Note méthodologique

Mentor : Benoit Letournel

Etudiante : Rima Haddad

1. Introduction
2. Choix des données
3. Modélisation & interprétation
4. Limites & amélioration.
5. Exigence de l’entreprise
6. Courbe d’apprentissage
7. **Introduction** :

La société financière, nommée **"Home Credit" est une société** qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt.

L’entreprise souhaite **développer un modèle de scoring de la probabilité de défaut de paiement du client** pour étayer la décision d'accorder ou non un prêt à un client potentiel en s’appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).

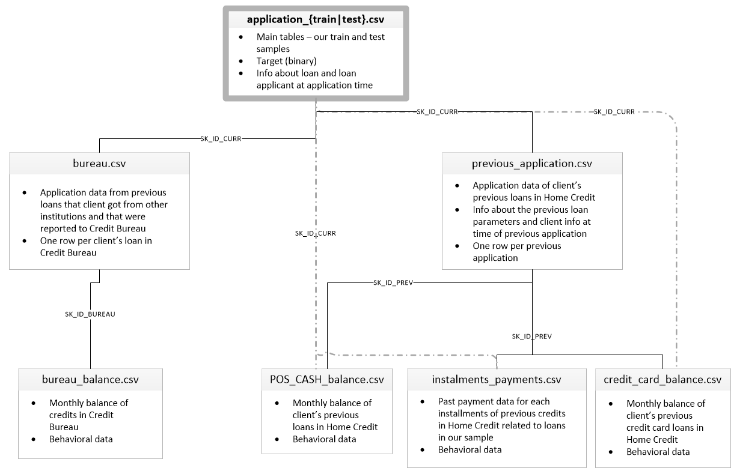
De plus, les chargés de relation client ont fait remonter le fait que les clients sont de plus en plus demandeurs de **transparence** vis-à-vis des décisions d’octroi de crédit. Cette demande de transparence des clients va tout à fait dans le sens des valeurs que l’entreprise veut incarner.

Elle décide donc de **développer un dashboard interactif** pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d’octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

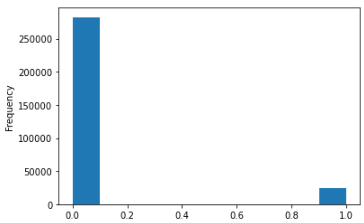
1. **Choix des données** :

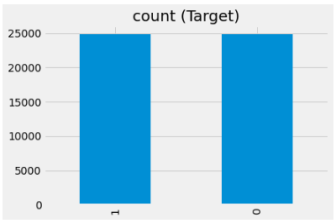
Nous disposons de plusieurs fichiers.

* Ci-dessous, un diagramme qui montre comment toutes les données sont liées.



* Il s’agit de données comportementales, données provenant d'autres institutions financières, etc.
* Dans notre travail, nous utilisons uniquement les données du fichier :
  + application\_train
* Nous nous tenons ainsi à un fichier qui devrait être plus maniable.
  + Cela nous permettra d'établir une base de référence que nous pourrons ensuite améliorer.
  + **Préparation des données :**
* Examen des valeurs manquantes dans notre dataset.
  + Le dataframe séléctionné app\_train possède122 colonnes.
  + Il y a 67 colonnes avec des valeurs manquantes.
* Application du One Hot Encoding sur les variables catégorielles.
  + Training Features shape: (307 511, 246)
* Anomalies dans les features
  + DAYS \_BIRTH : pas d’anomalies
  + DAYS\_EMPLOYED : il y a 55 374 anomalies que nous remplaçons par des NaN
* Recherche des corrélations avec la TARGET
  + DAYS \_BIRTH : est la feature la plus corrélée positivement avec la Target
  + EXT\_SOURCE\_1, EXT\_SOURCE\_2 et EXT\_SOURCE\_3 sont fortement corrélées négativement avec la TARGET
* Recherche des corrélations entre les features
  + Il y a une forte corrélation positive entre les features EXT\_SOURCE\_1 et DAYS \_BIRTH
* Problème de données déséquilibrées dans la TARGET 🡺 Rééquilibrage des données
  + Via la méthode Random Under Sampling.





0 : le prêt a été remboursé.

1 : le prêt n'a pas été remboursé.

1. **Modélisation :**

Afin de modéliser les probabilités de risque de défaut de remboursement de prêt, nous utilisons les deux modèles suivants :

* Logistic Regression Model
* Random Forest Model

3.1. Logistic Regression Model :

La régression logistique est un **modèle statistique** permettant d’étudier les relations entre un ensemble de **variables explicatives X**i et une**variable expliquée Y**. Il s’agit d’un **modèle linéaire** généralisé utilisant une **fonction logistique** comme fonction de lien.

