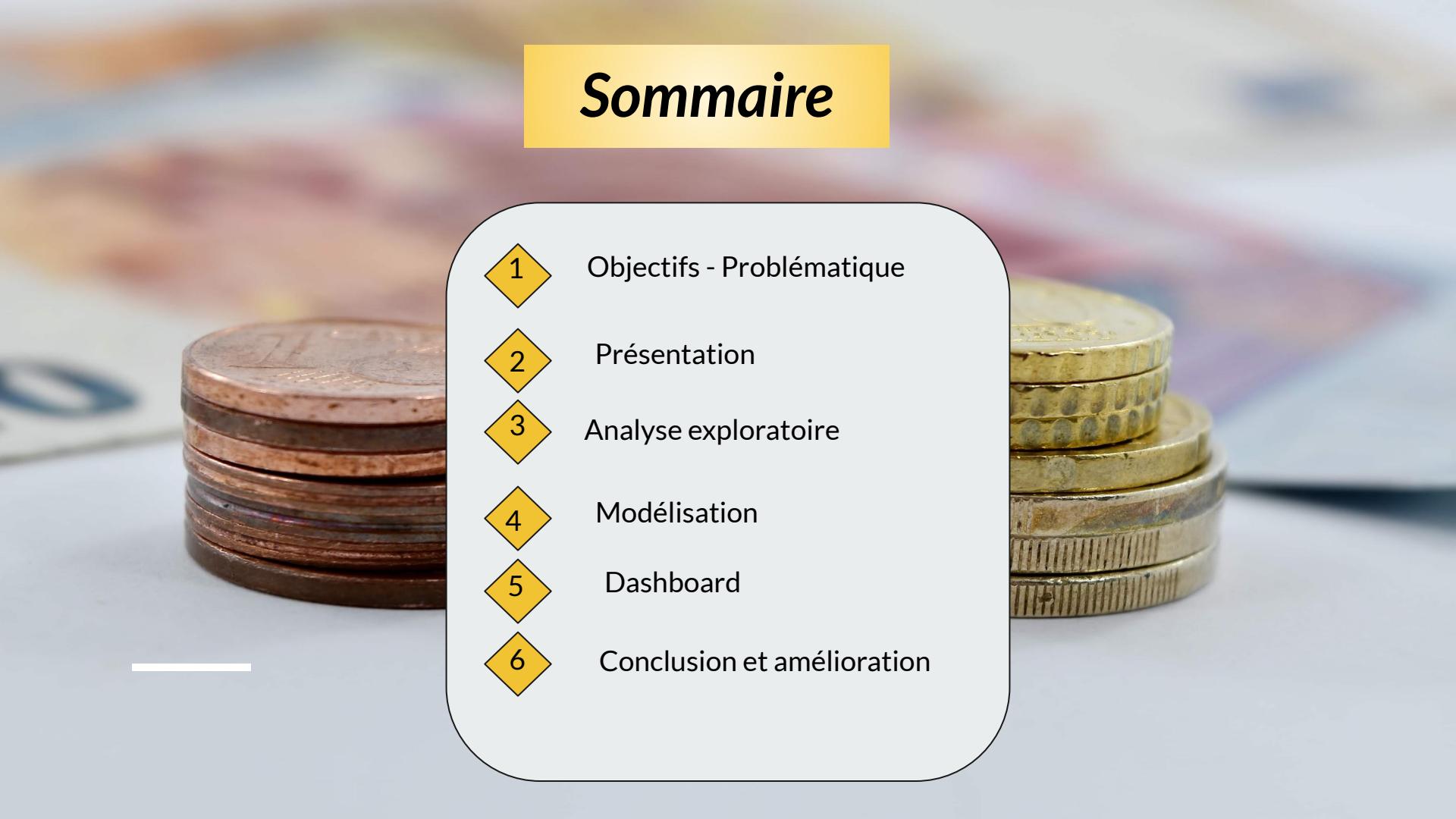




## P7: Implémentez un modèle de scoring

Mitra DADGAR - Data Scientist  
Openclassroom  
Novembre 2022

# Sommaire

- 
- 1 Objectifs - Problématique
  - 2 Présentation
  - 3 Analyse exploratoire
  - 4 Modélisation
  - 5 Dashboard
  - 6 Conclusion et amélioration

## Contexte

Data Scientist au sein d'une société financière, nommée "Prêt à dépenser", qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt

## Objectifs

- Étayer la décision d'accorder ou non un prêt à un client potentiel.
- Expliquer de façon la plus transparente possible les décisions d'octroi de crédit.
- Permettre aux clients de disposer de leurs données personnelles et de les explorer facilement.

Prêt à dépenser

## Mission

- Développer d'un modèle “scoring crédit” pour calculer la probabilité qu'un client rembourse son crédit
- Développer un dashboard interactif

# Présentation

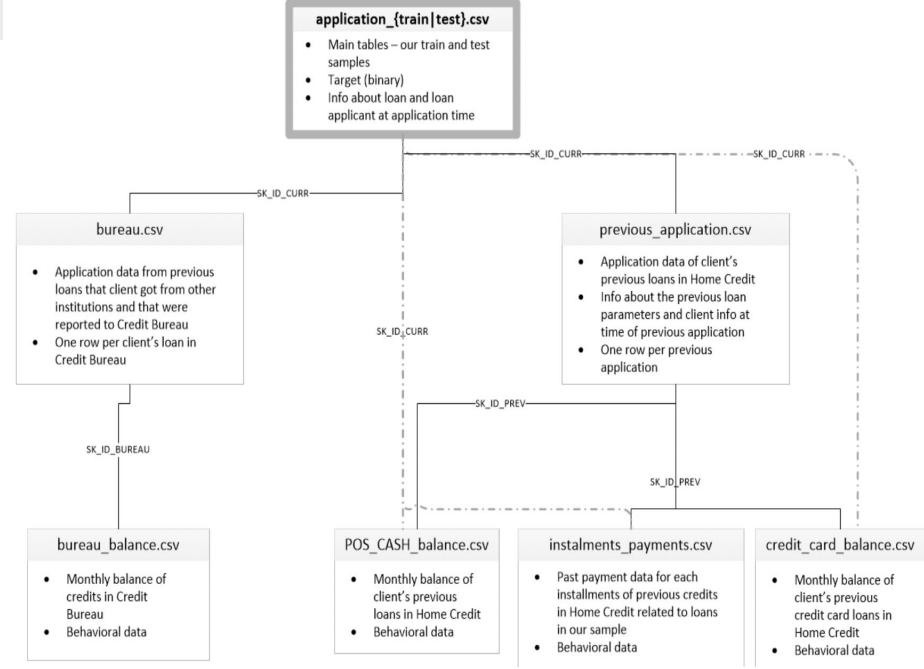
- 7 jeux de données
- Base de données principale : application\_train avec 122 features

```
app previous Shape: (1670214, 37)
installments payments Shape: (13605401, 8)
sample submission Shape: (48744, 2)
bureau Shape: (1716428, 17)
bureau balance Shape: (27299925, 3)
POS CASH balance Shape: (10001358, 8)
credit card balance Shape: (3840312, 23)
```

Training Data Shape: (307511, 122)  
Testing Data Shape: (48744, 121)

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...	FLAG_DOCUMENT_18	FLAG_DOCUMENT_19	...
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	...	0	0	
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	...	0	0	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	...	0	0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	...	0	0	
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	...	0	0	

