

LIVRABLE

Table des matières

| | |
|--|----|
| LIVRABLE | 1 |
| Introduction | 1 |
| Problèmes rencontrés et Gestion de Projet | 2 |
| Gestion de Projet : Scrumban pour Scrum et Kanban | 2 |
| Problèmes rencontrés liés aux données | 3 |
| Partie 1 : Création d'une base de données facilitant l'accès et la gestion des données | 3 |
| Problème 1 : distribution des données | 3 |
| Problème 2 : Algorithme gourmand et métriques non significatives | 3 |
| Le code permettant de déployer le modèle sous forme d'API | 7 |
| Note méthodologique | 8 |
| La démarche de modélisation | 8 |
| Calcul du coût de l'erreur de prédiction | 10 |
| L'interprétabilité | 10 |
| Conclusions et piste d'amélioration | 11 |

Introduction

Nous sommes DISF Consulting, une start-up Française qui s'occupe de fournir des outils de gestion de crédit pour des banques,

Dans le cadre de ce projet, nous avons pour mission de livrer à la société Home Crédit, un programme de machine Learning permettant de prédire la non-solvabilité des clients et de leur fournir un Dashboard, permettant au chargé de clients de répondre adéquatement aux différents clients selon leurs profils, et leurs habitudes financières.

Nous avons étudié leur base de donnée :

Home Crédit Default Risk dataset : <https://www.kaggle.com/c/home-credit-default-risk/data>
10 fichiers et 346 colonnes

Notre travail s'est structuré comme suite :

- Partie 1 : Création d'une base de données facilitant l'accès et la gestion des données
- Partie 2 : Création d'un modèle de scoring
- Partie 3 : Création d'un Dashboard interactif

Nous utiliserons une méthode de gestion Agile Scrumban une rencontre entre le Scrum et le Kanban, à partir d'un trello nous debuggerons les problèmes rencontrés (les tickets) suivant des sprints.

Problèmes rencontrés et Gestion de Projet

Gestion de Projet : Scrumban pour Scrum et Kanban

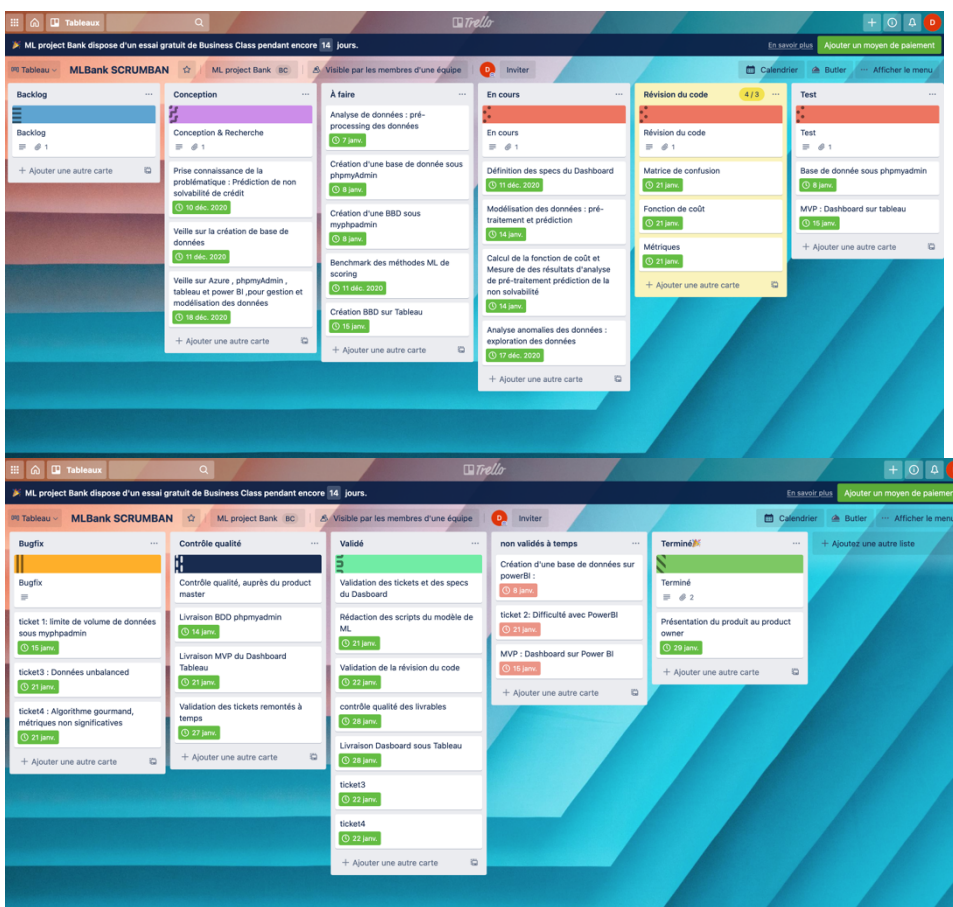
Scrum est une méthode de travail utilisée pour faire face à la complexité et livrer des produits qui ont la plus haute valeur possible. L'accent est donc mis sur la priorisation par la valeur métier en permettant une étroite participation entre le client et le consultant ; Cette démarche participative active est un atout fondamental. Elle garantit pour le client le juste équilibre entre l'investissement prévu et le produit finalement livré. L'étude du prototype permet l'évaluation des fonctionnalités réalisées, et facilite la réflexion commune sur l'opportunité de futurs développements.

