31/08/2021

CRÉDIT BANCAIRE

LES DÉFAUTS DE PAIEMENT

SOMMAIRE

SIMPLON

Pierre-Antoine SALISBURY

PROBLÉMATIQUE

QUELS CRITÈRES PERMETTENT DE DÉFINIR LES PROFILS LES PLUS À RISQUE QUANT AU DÉFAUTS DE PAIEMENT ?

INTRODUCTION

Choix de la base de données : nous avons opté pour un ensemble composé de trois datasets disponible sur Kaggle[1]. Il s'agit de trois fichiers composés de 164 colonnes au total. L'analyse des risques concernant le prêt bancaire est un cas d'école en data et permet d'explorer plusieurs compétences requises dans le métier de data analyst. Qui plus est, le jeu de données que nous avons à disposition prend en considération les contraintes liées au projet et nous permet de composer un schéma de quatre tables. A partir de cet ensemble, nous allons mettre en pratique tout ce que nous avons appris pendant cette formation et répondre aux exigences demandées pour le passage de notre certification développeur data. Il s'agira dès lors de concevoir et exploiter une base de données, tout en gérant la qualité du projet de manière agile à partir d'un tableau Kanban[2]. Concernant la conformité du processus de diffusion des

1 - Kaggle est une plateforme web qui met à disposition des bases de données. Elle organise des compétitions en science de la donnée. Sur cette plateforme, les entreprises proposent des problèmes en science de la donnée et offrent un prix aux datalogistes obtenant les meilleures performances. L'entreprise a été fondée en 2010 par Anthony Goldbloom. La société mère est Google.

2 - Un tableau Kanban est un outil de gestion de projet Agile conçu pour aider à visualiser le travail, limiter le travail en cours et maximiser l'efficacité (ou le flux). Ces tableaux peuvent aider les équipes Agile et DevOps à mettre de l'ordre dans leur travail quotidien. Les tableaux Kanban ont recours à des cartes, à des colonnes et à l'amélioration continue pour aider les équipes technologiques et de service à s'engager sur une quantité de travail appropriée, puis à la réaliser.

données, il est en accord ave la législation RGPD[3]. Nos données sont complètement anonymisées : nous n'avons pas d'informations relatives au nom, prénom, adresse,

téléphone qui permettraient d'identifier nos clients. Étant sociologue de formation, j'ai une appétence particulière pour l'analyse statistique de données. Ce projet nous permettra in fine de faire des corrélations et d'analyser des profils plus ou moins à risque.

Cette étude de cas vise à nous donner une idée de l'application de l'EDA (loan data exploratory risk analysis) dans un scénario commercial réel. Nous développerons une compréhension de base de l'analyse des risques dans les services bancaires et financiers afin de comprendre comment les données sont utilisées pour minimiser le risque de perdre de l'argent en prêtant à des clients. Les sociétés de prêt restreignent les accords de prêts aux personnes en raison de leurs antécédents de crédits insuffisants ou inexistants mais aussi en fonction de leur profil. Deux types de risques sont associés à la

3 - Le RGPD est un règlement de l'Union européenne qui constitue le texte de référence en matière de protection des données à caractère personnel. Il renforce et unifie la protection des données pour les individus au sein de l'Union européenne. Les principaux objectifs du RGPD sont d'accroître à la fois la protection des personnes concernées par un traitement de leurs données à caractère personnel et la responsabilisation des acteurs de ce traitement.

décision de la banque : si le demandeur est susceptible de rembourser le prêt, le fait de ne pas approuver le prêt entraine une perte d'activité pour la banque. Si le demandeur n'est pas capable de rembourser le prêt et donc de faire défaut, alors l'approbation du prêt peut entraîner une perte financière. Ainsi, l'objectif est de garantir que les candidats capables de rembourser leurs mensualités ne sont pas rejetés et inversement.

COMPRÉHENSION DES DONNÉES

Provenance des données : le propriétaire des données est indien "Gaurav Dutta », il vient de Hyderabad, Telangana en Inde et on peut supposer que la provenance des données concerne les micros crédits en Inde. Notre bdd comporte 8% de clients défectueux. Elles sont regroupées dans les trois CSV suivants :

'application_data.csv' contient toutes les informations du client au moment de la demande. Les données permettent de savoir si un client a des difficultés de paiement.

'previous_application.csv' contient des informations relatives au prêt : le prêt a été apporté, annulé ou refusé. Ainsi, un client peut avoir plusieurs prêt donc nous avons trois fois plus de prêts que de clients.

'description_columns.csv' : dictionnaire qui donne la signification de toutes les variables

```
round(df['cible'].value_counts(normalize=True)*100,1)
```

0 = 91,9 % soit 282 686 clients non défectueux
1 = 08,1 % soit 24 825 clients défectueux

Les données ci-dessous contiennent des informations sur les demandes de prêts. Elles présentent deux types de scénarios : le client en difficulté de paiement : il a un retard de paiement supérieur à X jours sur au moins une des Y premières échéances du crédit de notre échantillon. Tous les autres cas : le payment est effectué à temps. Lorsqu'un client demande un prêt, quatre types de décisions peuvent être prises par le client / l'entreprise :

Approuvé : la banque a accepté la demande de financement

Annulé : le client a annulé la demande en cours d'approbation. Il a changé d'avis et n'a par exemple pas obtenu le taux espéré.

Refusé : la banque refuse le prêt car le profil du client ne répond pas aux exigences.

Offre non utilisée : le prêt à été annulé par le client mais à différentes étapes du processus.

Nous mettrons à disposition plusieurs notebooks. Un notebook pour l'exploration des données, un notebook pour le preprocessing. Deux notebooks pour la création des tables CLIENT et CREDIT. Un notebook d'optimisation avec la fonction factorize. Un notebook pour faire une fusion avec merge. Un notebook pour l'analyse univariée. Un notebook pour la prédiction.

Pour l'analyse bivariée, nous utiliserons LibreOffice Calc, ce qui nous permettra aussi de faire des graphiques à partir de tableaux de proportionnalité.

OBJECTIFS

Une banque X nous a contacté pour analyser les données de ses clients. Elle souhaite réaliser une étude de cas pour identifier les facteurs qui peuvent influencer les défauts de paiement. Si un client a des difficultés la banque peut prendre des mesures telles que la diminution ou l'augmentation du taux de prêt voire du refus. Dès lors nous devons essayer d'analyser les facteurs déterminants ou variables déterminantes des clients défectueux et des paiements honorés. Ainsi, la banque pourra utiliser ces connaissances pour valuer les risques.

Voici en détail la mission proposée par la banque :

Vous devez d'abord nettoyer votre jeu de données et extraire au moins 20 variables pertinentes concernant les clients, les crédits, les types de crédits et les biens financés à partir des prêts bancaires. Pour se faire, notez bien que les valeurs nulles et valeurs aberrantes doivent être dûment mis en évidence afin de sélectionner les données pertinentes. Aussi, votre mission tout au long de votre étude est de conserver un maximum de données.

Parmi les variables vous devez impérativement extraire les informations suivantes :

Concernant les clients : des données sur la **cible**[4], l' âge , le genre, le niveau d'étude, le nombre d'enfants, le statut matrimonial, les revenus, le type d'entreprise et la catégorie professionnelle, sur l'ancienneté au sein de la banque et sur le logement.

Concernant le crédit : des données sur le montant accordé, le statut du contrat et le nombre de mois de décision.

Concernant le type de crédit, il suffit de recueillir l'ensemble des données concernant le type de crédit des clients sélectionnés.

Ensuite, votre objectif est de faire des corrélations entre la cible et les autres variables afin d'identifier les profils les plus risqués et les profils les plus en règle.

Vous devez trouver les corrélations les plus pertinentes à partir d'analyses univariées. Également, montrer que vous êtes capables de construire un raisonnement qui vous permette de procéder à une analyse bivariée. Cette analyse devra s'ajouter aux profils à risque que vous aurez identifié.

A l'aide d'un outil de datavisualisation comme Power BI proposer un dashboard interactif avec une vue complète qui permette de décrire analyser et mieux comprendre le différentes tables. Mettez en valeur le revenu, le genre, le niveau d'étude, les biens financés. Ainsi que ce que vous jugerez opportun.

---

[4] - La cible est la variable clé de notre analyse puisque qu'elle nous donne l'information sur le défaut de paiement : 0 pour client non défectueux, 1 pour client défectueux.

Pour finir proposez-nous un modèle simple (régression linéaire par exemple) qui permette de prédire le défaut de paiement d'un client à partir de variables déterminantes que vous aurez préalablement identifiées.

SAUVEGARDE DES DONNÉES

Afin d'assurer le processus de sauvegarde tout au long de notre projet nous avons opté pour quatre options afin qu'aucune possibilité de perte de donnée ne soit envisagée et que des issues de secours soient toujours possibles si un bug ou un problème se produirait.

Tout d'abord il s'agit d'enregistrer dans des fichiers tous les documents sur lesquels nous travaillons et faire régulièrement des sauvegardes notamment à l'aide du raccourci ctrl s.

Ensuite, nous créons des dossiers contenant des fichiers avec les différentes étapes du traitement de nos données. Par exemple nous avons un fichier pour les CSV originaux et un fichier pour les CSV nettoyés.

Puis, il s'agit de mettre en place une dropbox et un drive ou nous pouvons conserver en temps réel l'avancement de notre projet sur un cloud. Pour les bases de données - qui peuvent être relativement lourdes - nous avons créer un back up[5], c'est-à-dire une sauvegarde complète de données dans MySQL Server.

Enfin, à chaque étape de sauvegarde de fichier importante nous enregistrons les dossiers dans des clés USB et/ou un disque dur externe. Rappelons que la durée de vie d'un clé USB est de 10 ans mais qu'il est convenu de renouveler les sauvegardes sur de nouveaux supports tous les trois ans.

