



Projet 7 : Implémentez un modèle de scoring

Présenté par : Divine Tulomba, étudiante Data Scientist

Entreprise : RES GROUP

Année : 2024

Plan de Présentation

- 1) Problématique
- 2) Exploration de données
- 3) Modélisation
- 4) MIFlow
- 5) DataDrift
- 6) Déploiement
- 7) Conclusion

Plan de Présentation

1) Problématique

2) Exploration de données

3) Modélisation

4) MIFlow

5) DataDrift

6) Déploiement

7) Conclusion

1. Problématique

La société financière, nommée '**Prêt à dépenser**' propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt. L'entreprise souhaite mettre en œuvre un outil de **scoring crédit** pour calculer la probabilité qu'**un client rembourse son crédit**, puis classifie la demande en crédit **accordé** ou **refusé**. Elle souhaite donc développer un algorithme de classification en s'appuyant sur des sources de données variées.

- ✓ identité,
- ✓ données comportementales,
- ✓ données provenant d'autres institutions bancaires,
- ✓ etc.



Objectifs

- ✓ Analyse d'un jeu de données : Nettoyage
du jeu de données Recherche du
modèle optimal
- ✓ API
- ✓ Dashboard interactif
Visualisation score et interprétation
Informations du client Interprétation prédiction
modèle



Plan de Présentation

1) Problématique

2) Exploration de données

3) Modélisation

4) MIFlow

5) DataDrift

6) Déploiement

7) Conclusion

Les données

Home Credit Default Risk

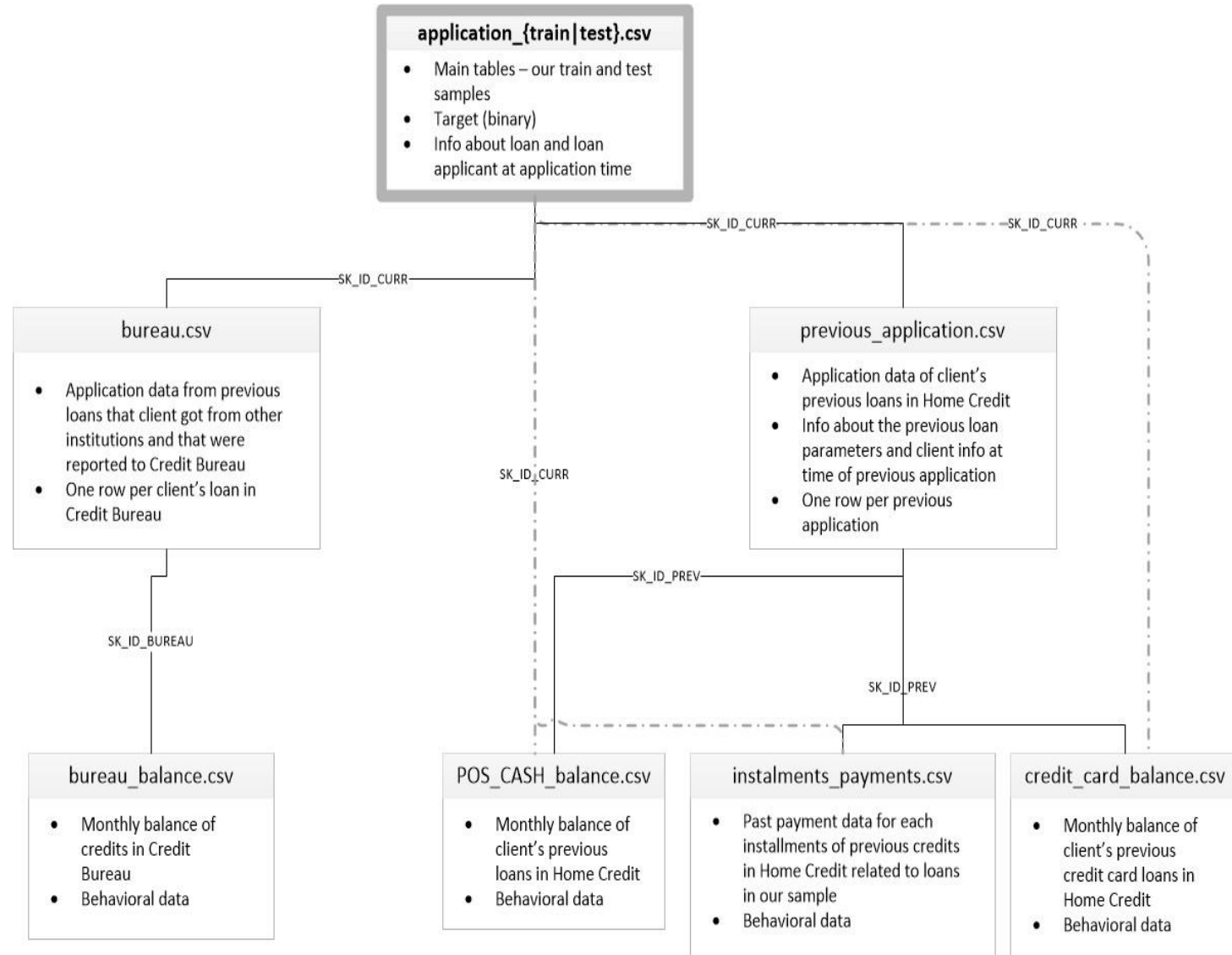
Il s'agit d'un jeu de données très connu sur Kaggle car ayant fait l'objet de nombreuses compétitions en termes de prédiction.

