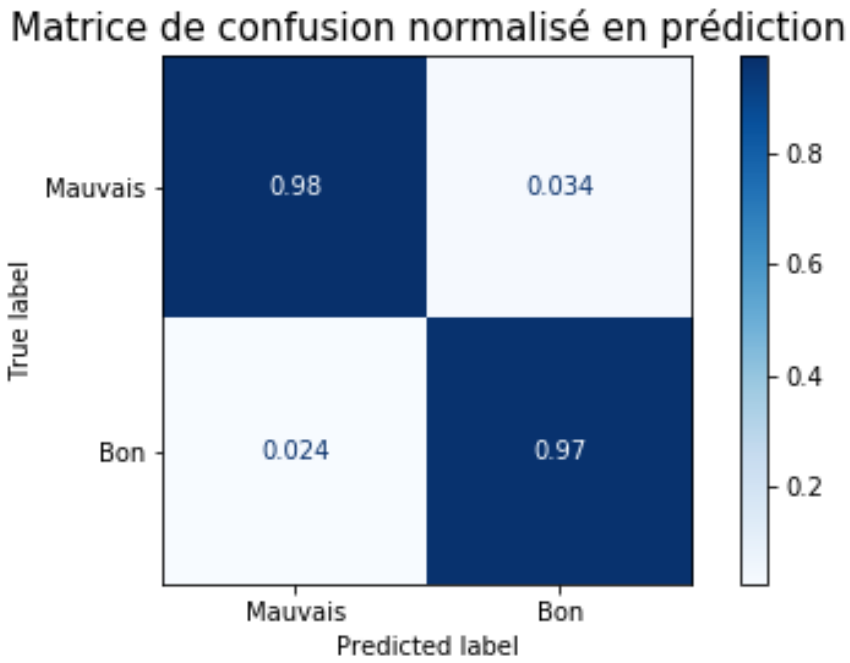


Figure 1 - Matrice de confusion -

98% de vrai négatifs
97% de vrai positifs
3,4% de faux positifs
2,4% de faux négatifs

Comme montré ci-dessus (**Figure 1**), la matrice à été normalisé en fonction des prédictions. Les pourcentages sont ceux réalisé par les prédictions. On a donc 98% de [5]vrai négatifs c'est à dire que dans notre échantillon, on a 98% de clients qui ont été prédit mauvais payeur sachant que ce sont réellement de mauvais payeur. On ne leur accordera pas de carte de crédit dans ce cas là.
Ce qui nous intéresse ici, ce sont les faux positifs. On a 3,4% de faux positifs dans notre échantillon. De plus, notre modèle repère mieux les vrai négatifs que les vrai positifs, c'est à dire que le modèle prédit légèrement mieux les vrai mauvais payeur que les vrai bon payeur a 1% prés.
On a donc un modèle qui représente très bien les objectifs ciblés.

Nous avons calculé certaines mesure qui permettent de valider le modèle :

[5] exemple : si négatif est l'absence de covid-19 chez une personne alors positif est la présence de covid-19 chez une autre personne. Un vrai négatif est une personne négative prédit comme négative par le modèle de prédiction. A l'inverse les faux positifs sont des individus négatifs qui on été prédit comme positifs par le modèle.

- Le **R²**, le coefficient de détermination. C'est la variance expliqué par le modèle. C'est un indicateur qui permet de savoir si un modèle représente bien les variations entre les valeurs prédites et les valeurs réelles.
- Le **Biais**, Le biais permet d'évaluer si les prédictions sont précises ou non et si le modèle a tendance à sur- ou sous-estimer les valeurs de la variable d'intérêt. Un Biais proche de 0 est gage d'une bonne précision du modèle.
- L'erreur moyenne absolu **MAE** est un indicateur qui juge si les prédictions sont proche des vrais valeurs. C'est la somme de la différence entre les prédictions et les valeurs réelle observés en valeur absolue, normalisé par le nombre d'observations.
- **RMSE** ou erreur quadratique moyenne qui représente la variance du modèle
- L'air de la courbe ROC, qui est l'air de pourcentage de vrai positif en fonction du pourcentage de vrai négatif. Elle juge des performance du modèle a avoir des prédictions correct. Une Aire proche de 0 indique que le modèle dont 100% des prédictions sont erronée et un AUC de 1 indique que le modèle dont 100% des prédictions sont correct.