Un modèle de régression logistique permet aussi de **prédire la probabilité**qu’un événement arrive ou non à partir de l’**optimisation des coefficients de régression**. Ce résultat varie toujours entre 0 et 1. Lorsque la valeur prédite est supérieure à un seuil, l’événement est susceptible de se produire, alors que lorsque cette valeur est inférieure au même seuil, il ne l’est pas.

3.2. Random Forest Model :

En plus du principe de bagging, les forêts aléatoires ajoutent de l’aléa au niveau des variables. Pour chaque arbre on sélectionne un échantillon bootstrat d’individus et à chaque étape, la construction d’un noeud de l’arbre se fait sur un sous-ensemble de variables tirées aléatoirement.

On se retrouve donc avec plusieurs arbres et donc des prédictions différentes pour chaque individu. Comment obtenir l’estimation finale?

* Dans le cas d’une classification : on choisit la catégorie la plus fréquente
* Dans le cas d’une régression : on fait la moyenne des valeurs prédites

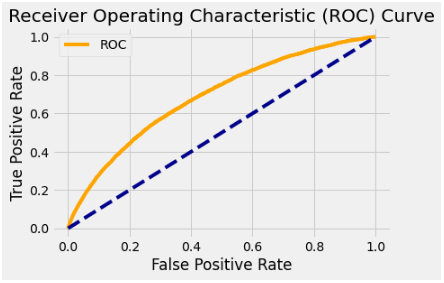
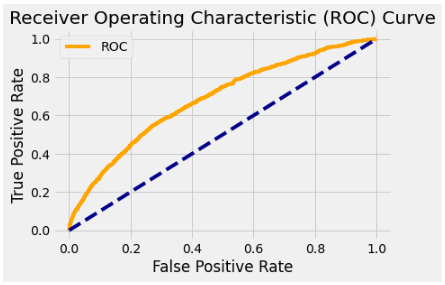
C’est un **algorithme** particulièrement **performant** pour les problématiques de **prédiction**. En particulier, nous pouvons l’utiliser quand nous avons un nombre de variables explicatives important. Nous pouvons aussi calculer l’importance de chaque variable pour présenter sa contribution au modèle.

* + **Choix du Modèle :**

Pour sélectionner le modèle final, nous nous basons sur la **fonction d’efficacité du récepteur**, plus fréquemment désignée sous le terme « **courbe ROC**[1](https://fr.wikipedia.org/wiki/Courbe_ROC#cite_note-1) » (de l’anglais ***receiver operating characteristic***, pour « caractéristique de fonctionnement du récepteur ») dite aussi *caractéristique de performance* (d'un test) ou *courbe sensibilité/spécificité*, est une [mesure](https://fr.wikipedia.org/wiki/Mesure_physique) de la [performance](https://fr.wikipedia.org/wiki/Performance) d'un [classificateur](https://fr.wikipedia.org/wiki/Classification_automatique) binaire, c'est-à-dire d'un système qui a pour objectif de catégoriser des éléments en deux groupes distincts sur la base d'une ou plusieurs des caractéristiques de chacun de ces éléments. Graphiquement, on [représente](https://fr.wikipedia.org/wiki/Repr%C3%A9sentation_graphique) souvent la mesure ROC sous la forme d'une [courbe](https://fr.wikipedia.org/wiki/Courbe) qui donne le taux de [vrais positifs](https://fr.wikipedia.org/wiki/Vrai_positif) (fraction des positifs qui sont effectivement détectés) en fonction du taux de [faux positifs](https://fr.wikipedia.org/wiki/Faux_positif) (fraction des négatifs qui sont incorrectement détectés).

Logistic Regression Model :

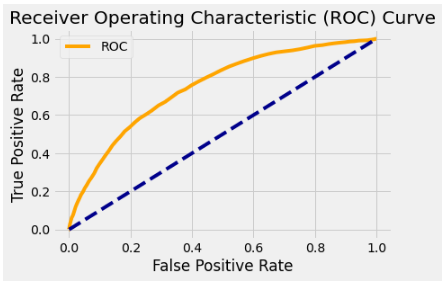
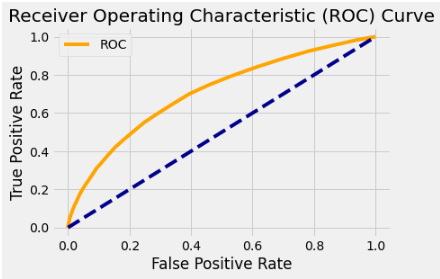
Données réequilibrées Données déséquilibrées



ROC\_AUC = 0.68 ROC\_AUC = 0.69

Random forest Model :