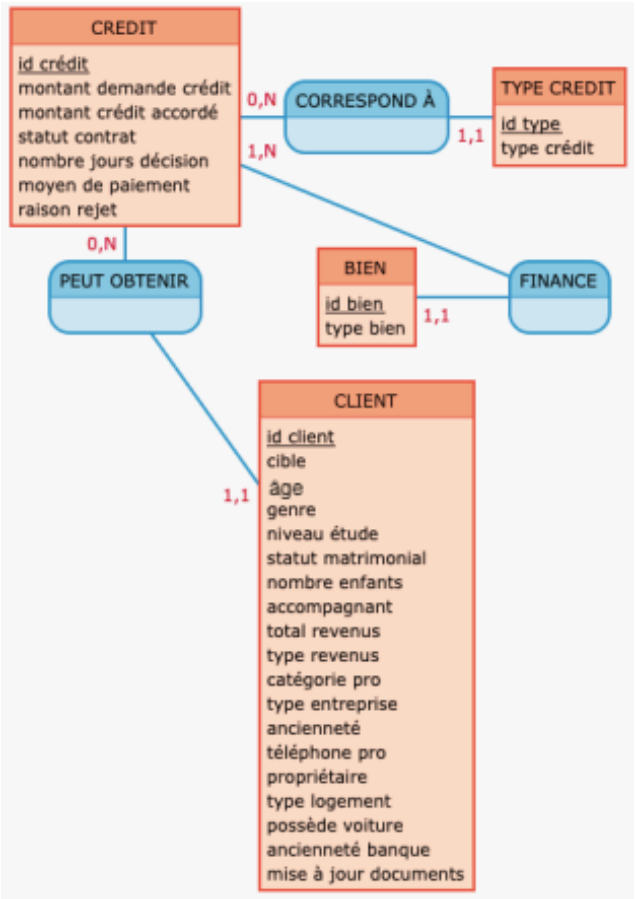
---

4 - Pour vider toutes les bases de données il faut appeler mysqldump avec l'option –alldatabases. Pour ne vider que des bases de données spécifiques, on le nomme sur la ligne de commande on utilise le database option. L'option –all databases fait que tous les noms sur la ligne de commande sont traités comme des noms de base de données. Sans cette option, mysqldump traite le prénom comme un nom de base de données et les suivants comme des noms de table. Avec --all-databases ou --databases, mysqldump écrit CREATE DATABASE et des USE instructions avant la sortie de vidage pour chaque base de données. Cela garantit que lorsque le fichier de vidage est rechargé, il crée chaque base de données si elle n'existe pas et en fait la base de données par défaut afin que le contenu soit chargé dans la même base de données d'où il provient. Pour recharger un fichier de vidage écrit par mysqldump qui se compose d'instructions SQL, on l'utilise comme entrée du client mysql .

I/ MODÉLISATION DE LA BASE DE DONNÉES

A) TROIS DIFFÉRENTS MODÈLES DE DONNÉES

Modèle conceptuel : le MCD nous permet de décrire l'application dans un langage de haut niveau (Entité-Association) qui ne tient pas compte du SGBD. La cardinalité est le nombre de participation d'une entité à une relation. Les cardinalités maximales et minimales traduisent les contraintes propres aux entités et relations. Une relation a au moins une clé candidate (chacun des attributs est renseigné, pas de valeurs NULL). On a 4 tables



Exemple de relation : un client peut obtenir un ou plusieurs prêts

Cardinalité : pas de plusieurs à plusieurs mais on peut en créer une avec une table BANQUE - imaginaire. Par exemple : plusieurs banques peuvent financer plusieurs clients.

Modèle logique : le MLD décrit les données dans un formalisme compatible avec un SGBD (schémas, tables, colonnes, clés primaires et étrangères).

CLIENT (id client, cible, âge, genre, niveau étude, statut matrimonial, nombre enfants, accompagnant, total revenus, type revenus, catégorie pro, type entreprise, ancienneté, telephone pro, propriétaire, type_logement, possède voiture, ancienneté banque, mise à jour documents)

CREDIT (id_credit, montant demande crédit, montant crédit accordé, statut contrat, nombre jours décision, moyen de paiement, raison rejet, id_client*, id type_credit*, id bien*)

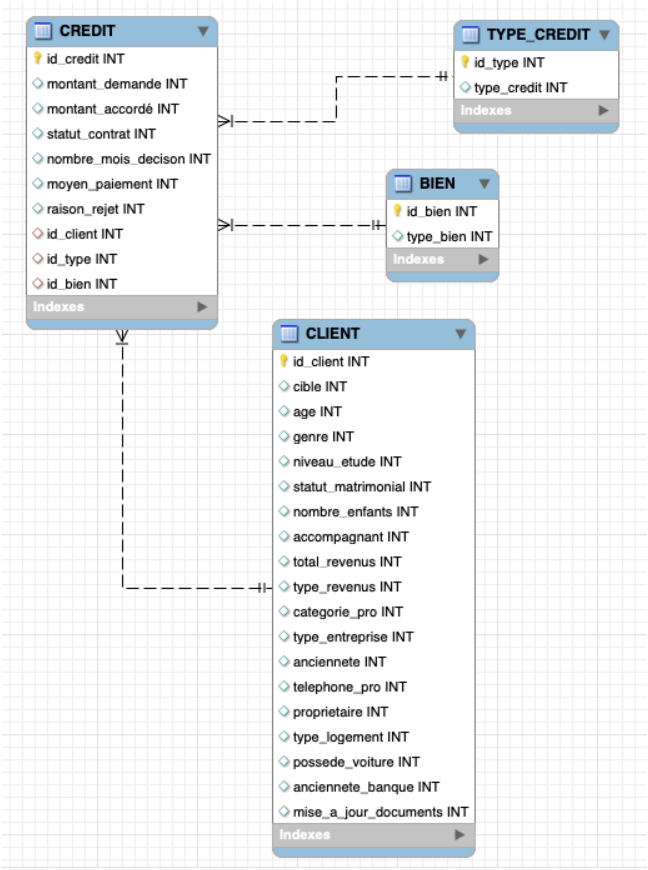
Exemple de lecture : id_client fait référence à la clé primaire de CLIENT.

TYPE CREDIT (id type, type credit)

TYPE BIEN (id bien, type bien)

Modèle physique de données : le MPD est l'implémentation du modèle logique dans le SGBD (affiner le MLD en un schéma pour un SGBD spécifique), utilisation de MySQL (create table …), types des attributs, index.

Les clés étrangères sont ajoutées à la table crédit qui est notre table centrale.



```
# Poucentage de valeurs manquantes supérieures à 40
# Fréquence relative des valeurs uniques sur la variable cible TARGET

nb_li=307511 # ou nb_li = df.shape[0]
i=0

for elem in application.isna().sum():
    if 100*elem/nb_li > 40:
        col=list(application.columns)[i]
        print("Nan sur",col,round(100*elem/nb_li),"%")
        print(application[application[col].isna()]['TARGET'].value_counts(normalize=True))
        print('################################')

    i=i+1
```

B) EXPLORATION

Beaucoup de colonnes ne sont pas directement exploitables car il s'agit de scores créés à partir des autres variables.

Comme nous avons 164 colonnes, l'idée est d'observer le pourcentage de valeurs manquantes ainsi que la corrélation avec notre variable cible. Ce qui va nous permettre lors du preprocessing de directement sélectionner les variables qui ont le plus de sens pour notre analyse.

Limite : nous pouvons faire des corrélations uniquement avec des variables numériques. Dans la suite de notre étude, parmi les variables sélectionnées, nous transformerons les variables catégorielles en variables numériques.

On a autour de 8% de défauts de paiement dans notre base de données.

```
EXT_SOURCE_1      1.000000
DAYS_BIRTH        0.600610
CODE_GENDER       0.306724
FLAG_EMP_PHONE    0.294147
DAYS_EMPLOYED     0.289848
Name: EXT_SOURCE_1, dtype: float64
```

Nous avons recodé les variables catégorielles en variables numériques et nous avons bien 122 colonnes pour faire nos corrélations.

On peut regarder par curiosité quelles variables influencent les EXT_SOURCE et on remarque que DAYS_BIRTH est considérablement corrélée à EXT_SOURCE_1

corr_target_test = corrMatrix['EXT_SOURCE_1'].abs().sort_values (ascending = False) [0:5]corr_target_test

On se rend compte que EXT_SOURCE_1 et un score normalisé à partir des variables clients les plus corrélées, c'est le même principe pour EXT_SOURCE_2 et EXT_SOURCE_3 mais cette fois-ci avec des corrélations sur l'habitation et l'emploi.

C) PRÉPROCESSING

Pour la table CLIENT, c'est le même principe que pour la table CREDIT :

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

On importe le CSV client et on indique vouloir garder toutes les colonnes.

application = pd.read_csv('/Users/p-asalisbury/Dropbox/Mon Mac (MacBook Air de P-A)/Desktop/Chef doeuvre/CSV_MySQL/CLIENT_MySQL.csv')
pd.set_option('display.max_columns',None)
application.head()

```
# On sélectionne les colonnes qui nous intéressent

df = pd.read_csv('/Users/p-asalisbury/Dropbox/Mon Mac (MacBook Air de P-A)/Desktop/Chef doeuvre/CSV_Originaux/

df.rename(columns={'SK_ID_CURR': 'id_client', 'TARGET': 'cible','DAYS_BIRTH':'age','CODE_GENDER': 'genre',
                   'NAME_EDUCATION_TYPE':'niveau_etude','NAME_FAMILY_STATUS':'statut_matrimonial',
                   'CNT_CHILDREN':'nombre_enfants','NAME_TYPE_SUITE':'accompagnant',
                   'AMT_INCOME_TOTAL':'total_revenus','NAME_INCOME_TYPE':'type_revenu',
                   'OCCUPATION_TYPE':'categorie_pro','ORGANIZATION_TYPE':'type_entreprise',
                   'DAYS_EMPLOYED':'anciennete','FLAG_EMP_PHONE':'telephone_pro',
                   'FLAG_OWN_REALTY':'proprietaire','NAME_HOUSING_TYPE':'type_logement',
                   'FLAG_OWN_CAR':'possede_voiture','DAYS_REGISTRATION':'anciennete_banque',
                   'DAYS_ID_PUBLISH':'mise_a_jour_documents',},inplace=True)

pd.set_option('display.max_columns',None)
df.head()
```

On compte les valeurs distinctes dans la colonne type_entreprise
df['type_entreprise'].value_counts()

