



# Implémentez un modèle de scoring

Projet 7 - Openclassroom

# Plan de la présentation

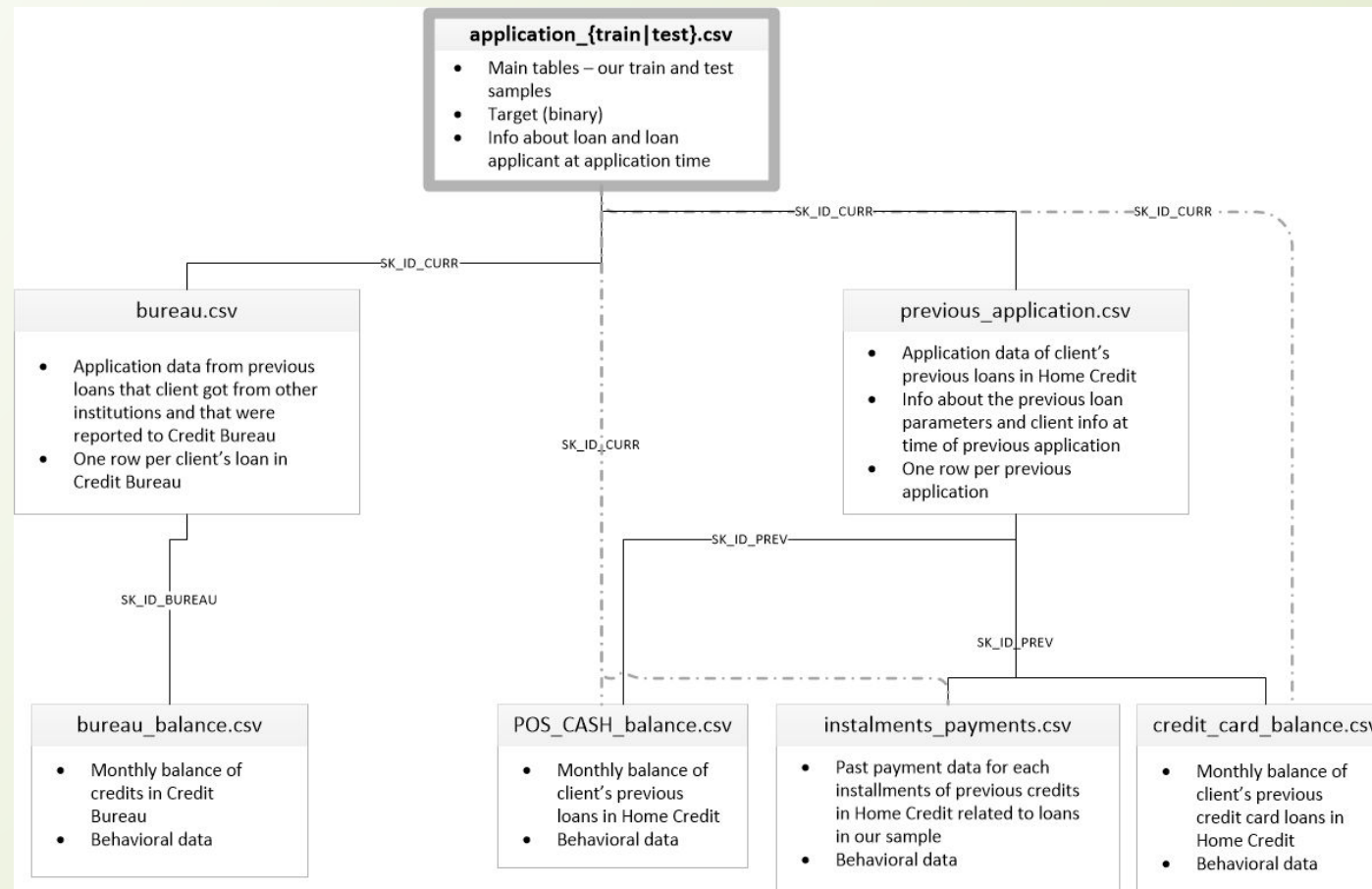
- I. Présentation de la problématique
  - i. Rappel de la problématique
  - ii. Présentation du jeu de données
- II. Modélisation effectuée
  - i. Utilisation d'une Baseline
  - ii. Définition d'une fonction de scoring
  - iii. Optimisation du modèle
- III. Présentation du dashboard
- IV. Conclusion

