Prêt à dépenser

Implémentez un modèle de scoring

Guillaume Rooman

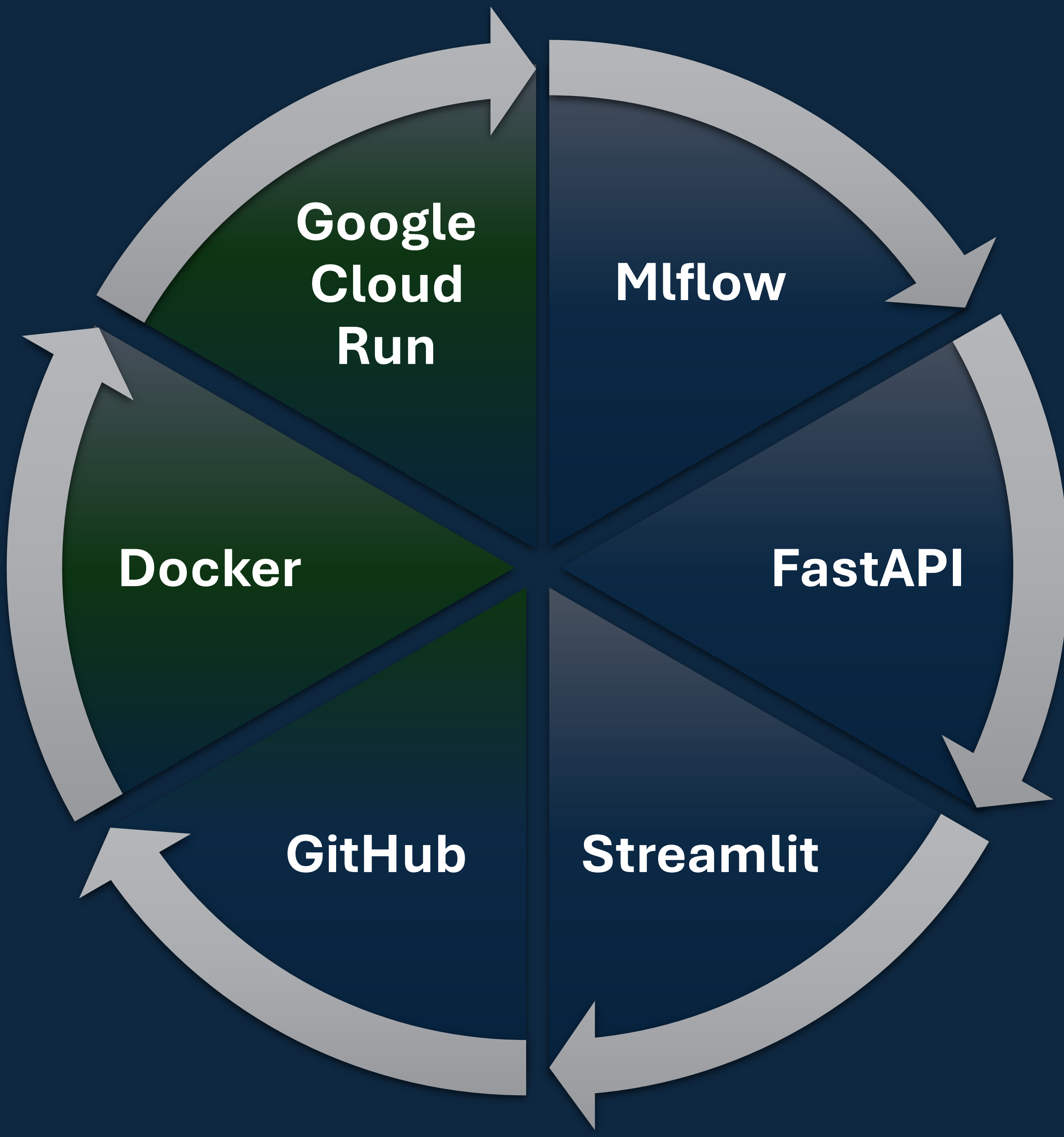
Data Scientist chez Prêt à dépenser

Sommaire

- Intro
- Stack
- Modèle : construction
- Modèle : analyse
- API
- Déploiement
- Pour aller plus loin

Introduction

Stack overview



Data

8 fichiers

- application_test.parquet
- application_train.parquet
- bureau.parquet
- bureau_balance.parquet
- credit_card_balance.parquet
- installments_payments.parquet
- POS_CASH_balance.parquet
- previous_application.parquet

Group by	Aggreg
SK_ID	min
	max
	mean
	median
	sum

```
df_global.info()
```

```
&lt;class 'pandas.core.frame.DataFrame'&gt;
Index: 356251 entries, 0 to 356254
Columns: 896 entries, index to CC_COUNT
dtypes: bool(133), float64(704), int64(43), object(16)
memory usage: 2.1+ GB
```

```
df_global["TARGET"].value_counts(normalize=True)
```

```
TARGET
0.0    0.91927
1.0    0.08073
Name: proportion, dtype: float64
```