On vérifie le pourcentage de valeurs manquantes par colonne et on indique vouloir garder toutes les lignes.

```
pd.set_option('display.max_rows',None)
round(df.isnull().sum() / df.shape[0] * 100,2)
```

On peut supprimer les lignes contenant des NaN dans accompagnant (que 0.42 %)
Aussi, on recode 'Other_A' et 'Other_B' en une variable 'Other' = 4

```
df = df.dropna(subset=['accompagnant'])
```

On vérifie toutes les données uniques dans la colonne accompagnant, les NaN ont bien été supprimés.

```
df['accompagnant'].unique()
```

On cherche a remplacer les variables catégorielles (discrètes) par des variables numériques (continues). On prend un exemple avec les colonnes genre, propriétaire et possède_voiture. Il y a deux possibilités :

Avec la fonction **replace** :

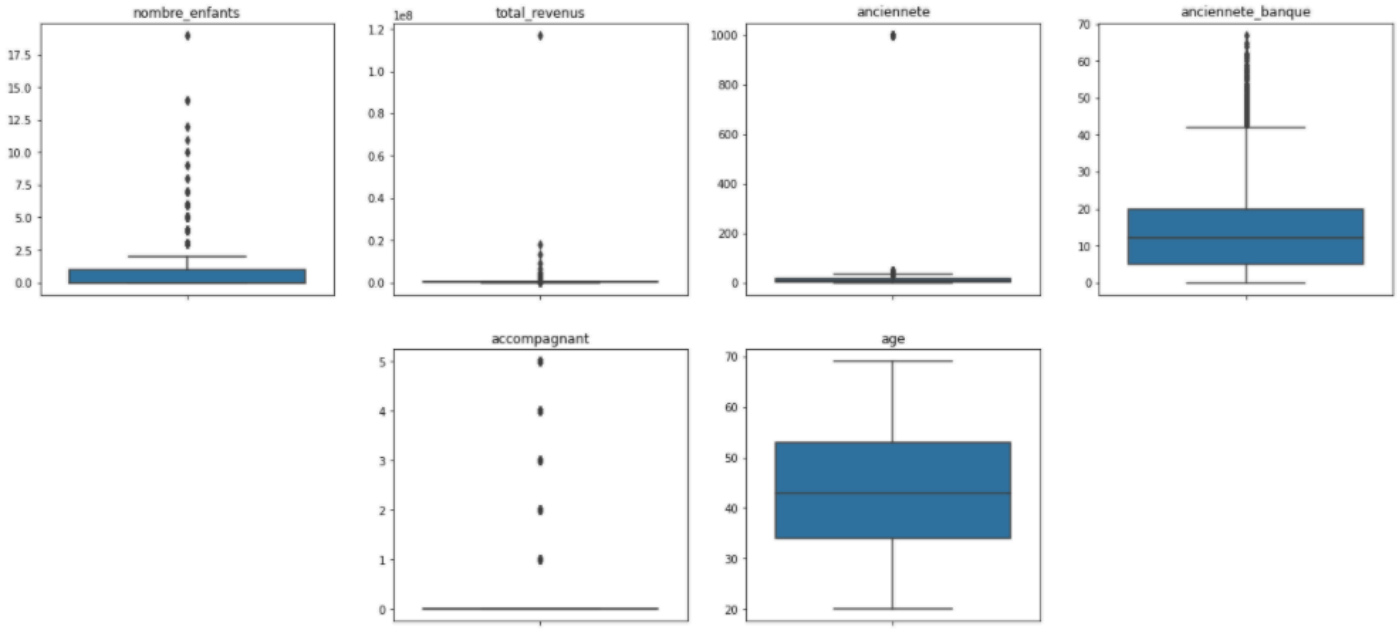
```
df.genre=df.genre.replace(["M","F","XNA"],[0,1,2])
df.proprietaire=df.proprietaire.replace(["Y","N"],[0,1])
df.possede_voiture=df.possede_voiture.replace(["Y","N"],[0,1])
```

Avec la fonction **factorise** qui permet de générer automatiquement des variables numériques (très utile quand on sait que pour type-entreprise il y a 58 valeurs différentes).

```
df['genre'] = pd.factorize(df.genre)[0]
df['proprietaire'] = pd.factorize(df.proprietaire)[0]
df['possede_voiture'] = pd.factorize(df.possede_voiture)[0]
```

On cherche des valeurs aberrantes qui pourraient biaiser notre analyse :

Ci-dessus les quatre premières colonnes semblent comporter des valeurs aberrantes. Les deux dernières non.



Pour la variables enfants on vérifie les clients qui ont plus de 10 enfants. Il n'y a rien d'aberrant.

```
df.loc[df['nombre_enfants'] >10]
```

On vérifie la catégorie professionnelle des clients qui ont un revenu > 10 millions par an. Il y a deux NaN qu'on peut supprimer, dont un ouvrier qui pourrait avoir hérité d'une grosse somme d'argent mais nous préférons le supprimer pour ne pas biaiser notre analyse avec une valeur extrême.

```
df.loc[df['total_revenus'] >10000000]['categorie_pro']
```

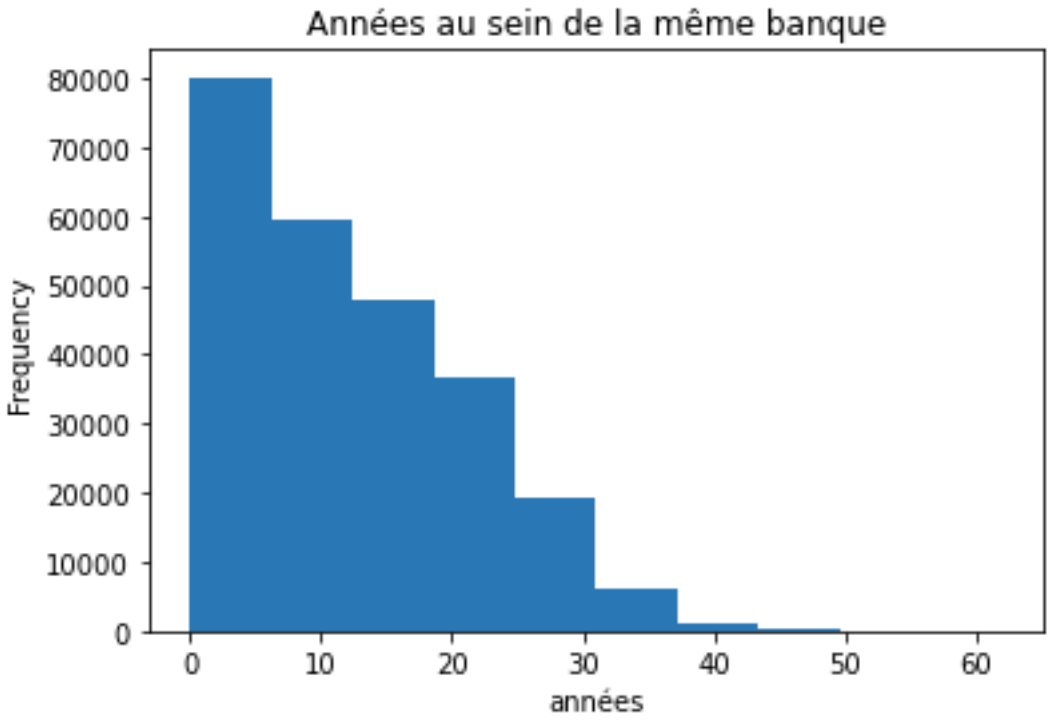
```
On remplace les valeurs aberrantes par des NaN (>1 milliard)
df.loc[df['total_revenus'] >10000000, 'total_revenus'] = np.nan
```

```
On remplace les valeurs aberrantes par des NaN (>10 millions)
On vérifie qu'elles ont bien été remplacées
On peut supprimer les lignes avec des NaN

df.loc[df['total_revenus'] >10000000, 'total_revenus'] = np.nan
df = df.dropna(subset=['total_revenus'])
df.loc[df['total_revenus'] >10000000]['categorie_pro']
```

Ancienneté banque

On fait un histogramme

```
df['anciennete_banque'].plot.hist(title = 'Années au sein de la même banque')
plt.xlabel('années')
```



Pour faire un fichier CSV à partir On exporte notre dataframe nettoyé au format CSV.

```
import csv
df.to_csv('CLIENT_MySQL.csv', index=False, quoting=csv.QUOTE_NONNUMERIC)
```

Exemple recodage de variable

On ajoute une nouvelle colonne qui s'appelle CATEGORIE_AGE. On prend la valeur plancher de « Age » pour obtenir des nombres entiers

```
application['AGE']=abs(application['DAYS_BIRTH'])
slots = ['0-20','20-30','30-40','40-50','50-60','60-70','70 and above']
```

```
bins = [0,20,30,40,50,60,70,100]
application['CATEGORIE_AGE']=pd.cut(application['AGE'],bins,labels=slots)
```

On vérifie le pourcentage de clients dans chaque catégorie d'âge pour qu'il soit bien équilibré

```
application['CATEGORIE_AGE'].value_counts(normalize=True)*100
```

Merge

Pour concevoir notre table finale CREDIT nous avons besoin de regrouper nos deux bases de données : CLIENT issue de application_data et CREDIT issue de previous_application. On importe les deux CSV puis on utilise la fonction MERGE[6]. On importe également les deux CSV créés à partir de MySQL : TYPE_CREDIT et BIEN.

```
dfmerge = pd.merge(dfclient, dfcredit, on="id_client")
dfmerge1 = pd.merge(dfmerge,dftype_credit,  on= "type_credit")
dfmerge2 = pd.merge(dfmerge1,dfbien,  on= "type_bien")
```

On supprime les colonnes dont on a pas besoin à l'aide de la fonction **drop** :

```
dfmerge2=dfmerge2.drop(columns=['cible','genre','age','niveau_etude','statut_matrimonial','nombre_enfants','accompagnant','total_revenus','type_revenu','categorie_pro','type_entreprise','anciennete','telephone_pro','proprietaire','type_logement','possede_voiture','anciennete_banque','mise_a_jour_documents','type_bien'])
```

Enfin, on les classe dans l'ordre souhaité puis on crée le CSV :

```
dfmerge2=dfmerge2[['id_credit','montant_demande_credit','montant_credit_accorde','statut_contrat','nombre_mois_decision','moyen_de_paiement','raison_rejet','id_client','id_type','id_bien']]
```

[6] - Fusionner des objets DataFrame ou Series nommés avec une jointure de style base de données. Un objet Series nommé est traité comme un DataFrame avec une seule colonne nommée. La jointure se fait sur des colonnes ou des index. Si vous joignez des colonnes sur des colonnes, les index DataFrame seront ignorés . Sinon, si vous joignez des index sur des index ou des index sur une ou plusieurs colonnes, l'index sera transmis. Lors de l'exécution d'une fusion croisée , aucune spécification de colonne sur laquelle fusionner n'est autorisée.