R^2 = 0.89
Biais = 0.003
MAE = 0.028
RMSE = 0.16
ROC_AUC_SCORE = 0.97

On s'aperçoit que le modèle est très performant, avec une performance remarquable de 0,97%.

○ Les variables importantes

Notre modèle nous donne d'autre informations utile à la compréhension. L'idée est de savoir également quelles variables jouent un rôle dans le choix de la prédiction.
On appelle Features Importance, l'importance des caractéristiques qui permettent au modèle de scindé les observations en deux groupes (ici les bon et mauvais payeurs).
Grace a une méthode connue sous le nom de **RFE** (**Recursive Feature Elimination**), qui permet de faire de la sélection d'attribut (variable), on ne garde dans ce modèle que :

- AMT_INCOME_TOTAL (le revenu total
- NAME_FAMILY_STATUS
- DAYS_BIRTH
- DAYS_EMPLOYED
- OCCUPATION_TYPE
- CNT_FAM_MEMBERS

(Référence : « Jeu de donnée de travail »)

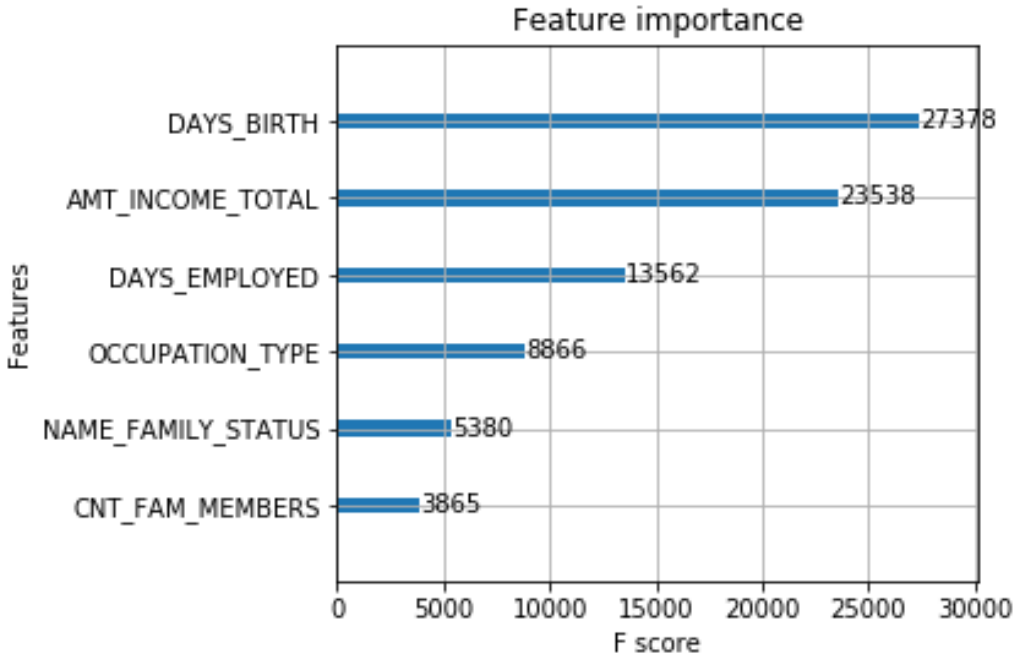


Figure 2 - Importance des variables -
DAYS_BIRTH est la variable qui sépare le mieux le jeu de donnée