5 rows x 122 columns

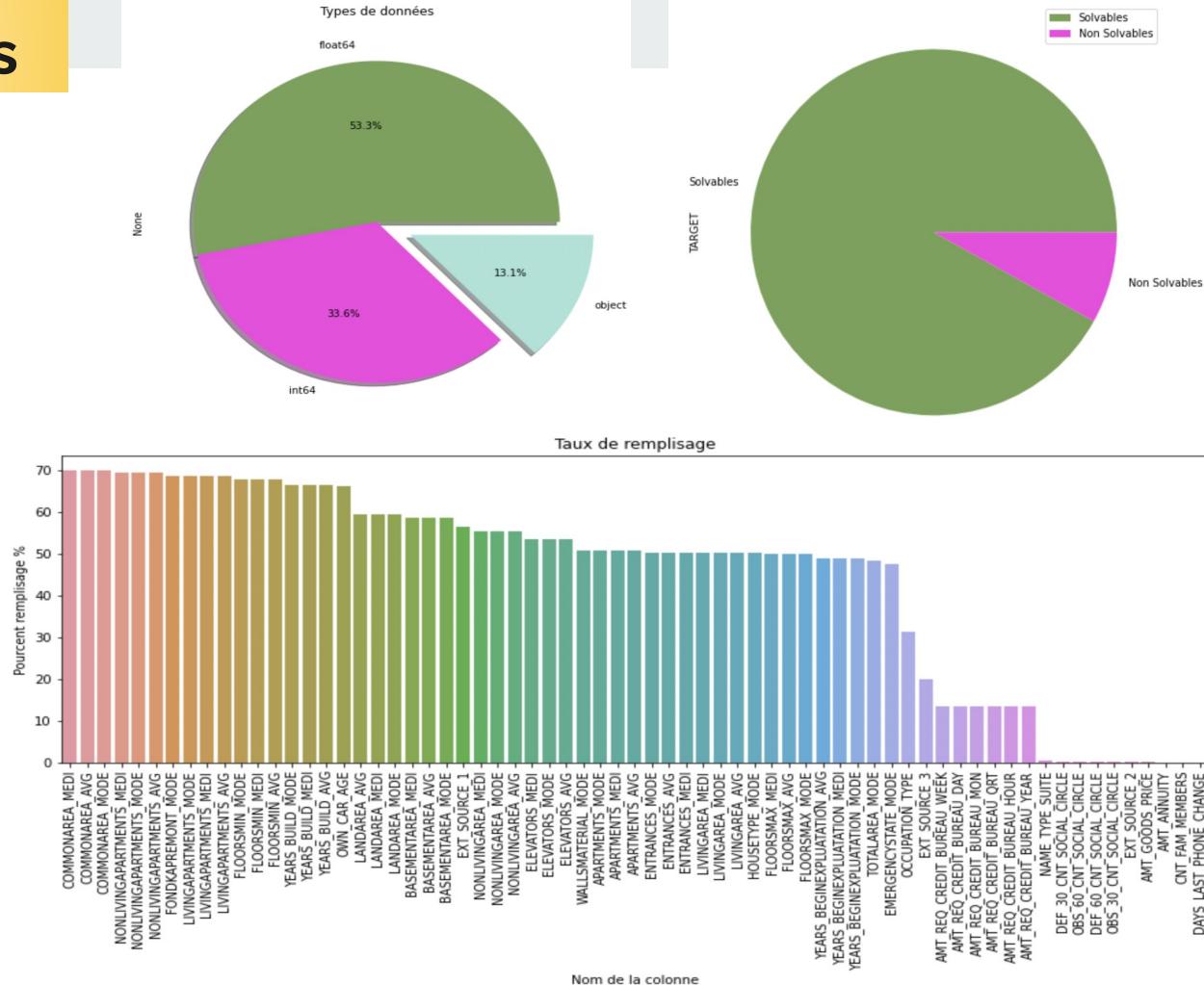


# Nettoyage données

Suppression des valeurs manquantes de plus 90%

There are 67 columns that have missing values.

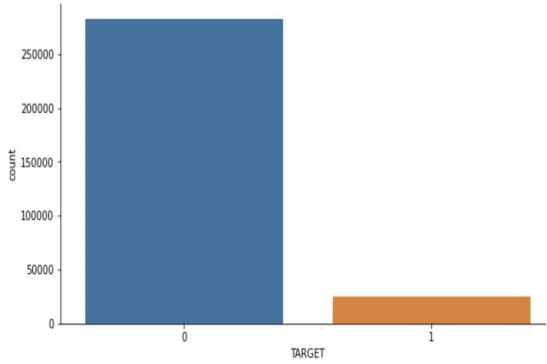
	Missing Values	% of Total Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4
NONLIVINGAPARTMENTS_AVG	213514	69.4
FONDKAPREMONT_MODE	210295	68.4
LIVINGAPARTMENTS_MODE	210199	68.4
LIVINGAPARTMENTS_MEDI	210199	68.4
LIVINGAPARTMENTS_AVG	210199	68.4
FLOORSMIN_MODE	208642	67.8
FLOORSMIN_MEDI	208642	67.8
FLOORSMIN_AVG	208642	67.8
YEARS_BUILD_MODE	204488	66.5
YEARS_BUILD_MEDI	204488	66.5
YEARS_BUILD_AVG	204488	66.5
OWN_CAR_AGE	202929	66.0
LANDAREA_AVG	182590	59.4
LANDAREA_MEDI	182590	59.4
LANDAREA_MODE	182590	59.4



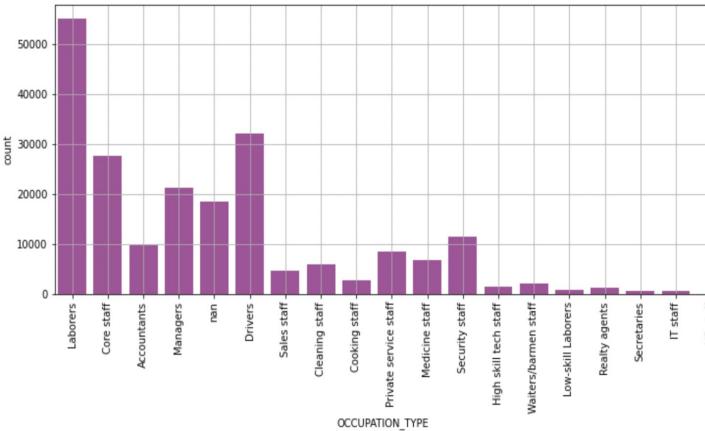
# Analyse Exploratoire

Distribution of the Target Column

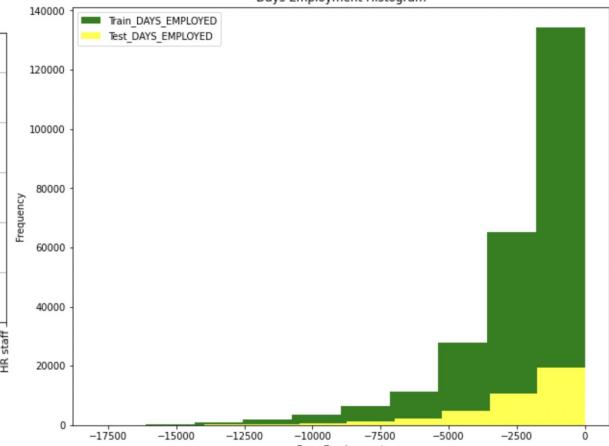
0 = loan was repaid on time, 1 = client had payment difficulties.



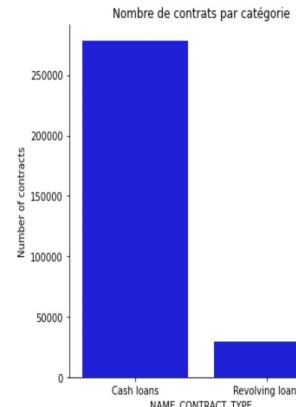
Les professions les plus représentées



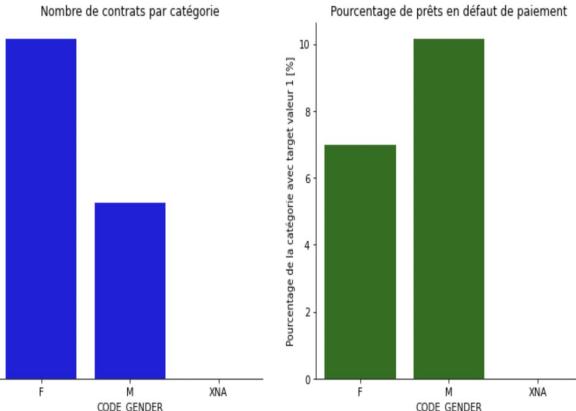
Days Employment Histogram



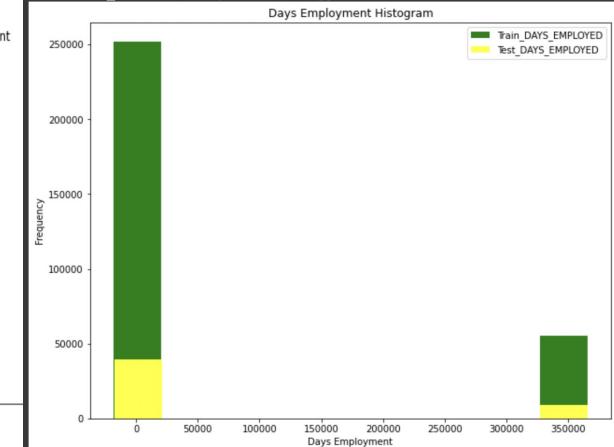
Contrats par catégorie de fonctionnalité 'NAME\_CONTRACT\_TYPE'



Contrats par catégorie de fonctionnalité 'CODE\_GENDER'