# Présentation de la problématique

# Rappel de la problématique

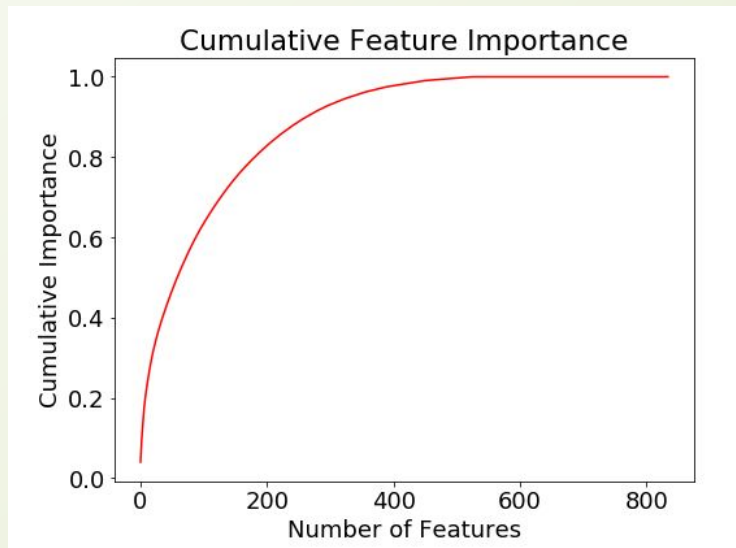
- ❑ Prêt à dépenser : Société financière qui propose des crédits à la consommation
- ❑ Développer un modèle de scoring de probabilité de défaut d'un client
- ❑ Pouvoir expliquer le refus ou l'acceptation d'une demande de crédit au client
- ❑ Utiliser un dashboard intuitif au vue des utilisateurs

# Présentation du jeu de données



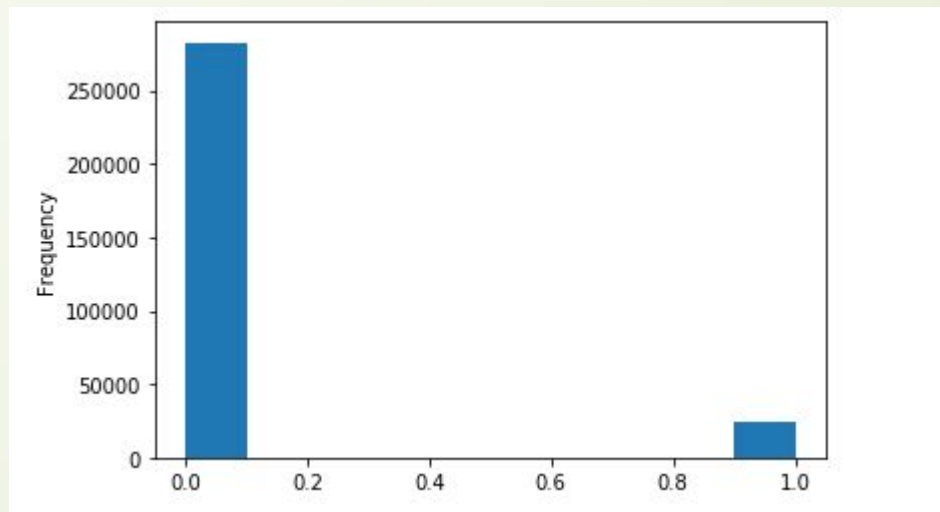
# Présentation du jeu de données

- ❑ Après agrégations des données 307511 données et 1465 features
- ❑ En enlevant les variables fortement corrélées => 854 features
- ❑ En enlevant les features pas importantes pour notre modèle => 338 features