Kanban est un mot japonais qui signifie l'étiquette, la carte. C'est aussi une approche, basée sur un système à flux tiré et utilisée dans le développement logiciel pour fluidifier le processus de création de valeur. Kanban a une approche plus appropriée pour le suivi de portefeuille projets, la maintenance applicative ou les équipes support.

Pour autant, avec Scrumban, on est davantage dans l'optimisation. Scrumban consiste en effet à marier les deux approches ou tendant vers un juste équilibre entre les deux pour une meilleure adaptation au contexte du projet ou du produit.

Trello : photographie ponctuelle

<https://trello.com/b/cu1zxVTj/homecredit-scrumban>



Problèmes rencontrés liés aux données

Partie 1 : Création d'une base de données facilitant l'accès et la gestion des données

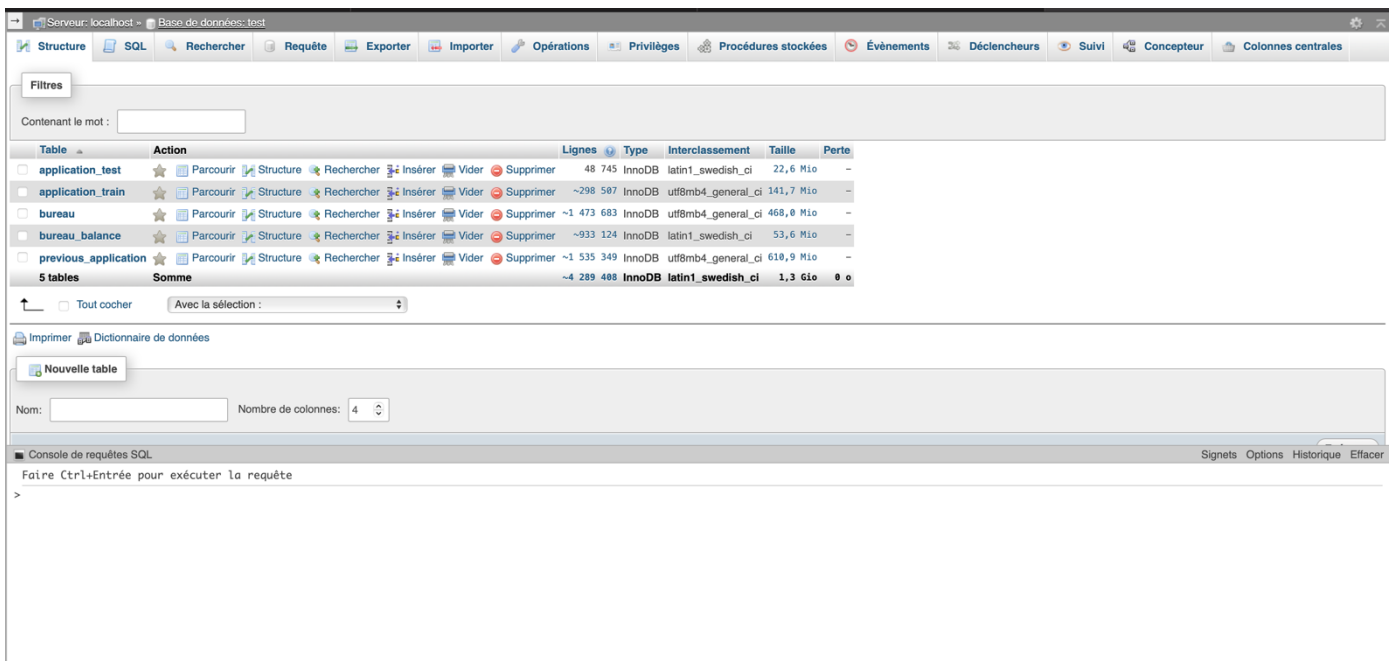
Piste 1 : Création base de données sous phpmyadmin