https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features

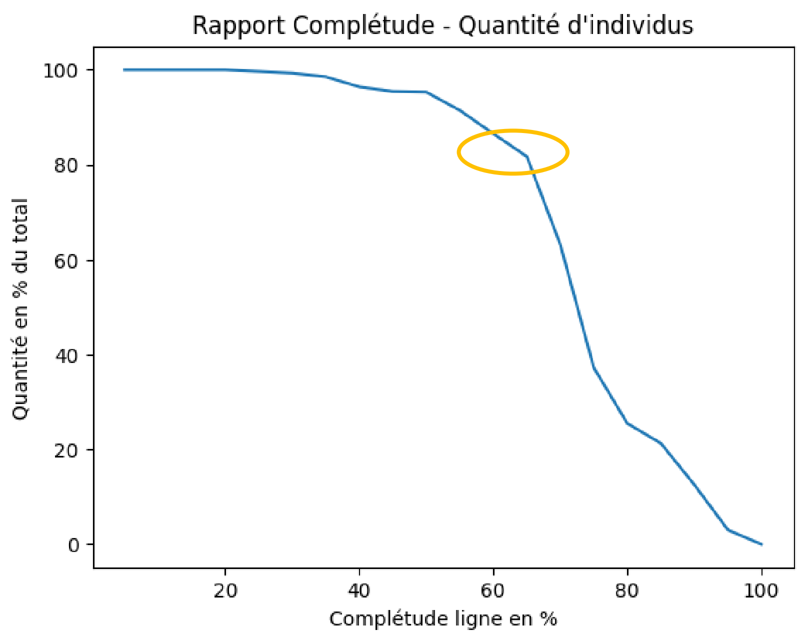
Construction modèle : overview

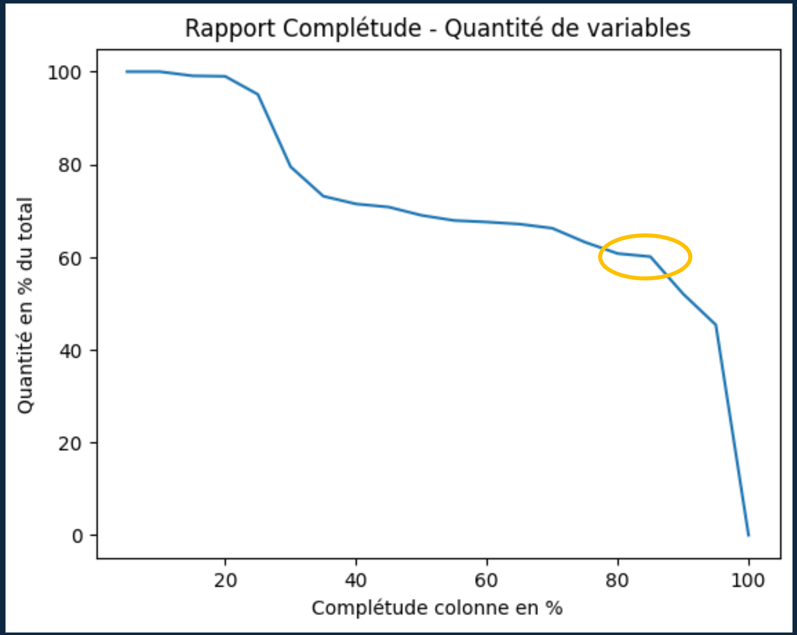
Expérience	Métrique	Réglages paramètres
Preprocess	ROC AUC	4
Process	ROC AUC	3 + Input
Algorithmes	ROC AUC + Custom	0
Tuning	Custom	2

```
{
  "run_name": "median_80_c
  "input_parquet": "C:\\Us
  "output_parquet": "C:\\U
  "completeness": 80,
  "impute": "median",
  "percent_outliers": 1,
  "target_col": "TARGET",
  "cv_threshold": 0.01
},
```

```
{
  "run_name": "Select40_
  "input_parquet": "C:\\
  "output_dir": "C:\\Use
  "n_select": 40,
  "cor_val": 0.7,
  "scaler": "robust",
  "target_col": "TARGET"
  "cache_dir": "C:\\User
},
```

Preprocessing

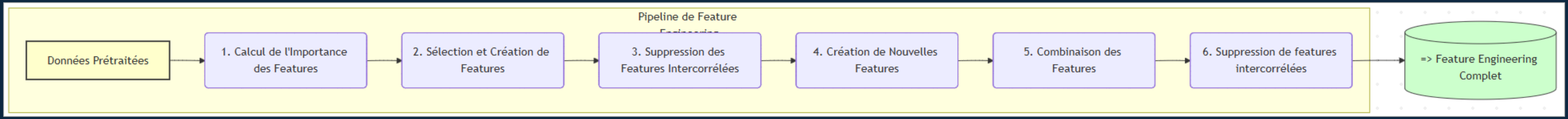




				Metrics		Parameters			
Run Name	Created	Duration	Source	final_feature_count	roc_auc	completeness	cv_threshold	impute	percent_outliers
median_85_comp_0p5_ou...	3 days ago	39.5s	preproce...	508	0.747471868...	85	0.05	median	0.5
median_90_comp_0p2_ou...	3 days ago	35.5s	preproce...	438	0.745010426...	90	0.1	median	0.2
					0.738538930...	80	0.01	median	1
					0.734446951...	80	0.01	median	1
					0.732860486...	90	0.1	median	0.2
					0.728437784...	85	0.05	median	0.5

```python
# Drop rows based on completeness
row_completeness = (1 - df_processed.isnull().sum(axis=1) / df_processed.shape[1]) * 100
rows_to_drop_completeness = df_processed[row_completeness < completeness*0.5].index.tolist()
if rows_to_drop_completeness:
    df_processed.drop(index=rows_to_drop_completeness, inplace=True)
    if verbose:
        print(f"Dropped {len(rows_to_drop_completeness)} rows due to completeness < {completeness*0.5}%")
```

Processing




```python
class FeatureEngineeringPipeline:
    def __init__(self, n_select: int = 50, cor_val: float = 0.7,
        self.n_select = n_select
        self.n_create = max(2, int(np.sqrt(n_select)))
        self.cor_val = cor_val
        self.target_col = target_col
```

```python
importance_df = pd.DataFrame({
    'feature': X.columns,
    'metric1_spearman': spearman_corr,
    'metric2_mdi': rfc.feature_importances_,
    'metric3_product': spearman_corr * rfc.feature_importances_
}).sort_values(by='metric3_product', ascending=False).reset_index(drop=True)
```

					Metrics		Parameters		
	Run Name	Created	Duration	Source	final_feature_cou	roc_auc_test	cor_val	n_select	scaler
	Select40_Cor50_Standard	4 days ago	17.8s	c:\Users\...	26	0.708232735...	0.5	40	standard
	Select45_Cor55_Standard	4 days ago	18.5s	c:\Users\...	27	0.710009252...	0.55	45	standard
	Select50_Cor60_Standard	4 days ago	20.9s	c:\Users\...	31	0.713816272...	0.6	50	standard
	Select40_Cor70_Robust	3 days ago	39.8s	process.py	36	0.717219370...	0.7	40	robust
	Select60_Cor65_Robust	3 days ago	45.7s	process.py	41	0.718538204...	0.65	60	robust
	Select80_Cor60_Robust	3 days ago	50.9s	process.py	42	0.723408775...	0.6	80	robust

Choix d'algorithme

				Metrics			
	Run Name	Cre	Duration	test_accuracy	test_auc	test_normalized_custom_score	test_recall_at_best_t
	Group: gradient_boosting 6	-		0.71604155...	0.71883838...	2.2510152586876084 (aver...	0.53755443134...
	processed_s50_c60_std__gradient_...	(	1.7min	0.873790117...	0.718211127...	4.024705556114052	0.261524822695...
	processed_s80_c60_robust__gradie...	(	3.6min	0.865385983...	0.728525825...	3.658911360851025	0.299623706491...
	processed_s50_c60_std__gradient_...	(	32.5s	0.747007131...	0.718607800...	3.2690755134059932	0.539007092198...
	processed_s80_c60_robust__gradie...	(	1.1min	0.712202063...	0.722414911...	2.632459240848847	0.602069614299...
	processed_s50_c60_std__gradient_...	(	18.4s	0.572784004...	0.707212100...	0.4026479335557029	0.737588652482...
	processed_s80_c60_robust__gradie...	(	36.0s	0.525080042...	0.718058551...	-0.48170805264996774	0.785512699905...
	Group: xgboost 6	-		0.71181795...	0.70121505...	1.9909163028322625 (aver...	0.52374129598...
	Group: catboost 6	-		0.68384492...	0.69879140...	1.5668875939094533 (aver...	0.55663486964...
	Group: random_forest 6	-		0.56014231...	0.71745283...	-0.45134163366474445 (av...	0.69030913557...
	processed_s50_c60_std__random_f...	(	5.3s	0.833418237...	0.713818401...	3.973265952267729	0.368794326241...