# Présentation du jeu de données

- Visualisation de la répartition de la valeur target



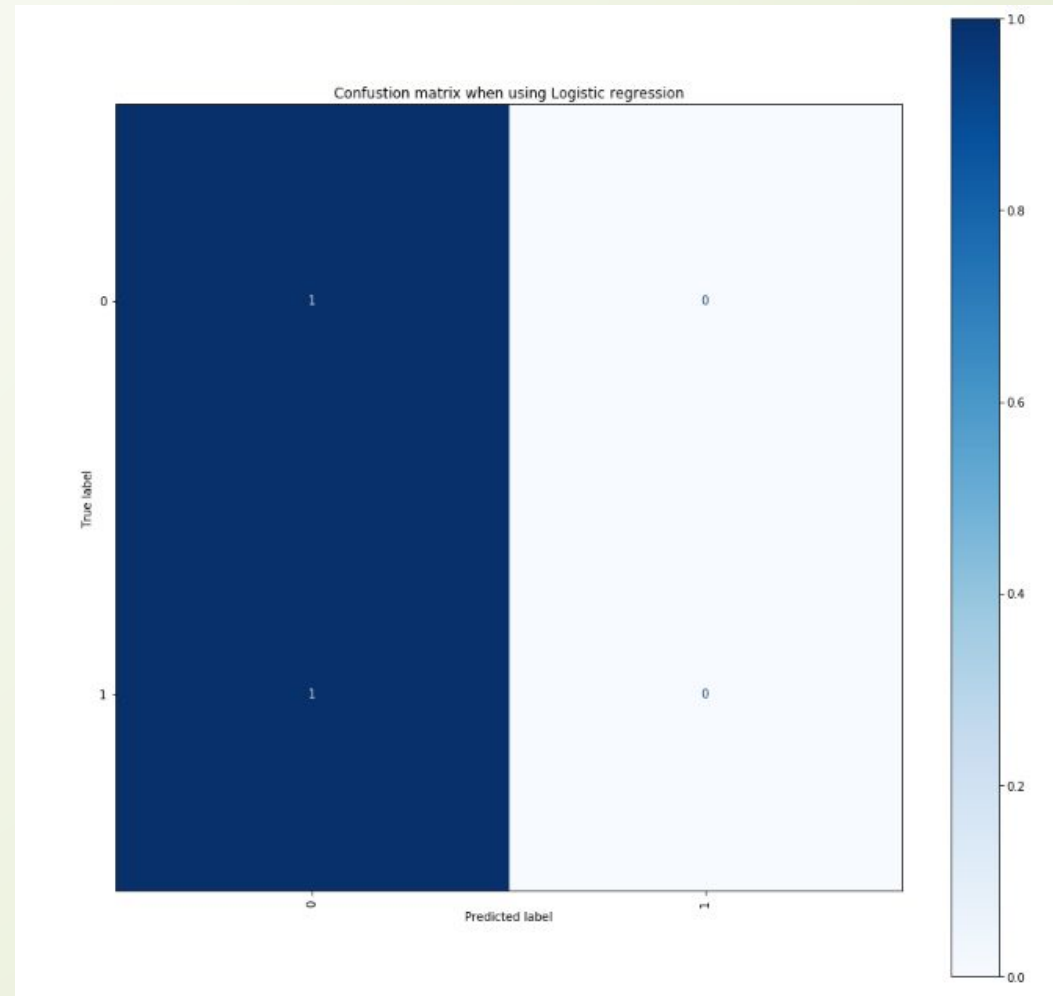
# Modélisation effectuée



# Utilisation d'une baseline

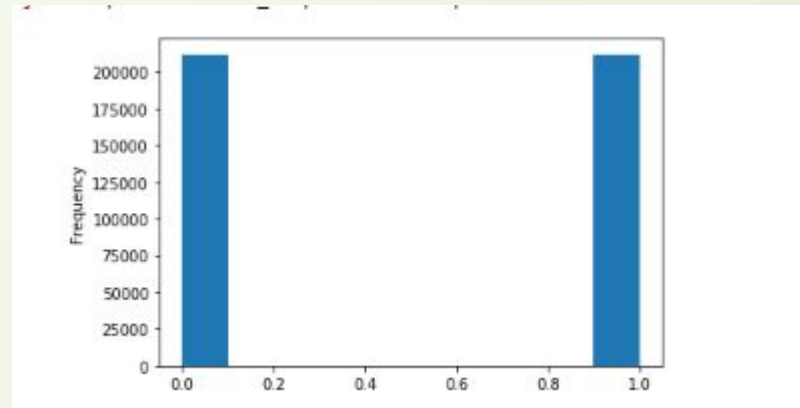
- Régression logistique =>
  - Accuracy = 91%
  - AUC score = 50%