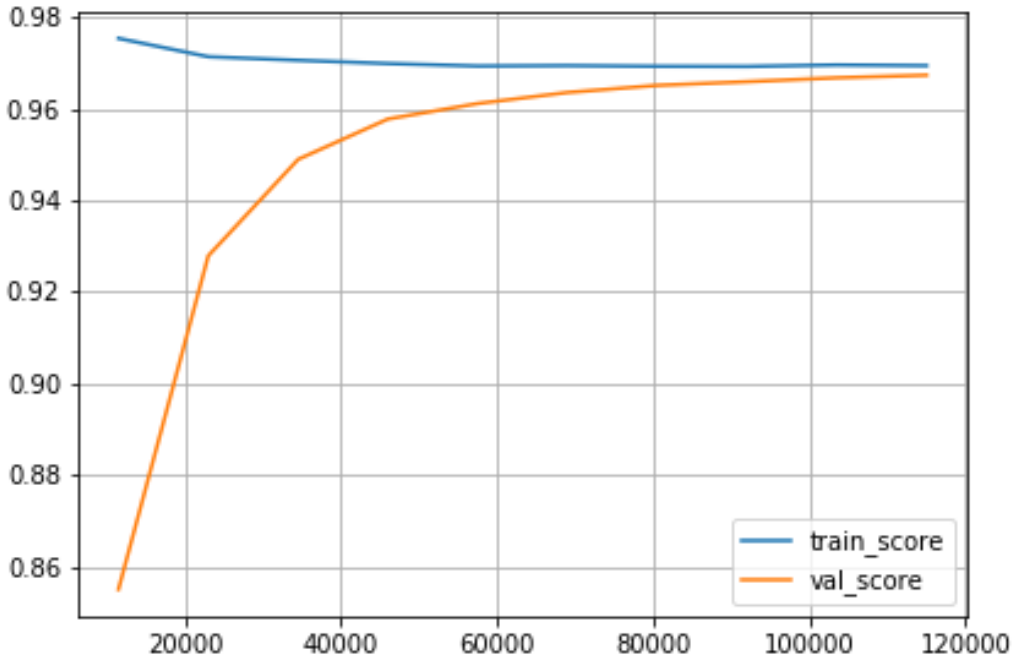
Ces variables sont celle qui permettent de définir si un client a un bon ou mauvais comportement. Grace à Extrême Gradient Boosting (XGB), on peut représenté l'importance des variables.

La **figure 2** (ci-dessus) nous montre que l'âge des clients est un facteur a prendre réellement en compte. En effet, on accorde pas de carte de crédit ou de credit a un client âgé de 30 ans comme on le ferai a un client âgé de 60 ans.
Comme on peut s'en douter, le revenu annuel est également très important.
La **figure 2**, nous montre l'importance que joue chaque variables dans la prise de décision du modèle.

Pour finaliser ce modèle nous allons analyse la [6]performance d'apprentissage de notre algorithme de machine learning dans le but de décelé éventuellement de l'overfitting.
L'overfitting, c'est l'incapacité du modèle a généraliser ses prédictions aux nouvelles données comme les données de test. Pour voir cela on utilise la courbe de train_score et la courbe de val_score pour validation score représenté dans la figure ci-dessous.
Plus la courbe de train_score et de val_score qui sont les courbes d'entrainements et de validations du modèle sont proche et plus le phénomène d'overfitting est absent.

[6] C'est la capacité d'un modèle à faire des prédictions non seulement sur les données que vous avez utilisées pour le construire, mais surtout sur de nouvelles données.

La figure ci-dessous nous montre le résultat :



- Learning Curve du modèle XGB -

Les deux courbes se touche donc il n'y a pas de sur-apprentissage des données.

Après l'étude des performances du modèles et des procédures de validations, nous pouvons enfin prédire les clients grâce aux informations représenté par les caractéristiques présenté sur la **figure 2**.

III. Prédiction et Boite Noire

Pour bien vous faire comprendre le processus nous allons utiliser un échantillon réduit qui servira d'exemple pour cette partie.
Nous avons choisit de ne représenter que **AMT_INCOME_TOTAL** et **DAYS_BIRTH** qui sont les variables les plus importantes dans ce modèle.
Les clients les plus riches sont susceptible d'être de mauvais payeur tandis que les moins riches sont susceptibles d'être des bon payeurs.