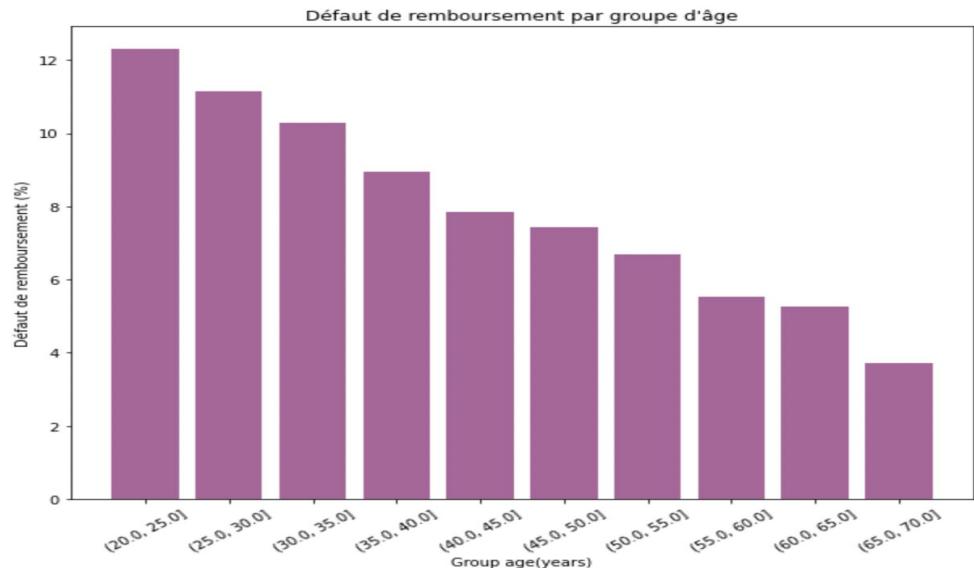
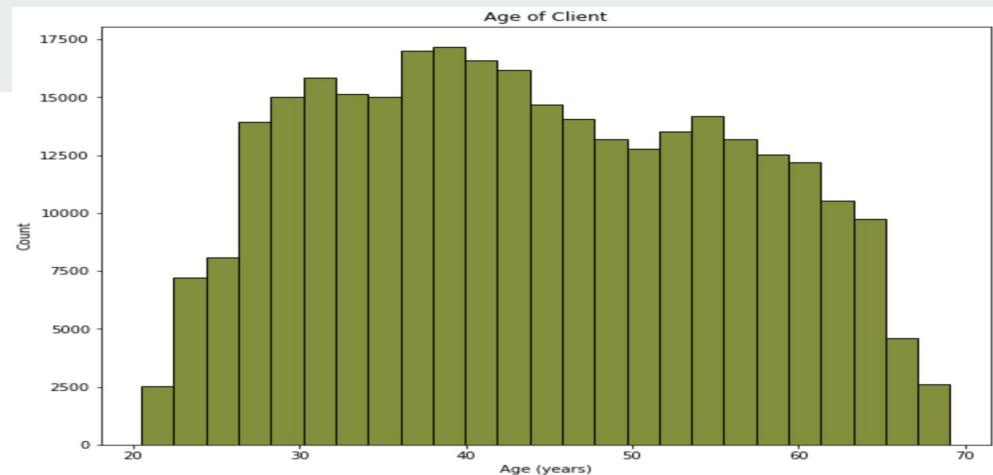
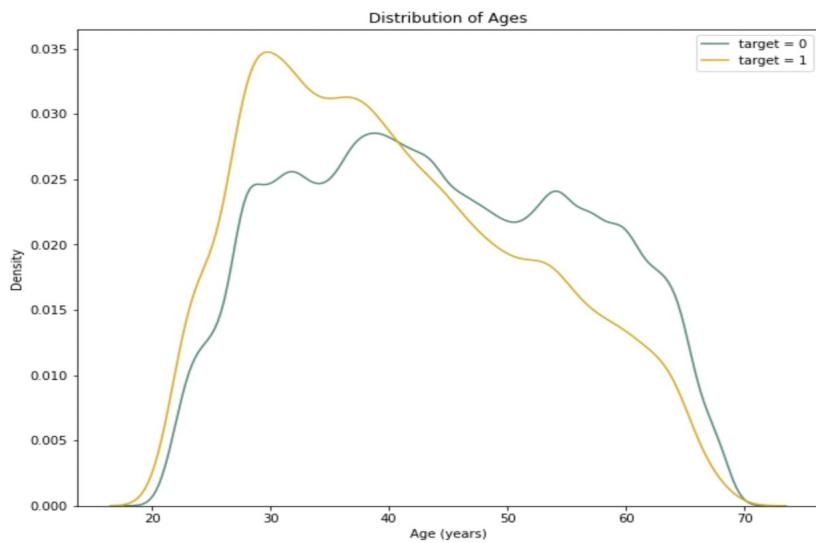


Name : DAYS\_EMPLOYED, Length: 9274, dtype: int64



# Analyse Exploratoire

Répartition de l'âge: À mesure que le client vieillit, il existe une relation linéaire négative avec la cible

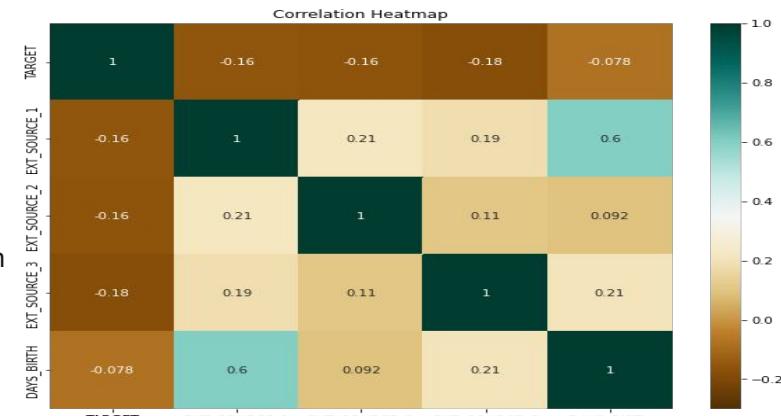
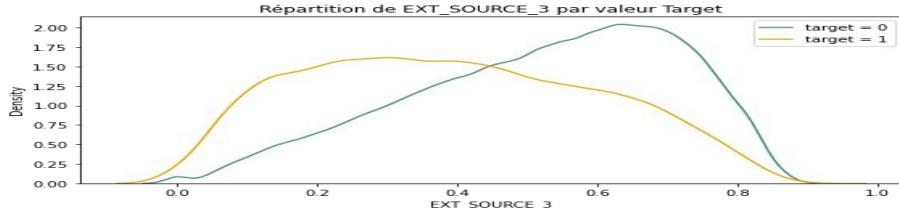
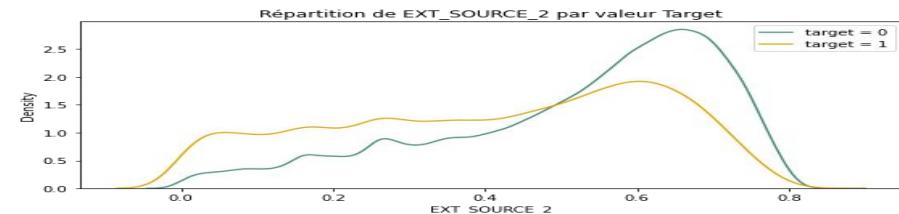
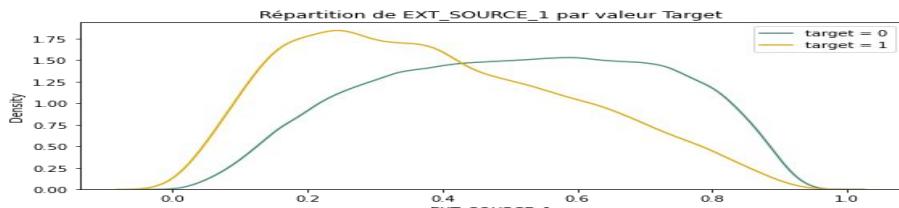


# Analyse Exploratoire

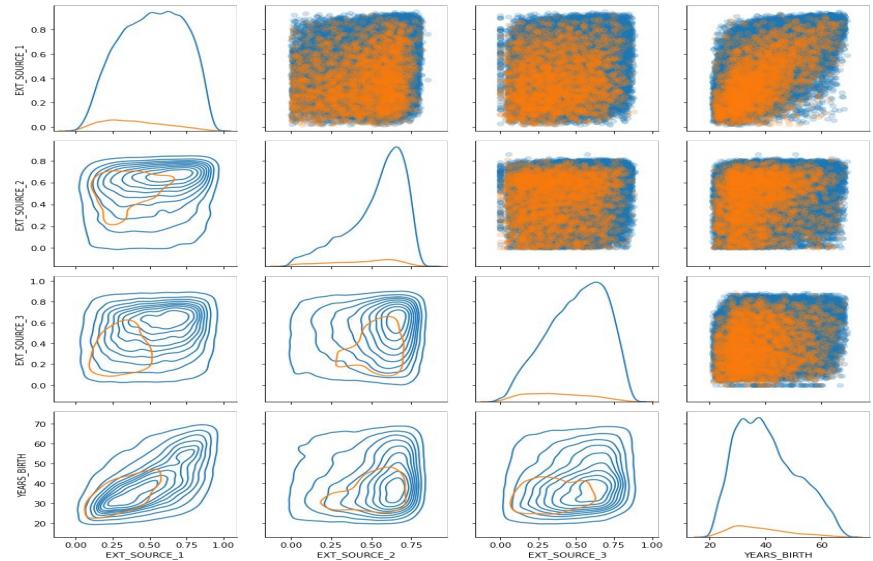
Les trois EXT\_SOURCE ont des corrélations négatives avec la cible, la valeur de EXT\_SOURCE augmente, le client est plus susceptible de rembourser le prêt.

