

Prêt **à**
dépendre



Credit Scoring

[GitHub Repository](#)

Context & Mission



→ **"Prêt à Dépenser"** is a financial company offering consumer credit to individuals with minimal or no loan history. The company aims to broaden the financial inclusion for the unbanked population.

With this intent in mind, financial, statutory & behavioral data were acquired to bridge the lack of loan history to:

- **Build a classifier model** which accepts or reject a credit application
- Make the model inference available through an **API**
- **(Future Work) provide a Dashboard** which:
 - Display the model decision
 - Assist customer relation managers in transparency by explaining the main factors that influenced the model decision
 - Facilitates navigation through customers's data

Constraints



Embrace a MLOps approach

- **ML Flow** → Track experiments, store models in a registry, and test ML serving.
- **Git & Github** → Manage and share code, integrate continuously.
- **CI/CD with GitHub Actions** → Automate API deployment in the cloud.
- **Testing** → Design and automate unit tests using PyTest
- **Logging** → Through the Loguru library

Deploy the API

Test evidently → As a tool to detect datadrift when the model is in production

Document the steps explored to build the classifier

Plan



Prêt à
dépendre



Modeling Approach & Results

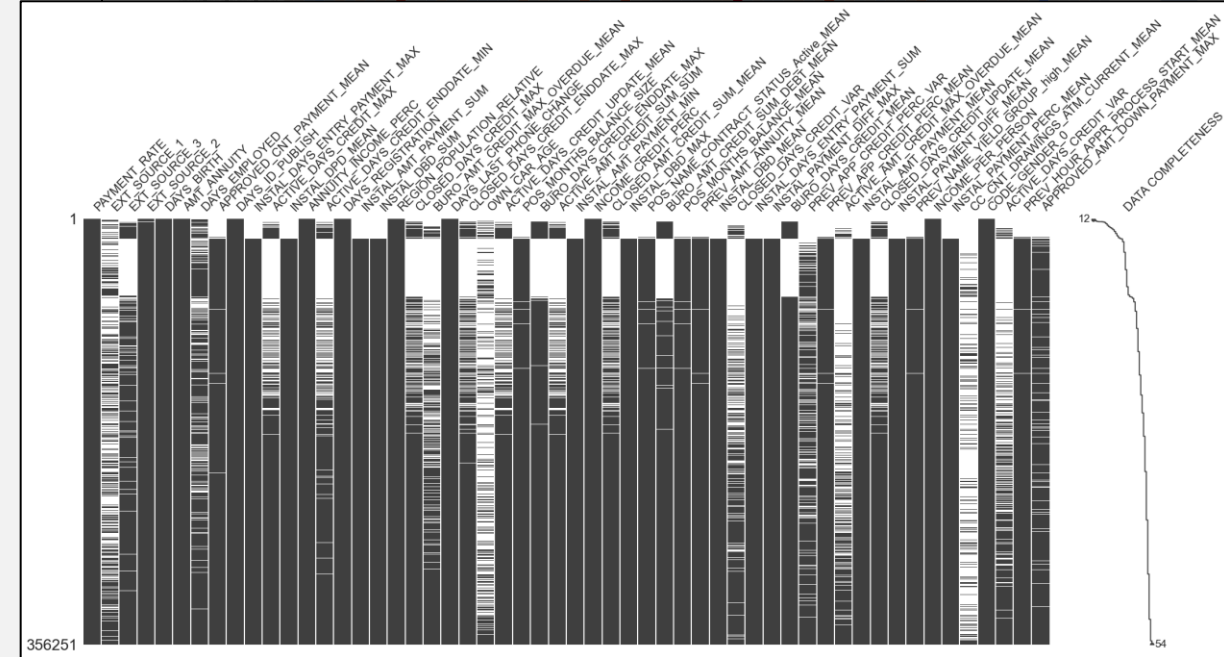
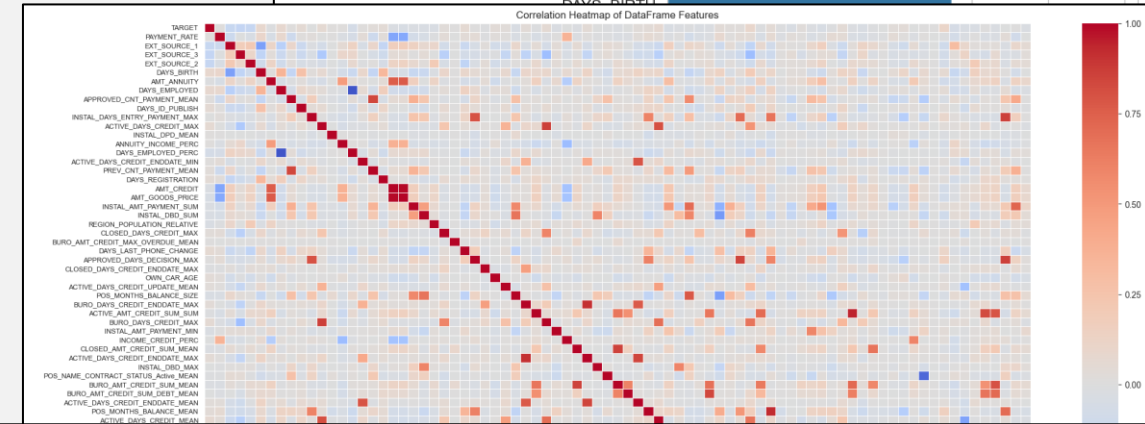
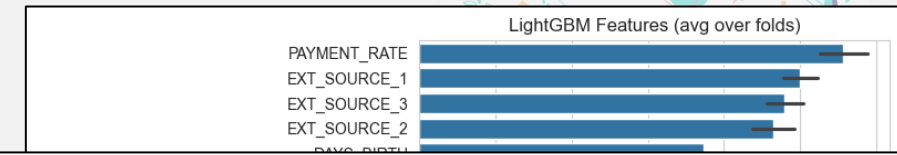
Feature engineering + Data Preparation