Factorize

Cette méthode d'optimisation est utile pour obtenir une représentation numérique d'un tableau lorsque tout ce qui compte est d'identifier des valeurs distinctes. Factoriser est disponible à la fois en tant que fonction de niveau supérieur pandas.factorize(), et en tant que méthode Series.factorize() et Index.factorize().

Quand on a peu de variables avec peu d'informations on peut utiliser la fonction ***replace*** mais dans notre cas - 15 variables de 2 à 58 valeurs uniques - on fait un ***factorize***. Attention tout de même de bien vérifier que nos variables ont été recodées comme on le souhaite. La fonction replace permet aussi une flexibilité appropriée en fonction du recodage choisi.

```
df['genre'] = pd.factorize(df.genre)[0]
df['niveau_etude'] = pd.factorize(df.niveau_etude)[0]
df['statut_matrimonial'] = pd.factorize(df.statut_matrimonial)[0]
df['accompagnant'] = pd.factorize(df.accompagnant)[0]
df['type_revenu'] = pd.factorize(df.type_revenu)[0]
df['categorie_pro'] = pd.factorize(df.categorie_pro)[0]
df['type_entreprise'] = pd.factorize(df.type_entreprise)[0]
df['proprietaire'] = pd.factorize(df.proprietaire)[0]
df['type_logement'] = pd.factorize(df.type_logement)[0]
df['possede_voiture'] = pd.factorize(df.possede_voiture)[0]
```

II/ MISE EN PLACE ET EXPLOITATION DE LA BDD

A) MYSQL WORKBENCH : CRÉATION DES TABLES, INSERTION DES DONNÉES ET AJOUT DES CLÉS ÉTRANGÈRES

A partir du logiciel MySQL Workbench (logiciel de gestion et d'administration de base de données. Nous allons pouvoir modéliser une base de données relationnelles puis créer, modifier analyser des tables en nous connectant au serveur MySQL.

Création de la base de donnée

Exemple avec la table crédit :

```
CREATE TABLE CREDIT (

 id_credit INT PRIMARY KEY NOT NULL,
 montant_demande INT,
 montant_accordé INT,
 statut_contrat INT,
 nombre_mois_decison INT,
 moyen_paiement INT,
 raison_rejet INT,
 id_client INT,
 id_type INT,
 id_bien INT);
```

On a bien les id des autres tables à la fin, ce qui nous permet d'insérer les Foreign Key.

Pour des petites tables, on peut directement insérer les données « à la main » de la manière suivante :

```
INSERT INTO TYPE_CREDIT VALUES (0,0), (1,1),(2,2),(3,3);
```

Exemple d'insertion de fichiers CSV :

```
set global local_infile=true;
SHOW GLOBAL VARIABLES LIKE 'local_infile';


LOAD DATA LOCAL INFILE  '/Users/p-asalisbury/Dropbox/Mon Mac (MacBook Air de P-A)/Desktop/Chef_doeuvre/CSV_MySQL/CLIENT_MySQL.csv'
```

```
INTO TABLE CLIENT
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Pour finir voici comment nous avons ajouté les clés étrangères :

```
ALTER TABLE CREDIT ADD FOREIGN KEY(id_client) REFERENCES CLIENT(id_client);
ALTER TABLE CREDIT ADD FOREIGN KEY(id_type) REFERENCES TYPE_CREDIT(id_type);
ALTER TABLE CREDIT ADD FOREIGN KEY(id_bien) REFERENCES BIEN(id_bien);
```

B) REQUÊTES SANS JOINTURE

Dans la table Client, trouver le nombre de lignes et proposer une méthode pour vérifier qu'il n'y a pas de doublons.

Quand une table contient beaucoup de lignes, difficile de détecter si il y a des doublons à l'oeil nu. Une méthode consiste à : compter le nombre de lignes de la table, supprimer les doublons si il y en a puis re-compter le nombre de lignes.

```
SELECT COUNT(*) FROM CREDITS.CLIENT;                résultat = 251034
SELECT DISTINCT COUNT (*) FROM CREDITS.CLIENT;    résultat = 251034
```

Nous n'avons pas de doublons, il est tout de même important de vérifier.

Pour la table CLIENT, proposer une requête qui permette de calculer la moyenne des revenus des clients qui ont :

a) Honoré leur prêt ou eu un défaut de paiement

```
SELECT ROUND (AVG(total_revenus)) FROM CREDITS.CLIENT;
Résultat = 175309
```

OU

```
SELECT ROUND (AVG(total_revenus)) FROM CREDITS.CLIENT WHERE cible = 0 OR cible = 1;
Résultat = 175309
```

b) Honoré leur prêt

```
SELECT ROUND (AVG(total_revenus)) FROM CREDITS.CLIENT WHERE cible = 0;
```
Résultat = 176344

c) Eu un défaut de paiement

```
SELECT ROUND (AVG(total_revenus)) FROM CREDITS.CLIENT WHERE cible = 1;
```
Résultat = 164416
La moyenne du total des revenus annuel est plus élevée chez les clients qui ont eu un défaut de paiement. On pourrait croire le contraire. Notons que la différence n'est pas extrêmement significative.

Proposer une requête pour extraire les 50 clients les plus récents dans la même banque.

```
SELECT id_client , anciennete_banque
FROM CLIENT
ORDER BY CLIENT.anciennete_banque ASC
LIMIT 50 ;
```

Résultat :	id_client	anciennte_banque
	101771	0
(x48)	…	…
	101647	0

Parmi les 50 premiers clients, tous on moins d'un an d'ancienneté.

Dans la table CRÉDIT, trouver la moyenne des crédits demandés et accordés

```
SELECT AVG(montant_demande) FROM CREDITS.CREDIT;
SELECT AVG(montant_accordé) FROM CREDITS.CREDIT;
SELECT MAX(montant_accordé) FROM CREDITS.CREDIT;
SELECT (montant_accordé) FROM CREDITS.CREDIT ORDER BY
CREDIT.montant_demande ;
```

La moyenne des crédits demandés est de 223219 roupies indienne soit 2548 euros.
La moyenne des crédits accordés est de 223257 roupies indienne soit 2549 euros .

Il n'y a pas de différence significative entre la moyenne des crédits demandés et la moyenne des crédits accordés. Globalement, lorsque le crédit demandé est accordé, il

est accordé au même montant que la demande. Pour simplifier notre table et notre analyse nous prendrons en compte uniquement le crédit accordé.

C) REQUÊTES AVEC JOINTURE

Jointure simple

Proposer une jointure simple qui permette d'afficher l'id client avec la cible = 0, le genre homme, le statut matrimonial divorcé et le statut contrat refusé. Ainsi nous pourrons quantifier le nombre de clients concernés.

```
SELECT CLIENT.id_client
FROM CLIENT
INNER JOIN CREDIT ON CLIENT.id_client = CREDIT.id_client
WHERE CLIENT.cible = 0 AND CLIENT.genre = 0 AND CLIENT.statut_matrimonial = 4
AND CREDIT.statut_contrat = 1;
```

Nombre de clients concernés : 2337

	id_client
0	100079
...	...
2336	455594

2337 rows

Jointure complexe

Proposer une jointure complexe qui permette d'afficher l'id client, la cible et le type de bien financé correspondant à bijoux. Comptabiliser le nombre de clients.