DAYS\_BIRTH a positivement corrélé avec EXT\_SOURCE\_1.

EXT\_SOURCE\_3 affiche la plus grande différence entre les valeurs de la cible, cette caractéristique a une certaine relation avec la probabilité qu'un demandeur rembourse un prêt.



Ext Source and Age Features Pairs Plot



# Analyse Exploratoire

## Feature Engineering:

### CREDIT\_INCOME\_PERCENT:

le pourcentage du montant du crédit par rapport au revenu d'un client

### ANNUITY\_INCOME\_PERCENT:

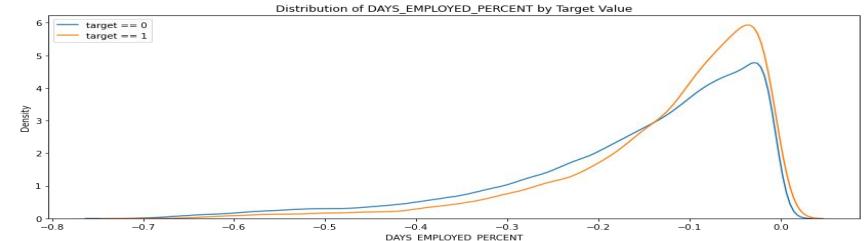
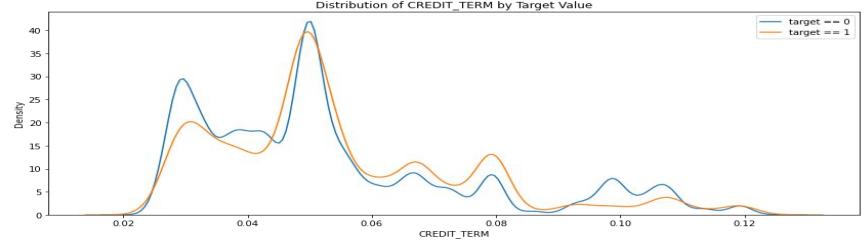
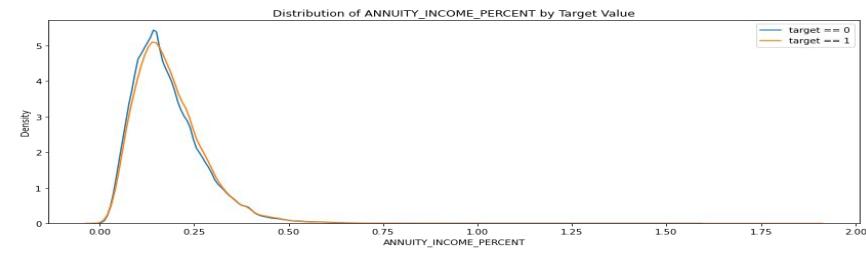
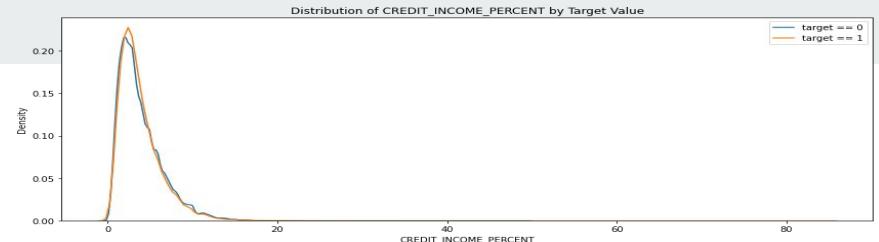
le pourcentage de l'annuité du prêt par rapport au revenu du client

### CREDIT\_TERM:

la durée du paiement en mois (puisque l'annuité est le montant mensuel)

### DAYS\_EMPLOYED\_PERCENT:

le pourcentage des jours employés par rapport à l'âge du client



# Encodage des variables

**Transformation des données :** Les variables catégorielles doivent être encodées pour être utilisables par les modèles.

- Label Encoding
- One-Hot Encoding

**Sample du dataset :** 30%

**Train\_test\_split:** (test\_size=0.2)

```
taille df sample 1: (92252, 253)
taille X_train: (73801, 252)
taille X_test: (18451, 252)
taille y_train: (92252,)
taille y_test: (18451,)
```



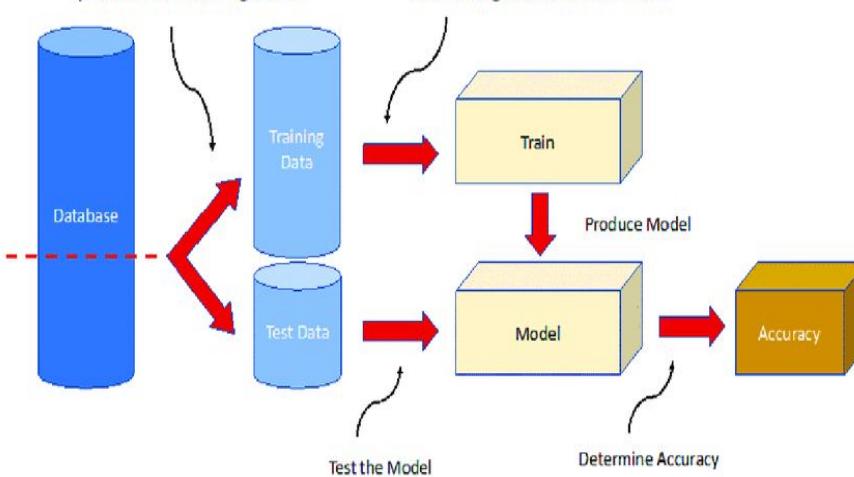
- **Un jeu de données déséquilibré :** 91 % des clients réguliers , 9 % des clients avec des défauts de paiement

## 4: Modélisation



# Modélisation

Split Dataset into Training and Test



- DummyClassifier
- Decision Tree
- Random Forest
- Logistic Regression
- Xgboost
- LightGBM
- Catboost

Étapes de méthodologie d'entraînement du modèle :

1: Entraînement sur training réduit avec cross validation avec optimisation des hyperparamètres

**Objectif :** sélectionner la bonne approche pour le déséquilibre

2: Même processus sur jeu complet

**Objectif :** Optimisation hyperparamètres pour chaque modèle

3: Entraînement du modèle choisi sur jeu données équilibré et échantillonnes 90%

**Objectif :** Choix du meilleur modèle meilleur F Beta Score obtenu en cross validation

4 : Interprétabilité du meilleur modèle

# les métriques

## Quel scoring adapté au problème métier?

Pour la classification binaire, les métriques pour estimer les erreurs entre  $y_{pred}$  et  $y_{test}$  sont :

**précision:** Quelle portion du target prédit sont du vrai classe ? Minimiser les faux positifs

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

**ROC AUC (Area Under the Curve) :** peut être comparé entre modèles

**Recall:** Quelle partie de la vraie classe est présente dans la classe prédict ? Minimiser les faux négatifs

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

**F1-score :** Accuracy équilibré, Généralisation de F1-score pour mettre plus de poids sur précision , pour mettre plus de poid sur recall (ex: beta=2)

$$F1 = \frac{2(\text{precision} \cdot \text{recall})}{\text{precision} + \text{recall}}$$

$$F1 = \frac{(2 \cdot \text{TP})}{(2 \cdot \text{TP} + \text{FP} + \text{FN})}$$

**Fbeta-score :** Moyenne harmonique pondérée

$$((1 + \beta^2) * \text{PRECISION} * \text{RECALL}) / (\beta^2 * \text{PRECISION} + \text{RECALL})$$

