Implémentez un modèle de scoring

Prêt à dépenser

PLE Coline

PROJET 7

04/12/2023

Plan de la présentation

- Présentation de la problématique et du jeu de données
- Présentation de la modélisation
- Présentation du pipeline de déploiement
- Présentation de l’analyse de la dérive des données
- Présentation /démo du dashboard interactif
- Discussion

RAPPEL DE LA PROBLEMATIQUE

PROBLEMATIQUE ET MISSIONS

Contexte:

Proposition de prêts à la consommation pour des clients ayant peu ou pas d'historique de prêts.

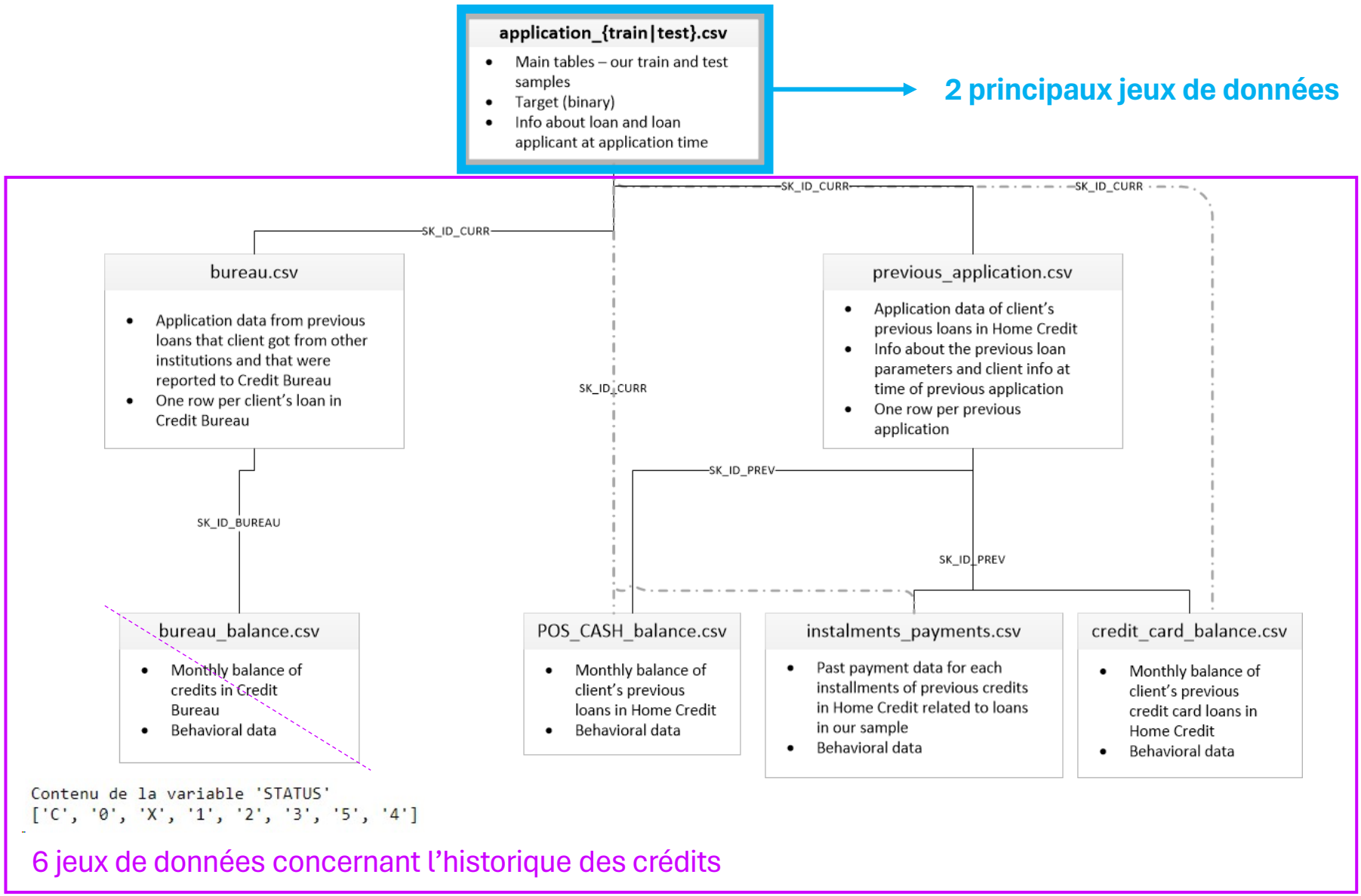
Besoins:

1. Modèle de classification calculant la probabilité qu'un client rembourse ou non son crédit pour la prise de décision.
2. Dashboard interactif pour un échange en toute transparence entre le chargé de relation client et le client.

Missions:

1. Construction d'un modèle de scoring donnant une prédiction sur la probabilité de faillite d'un client de façon automatique.
2. Création du dashboard interactif à destination des gestionnaires de la relation client.
3. Mise en production du modèle de scoring à l'aide d'une API, et du dashboard appelant l'API pour les prédictions.

PRESENTATION DES JEUX DE DONNEES



PREPARATION DES JEUX DE DONNEES

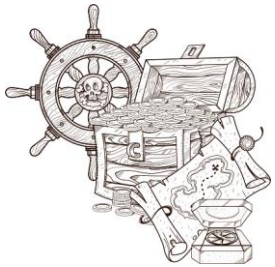
Ensemble des jeux de données

- Elimination des variables non pertinentes, redondantes ou fortement corrélées.
- Multiplication par -1 de toutes les variables négatives.
- Gestion des valeurs manquantes: Imputation par -1 (catégorielles) et -2 (numériques)

6 jeux de données concernant l'historique des crédits

- 1 seul type d'agrégation par variable (hors Montants crédits)
- Encodage manuel des variables catégorielles

2 principaux jeux de données



- Regroupement au niveau des secteurs d'activités, des emplois et du statut familial.
- Encodage manuel de certaines variables catégorielles (sexe, accompagnement, possession d'un bien immobilier ou d'une voiture…).
- Encodage avec le OneHotEncoder pour les autres variables catégorielles.
- Création de nouvelles variables:
  - Age du client, du téléphone et de la voiture (années).
  - Ancienneté dans le dernier emploi (années).
  - Comptage du nombre de façons de joindre le client.
  - Nombre de documents fournis.
  - Montant de l'annuité par rapport au revenu total du client (%).
  - Montant de l'annuité par rapport au montant du crédit (%).
  - Montant du crédit par rapport au revenu total du client (%).