# Utilisation d'une baseline

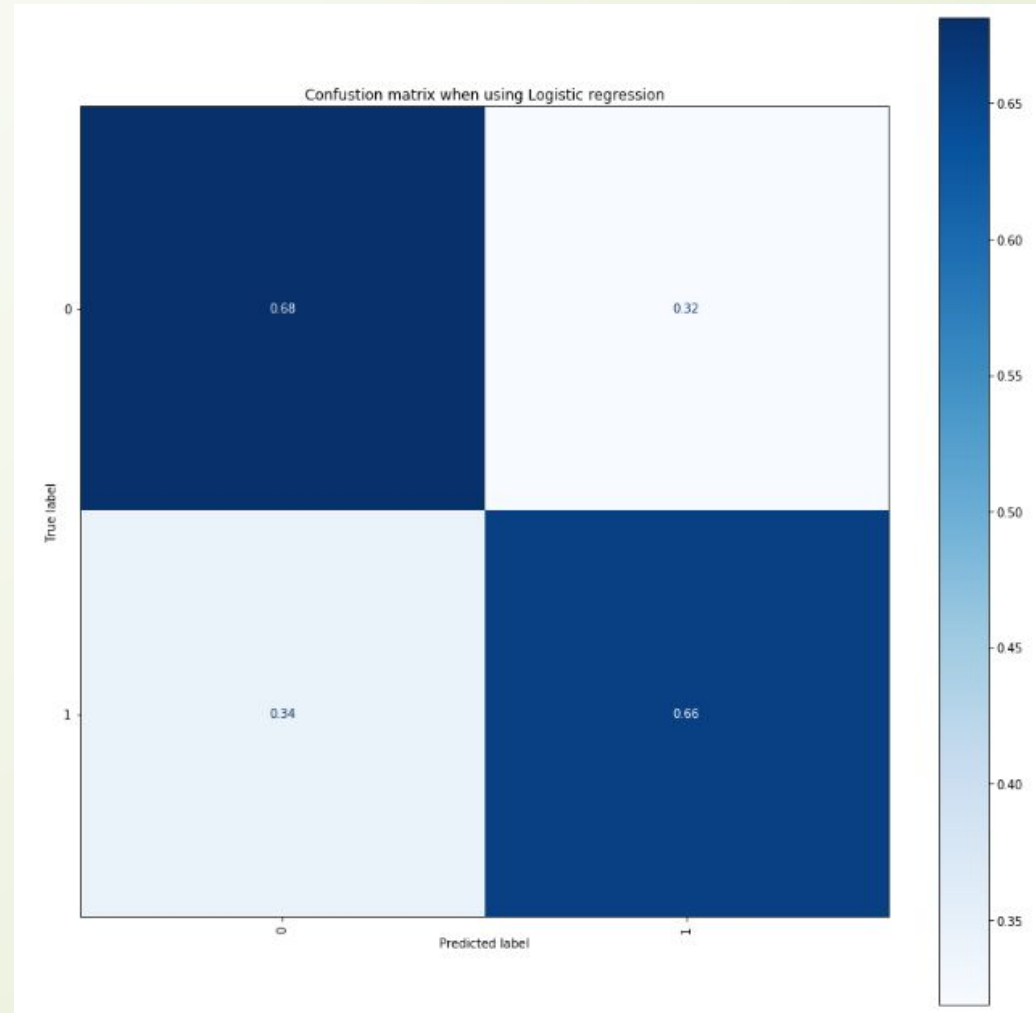


# Utilisation d'une baseline

- Over-sampling en utilisant SMOTE :