```
SELECT CLIENT id_client, CLIENT.cible, BIEN.type_bien
FROM CLIENT
INNER JOIN CREDIT ON CLIENT.id_client = CREDIT.id_client
INNER JOIN BIEN ON CREDIT.id_bien = BIEN.id_bien
WHERE BIEN.type_bien = 1;
```

Nombre de clients concernés : 2337

cible	genre	type_bien	
0	0	1	1
...	...	...	...
4481	0	0	1

4482 rows

III/ ANALYSE ET VISUALISATION

A) ANALYSE UNIVARIÉE ET BIVARIÉE

Analyse univariée

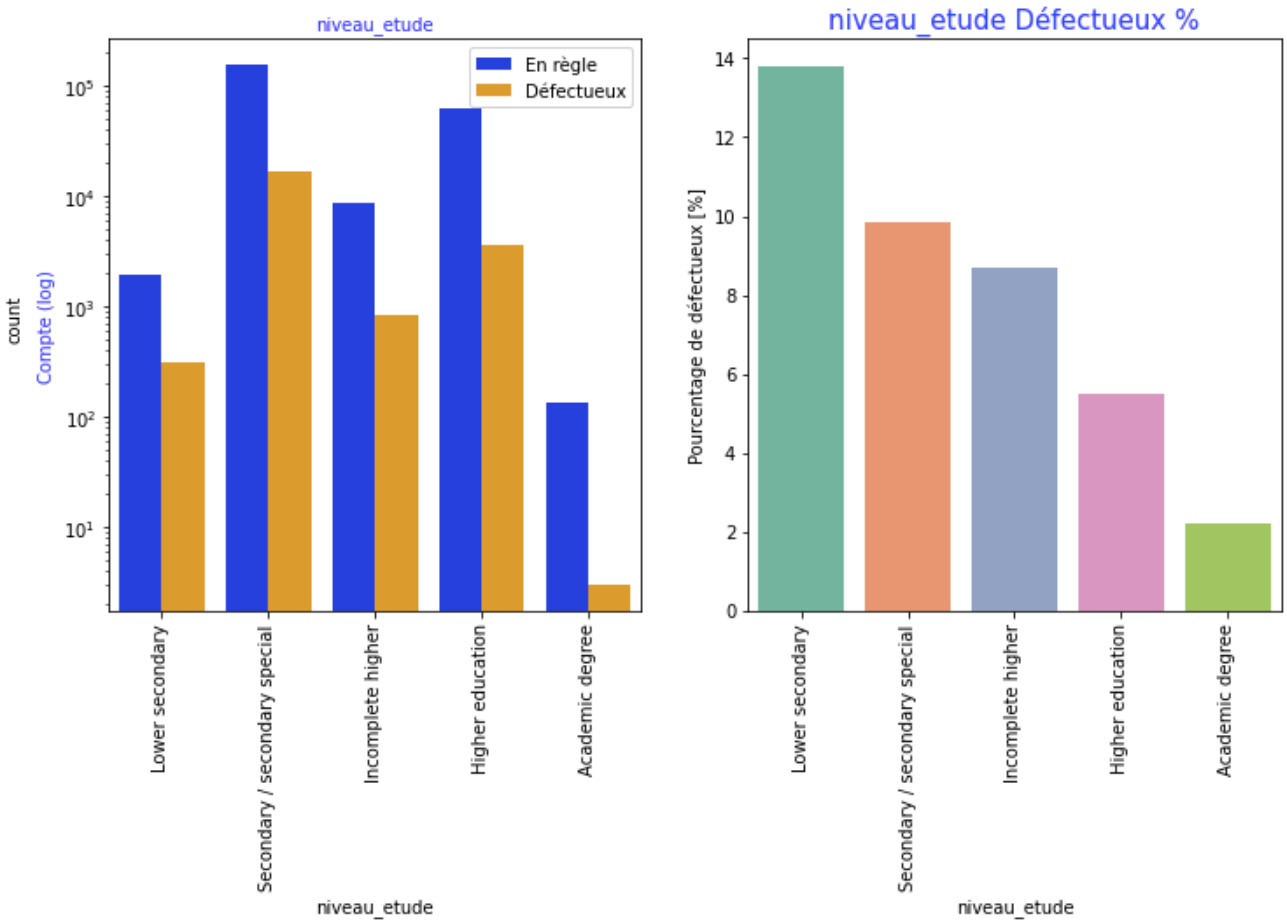
L'analyse univariée est une forme simple d'analyse statistique . Elle peut être inférentielle ou descriptive. Le fait essentiel est qu'une seule variable est impliquée. L'analyse univariée peut donner des résultats trompeurs dans les cas où l'analyse bivariée ou multivariée est plus appropriée dans certains cas.

Afin de répondre à la demande de notre client, nous allons étudier toutes les colonnes de la table CLIENT, une colonne pertinente de la table CREDIT et les colonnes des tables TYPE CREDIT et BIEN. Pour la table client nous proposons un extrait des graphiques les plus utiles : l'âge, le niveau d'étude, et la catégorie professionnelle. Notons que tous les graphiques sont accessibles sur le notebook analyse univariée.

Pour l'**âge**, nous avons préalablement recodé la colonne en catégories d'âges. Nous n'avons ni clients en dessous de 20 ans ni clients au dessus de 70 ans. Globalement, plus l'âge augmente moins il y a de défauts de paiement. La catégorie d'âge la plus à risque est celle des 20-30 ans : 11 % de clients défectueux alors que les 60-70 sont à 4%.

Concernant le **genre**, on constate une différence de 2,5 points de pourcentage. Selon notre dataset, les femmes honorent plus souvent leur.s crédit.s que les hommes.

Plus le **niveau d'étude** est élevé, plus les défauts de paiement sont rares.



Concernant la **situation matrimoniale**, les veufs, veuves sont les plus en règles. Les célibataires ont plus tendance à avoir des défauts de paiement. Point à noter on observe une différence entre mariage civil et religieux.

Lorsqu'il s'agit du **nombre d'enfants**, les clients ayant beaucoup d'enfants ont très souvent des défauts de paiement mais nous avons pas assez de données pour inférer les résultats. Cependant, il est toujours possible d'émettre une réserve au regard de l'importance des défaillances. On peut faire l'hypothèse que au delà de 6 enfants, les défauts de paiement vont croissant.

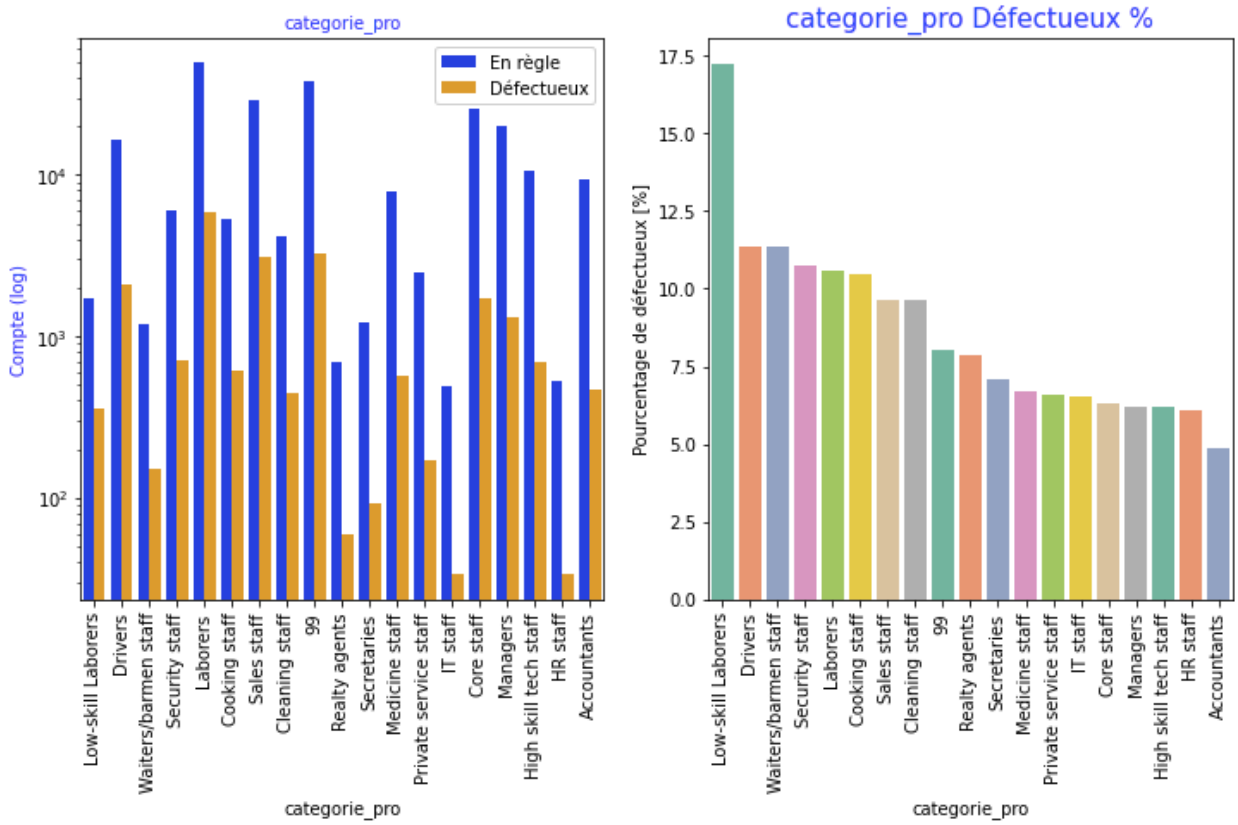
A propos de l'**accompagnateur**, pas de différence assez significative. Mais lorsqu'un

proche est identifié comme accompagnateur, il y a une légère tendance à respecter le contrat.

Concernant le **revenu**, on constate une dégressivité en fonction du revenu. Globalement les personnes dans les premiers déciles (faibles revenus) ont des défauts de paiement autour des 9 %. Les clients dans les derniers déciles (hauts revenus) sont autour des 7 %.

Pour les **types de revenus**, une catégorie très risquée avec 40 % de défauts de paiement : les femmes en congé maternité. Limite : on a pas les hommes. Congé maternité : sous groupe de femmes qui est surreprésenté parmi les défauts de paiement. Catégorie irréprochable avec aucun défaut de paiement : les hommes d'affaires, les titulaires d'une pension et les étudiants. ÉTUDIANT : sous groupe de jeunes qui pourtant remboursent leur prêt. Aussi, pour ces 4 catégories nous avons entre 5 et 12 individus à chaque fois ce qui nous permet pas vraiment d'inférer les résultats.

Intéressons nous à la catégorie professionnelle. Une catégorie est surreprésentée parmi les clients défectueux, celle des ouvriers peu qualifiés. Il y a une corrélation entre la hiérarchie du statut et les défauts de paiement. Un statut élevé est souvent synonyme de plus de rigueur . Néanmoins, le statut élevé n'exclut pas les défauts de paiement. Chez les ouvriers peu qualifiés, nous sommes à 17,4% de défauts de paiement alors que chez les comptables nous observons des défauts de paiement en dessous de la barre des 5%.



A propos du **type d'entreprise** 4 : banque et assurance , 1 : religieux : catégories où il y a le moins de défauts de paiement (- de 6%). 6 : secteur primaire, 2 : auto-entrepreneur : catégorie la moins rigoureuse (+ de 10%).

Concernant l'**ancienneté** au sein de la même **entreprise**, c'est flagrant ; plus l'ancienneté est élevée, moins il y a de défauts de paiement. On est au dessus des 10 % pour les nouveaux arrivants et on passe en dessous de la barre des 4 % pour les 20 ans et plus.

Le **téléphone** portable/professionnel ne semble pas être un indicateur pertinent. Même chose pour le fait d'être **propriétaire** ou non, on n'observe pas de différence significative.

Par contre, pour le **type logement** (si le client est propriétaire ou locataire d'un appartement ou d'une maison). Les locataires d'appartements ont plus de défauts de paiement.

Les clients qui possèdent une **voiture** ont un peu plus de défauts de paiement. Différence de 1 point de pourcentage.

Plus l'**ancienneté** au sein de la même banque est élevé moins les défauts de paiement sont nombreux. Cependant la différence est moins significative que l'ancienneté au sein de la même entreprise.

Enfin, pour la **mise à jour des documents** : Combien de jours avant la demande la personne a commencé son emploi actuel, temps seulement relatif à la demande. On remarque que les clients qui viennent de commencer leur emploi sont plus susceptibles d'avoir des défauts de paiement.

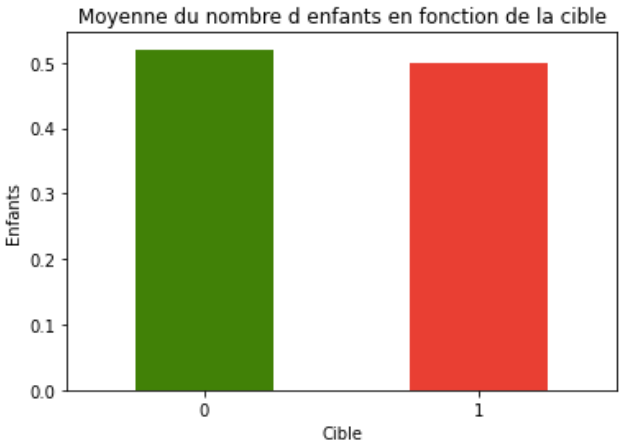
Connexion à MySQL Workbench via un notebook pour écrire en langage SQL et Python

[120]:
```
import mysql.connector
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rc
import seaborn as sns
%matplotlib inline
import mysql.connector as mysqlConnector
import sqlite3
import pandas as pd