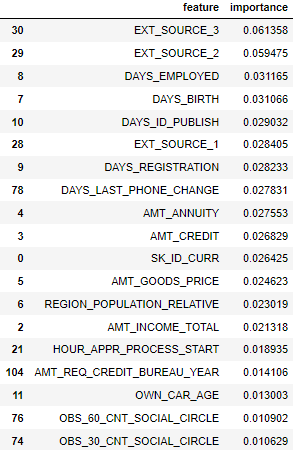
Données réequilibrées Données déséquilibrées

ROC\_AUC = 0.75 ROC\_AUC = 0.71

* + **Modèle retenu :**

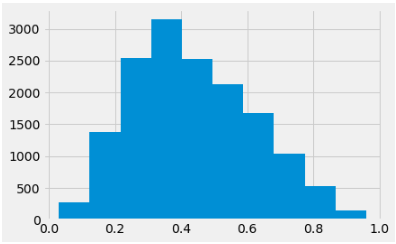
Le modèle correspondant au meilleur ROC AUC est  le modèle Random Forest sur données rééquilibrées. Il en résulte les caractéristiques « features » les plus importantes dans l’apprentissage de notre modèle sont les suivantes :



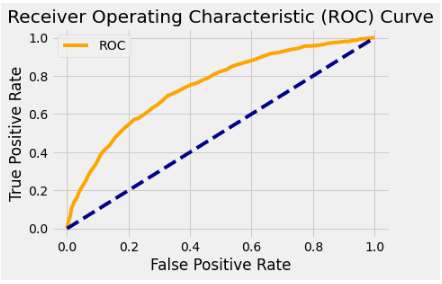
* + **Modèle final :**

Pour notre modèle final Random Forest sur données rééquilibrées, nous considérons uniquement les features importances. Ce sont les features qui ont un effet significatif dans l’apprentissage de notre modèle retenu et donc 19 variables au lieu de 246. Et afin d’estimer la fiabilité de notre modèle, nous appliquons la validation croisée « Cross Validation » qui est une méthode fondée sur une technique d’[échantillonnage](https://fr.wikipedia.org/wiki/%C3%89chantillonnage_(statistiques)). Nous avons utilisé la validation croisée à k = 5 blocs, comme suit : on divise l'échantillon original en {\displaystyle k}k échantillons (ou « blocs »), puis on sélectionne un des {\displaystyle k}k échantillons comme ensemble de validation pendant que les {\displaystyle k-1}k - 1 autres échantillons constituent l'ensemble d'apprentissage. Après apprentissage, on peut calculer une performance de validation. Puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les blocs prédéfinis. À l'issue de la procédure nous obtenons ainsi {\displaystyle k}k scores de performances, un par bloc. La moyenne et l'écart type des {\displaystyle k}k scores de performances peuvent être calculés pour estimer le biais et la variance de la performance de validation.

Distribution des probabilités :



ROC Curve :



ROC\_AUC = 0.74

Cross Validation (cv = 5) : [0.73617504, 0.73417518, 0.73413836, 0.72955738, 0.73050529]

1. **Améliorations sur le modèle & limites :**
   1. Améliorations :

La plupart des modèles de machine learning doivent être paramétrés pour donner les meilleurs résultats. Pour un [Random Forest](https://lovelyanalytics.com/2016/08/20/random-forest-comment-ca-marche/), on doit choisir le nombre d’arbres à créer et le nombre de variables à utiliser à chaque division d’un noeud. Si on paramètre à la main, cela peut vite s’avérer très coûteux en temps (et pas forcément très intéressant) …

C’est là que le Grid search intervient. C’est une méthode d’optimisation (hyperparameter optimization) qui va nous permettre de tester une série de paramètres et de comparer les performances pour en déduire le meilleur paramétrage.

Il existe plusieurs manières de tester les paramètres d’un modèle et le Grid Search est une des méthodes les plus simples. Pour chaque paramètre, on détermine un ensemble de valeurs que l’on souhaite tester.

* + Dans notre cas on veut tester :

bootstrap : [True, False], max\_depth : [5, 10, 15, 20, None], min\_samples\_leaf : [1, 2, 4], min\_samples\_split : [2, 5, 10], n\_estimators : [50, 100, 150, 200

Le Grid Search croise simplement chacune de ces hypothèses et va créer un modèle pour chaque combinaison de paramètres.