Le diagramme ci-contre présente l'ensemble des données et leur lien grâce à un identifiant unique pour chaque client/prêt

(« SK_ID_CURR/PREV »). Source:

<https://www.kaggle.com/code/willkoehrsen/start-here-a-gentle-introduction>

Les clients se composent en 2 datasets: application train et application test. Seul application train sera utilisée pour la modélisation et le dashboard. La partie « test » ne sera utilisée que pour l'analyse du data drift



Caractéristiques des fichiers

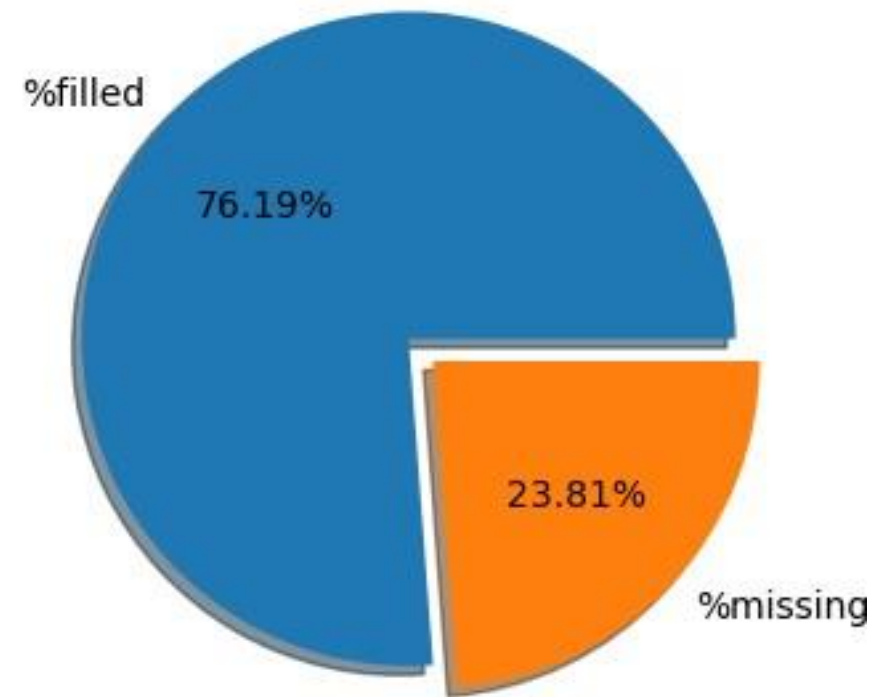
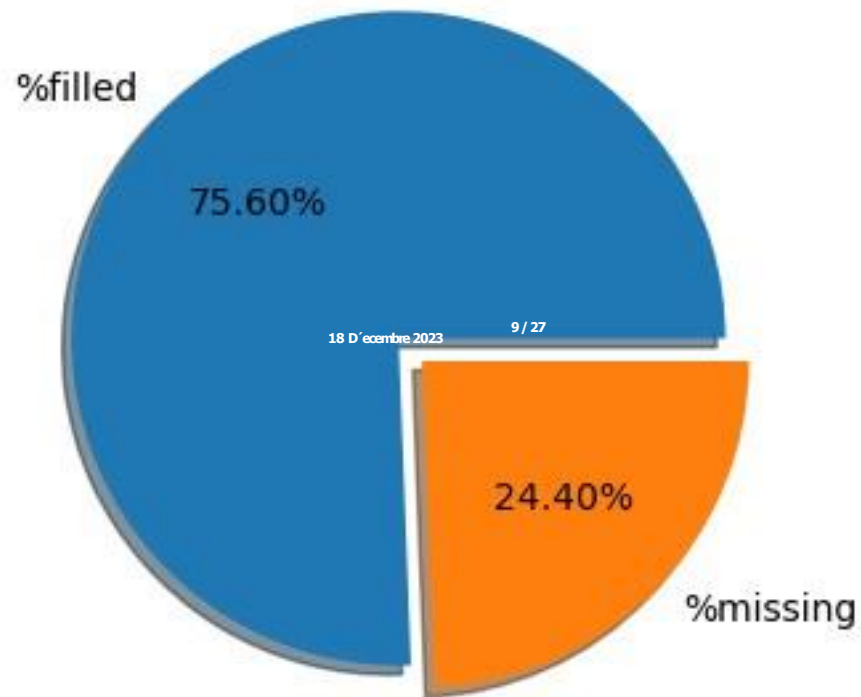
Nous avons 10 bases de données.

	nb_lignes	nb_cols	%missings	%doublons	object_dtype	float_dtype	int_dtype	bool_dtype	Mo_Memory
application_test.csv	48744	121	23.81	0.0	16	65	40	0	44.998
application_train.csv	307511	122	24.40	0.0	16	65	41	0	286.227
bureau.csv	1716428	17	13.50	0.0	3	8	6	0	222.620
bureau_balance.csv	27299925	3	0.00	0.0	1	0	2	0	624.846
credit_card_balance.csv	3840312	23	6.65	0.0	1	15	7	0	673.883
HomeCredit_columns_description.csv	219	5	12.15	0.0	4	0	1	0	0.008
installments_payments.csv	13605401	8	0.01	0.0	0	5	3	0	830.408
POS_CASH_balance.csv	10001358	8	0.07	0.0	1	2	5	0	610.435
previous_application.csv	1670214	37	17.98	0.0	16	15	6	0	471.481
sample_submission.csv	48744	2	0.00	0.0	0	1	1	0	0.744