JOINTURE DES JEUX DE DONNEES: 307511 clients et 98 variables (train) / 48744 clients et 97 variables (test)

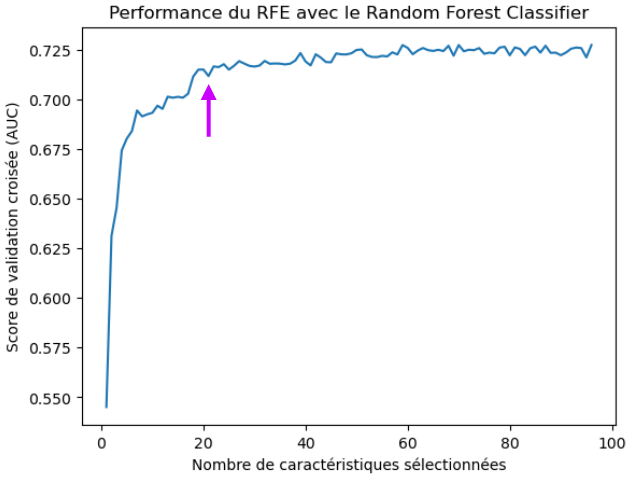
PRESENTATION DE LA MODELISATION:
Métriques métiers et RFE-CV

Le score métier = Métrique de référence

Postulat: Cout d'un FN 10x > FP

$$\frac{10 * FN + FP}{TP + TN + FP + 10 * FN}$$

Réduction du nombre de variables par RFE-CV (Random Forest Classifier -75000 clients)



Point d'inflexion/coude au niveau de 20 variables
→ Conservation des **30 premières variables**

Déséquilibre des classes ⟶ **Mauvais apprentissage en ML**
92% de clients sans risque
8% de clients à risque

PRESENTATION DE LA MODELISATION:
Gestion du déséquilibre des classes

La gestion du déséquilibre des classes (Réalisée sur un Random Forest Classifier)

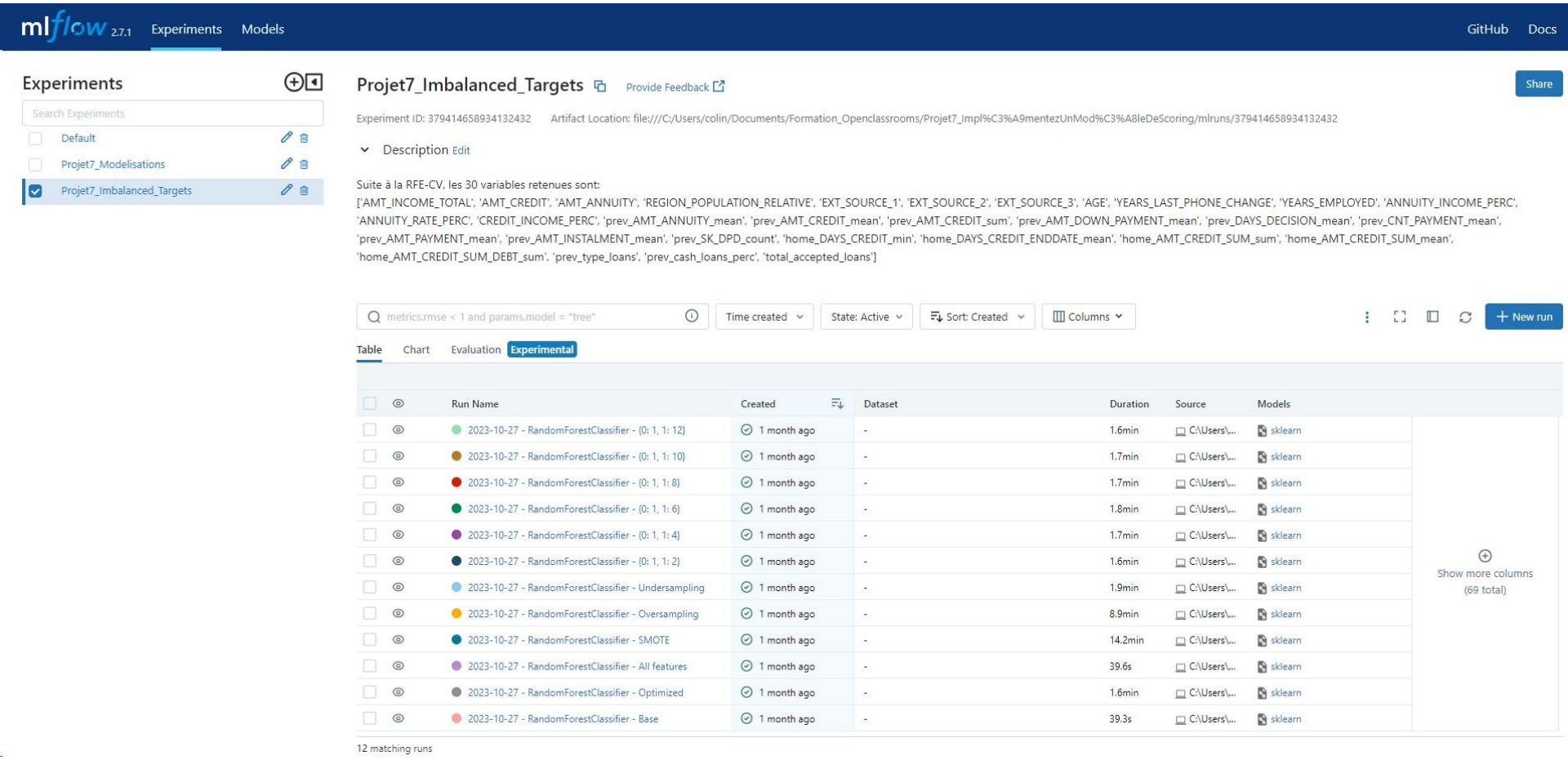
- SMOTE : Création synthétique d'échantillons dans la classe minoritaire.
- Oversampling: Sur-échantillonnage de la classe minoritaire.
- Undersampling: Sous-échantillonnage de la classe majoritaire.
- Attribution de poids différents aux 2 classes.