* + Les meilleurs paramètres : après application de la Grid Search CV (cv = 5)

Bootstrap : True, max\_depth : 15, min\_samples\_leaf : 4, min\_samples\_split : 10, n\_estimators : 200

* + Grid Best Score : c’est la moyenne des performances obtenues via la méthode de validation croisée : le k-fold, avec notre choix de fixer k = 5.

grid.best\_score\_ = 0.74

* + ROC AUC Score : de notre modèle avec les meilleurs paramètres

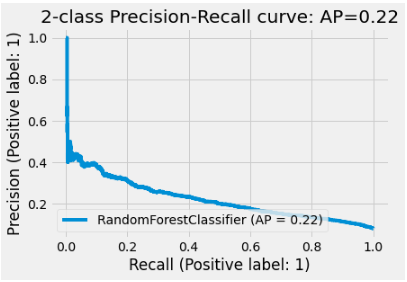
ROC\_AUC = 0.75

* 1. Limites :

Grid Search a aussi ses limites puisque c’est nous qui définissons à l’avance les paramètres que nous voulons tester.

1. **Décision d’octroie du prêt :**

Precision-Recall curve :



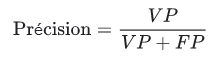
## Précision et rappel : un compromis permanent

Pour évaluer les performances d'un modèle de façon complète, nous devons analyser **à la fois** la précision et le rappel. Malheureusement, précision et rappel sont fréquemment en tension. Ceci est dû au fait que l'amélioration de la précision se fait généralement au détriment du rappel et réciproquement.

* La **précision** permet de répondre à la question suivante :

Quelle proportion d'identifications positives était effectivement correcte ?

La précision peut être définie comme suit :



* Le **rappel** permet de répondre à la question suivante :

Quelle proportion de résultats positifs réels a été identifiée correctement ?

Mathématiquement, le rappel est défini comme suit :



Courbe précision-rappel :

Une courbe qui donne différents seuils selon la valeur de la précision ou du rappel (recall).

* + Exigence de l’entreprise & critère de décision :

**L’entreprise émet le souhait très exigent que notre modèle doit détecter 90 % des clients qui ne remboursent pas leur prêt**, ce qui traduit par un Recall = 0.90 **:**

Precision = 0.11

**Recall = 0.90**

Thresholds = 0.34

Ainsi :

Si prédiction <= 0.34, alors le prêt est accordé

Sinon, le prêt est refusé

* + **Matrice de confusion :**

Une manière de tester la qualité d’un modèle est le calcul d’une matrice de confusion, c’est-à-dire le tableau croisé des valeurs observées et celles des valeurs prédites en appliquant le modèle aux données d’origine.

Seuil de classification par défaut (0.5)

|  |  |  |
| --- | --- | --- |
|  | **pred\_0** | **pred\_1** |
| **0** | **9628** | **4512** |
| **1** | **400** | **836** |

Seuil de classification ajusté (0.34)

|  |  |  |
| --- | --- | --- |
|  | **pred\_0** | **pred\_1** |
| **0** | **5253** | **8887** |
| **1** | **113** | **1123** |

Pour rappel :

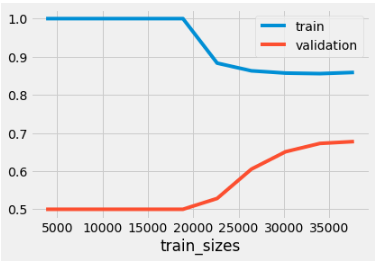
* Sur la diagonale de la matrice de confusion, on retrouve les **True Positive** (TP) et **True Negative** (TN).
* **True Positive** (TP) : la prédiction et la valeur réelle sont positives.
* **True Negative** (TN) : la prédiction et la valeur réelle sont négatives.
* En dehors de la diagonale, on retrouve les **False Positive** (FP) et **False Negative** (FN).
* **False Positive** (FP) : la prédiction est positive alors que la valeur réelle est négative.
* **False Negative** (FN) : la prédiction est négative alors que la valeur réelle est positive.

L’exigence que nous nous sommes fixés en concertation avec l’entreprise pour détecter 90% des clients qui ne remboursent pas, nous donne un nombre de False Positive égal à 113correspondant à notre seuil ajusté de 0.34. Cette exigence nous a permis donc de réduire ce chiffre quasiment par 3, car ce nombre de False Positive était égal à 400 pour le seuil de décision donné par défaut.

Néanmoins, ce choix a pour conséquence de passer à côté d’encore plus de clients qui remboursent, mais qui notre modèle détectera comme des clients qui ne remboursent pas, c’est à dire des False Negative.

En effet, avec notre exigence le nombre des False Negative passe de 4 512 à 8 887. Ainsi, le nombre des False Negative a pratiquement doublé.

1. **Courbe d’apprentissage :**



Cette courbe nous permet de constater qu’à partir d’un certain nombre de données, en l’occurrence ici environs 35 000 données, notre modèle ne gagne plus en performance qui devient constante. Cette information nous permet d’arrêter la récolte de données qui se trouve être généralement couteuse.