Feature Engineering → [LightGBM with Simple Features \(Kaggle\)](#)

- One hot categorical, Numerical Aggregation/Group By
- (356 000 individuals, 800 features)
- Retained the 80 most important features
- Highly correlated pairs deleted through lowest importance
- Missing Values Imputation → [Facebook AI Similarity Search \(FAISS\) KNN](#) + median because of outliers

→ (356 000 individuals, 50 features)

- Only numerical variables, no categorical ones
- Skewed Distributions
- High Range between Min/Max values
- High count of outliers

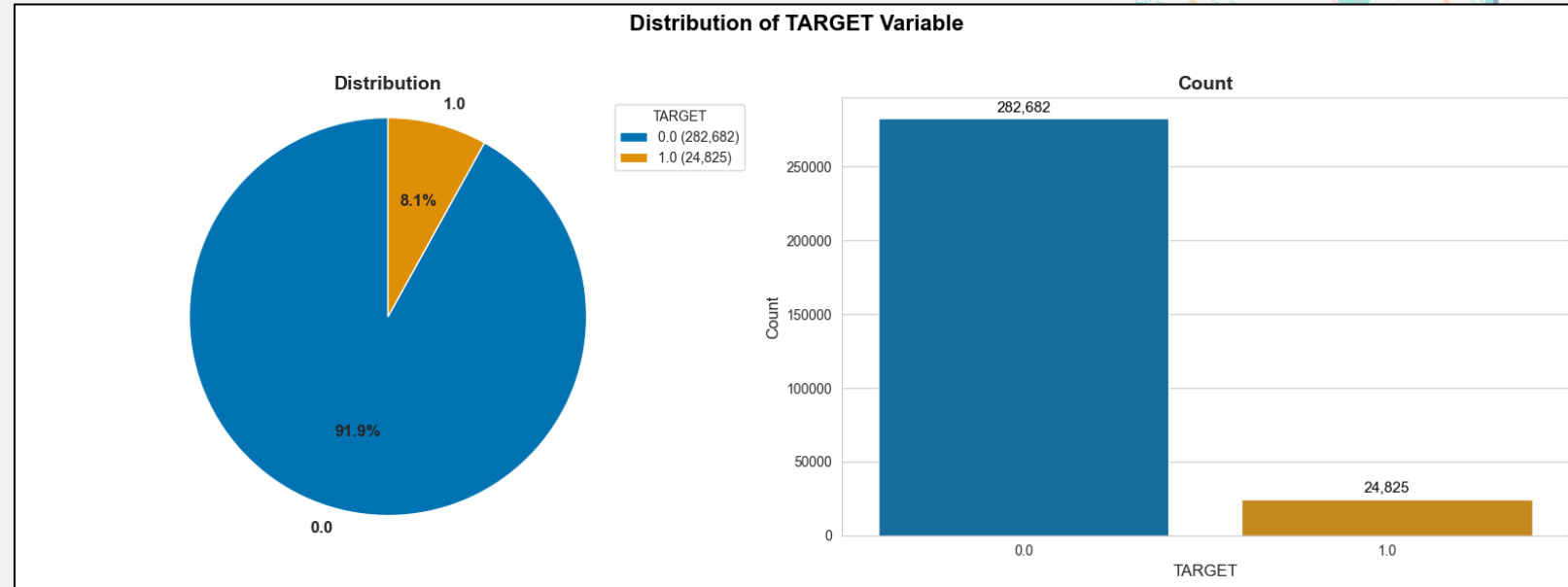


Target Distribution

Imbalance ratio → 11.38

Problem → Classifiers tend to predict the dominant class

3 Techniques used to address this issue



- **Weight Class Errors** → Increase the impact of errors made on the minority class when evaluating & decrease the ones made on the majority class
- **Resampling through SMOTETomek** → Combine over-sampling on the minority class using SMOTE & under-sampling on the majority class using Tomek links to get closer to class balance
- **Threshold optimization** → Finding the best threshold value to decide the cutoff class from probability prediction which default to 0,5.

Model Evaluation & Business Considerations

False Negative (FN) → Real loss of money, a client wrongly seen as capable of paying

False Positive (FP) → Hypothetical loss of money, a client capable of paying but seen as not capable