	SMOTE	RFC oversampling	RFC undersampling	RFC weight 2	RFC weight 4	RFC weight 6	RFC weight 8	RFC weight 10	RFC weight 12
Model	Pipeline	Pipeline	Pipeline	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier
Training Time	788.2608	471.5283	49.6051	76.5559	79.1562	78.189	78.2867	78.0988	75.2507
Train Accuracy	82.1311	94.366	86.5476	94.6876	94.39	92.8949	92.3974	91.1446	90.5332
Test Accuracy	75.889	87.4673	82.0805	85.934	86.8185	85.8755	85.669	84.7016	84.3016
Train AUC	0.9171	0.9891	0.9489	0.9896	0.9871	0.9831	0.9789	0.9734	0.9688
Test AUC	0.715	0.7477	0.7506	0.7516	0.7524	0.7496	0.7478	0.7459	0.7444
Train Recall	0.9037	0.9695	0.9046	0.9526	0.9549	0.9668	0.9654	0.9679	0.9666
Test Recall	0.5106	0.3072	0.4699	0.3811	0.3372	0.3623	0.3635	0.3907	0.3966
Train Precision	0.2992	0.5923	0.3654	0.6094	0.5951	0.5331	0.5156	0.4762	0.459
Test Precision	0.1697	0.2632	0.2176	0.2533	0.2579	0.2458	0.242	0.2331	0.2282
Train f1	0.4495	0.7353	0.5206	0.7433	0.7332	0.6872	0.6722	0.6383	0.6224
Test f1	0.2548	0.2835	0.2974	0.3043	0.2923	0.2929	0.2906	0.292	0.2897
Best threshold	0.4545	0.5657	0.6061	0.2121	0.3131	0.3636	0.4141	0.4444	0.4747
Train best score	0.2324	0.0768	0.1906	0.0847	0.0861	0.0929	0.0987	0.1093	0.1161
Test best score	0.4402	0.4182	0.4074	0.4072	0.414	0.4131	0.4142	0.4129	0.4139

CHOIX POUR LA GESTION DU DESEQUILIBRE DES CLASSES: Attribution d'un poids {0:1,1:2}

PRESENTATION DE LA MODELISATION:
Gestion du déséquilibre des classes

La gestion du déséquilibre des classes: Le tracking sur MLFlow



PRESENTATION DE LA MODELISATION:
Gestion du déséquilibre des classes

La gestion du déséquilibre des classes: Le tracking sur MLFlow

mlflow 2.7.1 Experiments Models

Projet7_Imbalanced_Targets ›

2023-10-27 - RandomForestClassifier - {0: 1, 1: 2}

Run ID: 6c34dbe671b84920bd945e9743c7940f Date: 2023-10-27 19:27:24

Status: FINISHED Lifecycle Stage: active

› Description Edit

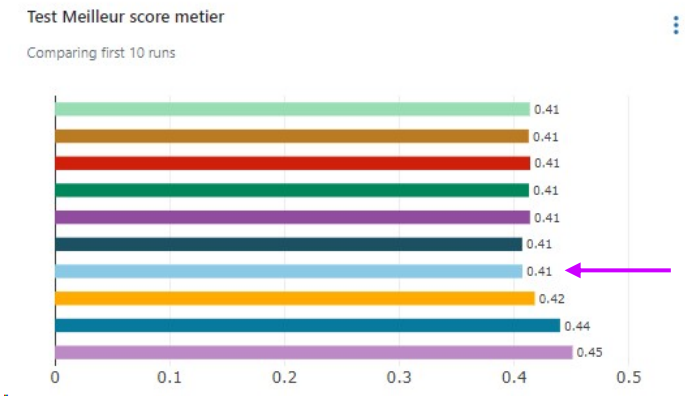
› Datasets

⌄ Parameters (19)

Name	Value	Name	Value
Number of Features	30	min_samples_leaf	1
bootstrap	True	min_samples_split	2
ccp_alpha	0.0	min_weight_fraction_leaf	0.0
class_weight	{0: 1, 1: 2}	n_estimators	400
criterion	gini	n_jobs	-1
max_depth	16	oob_score	False
max_features	log2	random_state	42
max_leaf_nodes	None	verbose	0
max_samples	None	warm_start	False
min_impurity_decrease	0.0		

⌄ Metrics (14)

Name	Value
Test AUC	0.752
Test Accuracy	85.93
Test Meilleur score metier	0.407
Test Precision	0.253
Test Recall	0.381
Test f1	0.304
Train AUC	0.99
Train Accuracy	94.69
Train Meilleur score metier	0.085
Train Meilleur seuil	0.212
Train Precision	0.609
Train Recall	0.953
Train f1	0.743
Training Time	76.56



MODELISATION:
Test de 4 modèles avec les paramètres de base

Paramètres *:

- Séparation du trainset: 80% en train et 20% en test
- Standardisation des données avec le MinMaxScaler()
- Random state = 42
- class_weight(s) = {0:1, 1:2} (LR, CatBoostClassifier, LGBMClassifier) / scale_pos_weight =2 (XGBClassifier)

	DummyClassifier	LogisticRegression	CatBoostClassifier	LGBMClassifier	XGBClassifier
Training Time	0.0081	8.1704	28.9202	1.2504	18.7926
Train Accuracy	91.9271	82.4433	84.1115	82.7786	83.5587
Test Accuracy	91.9272	82.5553	81.3456	82.1683	80.8676
Train AUC	0.5	0.6999	0.8652	0.8054	0.8676
Test AUC	0.5	0.7029	0.7677	0.7675	0.7616
Train Recall	0.0	0.3619	0.6885	0.5617	0.6985
Test Recall	0.0	0.3563	0.5223	0.5096	0.5156
Train Precision	0.0	0.1906	0.2936	0.2489	0.287
Test Precision	0.0	0.1902	0.2217	0.2287	0.2147
Train f1	0.0	0.2497	0.4117	0.345	0.4069
Test f1	0.0	0.248	0.3113	0.3157	0.3032
Best threshold	0.0808	0.2222	0.2121	0.2222	0.2121
Train best score	0.4676	0.4367	0.3141	0.3721	0.3146
Test best score	0.4676	0.4375	0.3961	0.3942	0.4018