connexion = mysql.connector.connect(host='127.0.0.1',
                         user='root',
                         password='xxxxxxxxxxxxxx',
                         database='CREDITS')
if connexion:
    print("Connexion à la base de données CREDITS réussie")
else:
    print("Connexion échouée")
```

```
Connexion à la base de données CREDITS réussie
```

Exemple de requête : on cherche à trouver le nombre moyen d'enfants en fonction de la cible.

```
query4 = 'SELECT cible, ROUND( AVG(nombre_enfants),2) AS MPF FROM CREDITS.CLIENT GROUP BY cible '

df4=pd.read_sql(query4,connexion)
df4.MPF.plot(xlabel='Cible', ylabel='Enfants',color=['g','r'], rot = 0,
                kind='bar',title ='Moyenne du nombre d enfants en fonction de la cible');
```



Analyse bivariée

L'analyse bivariée[7] est l'une des formes les plus simples d'analyse quantitative (statistique) . Elle implique l'analyse de deux variables (souvent désignées par X , Y), dans le but de déterminer la relation empirique entre elles.

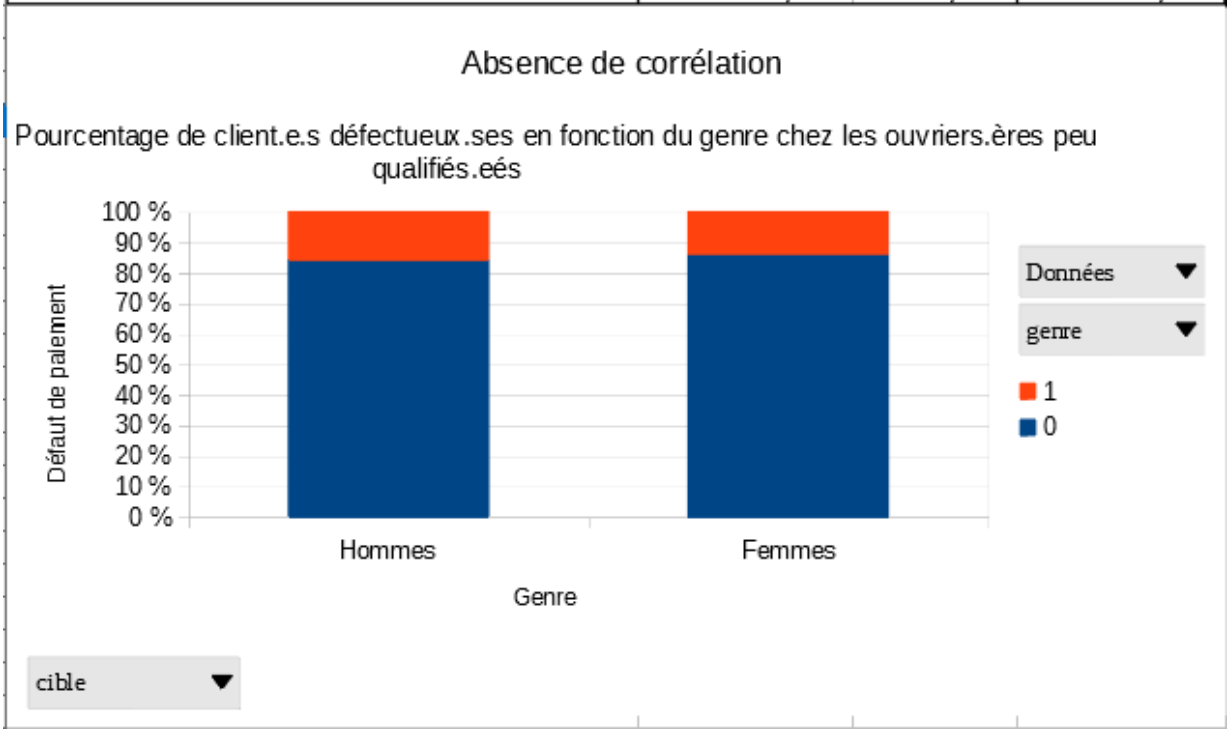
Croisons les **ouvriers peu qualifiés** avec le **genre** :

Somme - Ouvriés peu qualifiés	Données		
cible	0	1	Total Résultat
0	18915	3510	**22425**
1	4030	637	**4667**
Total Résultat	**22945**	**4147**	**27092**

Tendance à peine perceptible, les ouvriers ont légèrement plus de défauts de paiement que les ouvrières. Il n'y a pas de corrélation mais on observe une différence de 2 points de pourcentage entre les hommes et les femmes. On retrouve plus de défauts de paiement chez les hommes : 15,65% contre 13,65% pour les femmes. On est au dessus de la moyenne pour les deux.

[7] - Elle peut être utile pour tester des hypothèses d'association simples. L'analyse bivariée peut aider à déterminer dans quelle mesure il devient plus facile de connaître et de prédire une valeur pour une variable (éventuellement une variable dépendante) si nous connaissons la valeur de l'autre variable (éventuellement la variable indépendante).

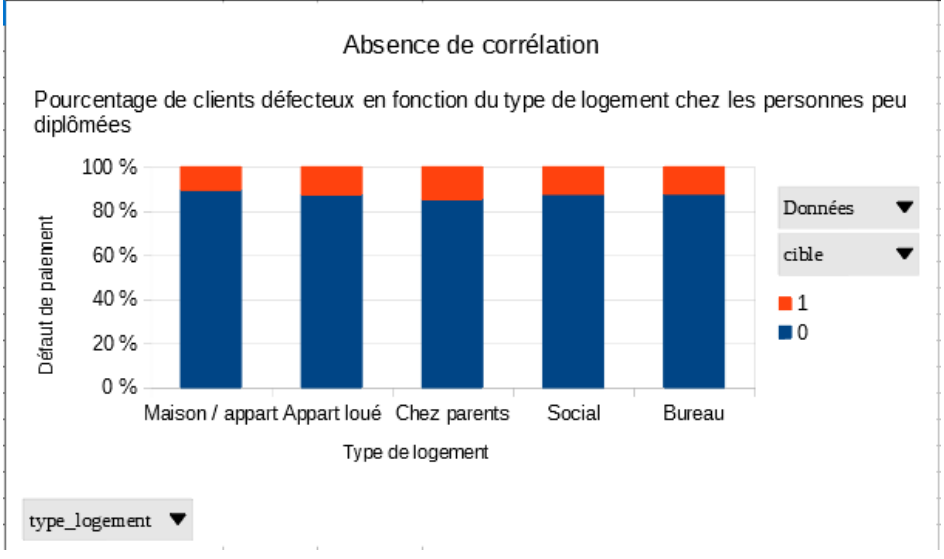
Somme - Ouvriés peu qualifiés	Données		
cible	0	1	Total Résultat
Hommes	84,35 %	15,65 %	100,00 %
Femmes	86,35 %	13,65 %	100,00 %
Total Résultat	**84,69 %**	**15,31 %**	**100,00 %**



Croisons le logement et le niveau d'étude :

Somme - niveau_etude	Données					
cible	0	1	2	3	4	Total Résultat
0	8586	207	567	729	108	**10197**
1	1005	30	99	102	15	**1251**
Total Résultat	**9591**	**237**	**666**	**831**	**123**	**11448**

Somme - niveau_etude	Données		
type_logement	0	1	Total Résultat
Maison / appart	89,52 %	10,48 %	**100,00 %**
Appart loué	87,34 %	12,66 %	**100,00 %**
Chez parents	85,14 %	14,86 %	**100,00 %**
Social	87,73 %	12,27 %	**100,00 %**
Bureau	87,80 %	12,20 %	**100,00 %**
Total Résultat	**89,07 %**	**10,93 %**	**100,00 %**

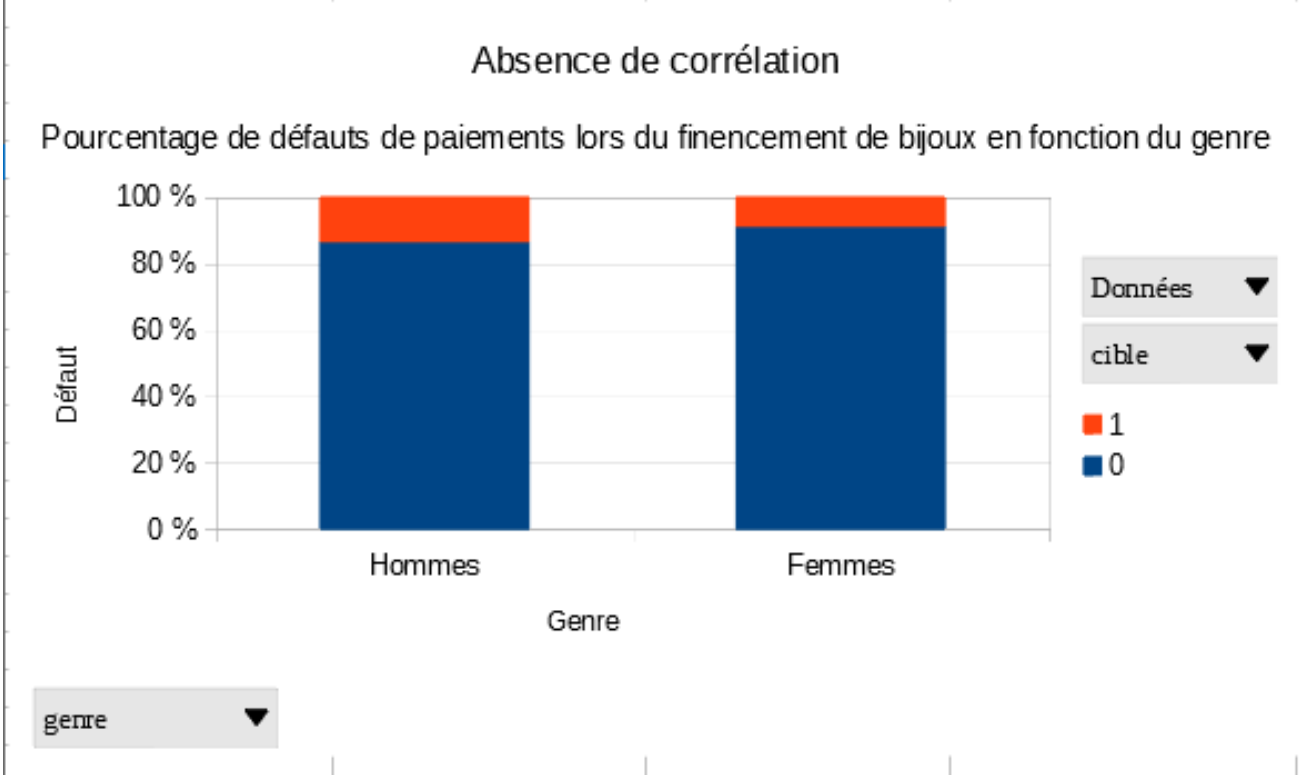


On observe une infime tendance : les clients propriétaires de leur logement sont plus réguliers dans le paiement de leur crédit.

Enfin, le financement de bijoux avec le genre :

Somme - type_bien	Données		
cible	0	1	Total Résultat
0	1056	2986	4042
1	161	279	440
Total Résultat	**1217**	**3265**	**4482**

Somme - type_bien	Données		
genre	0	1	Total Résultat
Hommes	86,77 %	13,23 %	100,00 %
Femmes	91,45 %	8,55 %	100,00 %
Total Résultat	**90,18 %**	**9,82 %**	**100,00 %**



Il n'y a pas de corrélation mais on observe une différence de 4,7 points de pourcentage entre les hommes et les femmes. On retrouve plus de défauts de paiement chez les hommes : 13,23% contre 8,55% pour les femmes. On est au dessus de la moyenne pour le deux. Pour le femme, le résultat est beaucoup plus proche de la moyenne à 8%. Une a une variable cachée qui est le genre : il est plus déterminant que le fait de financer des bijoux.

Pour nos trois graphiques, on mesure notre corrélation avec un coefficient de corrélation. On utilise la fonction du même nom. Si le résultat est négatif il y a une anti corrélation, ce qui est une forme de corrélation inversée. Plus le résultat est proche de 0 moins il y a de corrélation, plus le résultat est proche de 1 plus la corrélation est forte. On peut aussi calculer le coefficient de détermination ; c'est le coefficient de corrélation mis au carré. Dans nos trois exemples, nous n'avons pas trouvé de corrélation significative.

Si on fait l'hypothèse que le défaut de paiement est déterminé par le genre.

Par exemple pour le graphique sur le financement de bijoux, le coefficient de corrélation est de - 0,07 : il n'y a pas de corrélation. Avec le coefficient de détermination, 0,49%, des défauts de paiement lors du financement de bijoux est déterminé par le genre. On observe tout de même une différence de 4,7 points de pourcentage entre les hommes et les femmes. L'hypothèse n'est pas validé.

type_bien : 0 = NSP 1 = bijoux 2 = maison 3 = numérique 4 = medical 5 = véhicule 6 = vêtements accessoires 7 = audio/video 8 = détentes 9 = autres

On

Pour faire notre analyse bivariée, il a été utile de des tableaux croisés de fréquences en effectifs et en pourcentage d'effectifs. Nous avons également calculé le pourcentage par total de colonne et par ligne.

B) MODÈLE DE RÉGRESSION LINÉAIRE

On va écrire une fonction pour prédire si notre client va avoir un défaut de paiement ou non. On commence par s'assurer de la fiabilité de notre model.