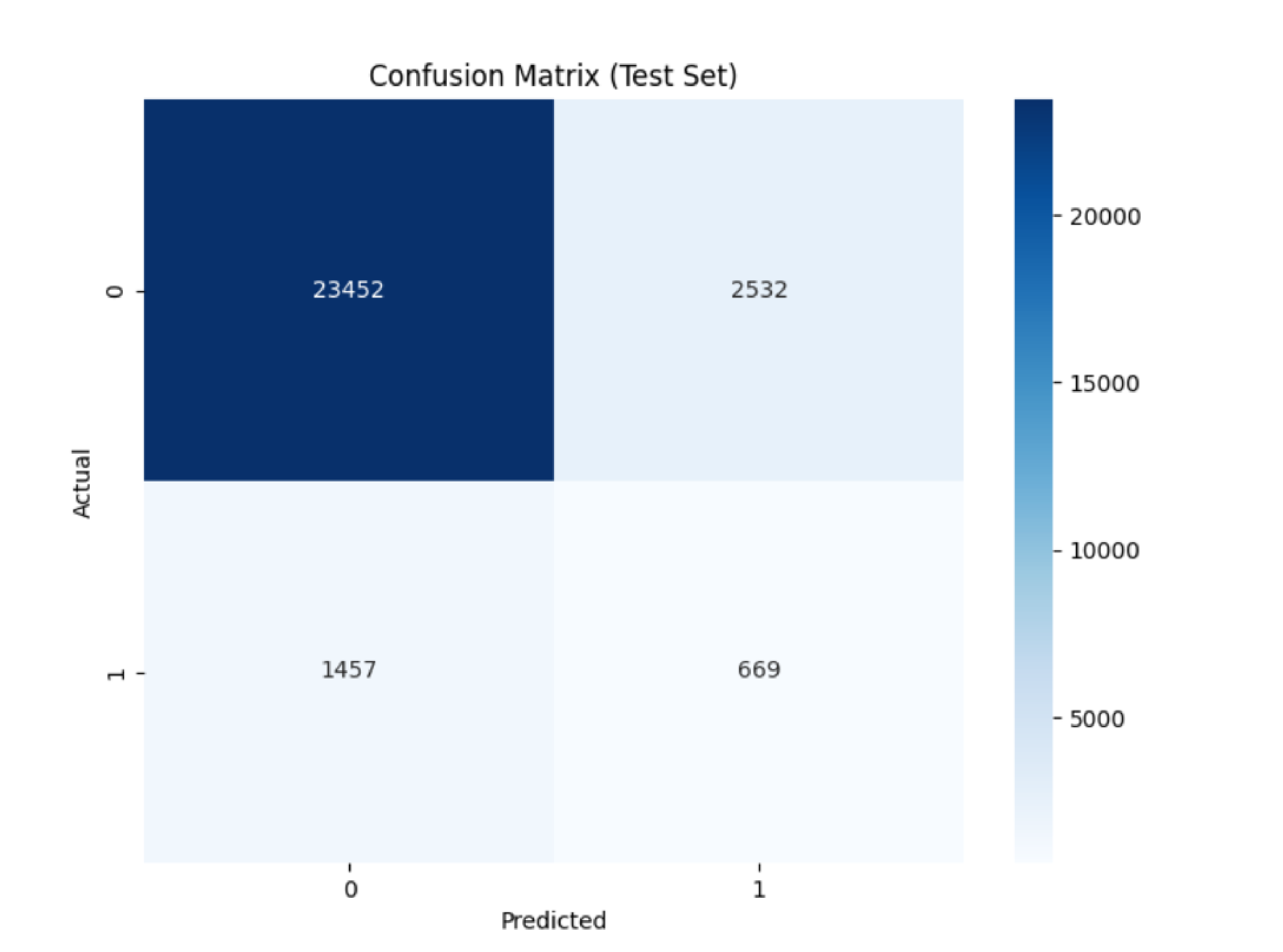
30 matching runs

Optimisation d’hyperparamètres

```python
def _compute_custom_and_normalized(y_true: np.ndarray, y_pred_bin: np.ndarray,
                                   pos_proportion: float) -> Tuple[float, float]:
    """Compute custom and normalized scores based on confusion matrix."""
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred_bin).ravel()
    custom = (2 * tp) + (1 * tn) - (1 * fp) - (10 * fn)
    n = len(y_true)
    pos_prop = max(pos_proportion, 1e-9)
    normalized = custom * (1.0 / max(1, n)) * (1.0 / pos_prop)
    return float(custom), float(normalized)


def _find_best_threshold_custom_score(y_true: np.ndarray, y_pred_proba: np.ndarray,
                                      pos_proportion: float) -> Tuple[float, float]:
    """Find the best threshold that maximizes the custom normalized score."""
    thresholds = np.linspace(0.01, 0.99, 99)
    best_t, best_score = 0.5, -np.inf

    for t in thresholds:
        y_pred_bin = (y_pred_proba >= t).astype(int)
        _, norm_score = _compute_custom_and_normalized(y_true, y_pred_bin, pos_proportion)
        if norm_score > best_score:
            best_score = norm_score
            best_t = t

    return best_t, best_score
```

Y[target].mean()

Analyse modèle : pourquoi ça marche ?



Soit +v la valeur que rapporte un prêt :