* LR: max_iter=500

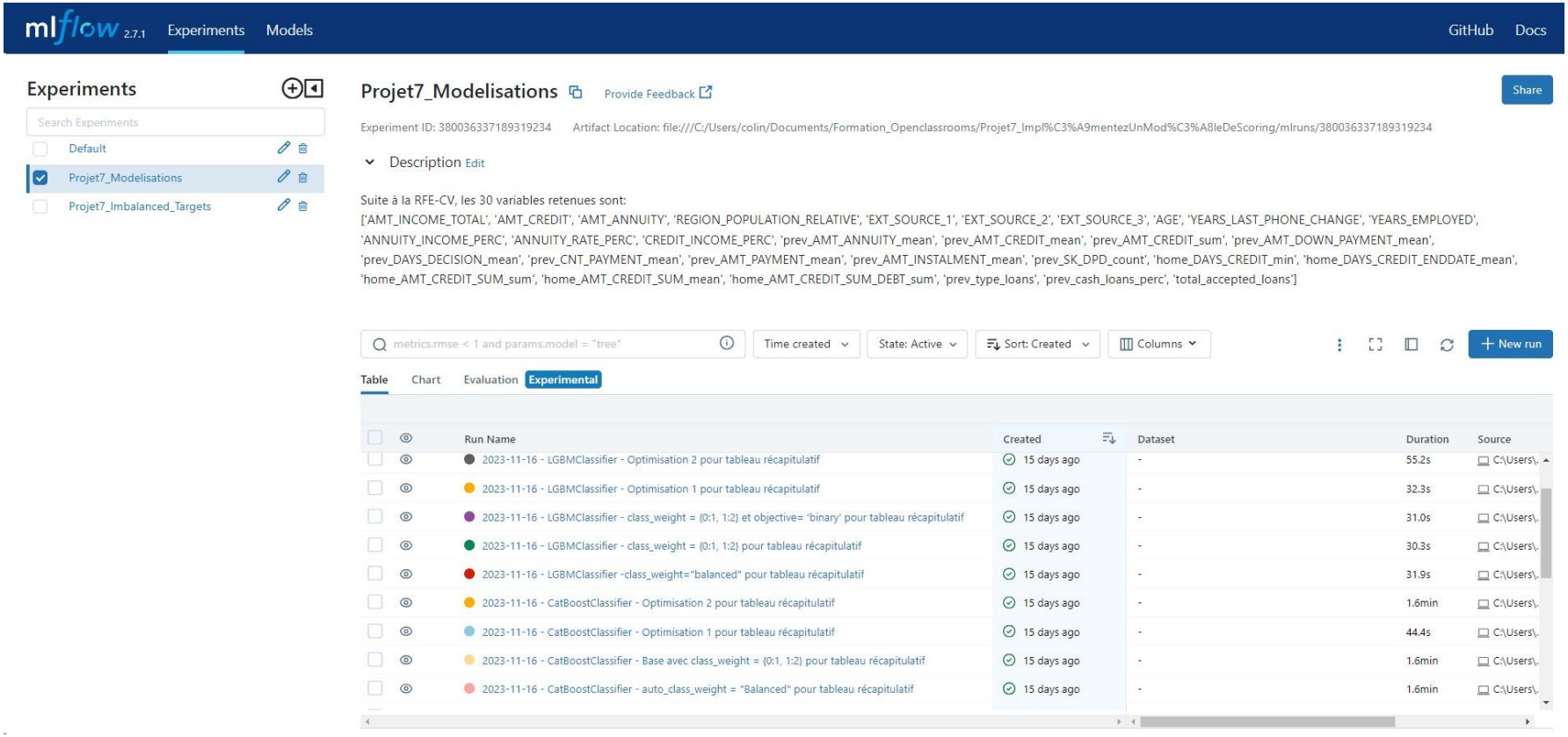
MODELISATION:
Test de 4 modèles avec optimisation des paramètres

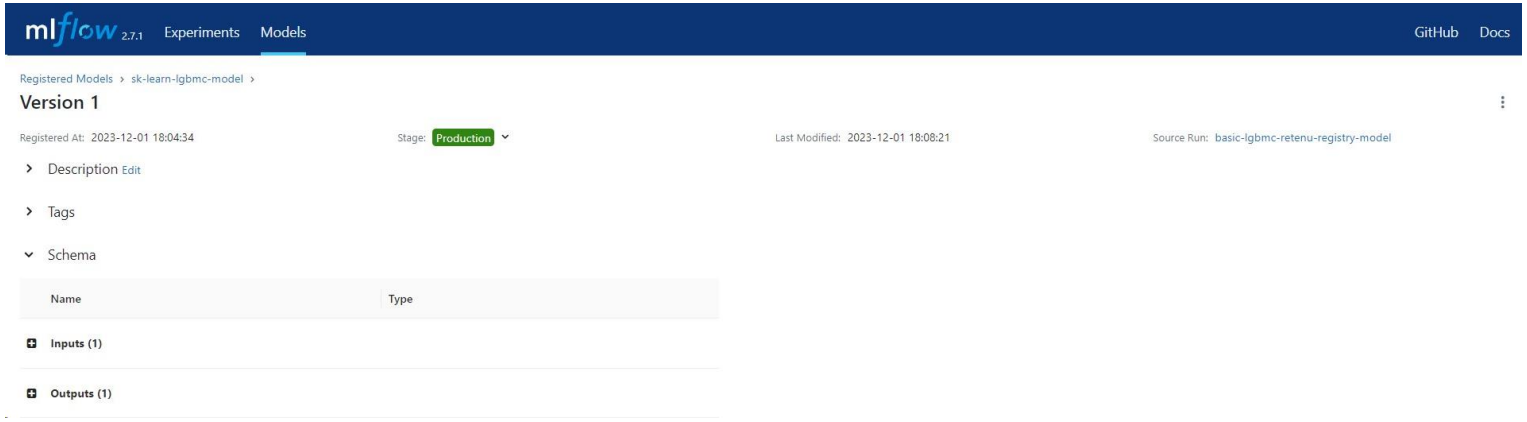
OPTIMISATION 1 (via GridSearchCV)	OPTIMISATION 2 (via GridSearchCV)
Logistic Regression random_state = 42, cv=5, scoring = scorer_metier, class_weight = {0:1, 1:2}, n_jobs=-1 **C** : [10, 20, 50, 100, **200**] **max_iter** : [500, 1000, **2500**, 5000]	Modèle peu performant de base Pas d'amélioration après recherche de meilleurs paramètres (cf slides supplémentaires)
CatBoostClassifier random_state = 42, cv=5, scoring = scorer_metier, class_weights = {0:1, 1:2}, n_jobs=-1, verbose=0 **iterations** : [50, 100, **200**] **depth:** [2, 4, **8**] **learning_rate:** [0.01, 0.05, **0.1**]	**CatBoostClassifier** random_state = 42, cv=5, scoring = scorer_metier, class_weights = {0:1, 1:2}, n_jobs=-1, verbose=0 **iterations** : [100, 200, 400, **1000**] **depth:** [**6**, 8, 10] **learning_rate:** [0.005, 0.01, 0.05, **0.1**]
LGBMClassifier random_state = 42, cv=5, scoring = scorer_metier, class_weights = {0:1, 1:2}, n_jobs=-1, objective = 'binary' **learning_rate** : [0.001, 0.01, 0.05, **0.1**] **n_estimators** : [25, 50, 100, **200**] **num_leaves** : [4, 8, **16**, 32]	**LGBMClassifier** random_state = 42, cv=5, scoring = scorer_metier, class_weights = {0:1, 1:2}, n_jobs=-1, objective = 'binary' **learning_rate** : [0.005, 0.01, 0.05, **0.1**] **n_estimators** : [100, 200, 400, **1000**] **num_leaves** : [4, 8, **16**, 32]
XGBClassifier random_state = 42, cv=5, scoring = scorer_metier, scale_pos_weight= 2, n_jobs=-1 **learning_rate**: [0.01, 0.05, **0.1**] **max_depth**: [2, 4, **8**] **n_estimators**: [50, 100, **200**]	Modèle overfittant le plus Long à entraîner (cf slides supplémentaires)