```
: from sklearn.neighbors import KNeighborsClassifier
```

```
: model = KNeighborsClassifier() # On vient de charger un model de classification
```

```
: y = rl['cible'] # On sélectionne la colonne cible
  X = rl.drop('cible', axis=1) # On sélectionne toutes les colonnes sauf cible
```

```
: model.fit(X,y)
  model.score(X,y)
```

```
0.8861540129710828
```

Secondary / secondary special = 0 Higher education = 1 Incomplete higher = 2 Lower secondary = 3 Academic degree = 4

Laborers = 0 Core staff = 1 Accountants = 2 Managers = 3 Drivers = 4 Sales staff = 5 Cleaning staff = 6 Cooking staff = 7 Private service staff = 8 Medicine staff = 9 Security staff = 10 High skill tech staff = 11Waiters/barmen staff = 12 Low-skill Laborers = 13 Realty agents = 14 Secretaries = 15 IT staff = 16 HR staf = 17

```
def defaut(model,niveau_etude=2, genre=0, age=31, statut_matrimonial=0, categorie_pro=0 ):

    x = np.array([niveau_etude, genre, age, statut_matrimonial, categorie_pro]).reshape(1,5)

    print(model.predict(x))
    print(model.predict_proba(x))

defaut(model)
```

On estime si notre client va avoir un défaut de paiement mais ce n'est pas suffisant ..
.. on calcule la probabilité qu'il a d'avoir un défaut de paiement