Problème : jeu de donnée volumineux ,

ticket 1 : jeu de donnée volumineux (> 40 Mo limite maximale sous phpmyadmin)

Solution 1: Génération d'un script sql de conversion des fichiers csv en sql

Via le site csv convert sql, <http://convertcsv.com/csv-to-sql.htm>



| Table | Action | Lignes | Type | Interclassement | Taille | Perte |
|----------------------|--|-------------------|---------------|--------------------------|----------------|------------|
| application_test | Parcourir Structure Rechercher Insérer Vider Supprimer | 48 745 | InnoDB | latin1_swedish_ci | 22,6 Mio | - |
| application_train | Parcourir Structure Rechercher Insérer Vider Supprimer | ~298 587 | InnoDB | utf8mb4_general_ci | 141,7 Mio | - |
| bureau | Parcourir Structure Rechercher Insérer Vider Supprimer | ~1 473 683 | InnoDB | utf8mb4_general_ci | 468,0 Mio | - |
| bureau_balance | Parcourir Structure Rechercher Insérer Vider Supprimer | ~933 124 | InnoDB | latin1_swedish_ci | 53,6 Mio | - |
| previous_application | Parcourir Structure Rechercher Insérer Vider Supprimer | ~1 535 349 | InnoDB | utf8mb4_general_ci | 610,9 Mio | - |
| 5 tables | Somme | ~4 289 488 | InnoDB | latin1_swedish_ci | 1,3 GiB | 0 o |

solution 2 : Création base de donnée sur Tableau.

Partie 2 : Création d'un modèle de scoring

Problème 1 : distribution des données

Ticket 3 : Données unblanced

La distribution des données est déséquilibrée, sans ré-équilibrage le modèle n'apprend pas correctement (il ne sait pas prédire les données déséquilibrée). Nous en avons fait l'expérience en appliquant le modèle de régression linéaire sur des données non équilibrée. La matrice de confusion ne sait pas prédire les valeur 1.

Problème 2 : Algorithme gourmand et métriques non significatives

Nous devons donc rééquilibrer les données avec une stratégie d'échantillonnage : Dupliquer simplement les données risque de provoquer du sur-apprentissage, c'est pourquoi il est préférable d'utiliser des algorithmes conçus expressément à cet effet comme SMOTE ou ADASYN.

Les données obtenues avec la matrice de confusion et la courbe AUC nous indique des valeurs acceptables d'un point de vue scientifique. Nous devons calculer le coût de l'erreur en terme économique.

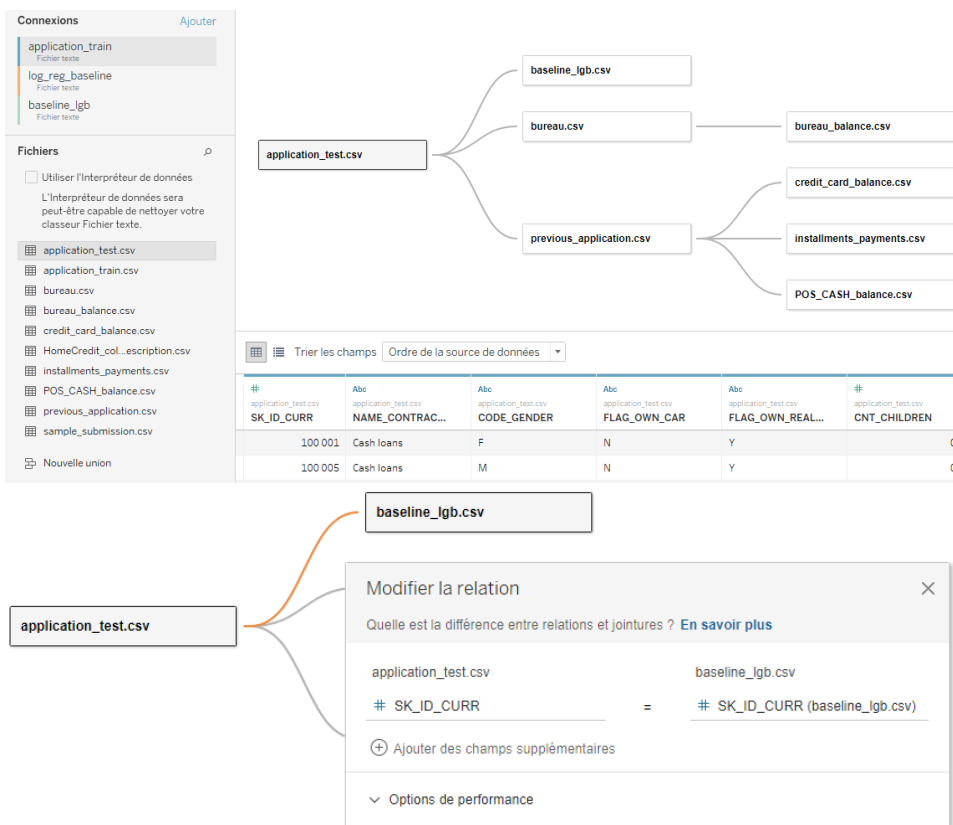
Sprint ML:



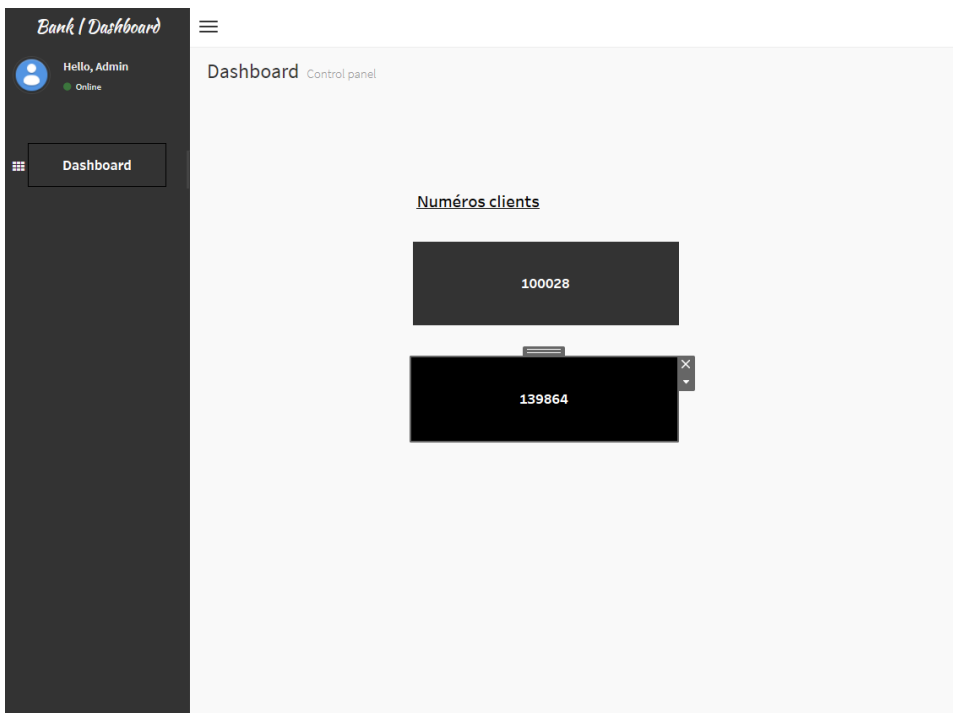
DashBoard (S + I)

Pour l'implémentation du Dashboard, nous nous sommes aidés d'un logiciel nommé Tableau, particulièrement adepte pour la visualisation des données. Avec sa version desktop, il fût possible pour nous d'accomplir les tâches suivantes :

- Importation des données csv sur le software Tableau (version Desktop) sous forme de tables
- Implémentation des différentes relations entre les tables