MODELISATION:
Choix du modèle final

	CatBoost Base auto_class_weight	CatBoost Base avec class_weight retenu	CatBoost 1ère opt	CatBoost 2ème opt	LGBMC Base class_weight	LGBMC Baseavec class_weight retenu	LGBMC 1ère opt	LGBMC 2ème opt
Training Time	69.8722	70.6693	20.8735	70.2175	4.6839	3.4623	5.2072	15.2404
Train Accuracy	81.1421	84.1115	82.1933	82.755	80.801	* 82.7786	80.5299	83.8078
Test Accuracy	78.4482	81.3456	81.1115	80.3262	80.4042	* 82.1683	79.9294	80.4107
Train AUC	0.8857	0.8652	0.8172	0.8607	0.8032	0.8054	0.805	0.8967
Test AUC	0.7641	0.7677	0.7669	0.7688	0.7674	0.7675	0.7679	0.7679
Train Recall	0.7899	0.6885	0.6056	0.7024	0.5872	0.5617	0.6066	0.7774
Test Recall	0.5813	0.5223	0.5321	0.5432	0.5482	0.5096	0.5519	0.539
Train Precision	0.2709	0.2936	0.2506	0.2764	0.23	0.2489	0.2311	0.3036
Test Precision	0.2052	0.2217	0.2213	0.2153	0.2172	0.2287	0.2131	0.2152
Train f1	0.4034	0.4117	0.3545	0.3967	0.3306	0.345	0.3347	0.4367
Test f1	0.3034	0.3113	0.3126	0.3083	0.3112	0.3157	0.3075	0.3076
Best threshold	0.5354	0.2121	0.2121	0.202	0.596	0.2222	0.202	0.202
Train best score	0.296	0.3141	0.3611	0.3196	0.3784	0.3721	0.3737	0.2786
Test best score	0.3985	0.3961	0.3947	0.3969	0.3947	0.3942	0.397	0.3977

MODELISATION:
Le tracking *via* MLFlow





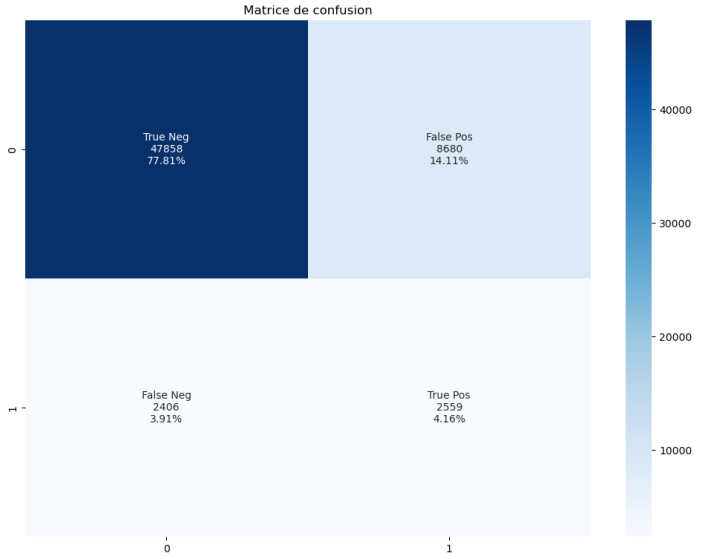
MODELISATION:
Le tracking *via* MLFlow

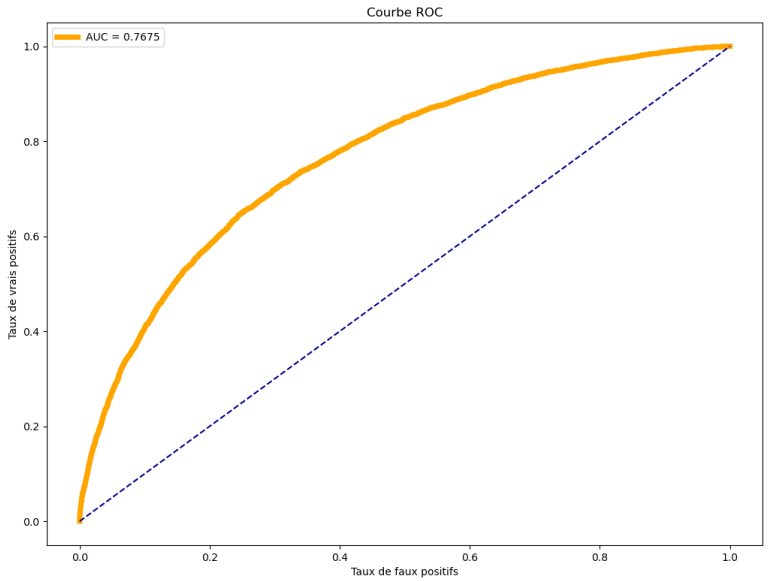
Métriques pour le LGBMClassifier retenu

Metrics (14)

Name	Value
Test AUC	0.768
Test Accuracy	82.17
Test Meilleur score metier	0.394
Test Precision	0.229
Test Recall	0.51
Test f1	0.316
Train AUC	0.805
Train Accuracy	82.78
Train Meilleur score metier	0.372
Train Meilleur seuil	0.222
Train Precision	0.249
Train Recall	0.562
Train f1	0.345
Training Time	3.378

Matrice de confusion et courbe ROC sur le test





MODELISATION:
Présentation du pipeline de déploiement

Lien du projet GitHub: https://github.com/colple/Implementez_un_modele_de_scoring

Implementez_un_modele_de_scoring / .github / workflows / api_heroku.yml

colple Update api_heroku.yml ✓

Code | Blame | 17 lines (14 loc) · 333 Bytes | Code 55% faster with GitHub Copilot

```yaml
name: Deploiement de l'API sur Heroku

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
        with:
          heroku_api_key: ${{secrets.HEROKU_API_KEY}}
          heroku_app_name: "modele-scoring-credits"
          heroku_email: "colineple@yahoo.fr"

```

Implementez_un_modele_de_scoring / .github / workflows / pytest_tests.yml

colple Create pytest_tests.yml ✓

Code | Blame | 24 lines (19 loc) · 405 Bytes | Code 55% faster with GitHub Copilot

```yaml
name: Tests

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: windows-latest
    steps:
      - uses: actions/checkout@v2

      - name: Set up Python
        uses: actions/setup-python@v2
        with:
          python-version: 3.9

      - name: Install dependencies
        run: |
          pip install -r requirements.txt

      - name: Run unit tests
        run: python -m pytest
```