## Problématique :

- La société ne doit pas se priver des potentiels clients qui ne présentent pas de risque
- Les clients à risque font perdre de l'argent à la société

## Solution:

- Limiter le nombre de faux négatifs
- Limiter dans une moindre mesure le nombre de faux positifs

## Explication fonction métier :

- TP : pas de gain, pas perte ni de gain
- TN : gain des intérêts du crédit
- FP : perte des intérêts du crédit
- FN : perte des intérêts du crédit et une partie du crédit

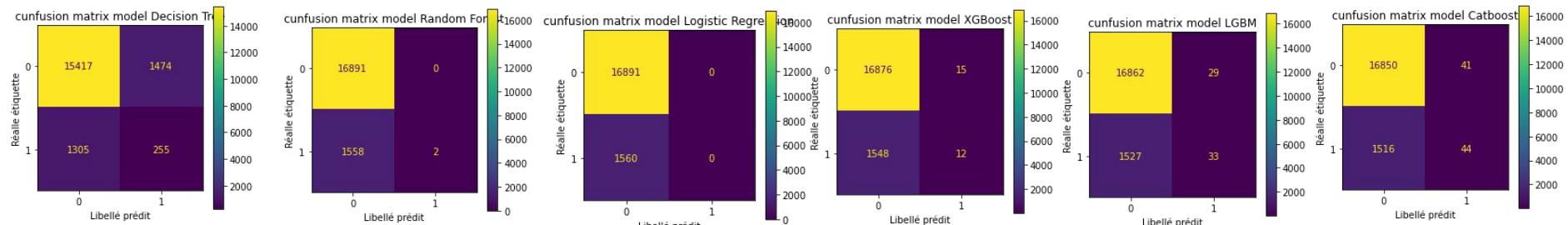
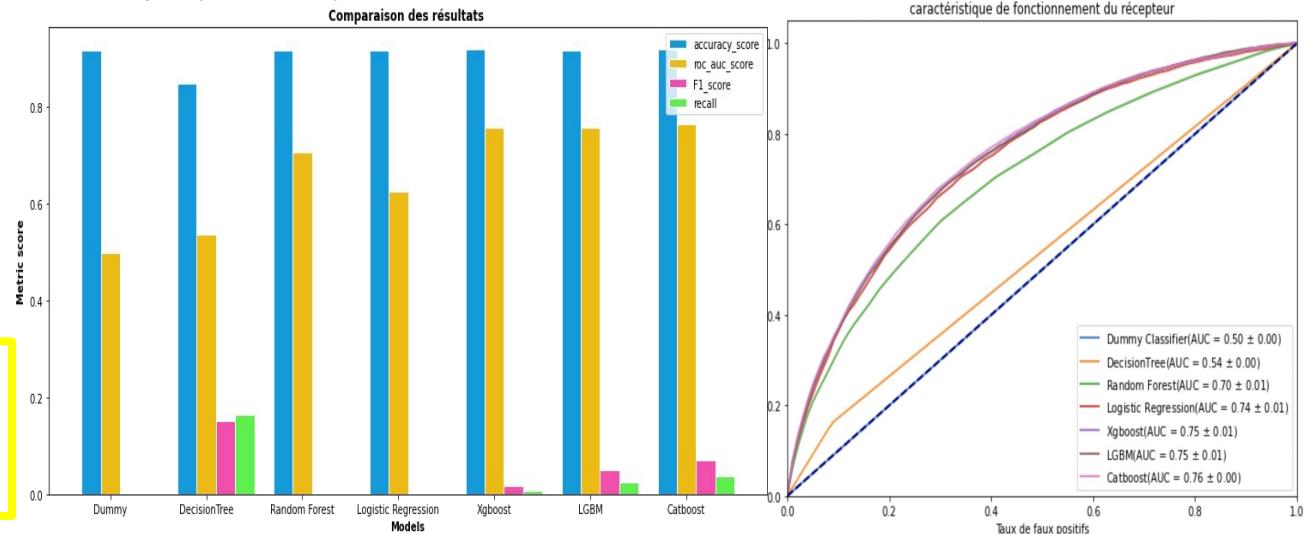
Il y 3 types de coûts à évaluer, la fonction de score est une somme des coûts dont l'objectif est d'être maximiser

		Predicted class	
		P	N
Actual Class	Sensitivity Recall P	True Positives (TP)	False Negatives (FN)
		False Positives (FP)	True Negatives (TN)
N Specificity	Precision		

# Comparaison de modèles

Matrix confusion et les tableaux performances Sample (frac = 0.3)

model_name	accuracy_score	roc_auc_score	F1_score	recall
Dummy	0.918	0.500	0.000	0.000
DecisionTree	0.848	0.537	0.151	0.165
Random Forest	0.918	0.707	0.001	0.001
Logistic Regression	0.918	0.625	0.000	0.000
Xgboost	0.919	0.758	0.016	0.008
LGBM	0.918	0.758	0.049	0.026
Catboost	0.919	0.765	0.071	0.038



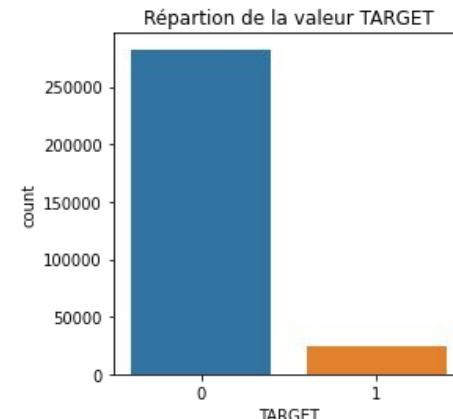
## Gridsearchcv best paramètres

Optimisation des hyper-paramètres par validation croisée avec Grid Search CV

model	
DummyClassifier	{'strategy': 'most_frequent'}
DecisionTree	{'criterion': 'entropy', 'max_depth': 5, 'max_features': 'sqrt'}
RandomForest	{'bootstrap': True, 'max_depth': 5, 'n_estimators': 100}
LogisticRegression	{'C': 0.0001, 'max_iter': 200, 'penalty': 'l2'}
XGB	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}
LGBM	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}
CatBoost	{'depth': 5, 'iterations': 20, 'learning_rate': 0.01}

## Ré-équilibration données

Les données du jeu d'entraînement sont déséquilibrées : environ 92 % de "TARGET = 0" pour 8 % de "TARGET = 1". 1 qui indique que le client n'a pas remboursé son crédit. (Défaillant, 8%) - 0 qui indique que le client l'a remboursé. (Non défaillant, 92%)

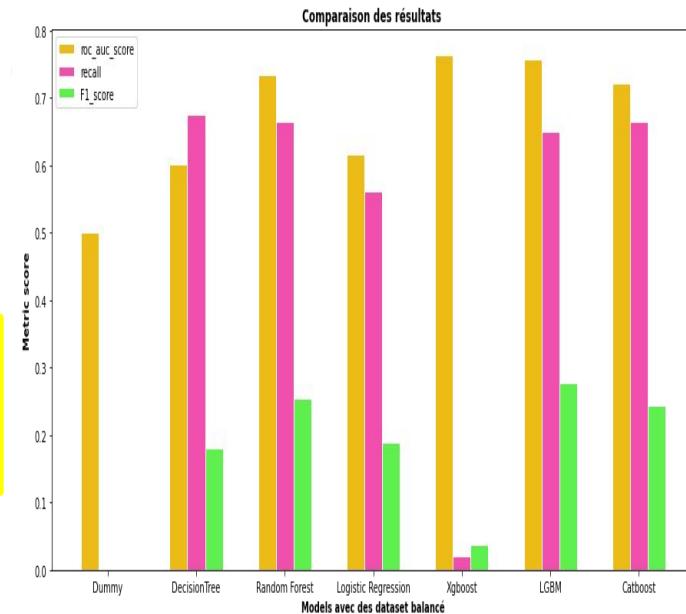


## Comparaison de modèles pour les données équilibrées

Gestion du problème du déséquilibre des classes : on peut indiquer à certains modèles le déséquilibre en réglant un hyperparamètre exemple : « class\_weight = 'balanced' »

la majorité des ROC AUC > 0.6 ou > 0.72, recall >= 0.5 et < 0.7, F1 score >= 0.2 et < 0.4 . On constate une amélioration du recall et du f1 score ,une amélioration de la capacité du modèle à détecter les FP et FN.