# Utilisation d'une baseline



# Définition d'une fonction de scoring

- Quelle fonction de scoring choisir:
  - AUC – ROC : pas intéressant on voudrait donner plus de poids aux faux négatifs
  - F-Bêta score : Un peu plus intéressant, mais ça correspond pas à la vraie mesure métier

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

$$F_{\beta} = \frac{(1 + \beta^2) \cdot \text{true positive}}{(1 + \beta^2) \cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}}$$

# Définition d'une fonction de scoring

▢  $annuityfactor = \frac{1 - (1+r)^{-1 * nyears}}{r}$

▢  $installements = \frac{amount}{annuityfactor}$

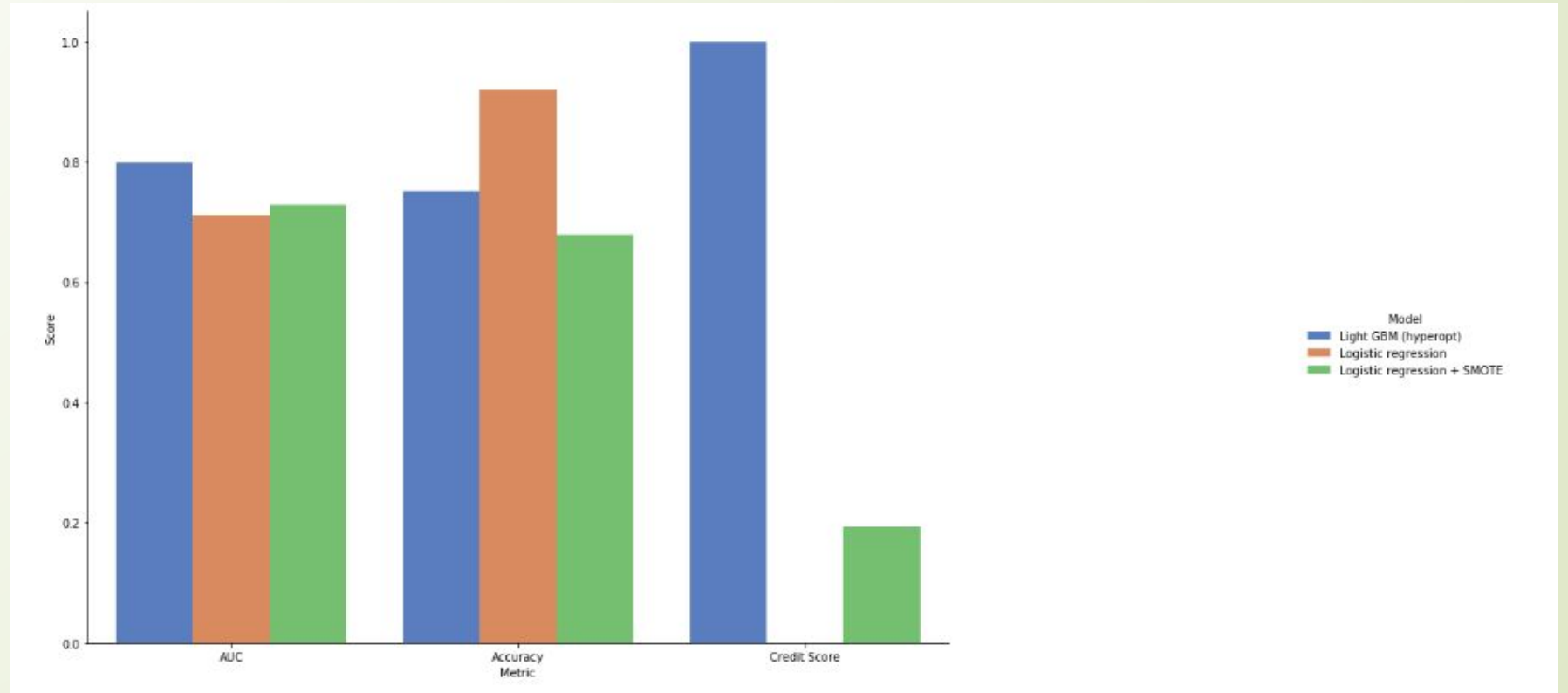
▢  $creditcost = installements * nyears - amount$