MODELISATION:
Présentation du pipeline de déploiement

Prêt à dépenser

build
succeeded 3 days ago in 5m 34s

Search logs

- Set up job — 1s
- Run actions/checkout@v2 — 15s
- Set up Python — 0s
- Install dependencies — 4m 47s
- Run unit tests — 25s

```
1  ▶ Run python -m pytest
6  ============================= test session starts =============================
7  platform win32 -- Python 3.9.13, pytest-7.4.0, pluggy-1.3.0
8  rootdir: D:\a\Implementez_un_modele_de_scoring\Implementez_un_modele_de_scoring
9  collected 10 items
10
11 test_pytest_tests.py ..........                                      [100%]
12
13 ============================== warnings summary ===============================
14 C:\hostedtoolcache\windows\Python\3.9.13\x64\lib\site-packages\plotly\express\imshow_utils.py:24
15   C:\hostedtoolcache\windows\Python\3.9.13\x64\lib\site-packages\plotly\express\imshow_utils.py:24: DeprecationWarning: `np.bool8` is a deprecated alias for `np.bool_`.  (Deprecated NumPy 1.24)
16     np.bool8: (False, True),
17
18 -- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
19 ======================= 10 passed, 1 warning in 23.17s ========================
```

MINGW64:/c/Users/colin/Documents/Projet_Github/Implementez_un_modele_de_scori...

```
   78f6b03..532a8f7  main -> main

colin@Coline MINGW64 ~/Documents/Projet_Github/Implementez_un_modele_de_scoring
(main)
$ git add Notebooks/notebook_5_rfecv_imbalanced.ipynb
warning: in the working copy of 'Notebooks/notebook_5_rfecv_imbalanced.ipynb', L
F will be replaced by CRLF the next time Git touches it

colin@Coline MINGW64 ~/Documents/Projet_Github/Implementez_un_modele_de_scoring
(main)
$ git commit -m "Modification du nom du notebook dans le notebook"
[main 226f79e] Modification du nom du notebook dans le notebook
 1 file changed, 1 insertion(+), 1 deletion(-)

colin@Coline MINGW64 ~/Documents/Projet_Github/Implementez_un_modele_de_scoring
(main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 401 bytes | 401.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/colple/Implementez_un_modele_de_scoring.git
   532a8f7..226f79e  main -> main
```

LA DERIVE DES DONNEES:
Présentation du datadrift

Prêt à dépenser

Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

30	8	0.267
Columns	Drifted Columns	Share of Drifted Columns

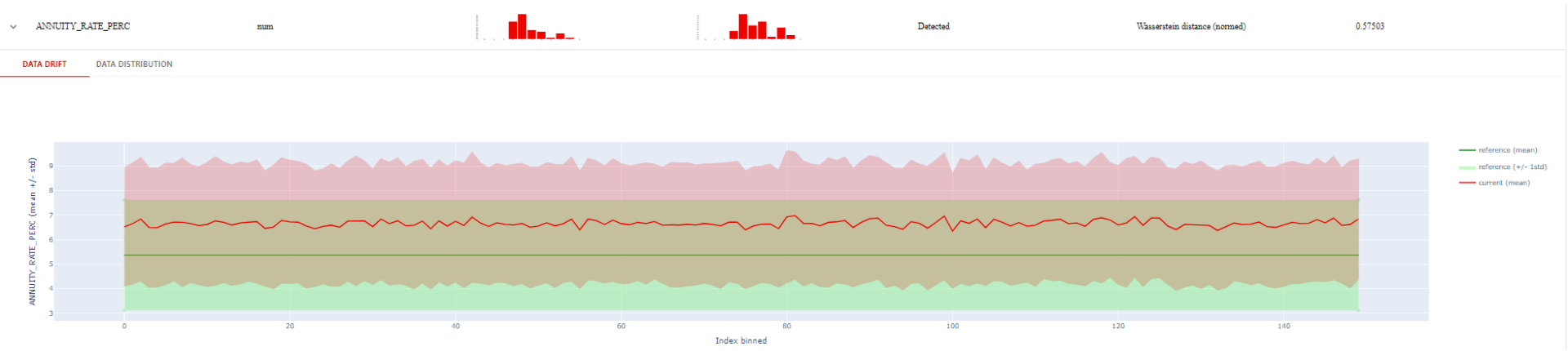
Data Drift Summary

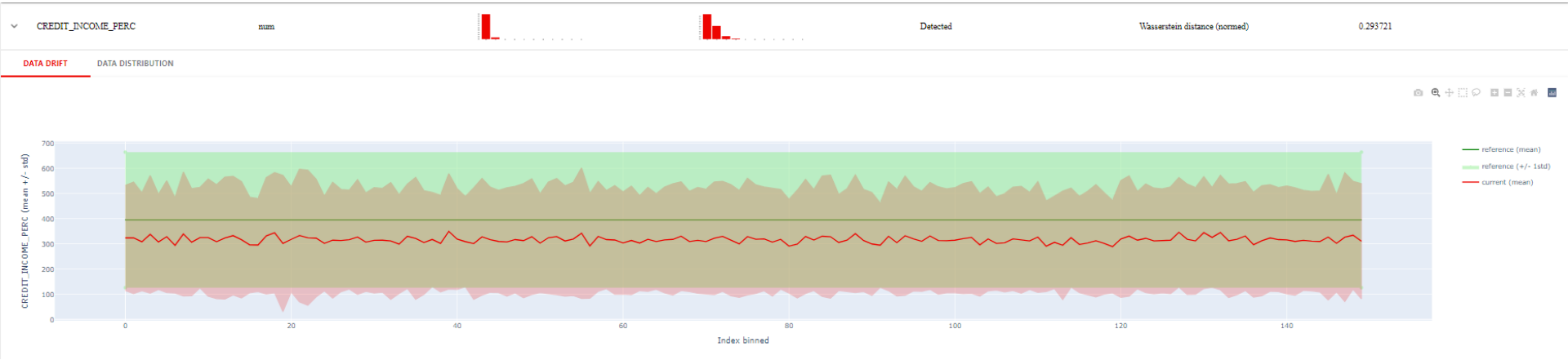
Drift is detected for 26.667% of columns (8 out of 30).

Search

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
ANNUITY_RATE_PERC	num			Detected	Wasserstein distance (normed)	0.57503
CREDIT_INCOME_PERC	num			Detected	Wasserstein distance (normed)	0.293721
EXT_SOURCE_1	num			Detected	Wasserstein distance (normed)	0.285364
AMT_CREDIT	num			Detected	Wasserstein distance (normed)	0.207334
AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.160558
prev_type_loans	num			Detected	Wasserstein distance (normed)	0.160321
YEARS_LAST_PHONE_CHANGE	num			Detected	Wasserstein distance (normed)	0.145448
total_accepted_loans	num			Detected	Wasserstein distance (normed)	0.112998
prev_AMT_PAYMENT_mean	num			Not Detected	Wasserstein distance (normed)	0.097969

LA DERIVE DES DONNEES:
Présentation du datadrift





- Pas de dérive significative sur l'ensemble des variables.
- 8 dérives dont 1 assez conséquente (Montant de l'annuité par rapport au montant du crédit).