Sur 28 810 clients

+ 23 452 valeurs en prêts gagnants

– 14 570 valeurs en prêts perdants

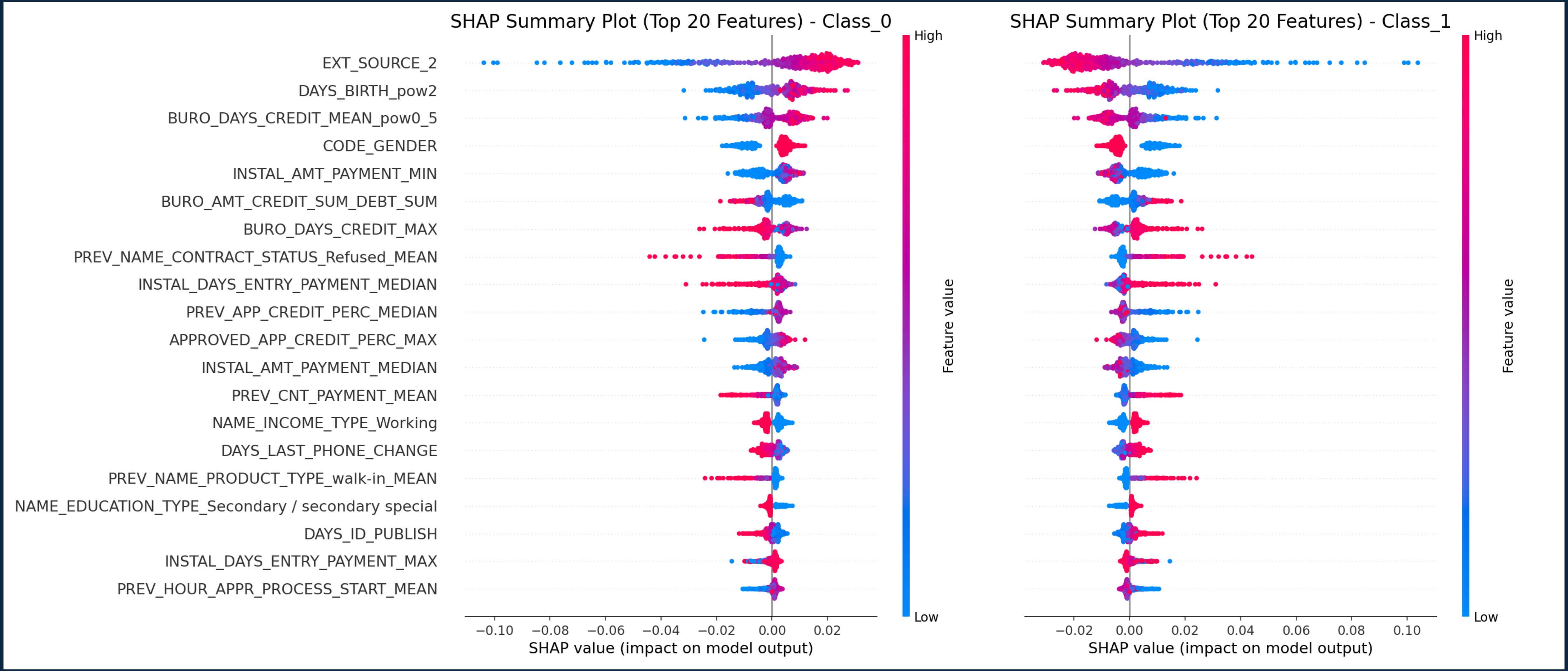
= 8882 valeurs de prêt

----