```
[0]
[[0.8 0.2]]
```

Pour un homme de 31 ans célibataire et ouvrier notre client a 80 % de chance d'honorer son prêt et 20 % d'avoir un défaut de paiement.

A partir de nos données, nous avons extrait de la connaissance sur le respect des mensualités des client de la banque X. Les informations recueillies nous paraissent sensée mais pour aller plus loin, il s'agit comparer notre analyse avec la réalité.

Confronter notre méthode au réel

On fait une recherche sur les facteurs de base[8] pris en compte pour le credit scoring[9]. Pour les particuliers : âge, nationalité (Français, Union Européenne, autre), situation familiale, régime matrimonial département de résidence, type d'habitat, situation de logement (locataire, propriétaire, hébergé), ancienneté dans le logement, catégorie socioprofessionnelle, situation professionnelle, ancienneté professionnelle, type de téléphones utilisés, utilisation de l'email, relations entre les co-emprunteurs (vie de couple, amis, famille, collègues…)

Cette technique doit désormais apporter encore plus d'informations aux établissements de crédit. Le scoring permet de discriminer[10] les emprunteurs qui seront capables de rembourser et ceux qui auront des chances d'être défaillants. Cette évolution ne s'arrête pas là car le crédit-scoring doit à terme mettre en valeur les acteurs qui sont intéressants ou non à prêter. Dans cette optique Wallis explique que le scoring est une méthode d'estimation de l'intérêt d'un crédit et non du risque. Dionne, Artis et Guillen abordent les notions de gains et de coût du crédit via les scores, ce qui modifie l'approche traditionnelle du scoring par rapport aux risques.

En lisant la liste, ils détiennent déjà un important nombre d'informations, mais vous noterez qu'on n'y voit pas la liste des crédits en cours. Cette liste est enrichie par les banques avec le profil client qu'ils détiennent. C'est une des raisons qui fait qu'il soit plus facile d'obtenir un prêt dans votre banque qu'ailleurs : ils détiennent plus d'informations, qui leur permet d'évaluer le risque de crédit que le demandeur de crédit représente.

Les clients qui se sont vus refuser un financement après un crédit scoring peuvent demander une deuxième évaluation, cette fois-ci non automatisée, où le client pourra apporter des informations supplémentaires qui lui permettront d'obtenir le crédit.

---

8 - https://ekonomia.fr/investir/avis-credit/credit-scoring-comment-les-banques-donnent-un-accord-de-credit/, José Da Silva.

9 - VAN PRAAG N, (1995), Credit management et credit scoring, Paris, Economica (Collection gestionpoche), p112

10 - Le crédit scoring repose sur la saisie d'informations sur le demandeur de crédit. Ces informations, détenues dans une base de données, sont encadrées par la Loi. La CNIL, Commission Nationale de l'Informatique et des Libertés, « est une institution indépendante chargée de veiller au respect de l'identité humaine, de la vie privée et des libertés dans un monde numérique ». Le point central de la sauvegarde du respect de notre vie privée réside dans l'impossibilité de l'utilisation des données n'ayant pas de rapport avec l'aspect économique et financier du crédit scoring pour disqualifier ou exclure le demandeur de crédit. La CNIL fixe ainsi les limites de ce qui peut servir en tant que données pour le crédit scoring.

Toutes les données utilisées pour le crédit scoring doivent être effacées après l'acceptation de la demande de crédit. Ces données ne peuvent pas être utilisées à d'autres fins que le contrat de crédit lui-même, sauf autorisation du demandeur de crédit : faites attention à ce que vous signez !

C) TABLEAU DE BORD AVEC POWER BI

Proposer un dashboard qui permette d'analyser simultanément les données des différentes tables.

Pour la table client, nous devons pouvoir sélectionner les différentes valeurs de la cible, du genre, du niveau d'étude, du nombre d'enfants et de la catégorie professionnelle.

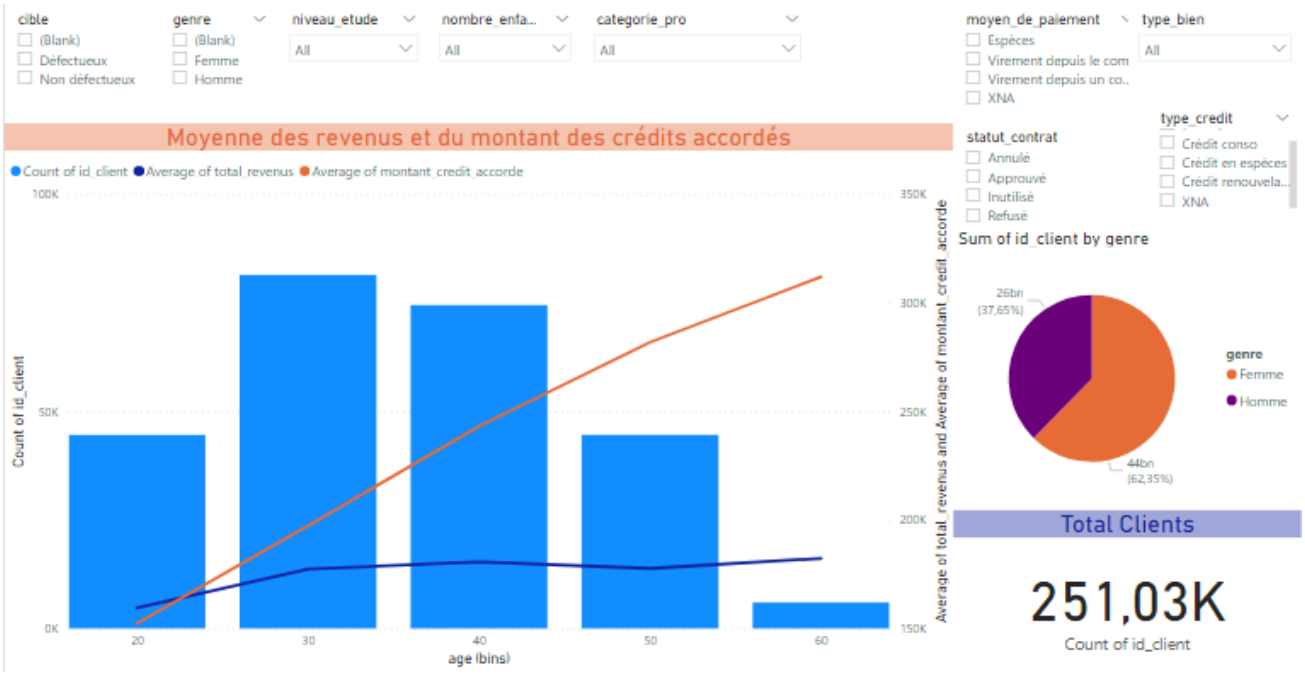
Pour la table crédit, le moyen de paiement et le statut du contrat.
Pour la table type crédit le type de crédit. Pour la table bien les différents biens.

On importe notre fichier CSV traité. On utilise power query pour transformer la structure de nos données en recodant les variables souhaitées. On crée les slicer : cible, niveau étude, genre, statut matrimonial que l'on met au format liste. On crée la carte : total clients (id_client). On crée des catégories d'âge (bins) avec line and stack column chart. Ce qui nous permet d'avoir à la fois un graphique linéaire et un histogramme sur le même visuel.

Nous proposons un histogramme dynamique avec la moyenne des revenus et du montant des crédits accordés par catégorie d'âges de 10 ans. On crée un diagramme avec le total des clients en fonction du genre et une carte qui comptabilise le nombre de clients en fonction des critères sélectionnés.

Lorsqu'on prend on considération l'ensemble de nos clients, les revenus on tendance à augmenter avec l'âge mais ce n'est pas flagrant. En revanche, on constate une nette augmentation de la moyenne des montants accordés en fonction de l'âge.

Certes les montants demandés sont plus élevés mais aussi, on peut supposer que la banque fait davantage confiance aux personnes plus âgés. En effet comme vu précédemment lors des analyses univariées les personnes plus âgées ont moins de défauts de paiement.



Slicer genre - femme - : quelle que soit la catégorie d'âge, on constate que le total des revenus moyen des hommes est toujours plus élevé que celui des femmes.

Slicer catégorie professionnelle : Pour les comptables (rappelons qu'ils font partie des clients les moins défectueux) la moyenne des crédits accordés est nettement plus élevée. On remarque un peu par hasard qu'il y a a peine 3% d'hommes parmi les comptables.

Slicer statut contrat : il n'y a pas de différence significative entre les contrats approuvés et refusés concernant les salaires. En revanche, on observe que les montants des crédits refusés sont beaucoup plus élevés que les autres statuts de contrat. On peut faire l'hypothèse que les crédits sont souvent refusés lorsque la demande est trop conséquente.

CONCLUSION

Au regard de nos observations, la banque est cohérente par rapport aux prêts qu'elle accorde. Les prêts sont plus facilement accordés aux profils qui ont des bonnes capacités de remboursement.

L'analyse univariée est sinon suffisante du moins très explicite. Elle permet de faire des corrélations entre le profil des client et la fiabilité des remboursements. Dans la grande majorité des cas, il est possible d'inférer les résultats.

L'analyse bivariée nous permet de trouver des variables cachées mais au regard de notre projet, pour le financement de bijoux, le genre est tout de même déterminant : contrairement aux hommes, les femmes qui financent des bijoux on des défaut de paiement proches de la moyenne à 8%.

Notre analyse permet à la banque de faire un profilage des client et d'établir une note en fonction de risque de défaut de paiement.

Pour aller plus loin

- On peut explorer les différentes catégories pro différenciées par un numéro pouvant aller de 1 à 12. Par exemple : industrie 1, industrie 2 … industrie 12 que nous avons redoré en industrie. Même chose avec affaires et commerce.

- Ajouter une table banque avec les différentes banques et une table Pays avec les différentes villes

- Utiliser d'autre modèles de prédiction pour faire du scoring

- Attribuer une note aux clients

BIBLIOGRAPHIE ET SITOGRAPHIE

- VAN PRAAG N, (1995), Credit management et credit scoring, Paris, Economica (Collection gestionpoche), p112.

- eKonomia, Économiser sur le crédit, *Crédit Scoring : comment les banques donnent un accord de crédit*, José Da Silva, 2019, [consulté le 27 juillet 2021]. Disponible sur : https://ekonomia.fr/investir/avis-credit/credit-scoring-comment-les-banques-donnent-un-accord-de-credit/.

- MySQL Documentation Officielle, *7.4.1 Dumping Data in SQL Format with mysqldump*, David Axmark, dernière version : 2021, [consulté le 16 juillet 2021]. Disponible sur https://dev.mysql.com/doc/refman/5.7/en/mysqldump-sql-format.html.

- MySQL Documentation Officielle, *6.2 Users and Privileges*, David Axmark, dernière version : 2021, [consulté le 16 juillet 2021]. Disponible sur https://dev.mysql.com/doc/workbench/en/wb-mysql-connections-navigator-management-users-and-privileges.html.

- MySQL Documentation Officielle, *6.2 5.1 Connecting to MySQL Using Connector/Python*, David Axmark, dernière version : 2021, [consulté le 16 juillet 2021]. Disponible sur https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html.

- SQL.SH, *Cours et tutoriels sur le langage SQL*, Tony Archambeau, 2014, [consulté le 15 juillet 2021]. Disponible sur https://sql.sh/.

ANNEXES

Explication des variables classées par tables

CLIENT

cible : 0 = paiement honoré 1 = défaut de paiement

age : en année.s puis en catégories d'âge pour l'analyse univariée

genre 0 = homme 1 = femme

niveau étude : 0 = secondaire 1 = études supérieures 2 = études supérieures incomplètes 3 = études secondaires incomplètes

statut matrimonial : 0 = célibataire 1 = mariage religieux 2 = mariage civil 3 = veuf.ve 4 = séparé

nombre d'enfants : de 0 à 19

accompagnant : 0 = non accompagné.e 1 = famille 2 = époux.se, partenaire 3 = enfant.s 4= autre.s

total revenus : de 25650 à 9 000 000 roupies

catégorie pro : 0 = ouvriers 1 = personnels de base 2 = comptables 3 = gestionnaires 4 = chauffeurs 5 = vendeurs.euses 6 = personnels de ménage 7 = personnels de cuisine 8 = personnel de service privé 9 = personnels de service médical 10 = personnels de sécurité 11 = personnels technique hautement qualifié 12 = personnel de restauration 13 = ouvriers peu qualifiés 14 = agents immobiliers 15 = secrétaires 16 = personnels informatique 17 = personnel RH

type entreprise : 0 = autre 1 = religion 2 = auto-entrepreneur 3 = médecine 4 = banque, assurance 5 = entretien 6 = primaire 7 = privé 8 = affaires 9 = transport 10 = commerce 11 = industrie 12 = public

ancienneté : de 0 à 49 ans

téléphone pro : oui ou non

propriétaire : oui ou non

type logement : 0 = maison, appartement, collocation (locataire) 1 = chez les parents
2 = logement social 3= appartement bureau

possède voiture : oui ou non

ancienneté banque : de 0 à 70 ans

mise à jour documents : nombres de jours entre l'application et l'enregistrement

CRÉDIT

montant demande crédit : de 3456 à 5 850 000
montant crédit accordé : de 3456 à 5 850 000
statut contrat : 0 = approuvé 1 = refusé 2 = inutilisé 3 = annulé
nombre mois décision : de 0 à 97

moyen de paiement : 0 = espèces 1 = XNA 2 = virement depuis le compte personnel
3 = virement depuis le compte de l'employeur

raison rejet : 0 = XAP 1 = HC 2 = limite 3 = client 4 = SCOFR 5 = SCO 6 = XNA
8 = vérification 8 = system

TYPE CRÉDIT

consumer loans (729151) = crédit à la consommation =0
cash loans (747553) = crédit en espèce =1
revolving loans (193164) = crédit renouvelable = 2

BIEN

type_bien : 0 = NSP 1 = bijoux 2 = maison 3 = numérique 4 = medical 5 = véhicule
6 = vêtements accessoires 7 = audio/video 8 = détentes 9 = autres