ID	AMT_INCOME_TOTAL	DAYS_BIRTH	Prediction
5008804	427500.0	32	0
5008808	270000.0	52	0
5008834	112500.0	30	1
5008891	297000.0	42	1
5010647	112500.0	24	0
5010675	225000.0	35	0
5022077	90000.0	52	1
5022102	94500.0	48	1
6836990	360000.0	44	0
6840222	103500.0	43	1

Figure 3 - Échantillons d'exemple -
Controverse des revenus et du comportement client.
0 : Mauvais payeur (mauvais comportement)
1 : Bon payeur (bon comportement)

Mais ce n'est pas tous, Dans le secteur bancaire, ce qui nous intéresse ce sont les causes et les interactions entre variables qui ont permis au modèle de prédire 0 ou 1.
Ce problème dans les modèles dits a effet de « Boîte noire », sont des modèles qui ne permettent pas de déceler le taux d'implications sur la prédiction final. On utilise donc une bibliothèque qui permet de contourner le problème. **Shap** est un outils qui permet de voir l'interaction entre chaque variables et de pouvoir décrire l'influence positive ou négative sur le choix final.

Voyons ensemble le client **5008834** :

Sur la **figure 4**, on voit l'importance des variables suivant les interactions qu'elles ont les unes avec les autres. Les interactions pour ce client montre que assemblé une a une elle réduise la probabilité de succès.
« base value » est la valeur de base du modèle tandis que la valeur en gras est la prédiction du modèle.
Chaque petit rectangle bleu montre l'effet de la caractéristique (plus il est grand, plus l'effet est grand et inversement).
Les valeurs **Shap** (c'est à dire les effets des caractéristiques), s'accumule pour donner une prédiction.

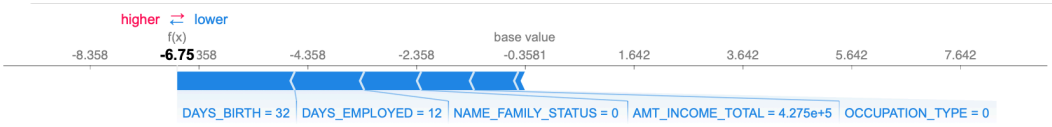


Figure 4 - Force plot client 5008834 -
Bleu : réduit la probabilité de réussite
Rouge : augmente la probabilité de réussite

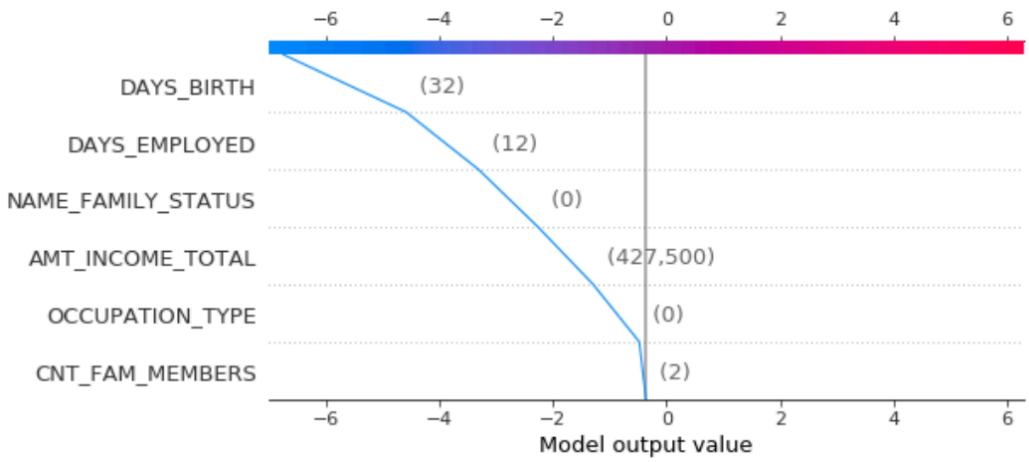


Figure 5 - Décision plot du client 5008834 -
Cumule les effets des caractéristiques et donne la prédiction associé

Ici, le client **5008834**, n'est pas un bon client sous entendu qu'il n'aura pas un bon comportement si on lui accordait un crédit ou une carte de crédit.