Assumption → "The cost incurred by a FN is ten times that of the cost incurred by a FP"

$$\text{Business Cost} = (FN \times 10) + FP$$

For the finetuning → We adjusted the cost function to dynamically increase false negative penalties when recall is low, ensuring accurate identification of default risks

$$\text{Business Cost} = \left(FN \times 10 \times \frac{1}{\text{Recall}} \right) \times FP$$

$$\text{Profit} = (6 \times TN - 4 \times FP) - (40 \times FN + 0 \times TP)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{FN Rate} = 1 - \text{Recall}$$



		Predicted	
		NEGATIVE	POSITIVE
Actual	NEGATIVE	Count of TN	Count of FP
	POSITIVE	Count of FN	Count of TP

Finding the Best Model

Tested Models

- Dummy Classifier (Baseline)
- Logistic Regression (solver: saga/sag & Penalty :Lasso, Ridge, ElasticNet)
- RandomForest Classifier
- XGBoost Classifier
- LightGBM Classifier

Best Hyperparameters Search with Optuna

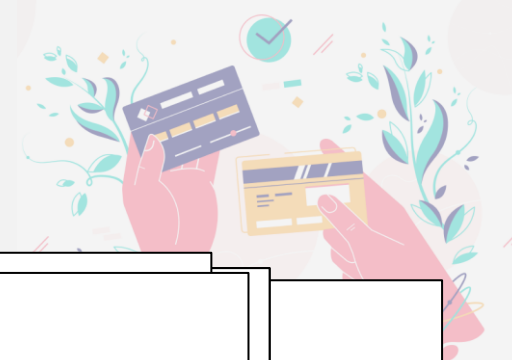
- Sampled 20% of the original dataset = 61 501 individuals
- Preprocessing Pipeline : RobustScaler + SMOTETomek ("Sampling Strategy" : [0.15, 0.25, 0.30, 0.35])
- Evaluation: StratifiedKFold (5 splits) + Cross Validation Score
- Scorer: Maximized (ROC AUC; - Business Cost)
- Sampler: TPE (Tree-structured Parzen Estimator) algorithm
- Trials: 15 & 45 for the Best Model