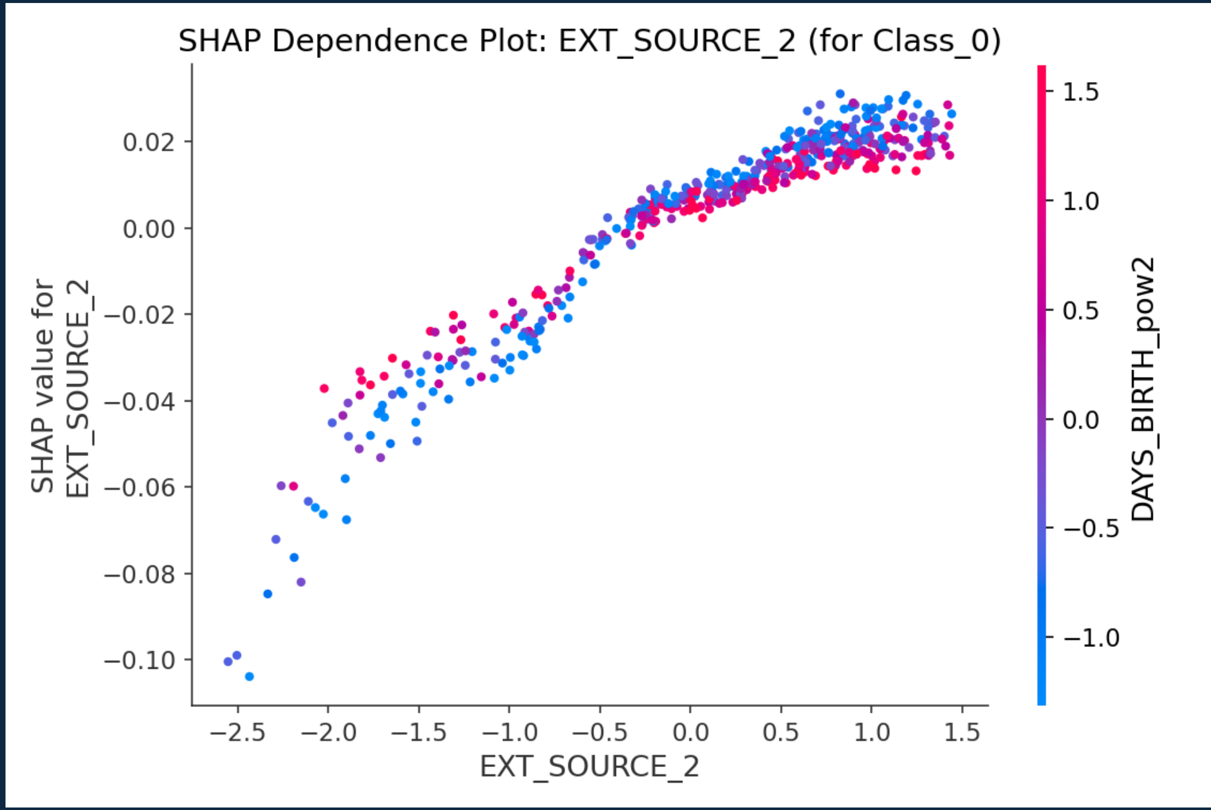
1 demande = 0.31 valeur

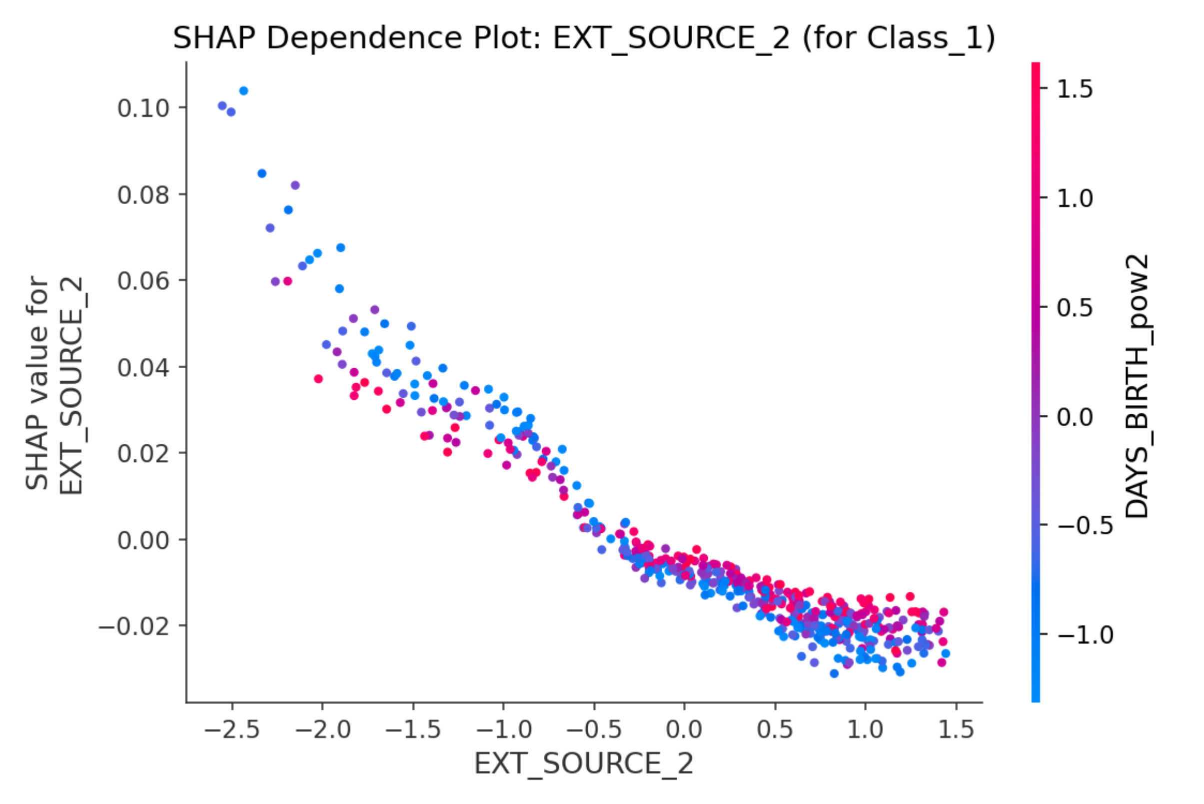
81.4% de prêt accordés

Analyse modèle



Analyse modèle