Le force plot est un bon moyen de se rendre compte de l'effet des caractéristiques ou variables sur la prédiction d'un client. Le décision plot (**Figure 5**) nous donne la même information que sur le force plot, mais est plus lisible si le nombre de caractéristique augmente.

Voici comment on peut justifier l'accord ou le refus de carte de crédit à travers un modèle complexe et difficile a traduire. Les interactions qui sont liés nous donnent une réponse qui prends en compte l'importance de chaque variable dans la prédiction en donnant un encadrement pour chaque variable. Si on reprends l'exemple du client 5008834, le fait qu'il ai 32 ans réduit la probabilité de succès du modèle (probabilité de succès = probabilité que le l'individus soit prédit comme ayant un bon comportement)

D'autre outils permettent de contourner l'effet boite noir et en particulier Lime.

B. Construction des données

I. Jeu de donnée 1 : informations clientèles

L'idée de se chapitre est d'expliquer comment nous avons construit notre dataset de travail (celui vu dans le chapitre 1). Pour construire un dataset exploitable plusieurs étapes sont à effectuer :

- Compréhension des données
- Nettoyages des données (valeur manquante, doublons, type des données etc…)
- Analyse univarié (études de chaque variables prise une à une)
- Analyse bivarié (études de chaque variables prise deux à deux)
- Construction du comportement du client (bon ou mauvais)

○ Jeu de donnée 1

En utilisant Pandas, on peut charger les données enregistré sous le nom « **application_record.csv** ». Il y a 438557 entré pour 18 variables (**Figure 6** ci-dessous).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 438557 entries, 0 to 438556
Data columns (total 18 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   ID                   438557 non-null  int64
 1   CODE_GENDER          438557 non-null  object
 2   FLAG_OWN_CAR         438557 non-null  object
 3   FLAG_OWN_REALTY      438557 non-null  object
 4   CNT_CHILDREN         438557 non-null  int64
 5   AMT_INCOME_TOTAL     438557 non-null  float64
 6   NAME_INCOME_TYPE     438557 non-null  object
 7   NAME_EDUCATION_TYPE  438557 non-null  object
 8   NAME_FAMILY_STATUS   438557 non-null  object
 9   NAME_HOUSING_TYPE    438557 non-null  object
 10  DAYS_BIRTH           438557 non-null  int64
 11  DAYS_EMPLOYED        438557 non-null  int64
 12  FLAG_MOBIL           438557 non-null  int64
 13  FLAG_WORK_PHONE      438557 non-null  int64
 14  FLAG_PHONE           438557 non-null  int64
 15  FLAG_EMAIL           438557 non-null  int64
 16  OCCUPATION_TYPE      304354 non-null  object
 17  CNT_FAM_MEMBERS      438557 non-null  float64
dtypes: float64(2), int64(8), object(8)
memory usage: 60.2+ MB
```

Figure 6 - Méthode info( ) -

Pour OCCUPATION_TYPE il y a que 304354 valeurs non nulles

On s'aperçoit que pour deux clients avec un **ID** différents nous avons les mêmes caractéristiques. On considère que ce sont les mêmes clients et qu'il faut supprimer tous les doublons.
Pour cela, on met en index de la data la colonne **ID** puis on supprime toutes les lignes qui se ressemble en utilisant « drop_dulicates() ».
On passe de 438 557 à 90 085 observations. On enregistre une perte de 348 472 observations (Soit prêt de 80% du jeu de donnée de base).

Pour **CNT_CHILDREN**, on s'aperçoit que les valeurs vont de 0 à 19 enfant. Dans la vie courante, 19 enfants dans une famille est rare. C'est pourquoi nous décidons de vérifier la variable **CNT_FAM_MENMBERS** :

```
Entrée [167]: 1 Info_CB.CNT_FAM_MEMBERS.value_counts()