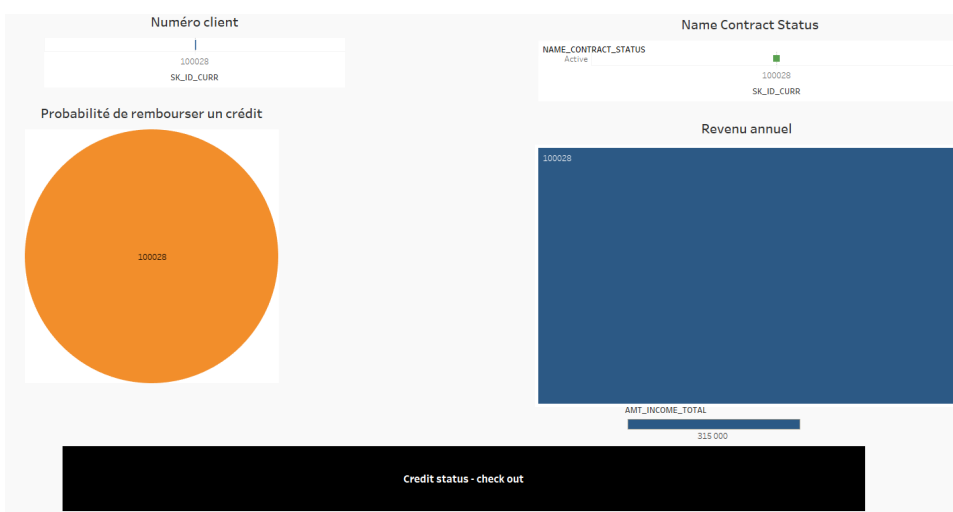


- Implémentation des différentes feuilles du Dashboard, illustrant :
 - De différents numéros clients venant de la table `application_test / train`
 - Nous avons décidé de mettre en évidence deux numéros de clients, en étant déjà connecté à un compte utilisateurs.



- Le statut du nom du contrat (actif or terminé), sa probabilité de remboursement du crédit, son revenu annuel
- Lorsqu'on clique sur un numéro de client, différents graphiques se présentent. Nous pouvons voir son Id, mais aussi la probabilité de rembourser un client avec différents code couleur. Rouge, Orange, Jaune et Vert.
- Nous avons aussi décidé de mettre en valeur ses revenus annuels et le statut de son crédit s'il est actif ou non.

L'admin a aussi la possibilité de jeter un coup d'œil au niveau des crédit de son client, ce qui nous amène à la deuxième partie du Dashboard.



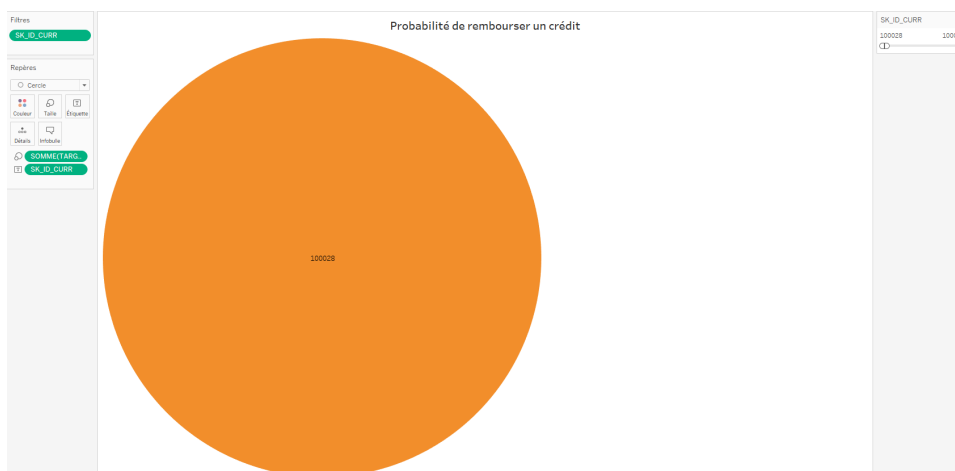
- Un résumé de différents crédits contractés, ainsi que le montant du crédit du prêt

Dans cette partie du Dashboard l'administrateur a accès à différentes informations tels que :

- Les crédits du client qui sont en cours ou terminés
- La somme des crédits cumulés
- Le montant actuel du crédit de prêt



L'outil Tableau a ainsi été particulièrement efficace lors de l'utilisation des requêtes à travers les relations entre différentes tables. Un exemple de cela étant l'utilisation de la relation entre la table `application_test` et `baseline_lgb` à travers la clé étrangère `SK_ID_CURR` pour relier les informations d'un numéro client d'`application_test` et son target dans `baseline_lgb` pour une visualisation sur le Dashboard plus centrée de clients en clients.