2. Taux de remplissage

Taux de completion (application_train)

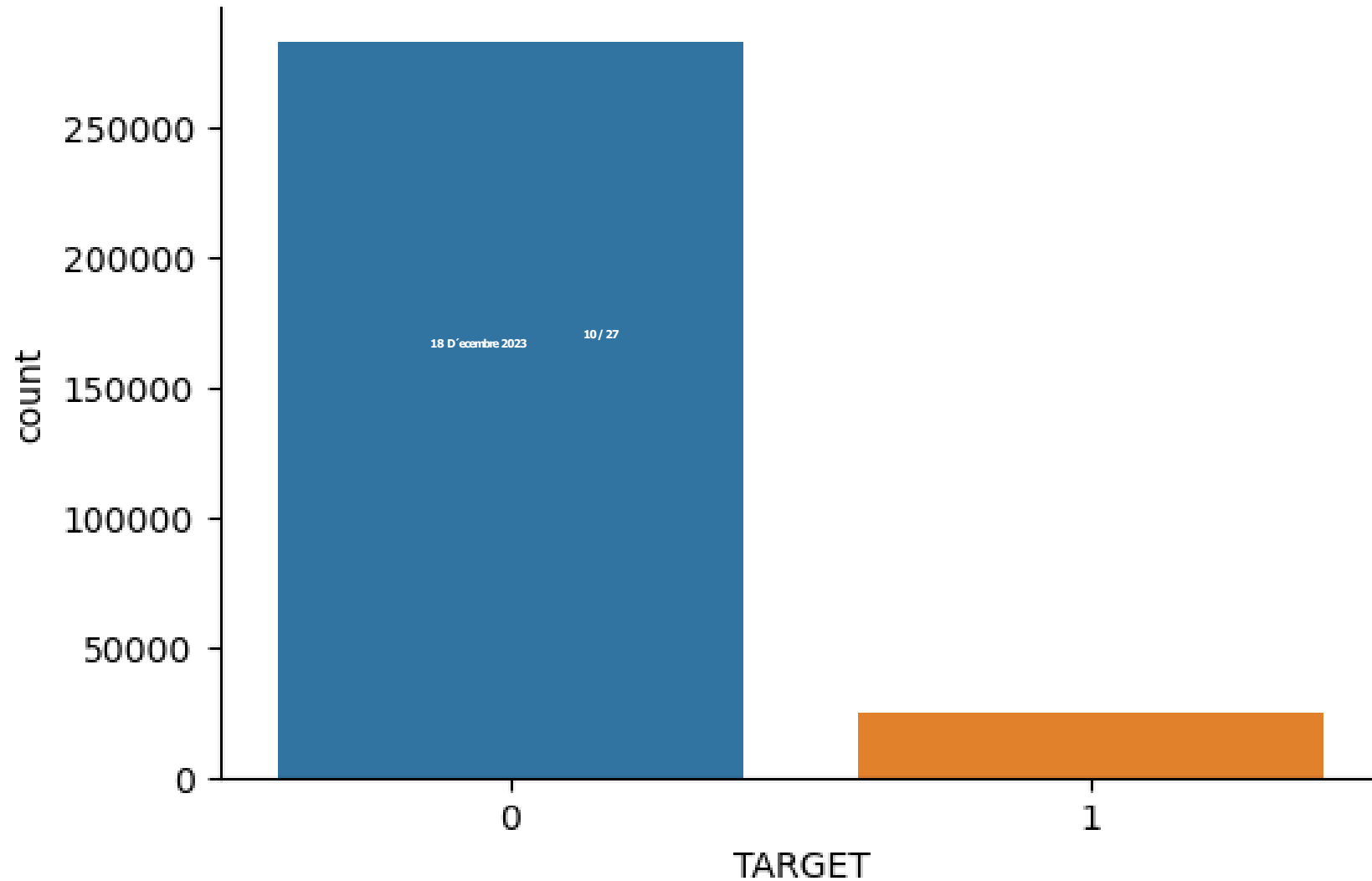
Taux de completion (application_test)