	AMT_CREDIT	AMT_ANNUIITY	cost	percent of credit
0	254700.0	28867.5	6295.514197	2.471737
1	337500.0	16875.0	17998.551323	5.332904
2	719365.5	36859.5	37464.837987	5.208039
3	408780.0	13185.0	33521.265320	8.200319
4	273636.0	15835.5	12674.310872	4.631814
...	...	...	...	...
230628	216000.0	8271.0	14947.889979	6.920319
230629	1214086.5	43609.5	89487.866336	7.370798
230630	1129500.0	58612.5	58108.392423	5.144612
230631	755856.0	31905.0	47536.615124	6.289110
230632	270000.0	13500.0	14398.841058	5.332904

# Optimisation du modèle

- Utilisation du Light GBM avec Hyperopt (en utilisant adaptive TPE)
- Définition de notre fonction de scoring (elle servira pour l'early stopping)
- Trials dépendant de la stabilité du score
- Utilisation de l'algorithme GOSS

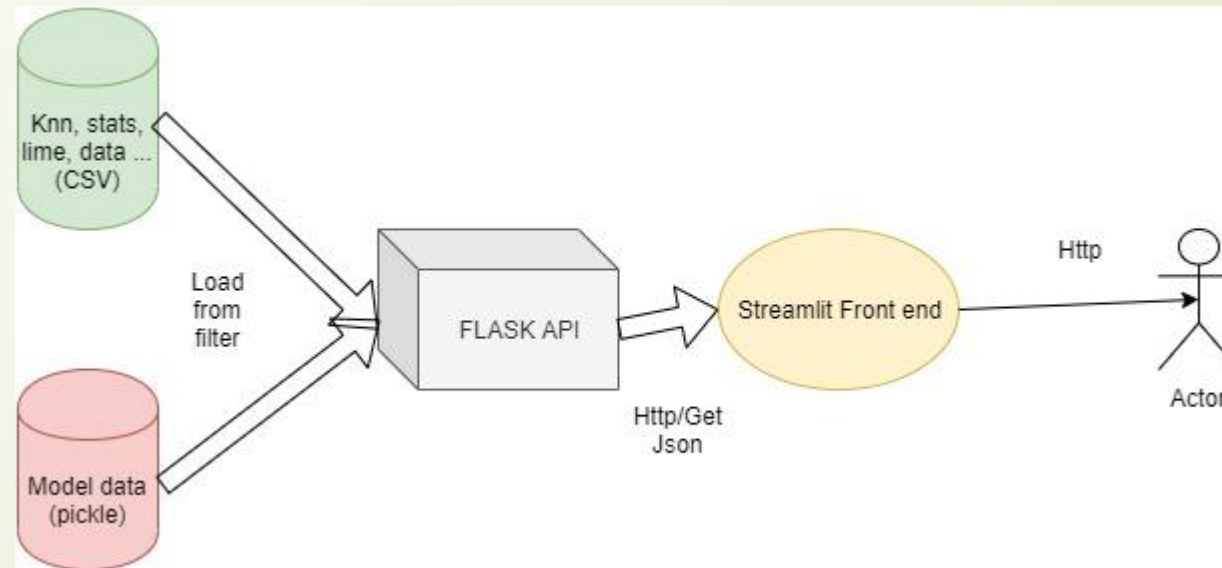
# Optimisation du modèle





# Présentation du dashboard

# Présentation du dashboard



# Présentation du dashboard

- <https://pret-a-depenser.herokuapp.com/>
- [https://pret-a-depenser-backend.herokuapp.com/get\\_all\\_clients](https://pret-a-depenser-backend.herokuapp.com/get_all_clients)

# Conclusion

# Conclusion

- ❑ Pousser un peu plus le nettoyage des données
- ❑ Utiliser une structure cloud pour améliorer la recherche hyperopt
- ❑ Utiliser un model stacking
- ❑ Améliorer la lisibilité des résultats (One hot encoding)
- ❑ Charger toutes les données dans le dashboard (en utilisant une meilleure architecture)
- ❑ Autoriser de renseigner des informations d'un client pour pouvoir avoir une prédiction