Out[167]: 2.0     47397
          1.0     18389
          3.0     15631
          4.0      7483
          5.0      1047
          6.0       106
          7.0        24
          9.0         2
          8.0         2
          11.0        1
          15.0        1
          14.0        1
          20.0        1
          Name: CNT_FAM_MEMBERS, dtype: int64
```

Figure 7 - Nombre de client par valeurs du nombre d'enfants -
Pour le nombre d'enfant = 20 on à qu'une famille

A la vue de ce graphique, on décide de supprimer les clients possédants [20, 14, 15, 11] membres dans leurs familles.
A ce stade on perds 80% des observation plus 4.

II. Jeu de donnée 2 : informations bancaires

On utilise pandas pour pouvoir changer les données. On a une colonne ID qui fait référence à la colonne **ID** de la data du jeu de donnée 1.
On a la variable **MONTHS_BALANCE** qui représente le mois de l'enregistrement. 0 est le mois en cours, -1 est le mois précèdent etc…

On a la variable **STATUS** qui représente les jours ou mois d'impayé de certain clients de la data du jeu de donnée 1. **Status 0**: arriéré de 1 à 29 jours, **Status 1** arriéré de 30 à 59 jours etc… **Status C** remboursé ce mois-ci, **Status X** pas de prêt pour ce moi-ci.

La figure ci-dessous nous montre un aperçu des données liées aux prêts.