	ROC_AUC_balanced	Recall_balanced	F1-Score_balanced
DummyClassifier()	0.5	0.0	0.0
DecisionTreeClassifier(class_weight='balanced')	0.6013	0.6748	0.1804
RandomForestClassifier(class_weight='balanced')	0.7336	0.6642	0.2551
LogisticRegression(class_weight='balanced')	0.616	0.5609	0.1893
XGBClassifier(class_weight='balanced')	0.763	0.0199	0.0385
LGBMClassifier(class_weight='balanced')	0.7575	0.6503	0.278
<catboost.core.CatBoostClassifier object at 0x7fdc7f6fc510>	0.7208	0.6656	0.2433



Les deux meilleurs modèles : LGBM et Catboost

# Adaptation de la métrique au métier

**Choix d'une métrique de performance adaptée:** utilisation du fbeta score (beta=2) pour relancer la recherche des hyper paramètres

La banque peut commettre 2 erreurs:

- Refuser un prêt dû Perdre le client (FP)
- Accorder un prêt indu Perdre la somme prêtée (FN)

**Solution:**

- Limiter le nombre de faux négatifs
- Limiter dans une moindre mesure le nombre de faux positifs

Le score F-bêta est la moyenne harmonique pondérée de la précision et du rappel. Le paramètre bêta détermine le poids du recall dans le score combiné.

model	best_param_dfl	roc_auc_dfl	f1_score_dfl	recall_score_dfl	fbeta_score_dfl
LGBMClassifier	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}	0.757	0.278	0.65	0.423
CatBoostClassifier	{'depth': 9, 'iterations': 100, 'learning_rate': 0.02}	0.75	0.267	0.664	0.416

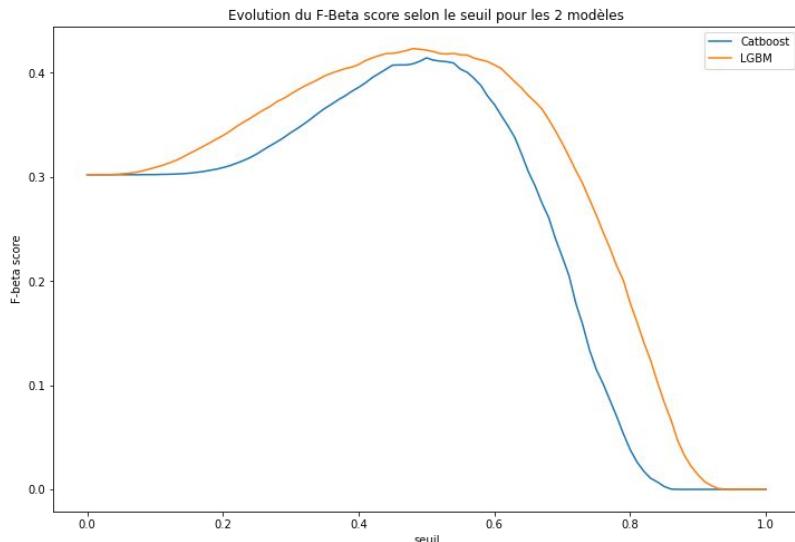
## Ré-apprendre les meilleurs modèles sur le train set complet

Sample (frac = 0.9)

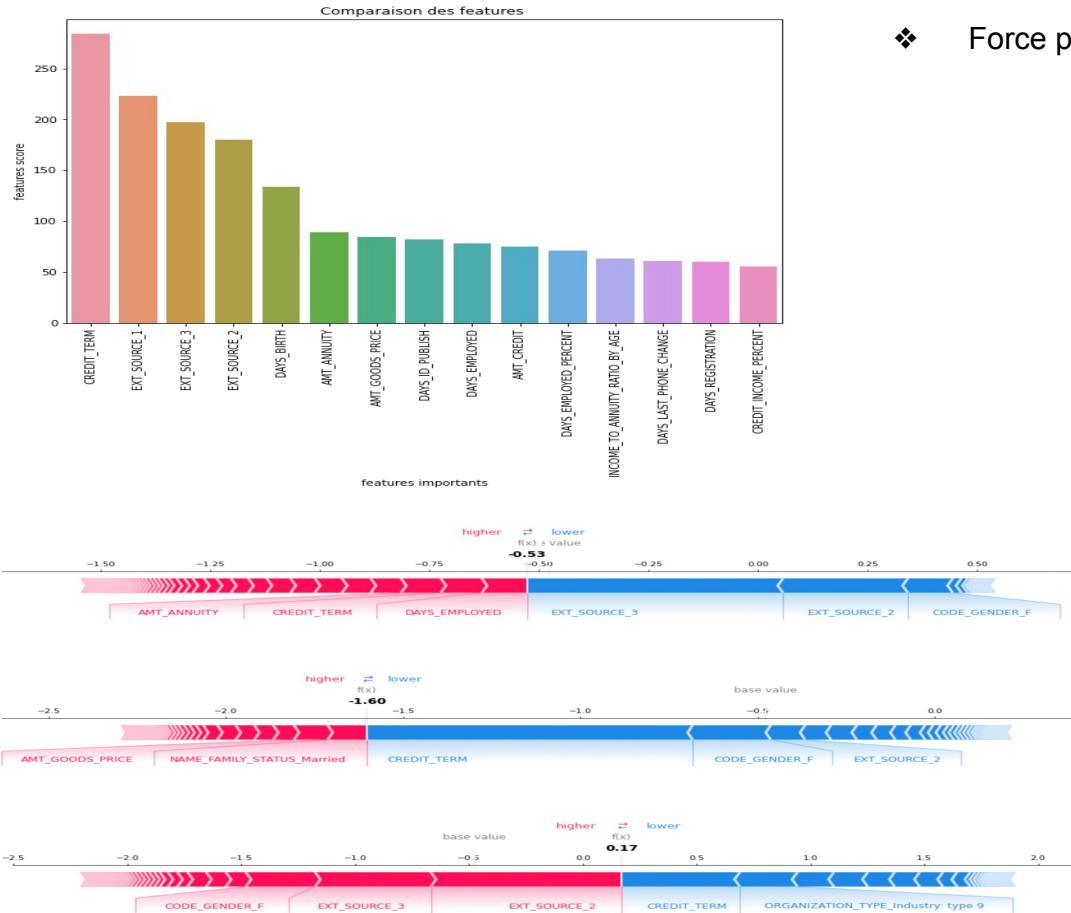
model	best_param_df2	roc_auc_df2	f1_score_df2	recall_score_df2	fbeta_score_df2
LGBMClassifier	{'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 100}	0.764	0.275	0.676	0.427
CatBoostClassifier	{'depth': 9, 'iterations': 100, 'learning_rate': 0.02}	0.748	0.263	0.673	0.414

## le seuil de décision

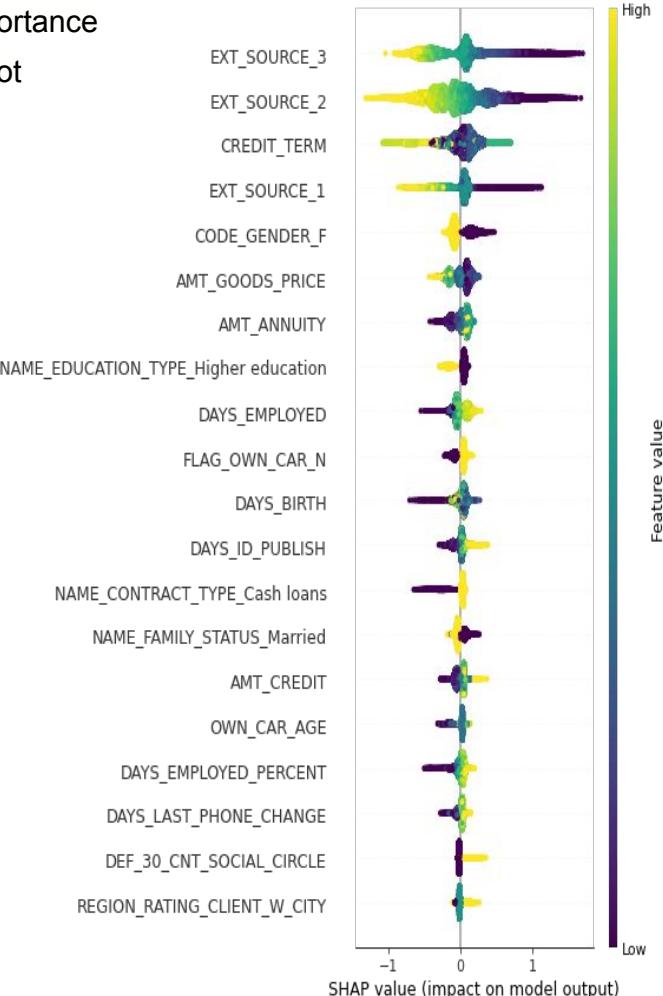
- ❖ Score max LGBM: 0.4233
- ❖ Seuil pour score max LGBM: 0.48
  