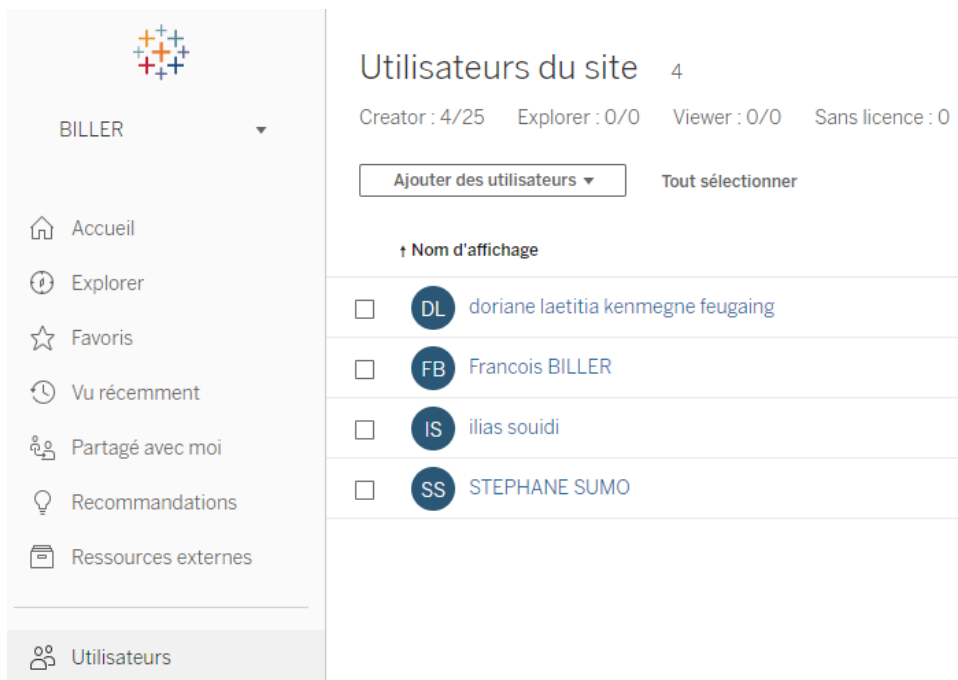


Déploiement du classeur sur Tableau online :

Tableau Desktop permet aussi de pouvoir relier notre source de données au serveur en ligne, ce qui nous a permis de déployer le classeur, qui est visible sur la plateforme tableau public grâce à ce lien :

https://public.tableau.com/profile/stephane.sumo#!/vizhome/Classeur1_16119141408490/Tableauprincipal

Mais aussi sur Tableau Online Pod (DUB01) :



Ce qui, comme illustré sur l'image ci-dessus nous a aussi été très favorable sur le travail d'équipe.

Améliorations possibles :

- Terminer la partie administration avec table d'utilisateurs, inscription, login géré par un administrateur
- Adapter le Dashboard à de différents appareils (desktop, tablette, portable)
- Centrer les différents éléments du Dashboard sur une même page
- Utiliser une barre de recherche avec filtre sur de différents clients
- Créer différentes visualisations dynamiques au rafraichissement d'une page

Le code permettant de déployer le modèle sous forme d'API

Pour pouvoir déployer notre modèle à partir de tableau, nous nous sommes aidés de l'API Javascript provenant du serveur de Tableau, qui héberge la visualisation des données.

Lors de la création de notre page web, l'appel du code de l'API fût possible grâce au script suivant:

```
<script type='text/javascript'
src='https://public.tableau.com/javascripts/api/viz v1.js'></script>
```

Et après ajout de différents paramètres permettant la gestion de la taille du tableau, des onglets à afficher, l'intégration finale au code html fût illustrée comme suit :

```

<div class="tableauPlaceholder" id="viz1611929292163" style="position: relative;">
  <noscript>
    <a href="#">
      
    </a>
  </noscript>
  <object class="TableauViz" style="display:none;">
    <param name="host_url" value="https://public.tableau.com/47" />
    <param name="embed_code_version" value="3" />
    <param name="site_root" value="" />
    <param name="name" value="Classeur1_16119141408490847;Tableauprincipal" />
    <param name="tabs" value="yes" />
    <param name="toolbar" value="yes" />
    <param name="static_image" value="https://public.tableau.com/47;static/47;images/47;Classeur1_16119141408490847;Tableauprincipal847;1.png" />
    <param name="animate_transition" value="yes" />
    <param name="display_static_image" value="yes" />
    <param name="display_spinner" value="yes" />
    <param name="display_overlay" value="yes" />
    <param name="display_count" value="yes" />
    <param name="language" value="fr" />
    <param name="filter" value="publish=yes" />
  </object>
</div>
<script type="text/javascript">
  var divElement = document.getElementById('viz1611929292163');
  var vizElement = divElement.getElementsByTagName('object')[0];
  if ( divElement.offsetWidth > 800 ) {
    vizElement.style.minWidth='1920px';
    vizElement.style.maxWidth='100%';
    vizElement.style.minHeight='1570px';
    vizElement.style.maxHeight=(divElement.offsetWidth*0.75)+'px';
  } else if ( divElement.offsetWidth > 500 ) {
    vizElement.style.minWidth='1920px';
    vizElement.style.maxWidth='100%';
    vizElement.style.minHeight='1570px';
    vizElement.style.maxHeight=(divElement.offsetWidth*0.75)+'px';
  } else {
    vizElement.style.width='100%';
    vizElement.style.minHeight='2300px';
    vizElement.style.maxHeight=(divElement.offsetWidth*1.77)+'px';
  }
  var scriptElement = document.createElement('script');
  scriptElement.src = 'https://public.tableau.com/javascripts/api/viz_v1.js';
  vizElement.parentNode.insertBefore(scriptElement, vizElement);
</script>