Analyse de la target

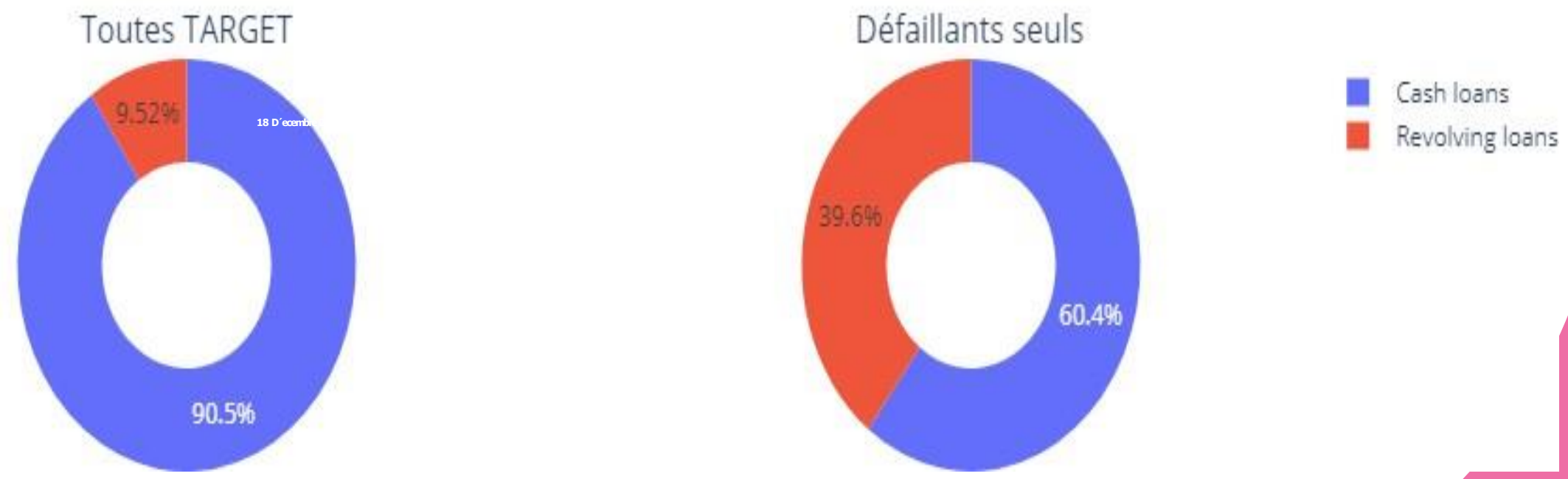
Distribution de la target

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



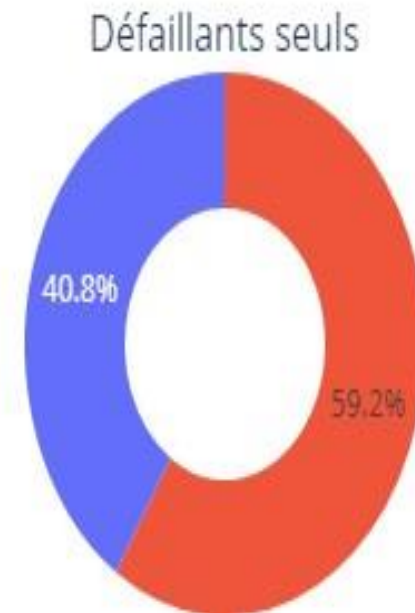
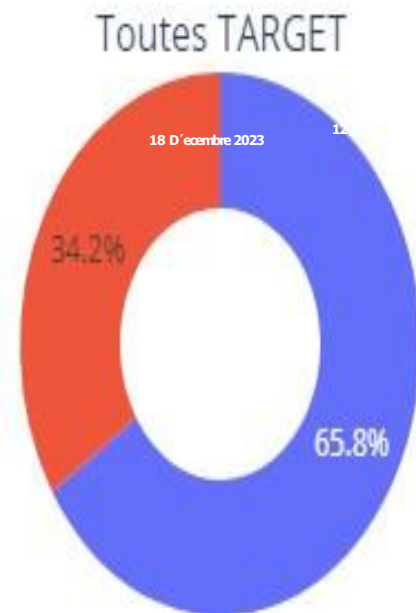
Distribution de la target : NAME CONTRACT TYPE