DASHBOARD INTERACTIF ET CONCLUSION

1. Version contenant les 48744 clients:

https://colple-implementez-un-m-applicationsstreamlit-interactif-e6beqf.streamlit.app/

2. Version ne comprenant que les 1000 premiers clients:
https://implementezunmodeledescoring-vuwuspwad8k6jktck7tawc.streamlit.app/

CONCLUSION

- ✓ Elaboration d'un modèle de ML pour la décision automatique d'octroi ou non du prêt.
- ✓ Développement du tableau de bord interactif pour le chargé de clientèle.
- ✓ Déploiement *via* un pipeline continu.
- ✓ Axe d'amélioration:
  - Plus de données permettant de réduire la dérive des données (réentraînement du modèle).
  - Utilisation d'Hyperopt à la place de GridSearchCV (utilisation des connaissances acquises des essais précédents pour guider la recherche vers les combinaisons qui sont plus susceptibles d'être les meilleures).

- Débat / Réflexion -

Informations supplémentaires: Présentation du jeu de données

```
# Informations sur le jeu de données
bureau.info(verbose = True, show_counts = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1716428 entries, 0 to 1716427
Data columns (total 17 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_CURR              1716428 non-null  int64
 1   SK_ID_BUREAU            1716428 non-null  int64
 2   CREDIT_ACTIVE           1716428 non-null  object
 3   CREDIT_CURRENCY         1716428 non-null  object
 4   DAYS_CREDIT             1716428 non-null  int64
 5   CREDIT_DAY_OVERDUE      1716428 non-null  int64
 6   DAYS_CREDIT_ENDDATE     1610875 non-null  float64
 7   DAYS_ENDDATE_FACT       1082775 non-null  float64
 8   AMT_CREDIT_MAX_OVERDUE  591940 non-null   float64
 9   CNT_CREDIT_PROLONG      1716428 non-null  int64
 10  AMT_CREDIT_SUM          1716415 non-null  float64
 11  AMT_CREDIT_SUM_DEBT     1458759 non-null  float64
 12  AMT_CREDIT_SUM_LIMIT    1124648 non-null  float64
 13  AMT_CREDIT_SUM_OVERDUE  1716428 non-null  float64
 14  CREDIT_TYPE             1716428 non-null  object
 15  DAYS_CREDIT_UPDATE      1716428 non-null  int64
 16  AMT_ANNUITY             489637 non-null   float64
dtypes: float64(8), int64(6), object(3)
memory usage: 222.6+ MB
```

- CREDIT_ACTIVE → Encodage manuel (**Sum** | **Count**)
- CREDIT_CURRENCY → Encodage manuel (**Sum** | **Count**)
- DAYS_CREDIT → **Min**
- CREDIT_DAY_OVERDUE → **Mean**
- DAYS_CREDIT_ENDDATE → **Mean**
- DAYS_ENDDATE_FACT → **Min**
- AMT_CREDIT_MAX_OVERDUE → **Mean**
- CNT_CREDIT_PROLONG → **Sum**
- AMT_CREDIT_SUM → **Mean & sum**
- AMT_CREDIT_SUM_DEBT → **Sum**
- AMT_CREDIT_SUM_LIMIT → **Mean**
- AMT_CREDIT_SUM_OVERDUE → **Mean**
- CREDIT_TYPE → Regroupement des prêts
- DAYS_CREDIT_UPDATE → **Mean**
- AMT_ANNUITY → **Mean**

Informations supplémentaires: Présentation du jeu de données

```
# Informations sur le jeu de données
previous_application.info(verbose=True, show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype
---  ------                       --------------    -----
 0   SK_ID_PREV                   1670214 non-null  int64
 1   SK_ID_CURR                   1670214 non-null  int64
 2   NAME_CONTRACT_TYPE           1670214 non-null  object
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object
 16  NAME_CONTRACT_STATUS         1670214 non-null  object
 17  DAYS_DECISION                1670214 non-null  int64
 18  NAME_PAYMENT_TYPE            1670214 non-null  object
 19  CODE_REJECT_REASON           1670214 non-null  object
 20  NAME_TYPE_SUITE              849809 non-null   object
 21  NAME_CLIENT_TYPE             1670214 non-null  object
 22  NAME_GOODS_CATEGORY          1670214 non-null  object
 23  NAME_PORTFOLIO               1670214 non-null  object
 24  NAME_PRODUCT_TYPE            1670214 non-null  object
 25  CHANNEL_TYPE                 1670214 non-null  object
 26  SELLERPLACE_AREA             1670214 non-null  int64
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object
 30  PRODUCT_COMBINATION          1669868 non-null  object
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB
```

- NAME_CONTRACT_TYPE → Encodage manuel (**Count**)
- AMT_ANNUITY → **Mean**
- AMT_CREDIT → **Mean & sum**
- AMT_DOWN_PAYMENT → **Mean**
- RATE_DOWN_PAYMENT → **Mean**
- NAME_CASH_LOAN_PURPOSE → Encodage manuel (**Count**)
- DAYS_DECISION → **Mean**
- NAME_TYPE_SUITE → Encodage manuel (**Count**)
- CNT_PAYMENT → **Mean**
- NFLAG_INSURED_ON_APPROVAL → **Most frequent**

Informations supplémentaires: Présentation du jeu de données

```
# Informations sur le jeu de données
installments.info(verbose=True, show_counts = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13605401 entries, 0 to 13605400
Data columns (total 8 columns):
 #   Column                  Non-Null Count     Dtype
---  ------                  --------------     -----
 0   SK_ID_PREV              13605401 non-null  int64
 1   SK_ID_CURR              13605401 non-null  int64
 2   NUM_INSTALMENT_VERSION  13605401 non-null  float64
 3   NUM_INSTALMENT_NUMBER   13605401 non-null  int64
 4   DAYS_INSTALMENT         13605401 non-null  float64
 5   DAYS_ENTRY_PAYMENT      13602496 non-null  float64
 6   AMT_INSTALMENT          13605401 non-null  float64
 7   AMT_PAYMENT             13602496 non-null  float64
dtypes: float64(5), int64(3)
memory usage: 830.4 MB
```

Mean

Mean

Mean

```
# Informations sur le jeu de données
pos_cash.info(verbose=True, show_counts = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10001358 entries, 0 to 10001357
Data columns (total 8 columns):
 #   Column                 Non-Null Count     Dtype
---  ------                 --------------     -----
 0   SK_ID_PREV             10001358 non-null  int64
 1   SK_ID_CURR             10001358 non-null  int64
 2   MONTHS_BALANCE         10001358 non-null  int64
 3   CNT_INSTALMENT         9975287 non-null   float64
 4   CNT_INSTALMENT_FUTURE  9975271 non-null   float64
 5   NAME_CONTRACT_STATUS   10001358 non-null  object
 6   SK_DPD                 10001358 non-null  int64
 7   SK_DPD_DEF             10001358 non-null  int64
dtypes: float64(2), int64(5), object(1)
memory usage: 610.4+ MB
```