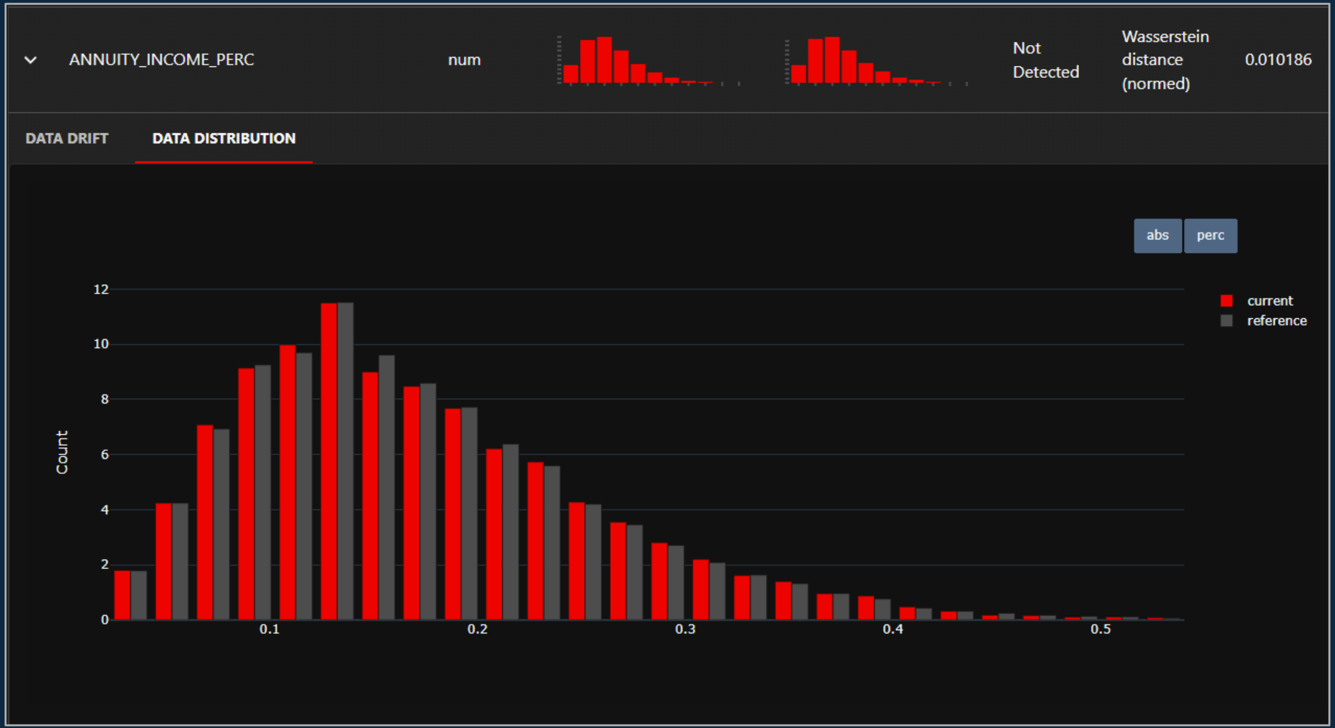




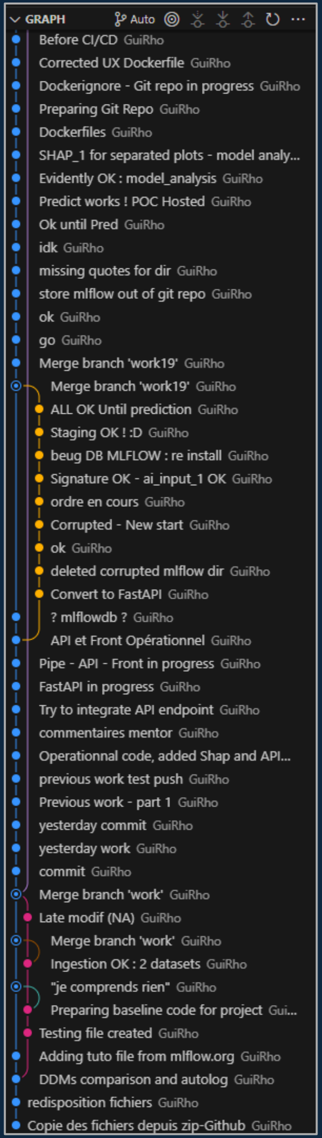
Dataset Drift

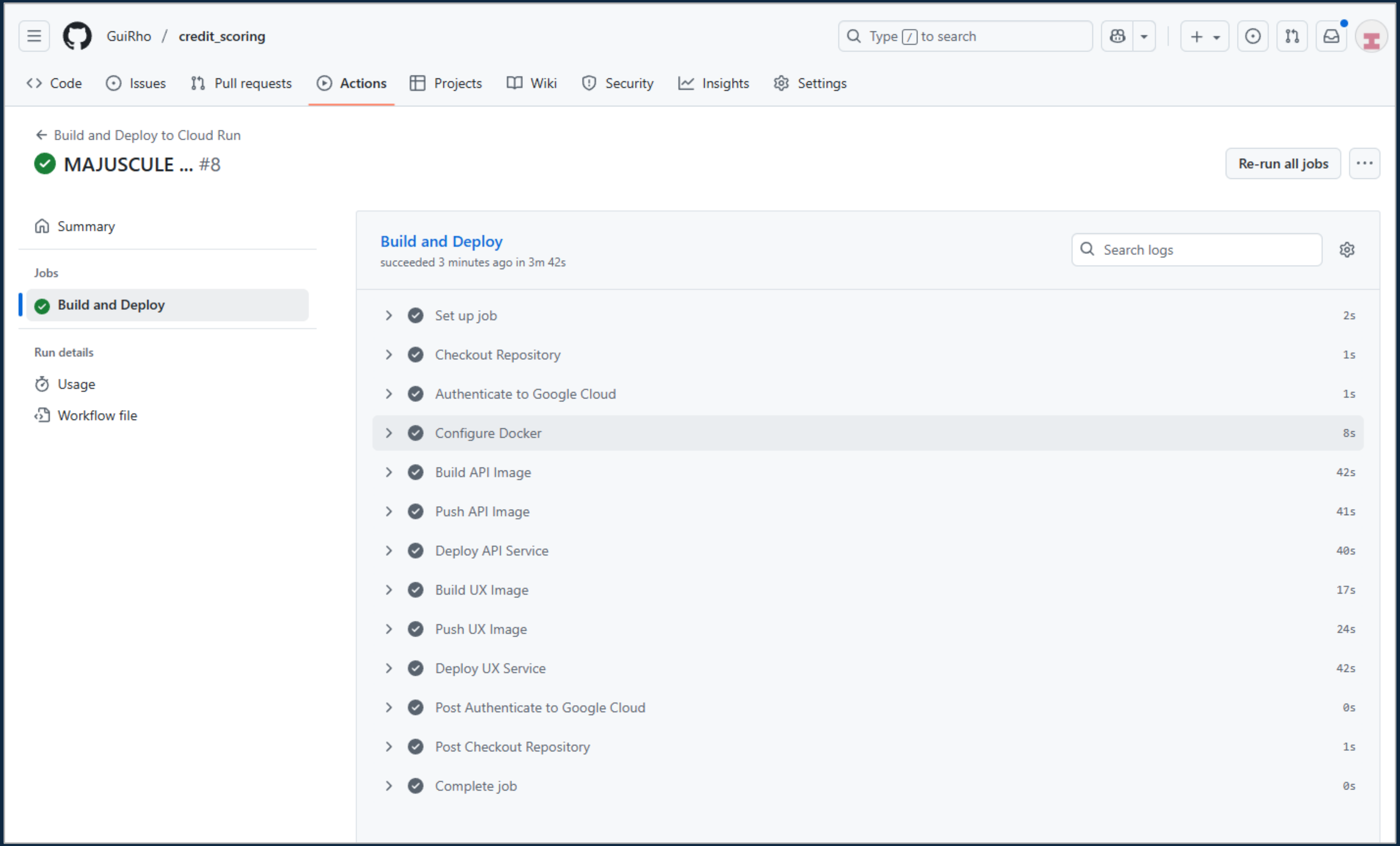
Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

