1. **Provide your team background and organization description (if applicable).**

Je participe régulièrement à des compétitions de ML (surtout sur Kaggle).

2. **Explain why you participated in the Alim' Confiance challenge.**

Je suis tombé sur Trustii par hasard. Le challenge Alim'Confiance était sympathique et proche d'un concours Kaggle sur lequel j'avais fait 4ème sur plus de 1000 équipes, donc je me suis dit que je pouvais gagner la 1ère place et les 1000€.

3. **Describe how you built your winning model and elaborate on the technical and modeling choices you made.**

La solution est composée de 5 notebooks :

- N1 et N2 : création des features
- N3a et N3b : entrainements de modèles (LGBM, XGBOST et CATBOOST) et inférence sur les données de test
- N4 : soumission

Parmi les features créées, 3 groupes sont intéressants. Les autres sont classiques (tfidf …). Ces groupes sont :

- **Des données externes** en open-source sur le numéro de SIRET (notebook n°1)
- **Des données de géolocalisation**. Un algo de geocoding est utilisé pour combler les latitudes/longitudes manquantes, puis des données sont calculées. Exemple : combien de restaurant ont eu un controle « Satisfaisant » dans un rayon de 1km, 3km, 5km ?
- **Des données qui exploitent un biais de vos données de train et test**. Le biais vient du fait que dans le test.csv, il y a des contrôles pour des mêmes établissements et le même jour que dans le train.csv. Au début du concours, je l'ai remarqué et j'ai mis en place un postprocessing à la fin de mon traitement pour écraser mes prédictions avec des corrections manuelles. J'ai vu que ça augmentait nettement mon score de validation croisée (CV) et mon score public. Donc je suis allé plus loin : au lieu de faire des corrections manuelles, j'ai créé des features. Exemple : quand un contrôle à un jour donné pour un établissement a lieu, est-ce qu'on en a vu un autre dans le train.csv ? Si oui, quel était le résultat ? Est-ce qu'il y a eu des contrôles avant/après ce jour ? Si oui, quels étaient les résultats ? → Ces données ont rendu ma solution beaucoup plus performante mais dans la vraie vie, elles ne sont pas utilisables. C'est un biais qui à mon sens aurait dû être évité par l'organisation.

En tout, plus de 1000 features sont créées, mais la majorité est inutile ! Ma feature selection a été faite à la main après quelques itérations de XGBOOST. Je ne voulais pas en garder plus de 300-400 pour ne pas avoir des temps d'entrainements trop longs. Cette partie peut être améliorée en étant plus rigoureux.

Modélisation :

- Plusieurs LightGBM, XGBOOST et CATBOOST ont été entrainés avec des paramètres légèrement différents. Les features de XGB et CATBOOST sont les mêmes. Le LightGBM en utilise 2 de plus (dates relatives à la création de l'entreprise), que j'ai calculées à la fin du concours. Comme ça n'augmentait pas la CV d'xgb et catboost, je ne les ai gardé que pour les lgbm.
- A noter qu'avoir autant de modèles n'est pas crucial pour faire un bon score. L'essentiel est d'avoir une diversité d'algo (lightgbm, xgb et catboost). Dans les faits, j'ai tout gardé parce que l'entrainement était assez rapide.
- Les poids pour l'ensembling final peuvent être améliorés. Je les ai obtenus avec optuna en optimisant mon score de CV, mais mon programme avait un problème (trop de cpu alloués, donc les poids renvoyés ne correspondaient pas au meilleur score).
- J'ai optimisé l'accuracy pour tous mes modèles, mais optimiser le f1-score a eu l'air de mieux fonctionner sur le classement privé.

4. What CPU/RAM resources you used to build your model

7 CPU et 1 GPU P100.