Répartition de la variable NAME_CONTRACT_TYPE



Distribution de la target : CODE GENDER

Répartition de la variable CODE_GENDER

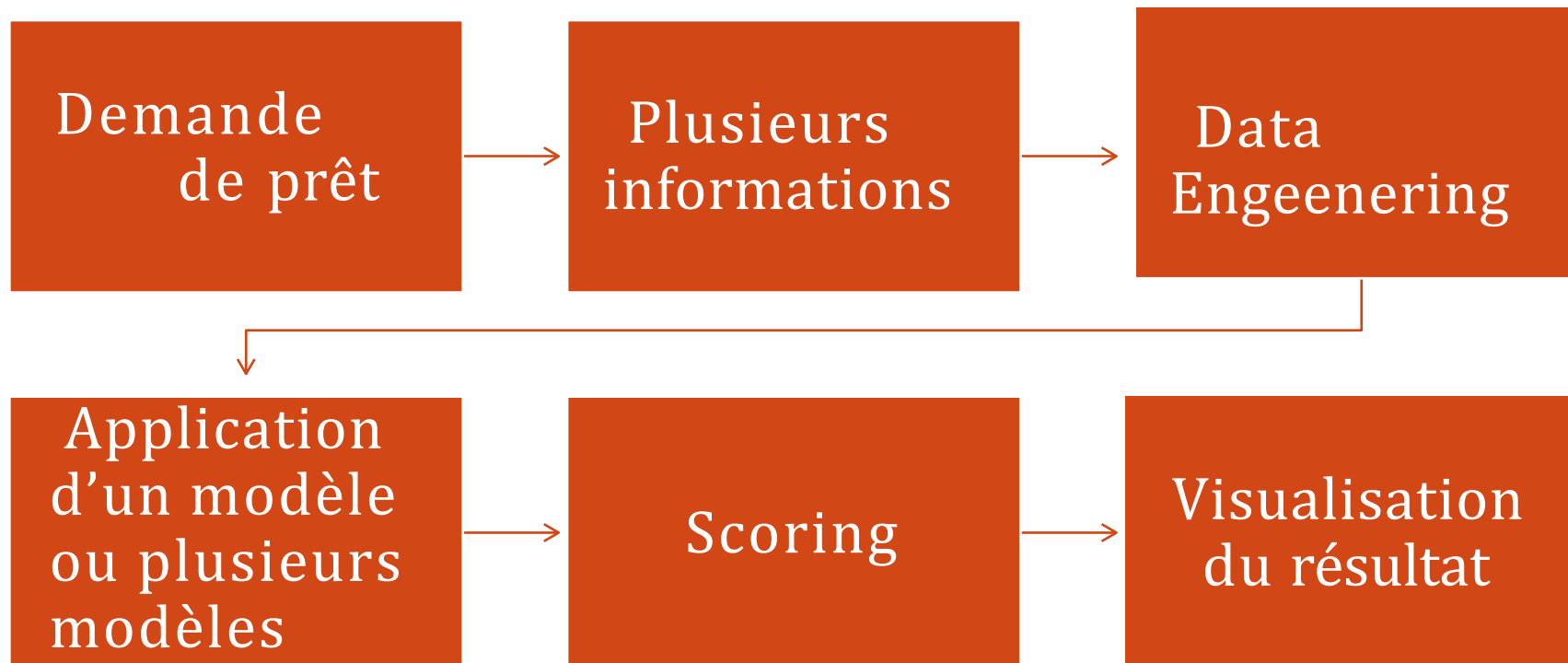


F
M
XNA

Plan de Présentation

- 1) Problématique
- 2) Exploration de données
- 3) Modélisation
- 4) MIFlow
- 5) DataDrift
- 6) Déploiement
- 7) Conclusion

SUIS-JE SOLVABLE ? : LE PRINCIPE



3. Modélisation : feautres engeneering (1/2)

1. Utilisation du **Kernel LightGBM** avec des caractéristiques simples comme base pour la majorité des tâches d'ingénierie des caractéristiques.
2. À l'aide d'un code synthétique, la génération d'un jeu de données relativement propre incluant :
 - ❑ **Encodage one-hot** des colonnes catégorielles du dataframe principal (application).
 - ❑ **Prétraitement** (nettoyage de certaines valeurs aberrantes) et **fusion de tous les ensembles de données** en un unique dataframe de niveau client, en supprimant les caractéristiques catégorielles.
 - ❑ **Création de nombreux indicateurs pertinents**, notamment des calculs supplémentaires sur les colonnes existantes (min, max, sum, mean), ainsi que l'ajout de pourcentages pour "standardiser" les valeurs importantes.

Modélisation : feautres engeneering (2/2)

3. Résultat : **un dataframe de 797** caractéristiques et 356 254 lignes/clients, avec une réduction de la mémoire utilisée **jusqu'à 65%** en convertissant les types de valeurs numériques.
4. Gestion des valeurs **NaN ou infinies restantes** : remplacement par la médiane de la colonne concernée pour **garantir la compatibilité** avec les modèles.
5. Split de train des données pour retrouver **le périmètre train/test grâce** à l'identifiant.

Modélisation : démarche (1/3)

- Train / test split (taille du jeu **de test** = 25%)
- Sous échantillonnage pour équilibrer le jeu d'entraînement par RandomUnderSampling:

Nombre d'éléments « target »	Jeu d'entraînement original	Jeu après sous échantillonnage
Valeur 0 (crédit accepté)	17 942	17 942
Valeur 1 (crédit refusé)	203 863	17 942

Modélisation : démarche (2/3)

- Définition d'une fonction de coût pour prendre en compte le fait qu'un faux négatif coûte 10 fois plus qu'un faux positif :
 - Création d'une **matrice de confusion** à partir de la valeur réelle et de la valeur prédite
 - Création de la fonction $\text{cost} = (10 \times \text{faux négatif} + \text{faux positif}) / \text{taille du dataframe}$
 - Cette fonction est ensuite utilisée avec **make scorer** pour être minimisée dans la recherche d'hyperparamètres de chaque modèle ainsi que dans une fonction permettant de trouver le seuil optimal de probabilité qui permettra de déterminer **le point d'équilibre** pour la décision d'octroi de crédit.
- Prédiction des résultats avec des modèles **XG BOOST & LightGBM** : donner une première évaluation de notre score, évaluer la pertinence de nos modèles futurs.

Modélisation : démarche (3/3)