43	0	0.0
Columns	Drifted Columns	Share of Drifted Columns

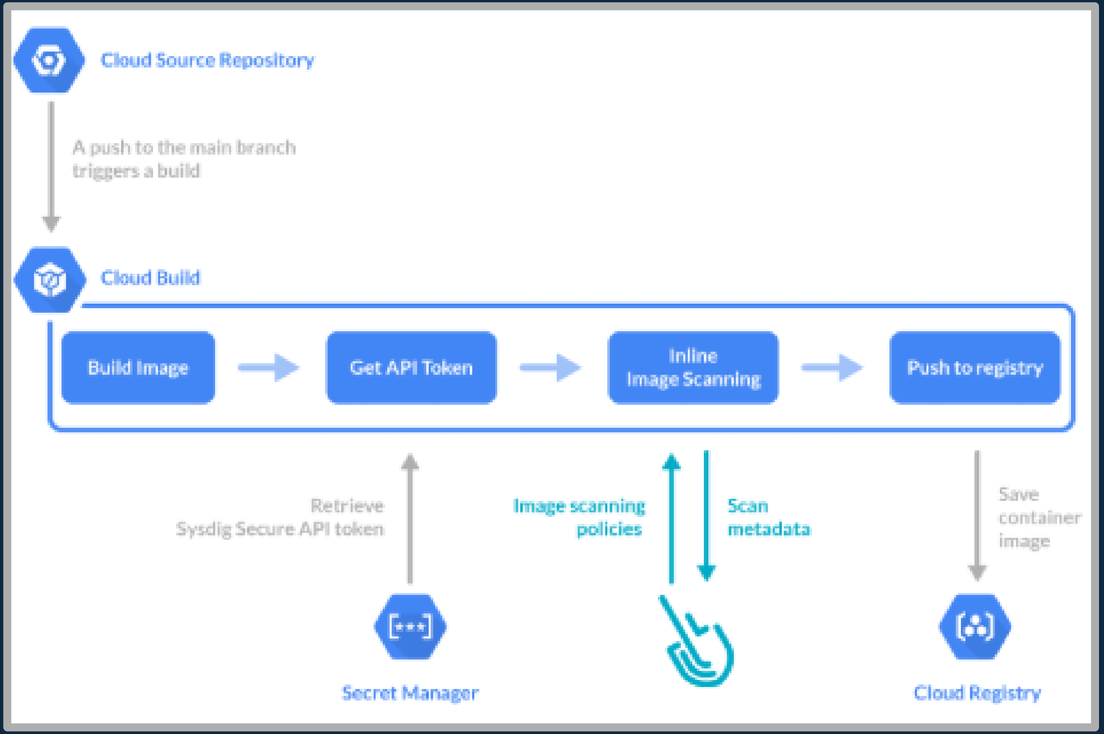


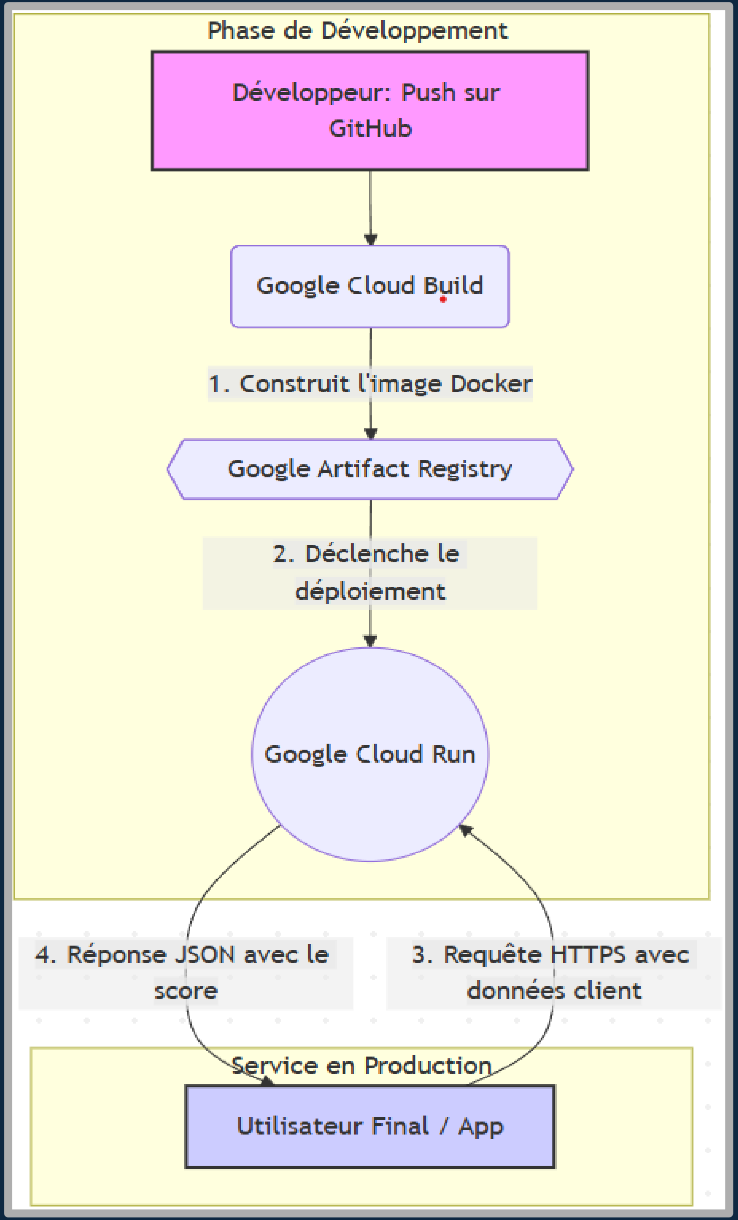
CI/CD


GRAPH
Auto
Before CI/CD GuiRho
Corrected UX Dockerfile GuiRho
Dockerignore - Git repo in progress GuiRho
Preparing Git Repo GuiRho
Dockerfiles GuiRho
SHAP_1 for separated plots - model analy...
Evidently OK : model_analysis GuiRho
Predict works ! POC Hosted GuiRho
Ok until Pred GuiRho
idk GuiRho
missing quotes for dir GuiRho
store mlflow out of git repo GuiRho
ok GuiRho
go GuiRho
Merge branch 'work19' GuiRho
Merge branch 'work19' GuiRho
ALL OK Until prediction GuiRho
Staging OK ! :D GuiRho
beug DB MLFLOW : re install GuiRho
Signature OK - ai_input_1 OK GuiRho
ordre en cours GuiRho
Corrupted - New start GuiRho
ok GuiRho
deleted corrupted mlflow dir GuiRho
Convert to FastAPI GuiRho
? mlflowdb ? GuiRho
API et Front Opérationnel GuiRho
Pipe - API - Front in progress GuiRho
FastAPI in progress GuiRho
Try to integrate API endpoint GuiRho
commentaires mentor GuiRho
Operationnal code, added Shap and API...
previous work test push GuiRho
Previous work - part 1 GuiRho
yesterday commit GuiRho
yesterday work GuiRho
commit GuiRho
Merge branch 'work' GuiRho
Late modif (NA) GuiRho
Merge branch 'work' GuiRho
Ingestion OK : 2 datasets GuiRho
"je comprends rien" GuiRho
Preparing baseline code for project Gui...
Testing file created GuiRho
Adding tuto file from mlflow.org GuiRho
DDMs comparison and autolog GuiRho
redisposition fichiers GuiRho
Copie des fichiers depuis zip-Github GuiRho


GuiRho / credit_scoring
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Build and Deploy to Cloud Run
MAJUSCULE ... #8
Re-run all jobs
Summary
Jobs
Build and Deploy
Run details
Usage
Workflow file
Build and Deploy
succeeded 3 minutes ago in 3m 42s
Search logs
Set up job 2s
Checkout Repository 1s
Authenticate to Google Cloud 1s
Configure Docker 8s
Build API Image 42s
Push API Image 41s
Deploy API Service 40s
Build UX Image 17s
Push UX Image 24s
Deploy UX Service 42s
Post Authenticate to Google Cloud 0s
Post Checkout Repository 1s
Complete job 0s