Count

```
# Informations sur le jeu de données
credit_cards.info(verbose=True, show_counts = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3840312 entries, 0 to 3840311
Data columns (total 23 columns):
 #   Column                      Non-Null Count    Dtype
---  ------                      --------------    -----
 0   SK_ID_PREV                  3840312 non-null  int64
 1   SK_ID_CURR                  3840312 non-null  int64
 2   MONTHS_BALANCE              3840312 non-null  int64
 3   AMT_BALANCE                 3840312 non-null  float64
 4   AMT_CREDIT_LIMIT_ACTUAL     3840312 non-null  int64
 5   AMT_DRAWINGS_ATM_CURRENT    3090496 non-null  float64
 6   AMT_DRAWINGS_CURRENT        3840312 non-null  float64
 7   AMT_DRAWINGS_OTHER_CURRENT  3090496 non-null  float64
 8   AMT_DRAWINGS_POS_CURRENT    3090496 non-null  float64
 9   AMT_INST_MIN_REGULARITY     3535076 non-null  float64
 10  AMT_PAYMENT_CURRENT         3072324 non-null  float64
 11  AMT_PAYMENT_TOTAL_CURRENT   3840312 non-null  float64
 12  AMT_RECEIVABLE_PRINCIPAL    3840312 non-null  float64
 13  AMT_RECIVABLE               3840312 non-null  float64
 14  AMT_TOTAL_RECEIVABLE        3840312 non-null  float64
 15  CNT_DRAWINGS_ATM_CURRENT    3090496 non-null  float64
 16  CNT_DRAWINGS_CURRENT        3840312 non-null  int64
 17  CNT_DRAWINGS_OTHER_CURRENT  3090496 non-null  float64
 18  CNT_DRAWINGS_POS_CURRENT    3090496 non-null  float64
 19  CNT_INSTALMENT_MATURE_CUM   3535076 non-null  float64
 20  NAME_CONTRACT_STATUS        3840312 non-null  object
 21  SK_DPD                      3840312 non-null  int64
 22  SK_DPD_DEF                  3840312 non-null  int64
dtypes: float64(15), int64(7), object(1)
memory usage: 673.9+ MB
```

Informations supplémentaires:
Résultats obtenus après la première optimisation

	DummyClassifier	LogisticRegression	CatBoostClassifier	LGBMClassifier	XGBClassifier
Training Time	0.0082	23.9292	8.6766	1.3275	57.8648
Train Accuracy	91.9271	82.2002	82.1933	80.5299	85.4415
Test Accuracy	91.9272	82.5196	81.1115	79.9294	81.131
Train AUC	0.5	0.7048	0.8172	0.805	0.9247
Test AUC	0.5	0.7073	0.7669	0.7679	0.7672
Train Recall	0.0	0.3745	0.6056	0.6066	0.8255
Test Recall	0.0	0.3738	0.5321	0.5519	0.5299
Train Precision	0.0	0.1917	0.2506	0.2311	0.3363
Test Precision	0.0	0.1954	0.2213	0.2131	0.2211
Train f1	0.0	0.2536	0.3545	0.3347	0.4779
Test f1	0.0	0.2567	0.3126	0.3075	0.312
Best threshold	0.0808	0.2222	0.2121	0.202	0.202
Train best score	0.4676	0.4348	0.3611	0.3737	0.2417
Test best score	0.4676	0.4328	0.3947	0.397	0.3952

Informations supplémentaires:
La stabilité des variables

93	88	0	5	0
Tests	Success	Warning	Fail	Error

All tests

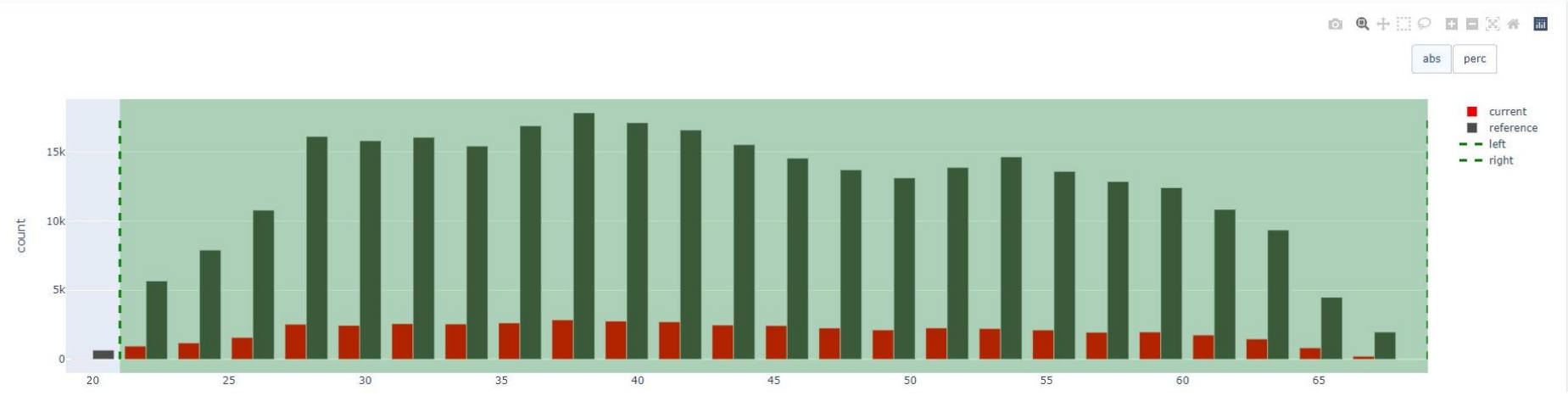
Number of Rows

The number of rows is 48744. The test threshold is eq=3.08e+05 ± 3.08e+04.

Share of Out-of-Range Values

The share of values out of range in the column **AGE** is 4.1e-05 (2 out of 48744). The test threshold is eq=0 ± 1e-12.

DETAILS



Share of Out-of-Range Values

The share of values out of range in the column **ANNUITY_INCOME_PERC** is 2.05e-05 (1 out of 48744). The test threshold is eq=0 ± 1e-12.

DETAILS

Share of Out-of-Range Values

The share of values out of range in the column **REGION_POPULATION_RELATIVE** is 2.05e-05 (1 out of 48744). The test threshold is eq=0 ± 1e-12.

DETAILS

Share of Out-of-Range Values

The share of values out of range in the column **prev_type_loans** is 2.05e-05 (1 out of 48744). The test threshold is eq=0 ± 1e-12.

DETAILS