ML Flow → Recorded each run, save plots as artifacts and register the fitted model inside the Model Registry

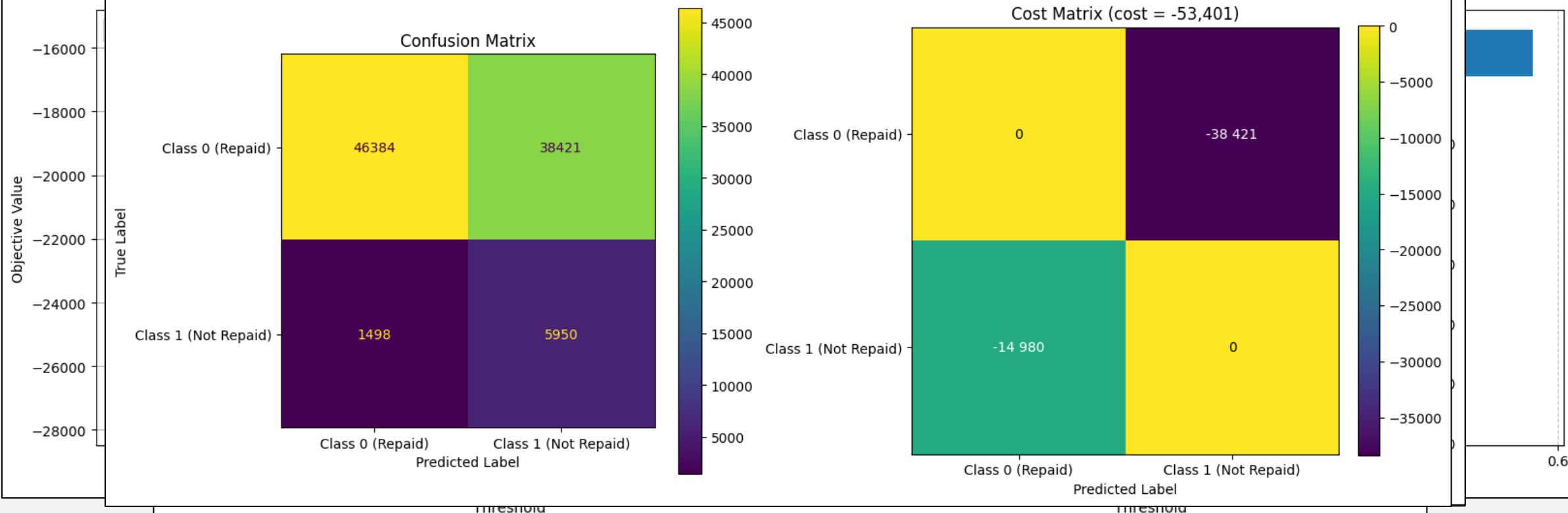
Metrics → Precision/Recall, FNR, F2 Beta Score (Beta =2), ROC AUC, FN, FP, Business Cost, Profit, Business Cost Std