- ❖ Score max Catboost: 0.4144
- ❖ Seuil pour score max Cat Boost: 0.5



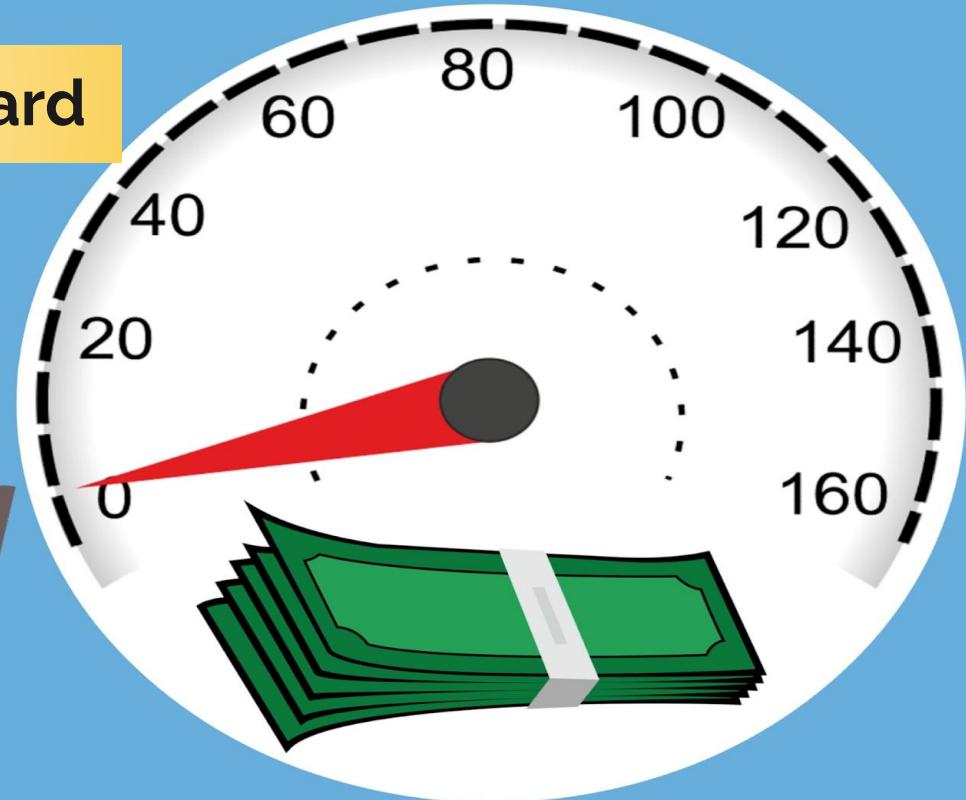
# Interprétabilité du meilleur modèle



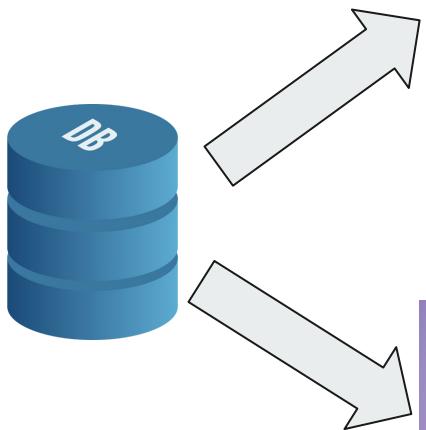
- ❖ Feature importance
- ❖ Summary plot
- ❖ Force plot



## 5: Présentation du Dashboard



# Architecture de l'application



API :

- FastAPI



<https://ocr-p7-api.herokuapp.com/>

Dashboard :

- Streamlit



<https://ocr-p7-dashboard.herokuapp.com/>



Prédiction pour l'octroi d'un crédit à partir d'un modèle

Mise en ligne :

- Heroku



Versioning:

- GitLab et Github

<https://github.com/mitraddgr/>



## 6: Conclusion

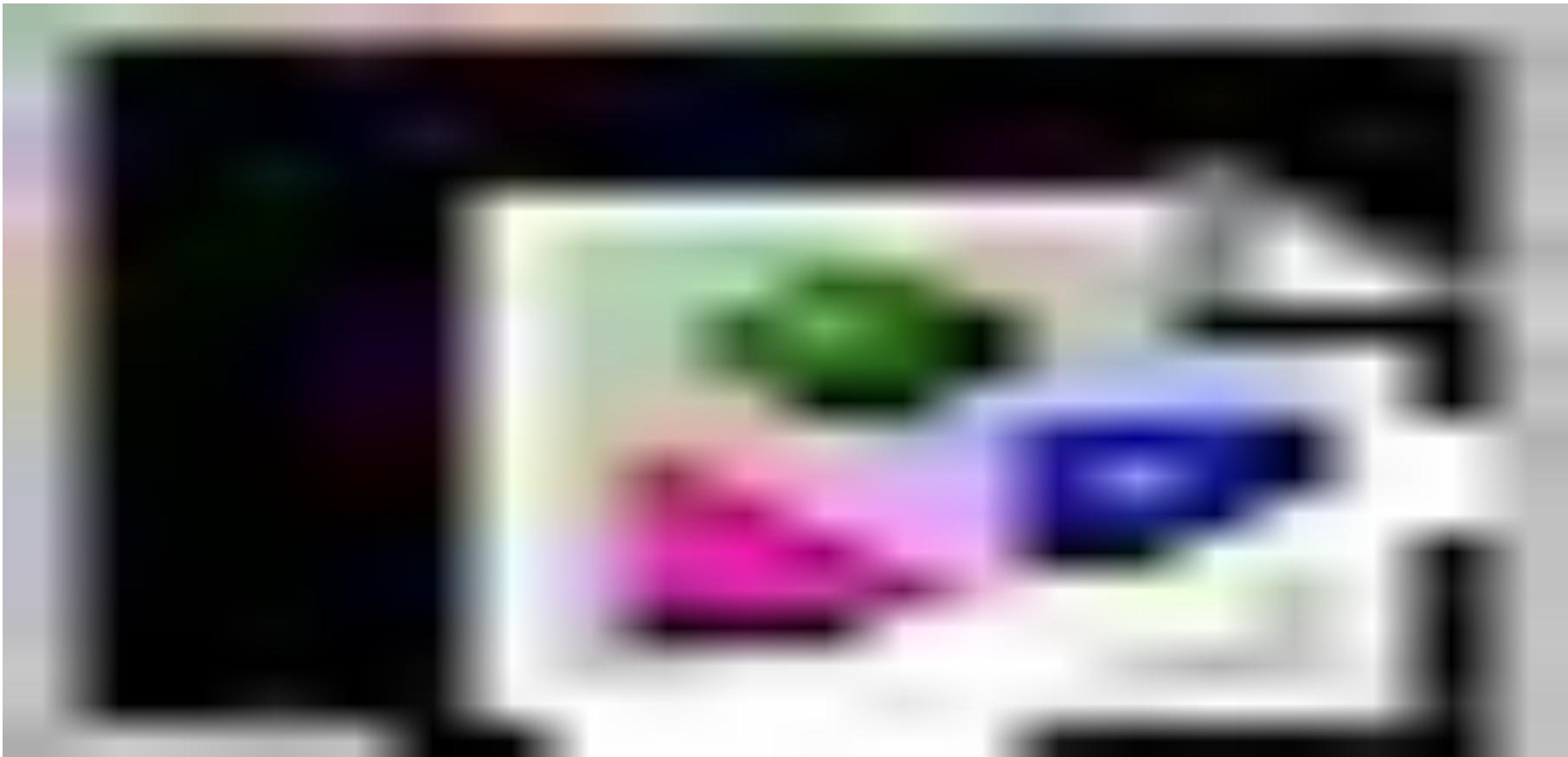
- ❑ Étudier d'autres algorithmes de classification Light Gbm , Cat Boost
- ❑ Les meilleurs modèles sont Lgbm et Catboost
- ❑ Le ré-échantillonnage des données permet de corriger le déséquilibre des classes
- ❑ Utilisation d'une méthode d'échantillonnage des données plus performante (Class Weight)
- ❑ Utilisation d'un métrique métier et fixation d'un seuil de solvabilité optimum.