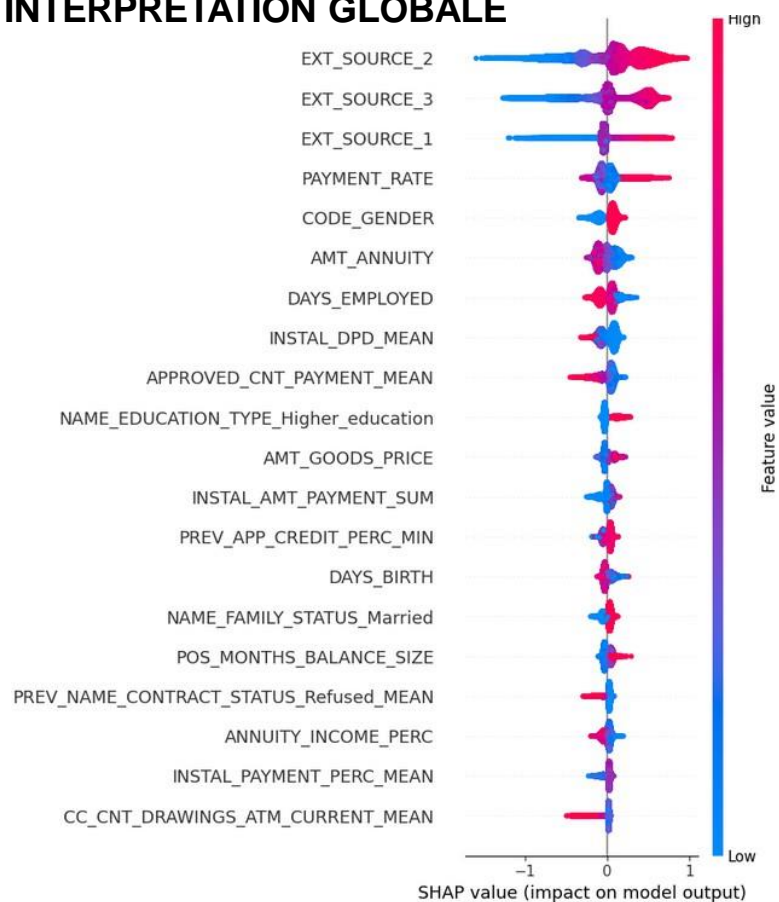
- Choix des modèles utilisés: les modèles **XG Boost Classif** et **LightGBM Classif** sont ceux qui ont obtenu les meilleurs résultats sur ce jeu de données.
- Recherche d'hyperparamètres via **GridSearch Cv** sur ces modèles en jouant sur la vitesse d'apprentissage (learning rate), le nombre d'arbres à entraîner (n_estimators) , la profondeur de l'arbre de décision (max depth) et la taille du jeu de données utilisé (en nombre d'éléments et de colonnes).
- Choix du modèle et des paramètres retenus selon les métriques établis sur jeu de test et d'entraînement (score custom, **AUC**, **accuracy**, ...) et le temps nécessaire.

Modélisation: features importance

SHAP

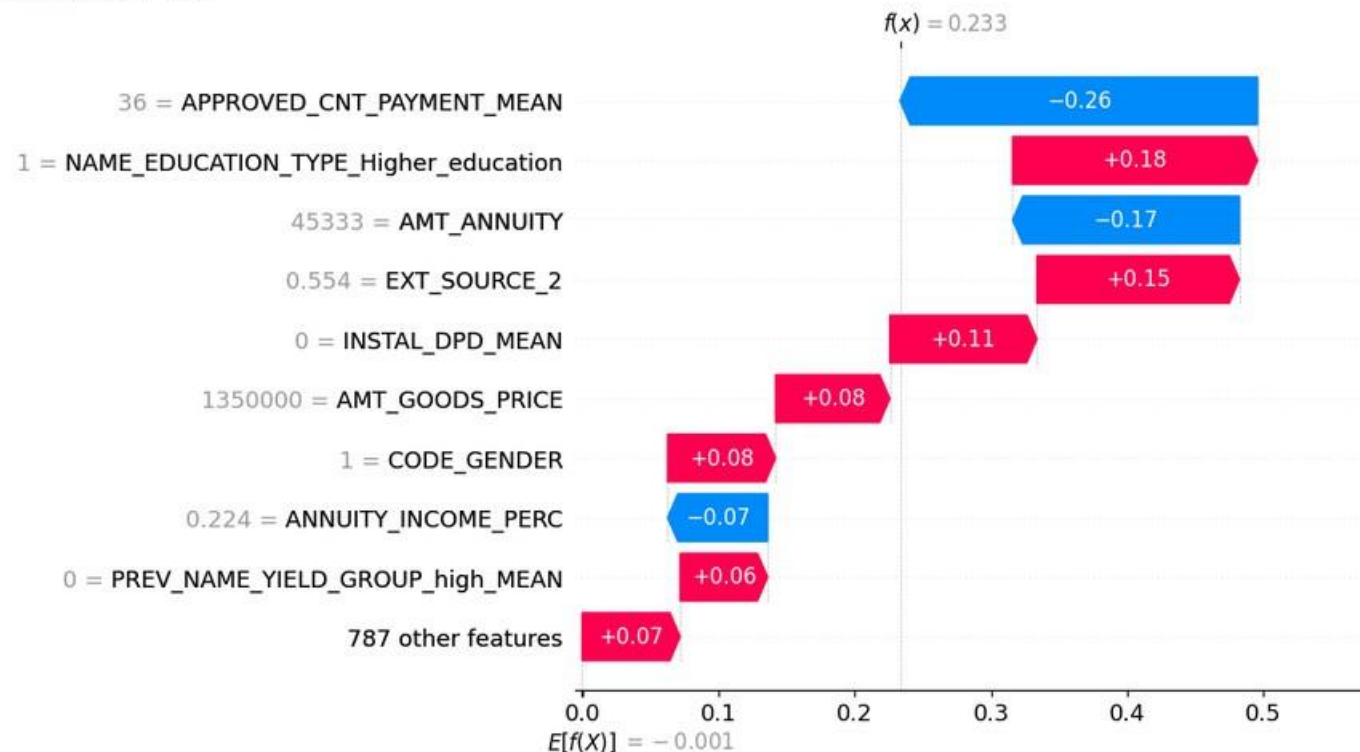
- Mesure des features importances globale et locale avec la librairie SHAP qui permet de calculer et mettre en forme les shapley values des features utilisés par notre modèle