```

Le détail du code est visible sous notre lien github :
https://github.com/sas03/Banque_ML/blob/master/index.html

Note méthodologique

La démarche de modélisation

(Idées - A modifier, fait par D,S)

Utilisation du machine learning pour la modélisation des données pour des prises de décision stratégiques. Etapes qui s'effectue après avoir

- Sélectionner les données pertinentes
- Nettoyer les données pertinentes
- Transformer les données pertinentes

Pour représenter le comportement pour résoudre la problématique de savoir si un client est plus susceptible de rembourser son crédit ou non.

Cela passe par l'apprentissage du modèle grâce à des données d'entraînement.

- Les données
- Prédiction du remboursement du crédit ou non
- Algorithme d'apprentissage
- La mesure de performance de l'algorithme

OnehotEncoding versus LabelEncoding

Les machines comprennent les nombres, pas les textes. Pour ces raisons, l'on doit convertir chaque catégorie de textes en des nombres pour que les machines puissent les interpréter avec des expressions mathématiques.

Des colonnes combinant les variables numériques et les variables catégoriques.

Pour cela, l'on doit donc utiliser un codage catégorique pour convertir les variables catégoriques en variables numériques.

Label encoding: Un integer unique est assigné à chaque label sur un ordre alphabétique.

- Inconvénient de la méthode : La transformation des variables catégoriques en variables numériques se fait sur un ordre mathématique, et pour cette raison, peut créer des relations de supériorité entre nombres qui n'existe pas sur les chaînes de caractères, affectant la performance du modèle.

One hot encoding: Créer de nouvelles colonnes dépendant du nombre de variable unique dans la colonne catégorielle. Chaque valeur unique de la colonne avec les variables catégoriques sera ajoutée comme une nouvelle colonne

- Avantage : Au contraire du label encoding, le One hot encoding résoud le problème de supériorité de variables catégorielles, avec chaque catégorie représentée par un vecteur binaire (0 et 1).

Gestion des valeurs manquantes

Les algorithmes de machine learning n'aiment pas l'absence de valeur pour une variable, cela peut provoquer des biais dans les calculs de prédiction. Supprimer les lignes avec les valeurs manquantes n'est pas une solution envisageable car cela reviendrait à supprimer les 2/3 des lignes du jeux de données !

Nous remplaçons les valeurs vide par la valeur médiane de la variable (SimpleImputer). En choisissant la médiane nous modifions moins la distribution des valeurs d'une variable

Mise à l'échelle des données

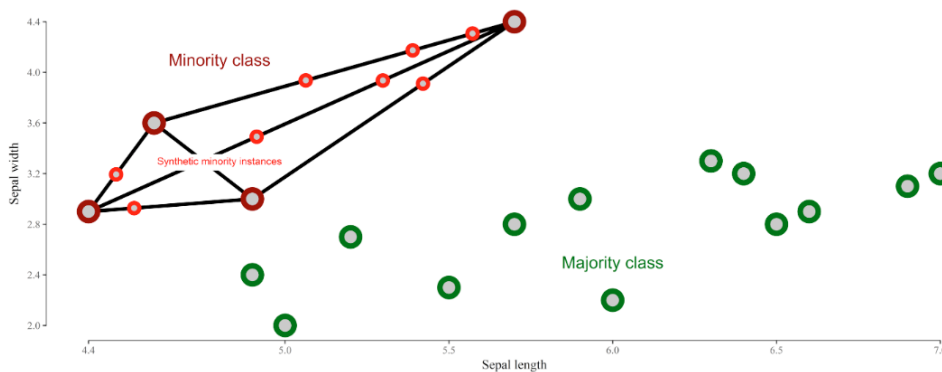
Les différences dans les échelles entre les variables d'entrée peuvent augmenter la difficulté du problème à modéliser. Un modèle avec de grandes valeurs de poids est souvent instable, ce qui signifie qu'il peut souffrir de mauvais résultats pendant l'apprentissage et d'une sensibilité aux valeurs d'entrée entraînant une erreur de généralisation plus élevée. C'est particulièrement le cas avec la régression linéaire et logistique et l'apprentissage profond. Un redimensionnement des variables d'entrée est nécessaire. Nous choisissons de normaliser les valeurs des variables des jeux de donnée en entrée sur une échelle de valeur de 0 à 1 (MinMaxScaler)

Équilibrage des données

La distribution des données cible (y) est déséquilibrée, sans ré-équilibrage le modèle n'apprend pas correctement (il ne sait pas prédire les données déséquilibrée). Nous en avons fait l'expérience en appliquant le modèle de régression linéaire sur des données non équilibrée. La matrice de confusion ne sait pas prédire les valeur 1.