## Amélioration

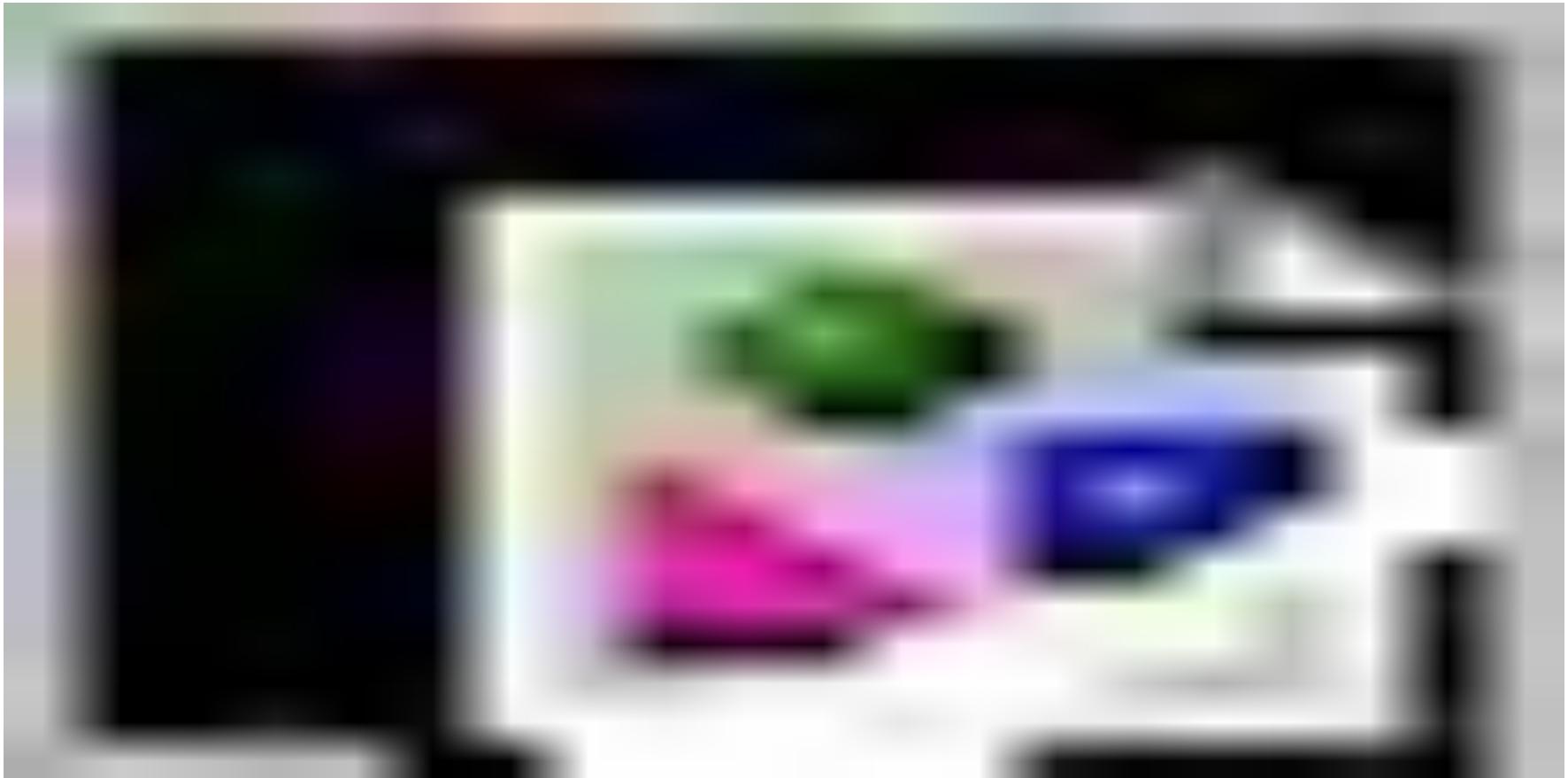
- ❑ Feature engineering en créant de variables plus pertinentes
- ❑ Optimisation plus fine des hyper-paramètres
- ❑ Faire évoluer les scoring extérieur en même temps que les features sont modifiées
- ❑ Graphes interactifs
- ❑ Modification de la métrique créée, avec l'aide d'un expert métier

# CONCLUSION

# Présentation du API



# Présentation du Dashboard



## Quel est l'impact du déséquilibre de classe :

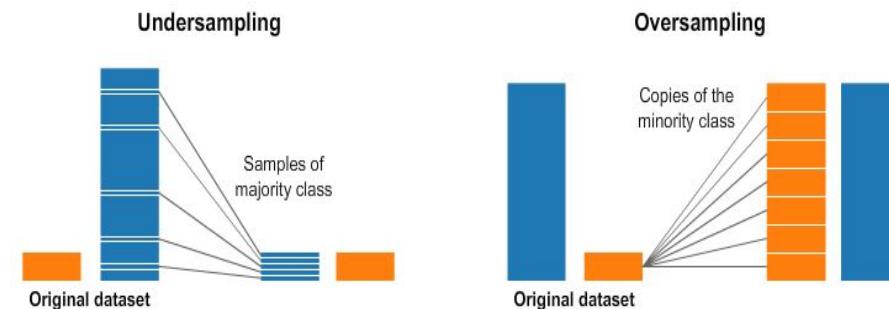
- Ce déséquilibre de classe augmente nettement la difficulté de l'apprentissage par l'algorithme de classification
- l'algorithme n'a que peu d'exemples de la classe minoritaire sur lesquels apprendre
- Il est donc biaisé
- La précision pour minimiser le taux d'erreurs parmi les exemples prédis positifs par le modèle
- Le rappel pour tenter de détecter un maximum de positif
- Le F1-score pour trouver un compromis entre la précision et le recall. Lorsqu'il est aussi coûteux de manquer un positif que de déclarer un faux positif

## l'utilisation de SMOTE:

- Le SMOTE est une technique très utile pour rééquilibrer les données numériques en entrée d'un modèle de Machine Learning. Il permet d'éviter le surapprentissage du modèle en densifiant les individus minoritaires de façon homogène et peut améliorer significativement les performances des modèles. Pour utiliser le SMOTE sans danger, il est indispensable de se souvenir des cinq règles suivantes :
- Les paramètres optimaux du SMOTE dépendent des données et doivent donc être optimisés. Le plus simple est de les optimiser en même temps que votre modèle.
- Les variables numériques doivent être normalisées.
- Les variables discrètes doivent être retraitées entre le SMOTE et le modèle.
- Les variables catégorielles doivent être conservées telles quelles et ne pas être encodées.
- Le SMOTE est réservé à l'entraînement du modèle et ne doit surtout pas être appliqué aux données de validation et de test.

**Resampling** **Le sous-échantillonnage (undersampling):** Parmi les individus majoritaires, on en retire une partie afin d'accorder plus d'importance aux individus minoritaires. Cette approche permet de diminuer la redondance des informations apportées par le grand nombre d'individus majoritaires.

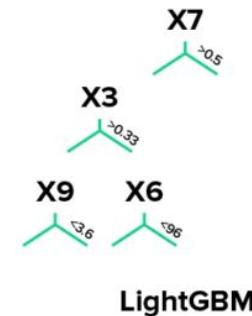
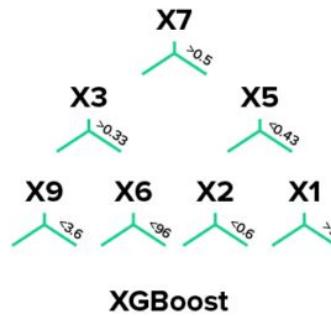
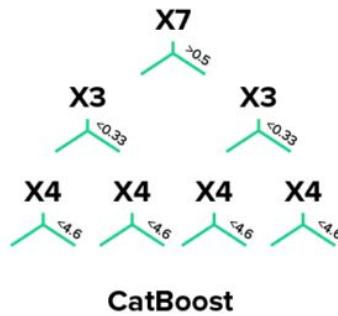
**Le sur-échantillonnage (oversampling):** Le nombre d'individus minoritaires est augmenté pour qu'ils aient plus d'importance lors de la modélisation. Différentes solutions sont possibles, comme le "clonage" aléatoire ou le SMOTE.



## LightGBM:

- basée sur les arbres de décision
- utilisée dans la classification et la régression
- optimisé pour des performances élevées dans les systèmes distribués.
- créer des arbres de décision prenant les feuilles en charge, ce qui signifie que, selon une condition donnée, une seule feuille est fractionnée,, en fonction du gain
- il divise la feuille de l'arbre la mieux adaptée tandis que d'autres calculs de renforcement divisent la profondeur de l'arbre en deux parties
- 

### Tree growth examples:



### Précision

- Quelle portion du cluster prédict sont du vrai classe ?

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

### Recall

- Quelle portion du vrai classe sont présent dans le cluster prédict ?

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

### F1 Score

- accuracy « équilibré » :

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

- Cluster prédict : TP + FP
- (Vrai classe : TP + FN)