INTERPRETATION GLOBALE



INTERPRETATION LOCALE

Client numero : 13017

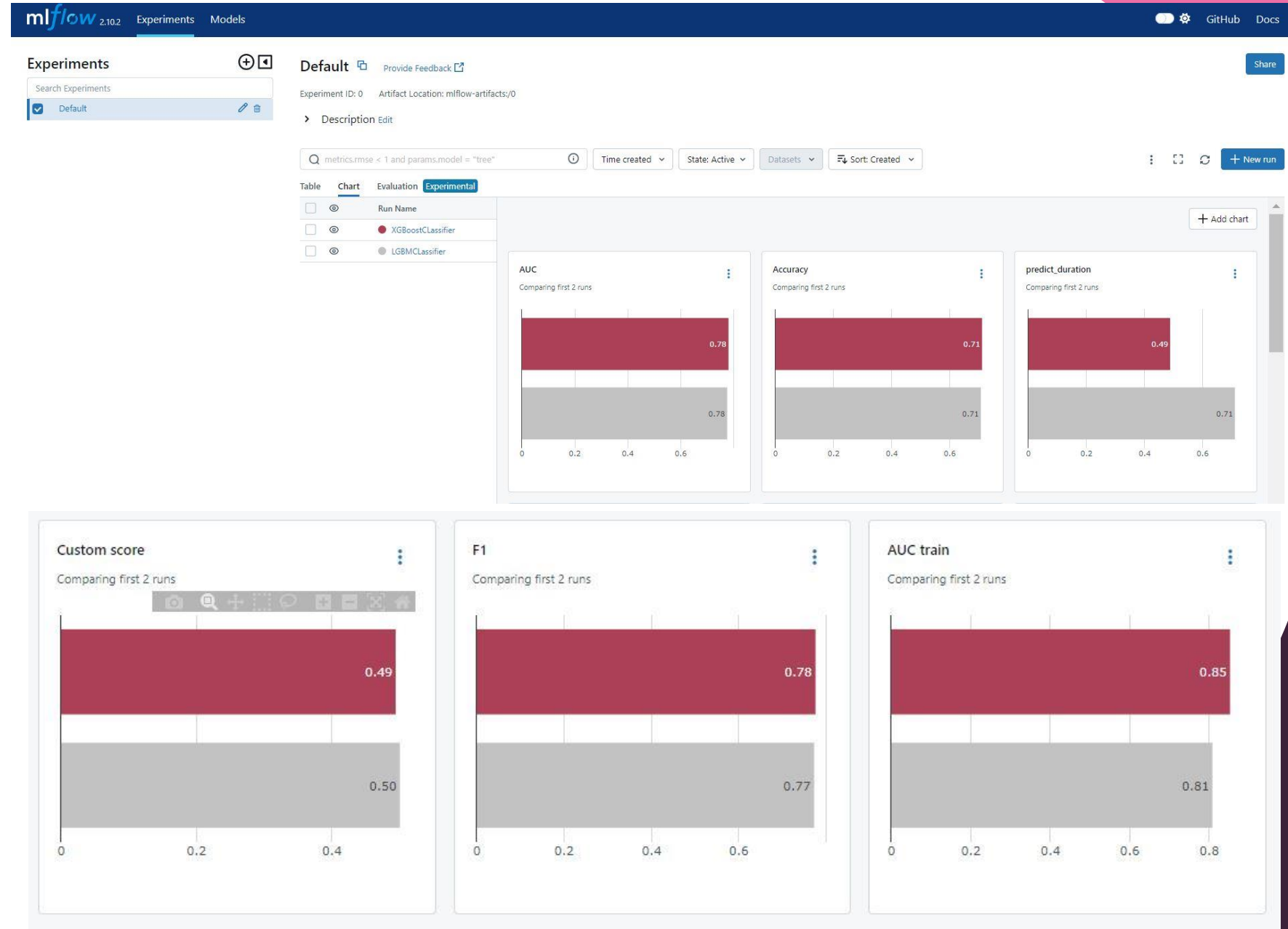


Plan de Présentation

- 1) Problématique
- 2) Exploration de données
- 3) Modélisation & Résultats
- 4) MIFlow
- 5) DataDrift
- 6) Déploiement
- 7) Conclusion

4. Modélisation: tracking et résultats avec le ML Flow

- Après obtention des meilleurs hyperparamètres de chaque modèle (LightGBM et XG Boost), on log les résultats de chaque modèle avec ses hyperparamètres dans ML Flow pour pouvoir faire une comparaison des modèles
- Log des paramètres utilisés et du modèle



Plan de Présentation

- 1) Problématique
- 2) Exploration de données
- 3) Modélisation & Résultats
- 4) MIFlow
- 5) DataDrift
- 6) Deploiement
- 7) Conclusion

5. Analyse du data drift (1/2)

➤ Métriques et seuil


- Colonnes numériques: divergence de Kullback-Leibler / seuil=0,1 (défaut)
- Colonnes catégoriques: indice de stabilité de population (PSI) / seuil=0,2

➤ Résultats:

798
Columns

17
Drifted Columns

0.0213
Share of Drifted Columns

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
> BURO_CREDIT_ACTIVE_Bad debt_MEAN	num			Not Detected	K-S p_value	1
> BURO_CREDIT_CURRENCY_currency 1_MEAN	num			Not Detected	K-S p_value	1

Analyse du data drift (2/2)

➤ Interprétation :