```
Entrée [169]:
1 Dossier_Credit = pd.read_csv("credit_record.csv")
2 Dossier_Credit
```

Out[169]:

	ID	MONTHS_BALANCE	STATUS
0	5001711	0	X
1	5001711	-1	0
2	5001711	-2	0
3	5001711	-3	0
4	5001712	0	C
...	...	...	...
1048570	5150487	-25	C
1048571	5150487	-26	C
1048572	5150487	-27	C
1048573	5150487	-28	C
1048574	5150487	-29	C

1048575 rows × 3 columns

Figure 8 - Dataset représentant pour certain client, le détail des prêts -
Plusieurs client sont représenté par plusieurs lignes car ils correspondent a plusieurs mois de prêts.

III. AFDM : Analyse factorielle de données mixte

La data de travail est maintenant prête à l'emploie, on se décide a représenter ces observations. Pour cela on va utiliser ce que l'on appelle une AFDM (Analyse factorielle des données mixtes), car nous avons dans la data frame des variables qualitatives et quantitatives. Le principe est très simple, on normalise les valeurs quantitatives en les

centrant et en les réduisant pas leur écart-types, et on pondère par la moyenne les valeurs qualitative dont on a récupéré leur fréquence d'apparition.
Avant de réaliser ça, on dois passer par une phase dite la phase d'encodage des valeurs qualitatives.
Après l'encodage et la normalisation, on affiche les valeurs dans les 6 premiers plan factorielle dans les figures ci-dessous :

Pour pouvoir en déduire certaine informations on utilise en simultanée les [7]cercles de corrélations correspondants au axe factorielle ci-dessous :

On peut réduire le jeu de donnée en disant que l'axe F1 représente le nombre de personne dans la famille et retiré les deux variables **CNT_FAM_MEMBERS** et **CNT_CHILDREN**.

Pour plus d'informations veuillez vous rendre sur ce liens : ***« liens des corrélations detaillés »***

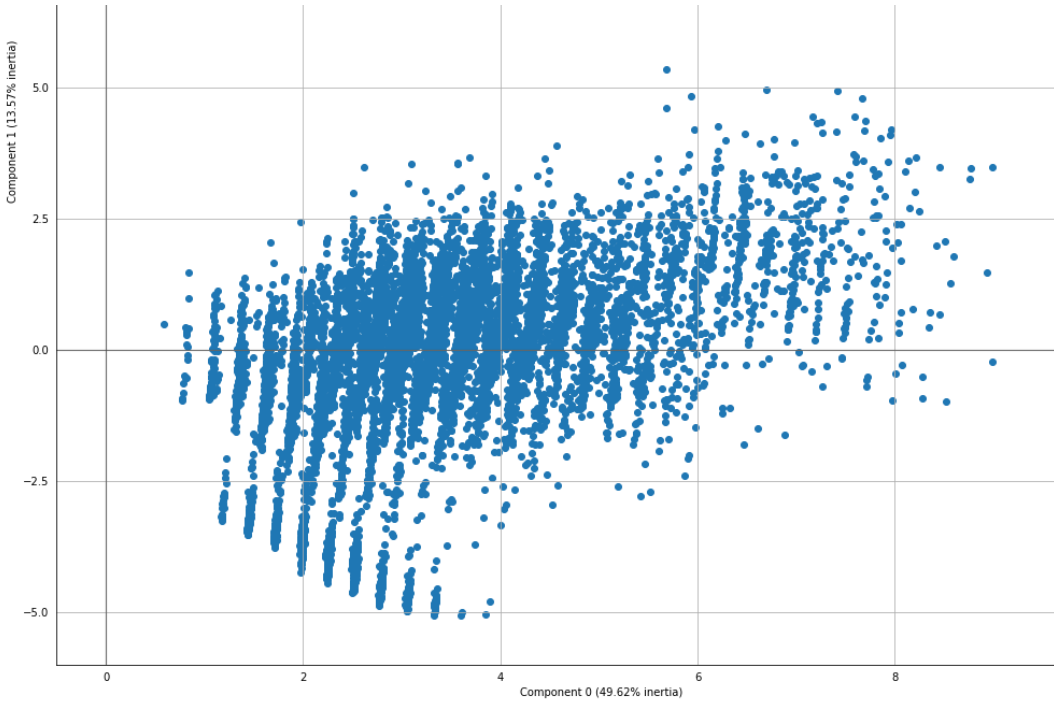


Figure 9 - Plan factorielle 1 -
On comptabilise 63,2% de l'inertie total

[7] Pour plus d'informations veuillez vous rendre sur ce liens : http://www.jybaudot.fr/Analdonnees/cerclecor.html

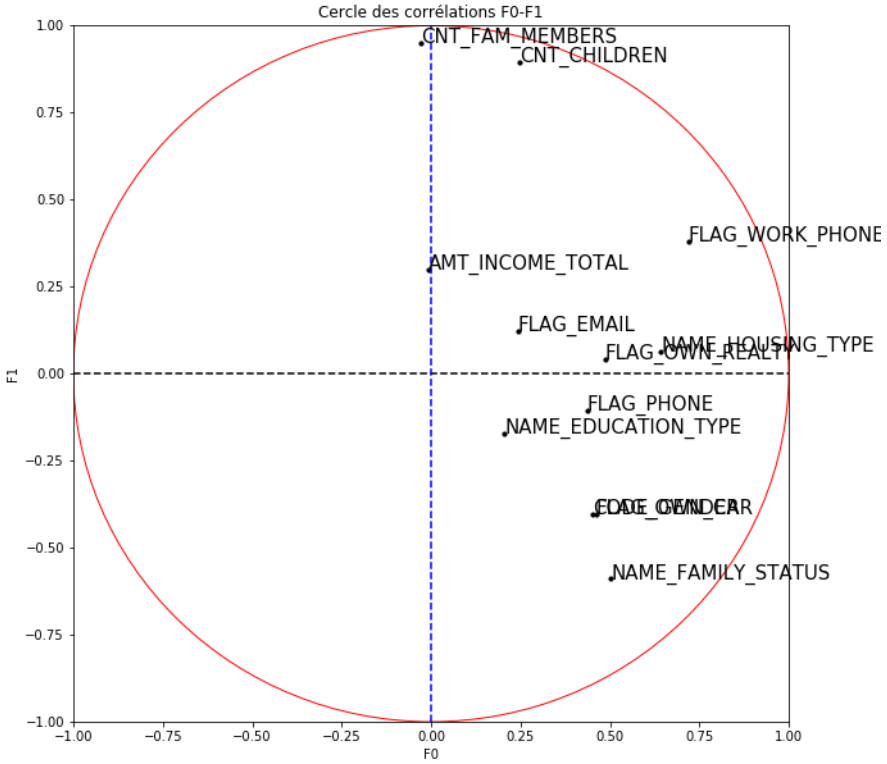


Figure 10 - Cercle des corrélations du premier plan factorielle -
CNT_FAM_MEMBERS et CNT_CHILDREN sont corrélé avec F1

IV. Algorithme du comportement d'un individu

Pour positionner le problème dans un apprentissage supervisé, on se doit de pouvoir étiqueter les clients (dont on possèdent le détails de leur opérations) sur leur comportement.
0 : mauvais comportement et donc refus de carte de crédit.
1 : bon comportement et donc acceptation de carte de crédit.

L'idée est de savoir comment diriger nos analyses pour éviter les impertinences. D'après le fichier national des incidents de remboursement des crédits aux particuliers, qui stipule les clauses d'inscriptions suivantes :

- Non remboursement de deux mensualités successive de l'emprunt en cours.
- Découvert supérieur a *500* unités pendant plus de 60 jours sans régularisation.