Google cloud



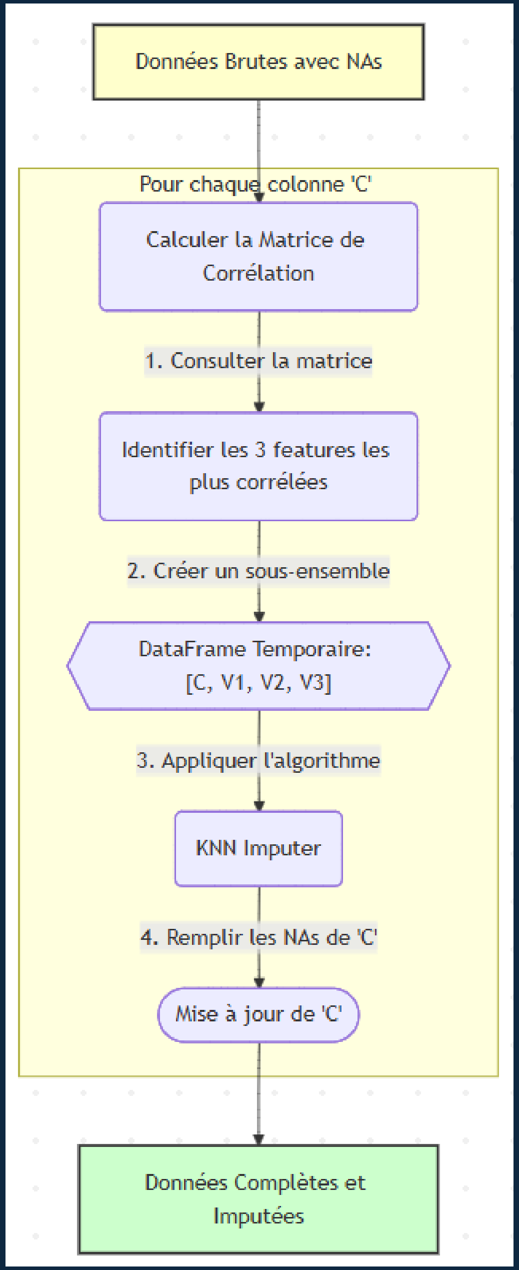


API : demo

URL : https://credit-scoring-ux-257261936398.europe-west1.run.app

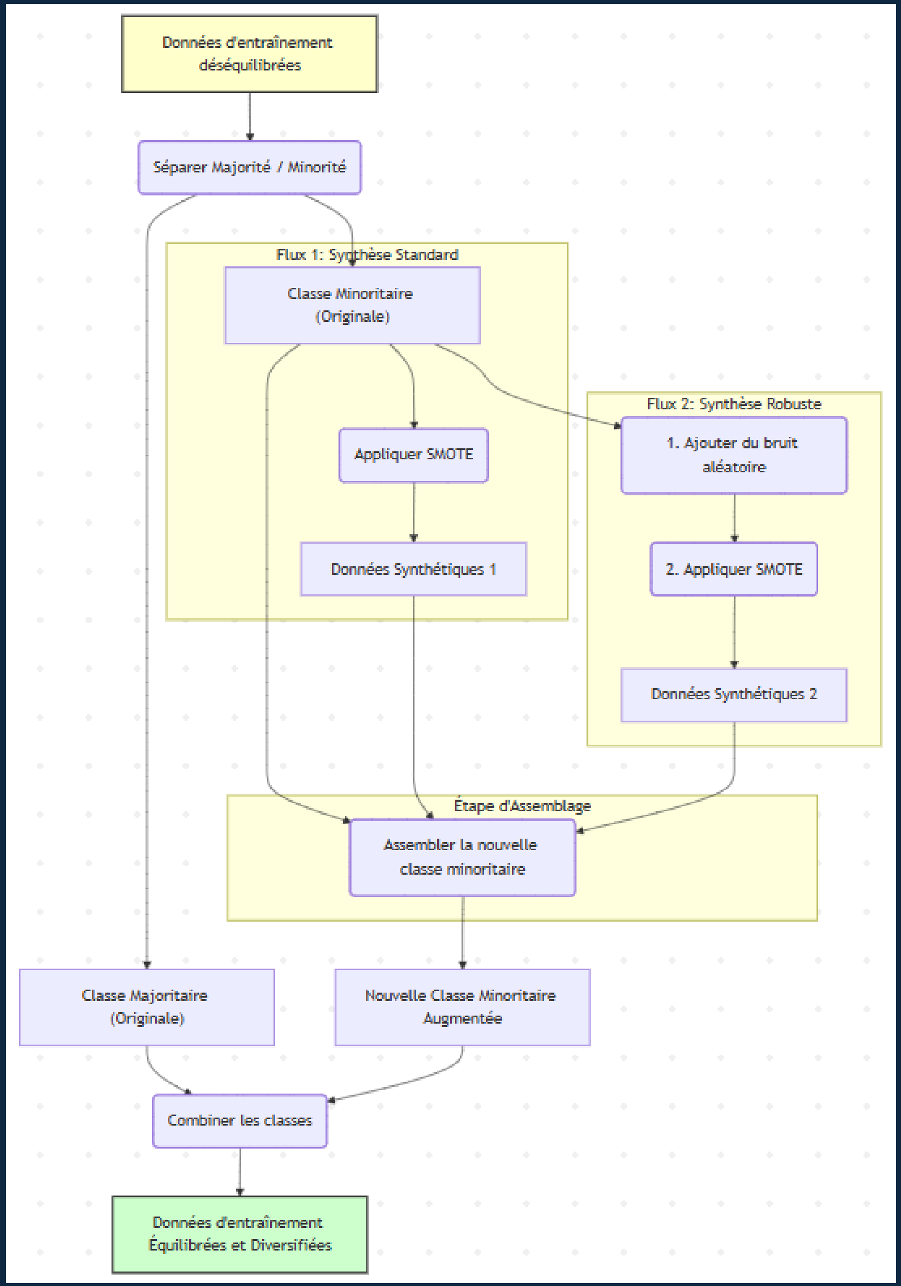
KNN

Pour aller plus loin

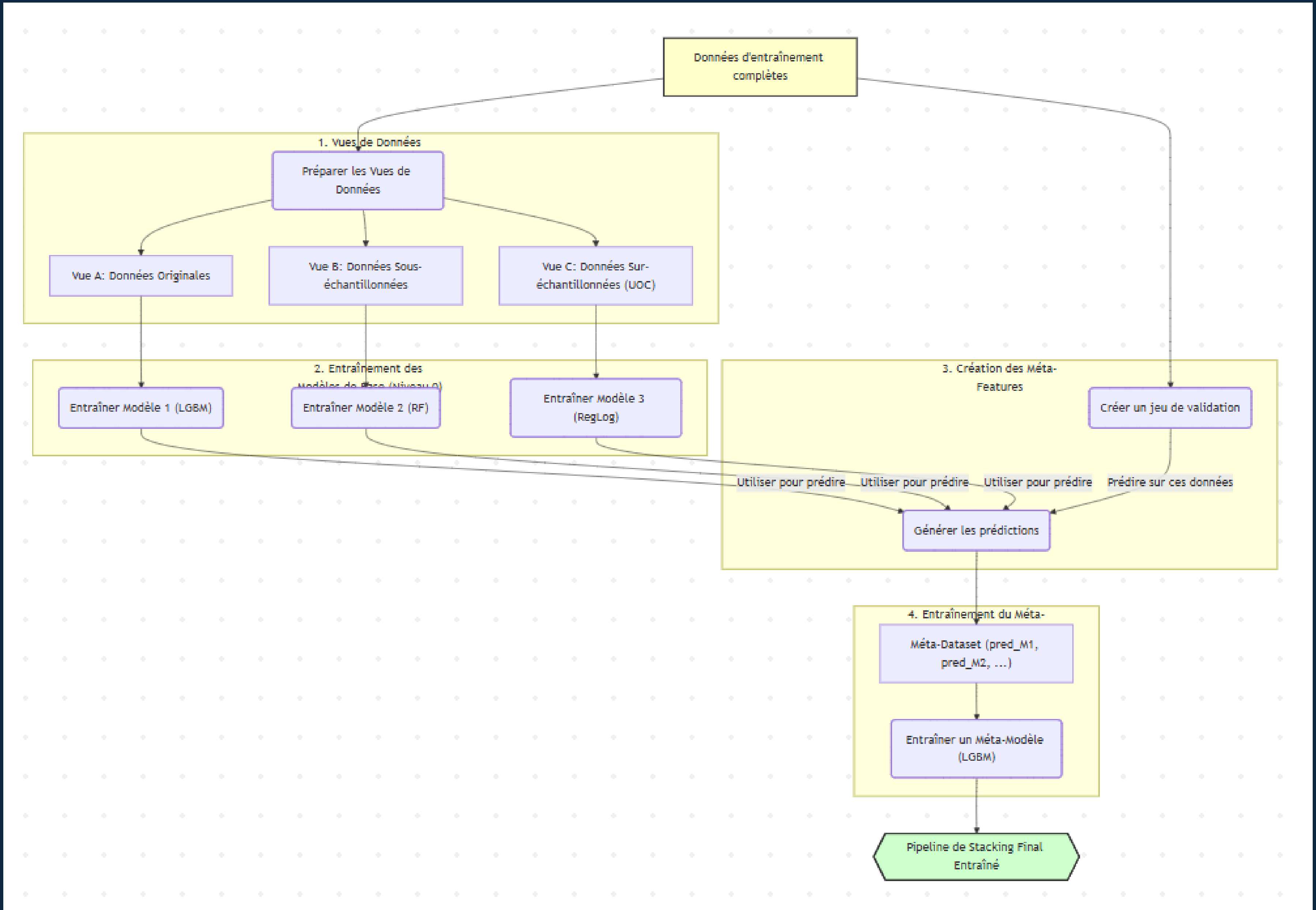


- DL model
- Imputation KNN ciblée
- Algorithme de rééquilibrage hybride
- Stacking model : données à version

Imbalance



Stacking model



Conclusion