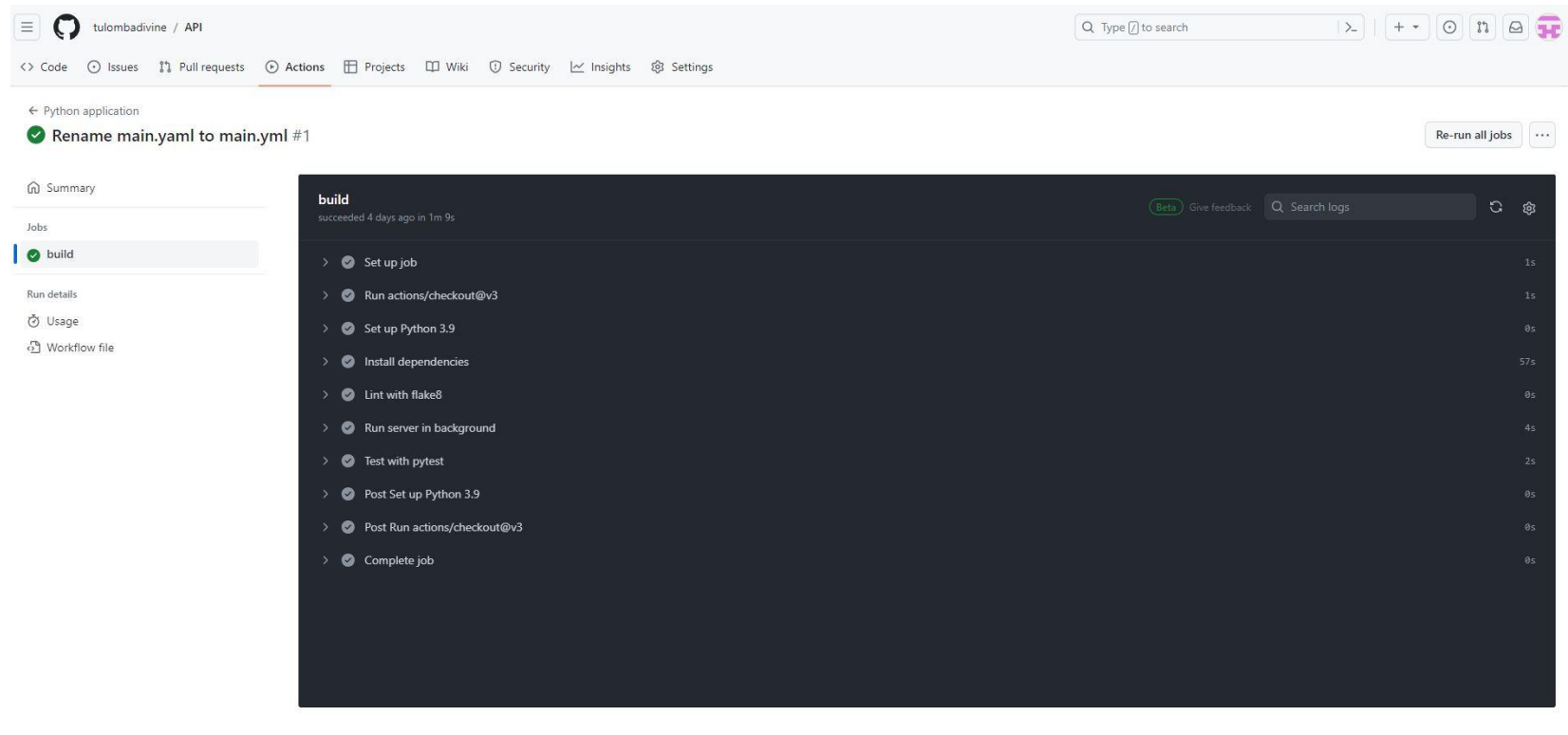
- Le data drift est limité entre les 2 jeux de données avec 2.13% des colonnes « drifted »
- Les colonnes drifted concernent des informations très spécifiques à chaque client sur leurs crédits précédents (min / max / moyenne / somme sur le nombre de mois, les montants des mensualités,...)
- Colonnes catégoriques ayant le plus de drift: type de prêt (cash / revolving) -> le jeu référence contient des prêts « revolving » alors que le courant n'en contient pas.

Plan de Présentation

- 1) Problématique
- 2) Exploration de données
- 3) Modélisation & Résultats
- 4) MIFlow
- 5) DataDrift
- 6) **Deploiement**
- 7) Conclusion

6. Déploiement de la solution

- Mise en place de 3 repositories Github :
- Développement de l'API avec Fast API : <https://github.com/tulombadivine/API>
- Dashboard avec Streamlit : <https://github.com/tulombadivine/Dashboard>
- Déploiement sur Heroku
- Mis en place du pipeline de déploiement continu avec des tests unitaires sur github actions



Demonstration :

- API:

- <https://github.com/tulombadivine/API>
- <https://app-opc-01e0e62f2bf5.herokuapp.com>

- Dashboard:

- <https://github.com/tulombadivine/Dashboard>
- <https://app-dashopc-a46c6003cb21.herokuapp.com>

Plan de Présentation

- 1) Problématique
- 2) Exploration de données
- 3) Modélisation & Résultats
- 4) MIFlow
- 5) DataDrift
- 6) Déploiement
- 7) Conclusion & Limites

7. Limites et points d'amélioration

- Augmentation des capacités : stockage de la base de données en dehors de l'application pour pouvoir intégrer l'ensemble des données (test + train) plutôt qu'un échantillon
- Complément d'informations sur les descriptions des colonnes: être en capacité de fournir des explications précises sur chaque feature utilisée et les intégrer dans le dashboard afin que l'interprétation du résultat soit plus parlante pour le chargé de relations client
- Mettre en place une sélection des variables pour avoir un modèle plus parcimonieux.

Merci pour votre attention