- Absence de régularisation.
- Dossier de surendettement.

Si un client comptabilise un nombre de (STATUS = 1) >= 1 alors ce client est automatiquement inscrit sur le fichier national des incidents de remboursement des crédits aux particuliers et de ce fait sera considéré comme un mauvais payeur.

Pour le reste des clients on choisit de noté la probabilité des impayés de 1 à 29 jours, ce qui nous amène a la figure suivante :

STATUS ID	0	1	2	3	4	5	C	X	Month	Prob_0
5008804	1	1	0	0	0	0	13	1	16	6.67
5008806	7	0	0	0	0	0	7	16	30	50.00
5008808	2	0	0	0	0	0	0	3	5	100.00
5008812	14	0	0	0	0	0	0	3	17	100.00
5008815	6	0	0	0	0	0	0	0	6	100.00

Figure 11 - Nombre de STATUS et probabilité de STATUS 0 -
Un STATUS 0 est réglé avec un C (remboursement)

Un client avec une Prob_0 < 50 est un client qui n'a pas beaucoup d'impayé et que c'est un client qui gère son compte. Pour ceux qui ont une Prob_0 > 50 on considère qu'il ne savent pas gérer leur compte.
Sur la figure 11 on voit que le premier et deuxième clients sont de bon client et que le reste sont de mauvais clients.

C'est à partir de ces informations que nous avons pu construire les étiquettes de bon ou de mauvais client qui on permis de réaliser les prédictions sur le comportement d'un individus.

Conclusion du rapport

Nous avons pu créer un programme permettant de prédire le comportement d'un individu à l'aide d'un algorithme qui est très répandu sur la sphère de la data science qui n'est autre que le modèle Extreme Gradient Boosting. La particularité de ces modèles dits d'ensemble utilisent comme base des modèles de classifications dits faibles, permettant ainsi de les généraliser. De ce fait on améliore les prévisions mais on perd en « interprétabilité ».
Tous l'enjeu de ce rapport était de pouvoir construire ce modèle et de le rendre interprétable. Dans le secteur bancaire on doit pouvoir justifier de l'acceptation ou du rejet.
Ces analyses nous auront permis de voir que la phase de nettoyage de donnée est très importante car 80% des donnée n'était pas exploitable.

Les représentations dans l'espace factorielle nous permet de réduire le nombre de variables et de voir les corrélation possibles entre elle (idem pour les observations).
On aurait pu ajouter un code couleur sur les graphiques pour une meilleur compréhension visuel, mais j'ai choisit de ne pas le faire car les données sont trop irrégulières.
L'importance du poids des variables est un enjeu de taille mais dans le secteur bancaire le revenu ne dois pas avoir le même poids que le genre. On a donc conservé les poids et de ce fait le déséquilibre des données.

Nous avons répondu a la problématique. Bien entendu, on aurait pu réaliser ce travail de manière complètement différente. Ce n'est la que le fruit de mes analyses et c'est ça qui fait la force et la faiblesse de la data science.