SMOTE : est une méthode utilisée visant à re-équilibrer les données (génération de lignes pour rééquilibrer le nombre de valeur cible à 1 par rapport à 0)

l'algorithme d'over-sampling le plus connu, l'algorithme parcourt toutes les observations de la classe minoritaire, cherche ses k plus proches voisins puis synthétise aléatoirement de nouvelles données entre ces deux points.



- Partie machine learning (suite et fin F)

Lorsque cette méthode est appliquée le nombre de ligne $y = 1$ est égal au nombre de ligne $y = 0$ par l'ajout de ligne supplémentaire.

Calcul du coût de l'erreur de prédiction

Afin d'estimer la validité du modèle au regard du contexte métier il est intéressant de calculer le coût métier des mauvaises prédictions faites par le modèle. Ici l'appréciation sera économique : quel est le coût économique d'une mauvaise prédiction.

Il est la conséquence des :

- 1) Faux Positif : Bon emprunteur prédit comme étant mauvais. C'est un manque à gagner en « nouveau emprunter » pour la société.
- 2) Faux Négatif : Mauvais emprunteur prédit comme étant bon. C'est une perte pour la société car le prêt ne sera pas remboursé.

Nous pouvons calculer ce coût pour une taille de population (ici 500.000 client) à partir de la précision du modèle (0,71), et le ration de la sous population cible (ici 0,08)

Nous connaissons la valeur moyenne d'un prêt : 599 000

Nous pouvons donc calculer le manque à gagner en prêt pour les faux positifs (167961) : 100 millions, rapporté au montant total des prêts = 0.05%

Nous pouvons aussi calculer les pertes dues aux faux négatifs (13559), en estimant que la perte moyenne d'un prêt et la moitié du montant du prêt (l'emprunteur en rembourse que la moitié) : 4 millions soit 0,0022 % rapporté au montant total des prêts.

Dans ces conditions économiques la précision du modèle est acceptée, puisque le calcul du coût de l'erreur de prédiction est négligeable.

L'interprétabilité

L'interopérabilité d'un modèle de machine learning signifie qu'un humain peut comprendre la cause de la décision. Le modèle de régression logistique que nous avons utilisé s'explique mathématiquement.

Quelles métriques utiliser ?

L'erreur classique dans le cas de classes déséquilibrées est de croire que votre modèle est performant parce qu'il possède une exactitude proche de 1. Cela ne pourrait pas être plus faux !

Effectivement, ici 90% de nos observations appartiennent à la même catégorie et le modèle prédit toujours cette catégorie nous avons obtenu une exactitude de 90%, mais considéré ce résultat serait complètement naïf. Il vaut mieux utiliser d'autres métriques plus adaptées, comme :

$$Precision = \frac{TP}{FP + TP}$$

· La **précision** :

$$Rappel = \frac{TP}{FN + TP}$$

Le **rappel** :

$$F1 = \frac{2 * Precision * Rappel}{Precision + Rappel}$$

Le **F1-Score** :

L'**AUC Précision-Rappel (AUC PR)** : aire normalisée sous la courbe paramétrique définie par la Précision et le Rappel en fonction du seuil de décision.

La **matrice de confusion** :

| | | Classe réelle | |
|----------------|---|------------------------------------|------------------------------------|
| | | - | + |
| Classe prédite | - | True Negatives (vrais négatifs) | False Negatives (faux négatifs) |
| | + | False Positives (faux positifs) | True Positives (vrais positifs) |

Conclusions et piste d'amélioration

Cet exercice nous a permis d'appréhender comment le Machine Learning peut s'intégrer dans un processus numérique de décision d'octroi d'un crédit.

Nous avons manqué de temps parce que nous l'avons mal géré. Nous aurions pu tester d'autre modèle notamment comprendre pourquoi le modèle de random forest que nous avons implémenté donne de mauvais résultat, as-t-on utilisé les bons paramètres.

Mieux utiliser Tableau, afin de fournir un tableau de bord métier est une autre piste d'amélioration