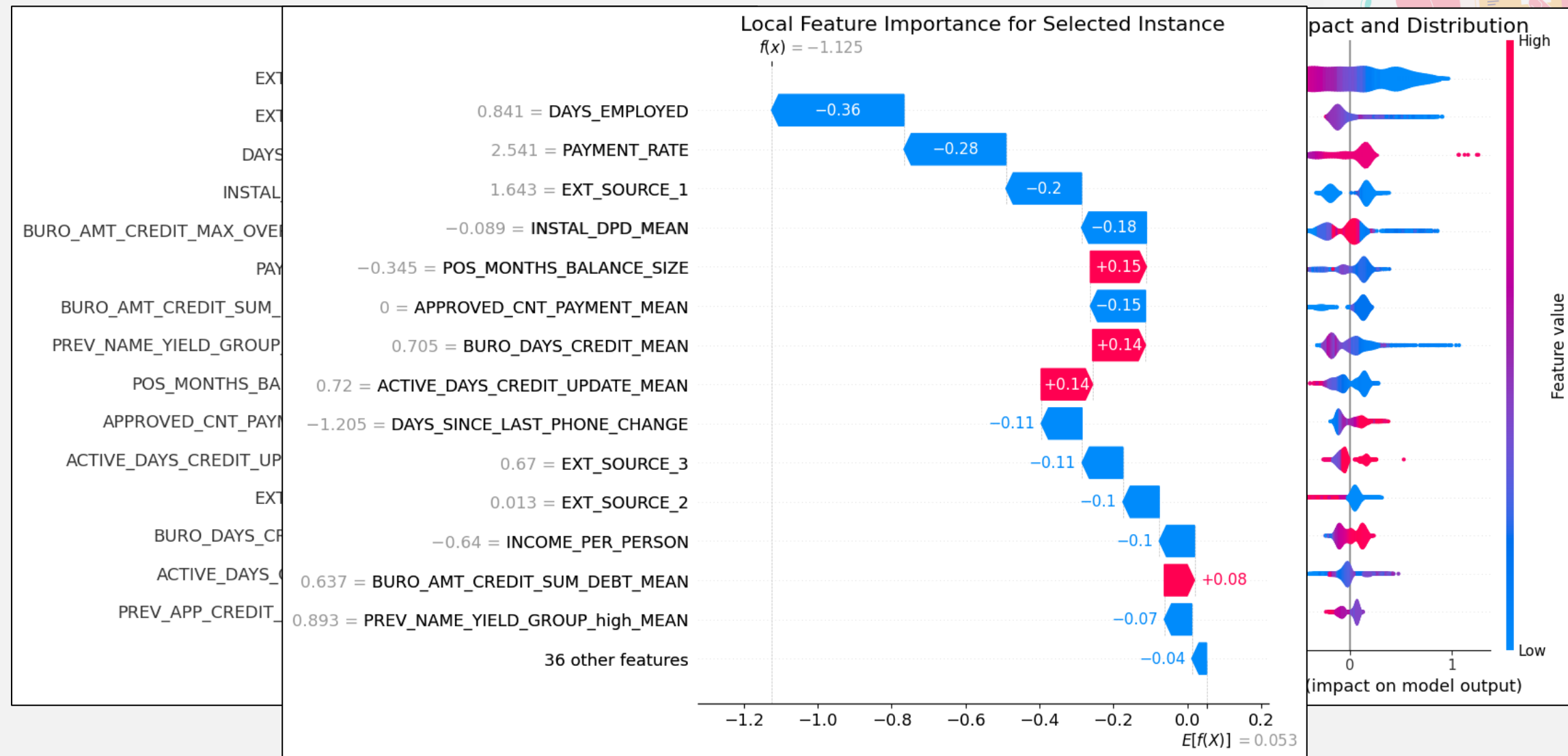
LightGBM



Confusion-Cost Matrices for LGBMClassifier (Scorer: business) at Threshold 0.48



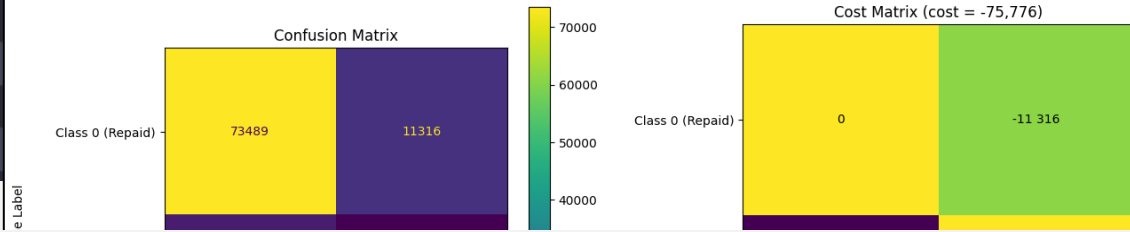
Feature Importance



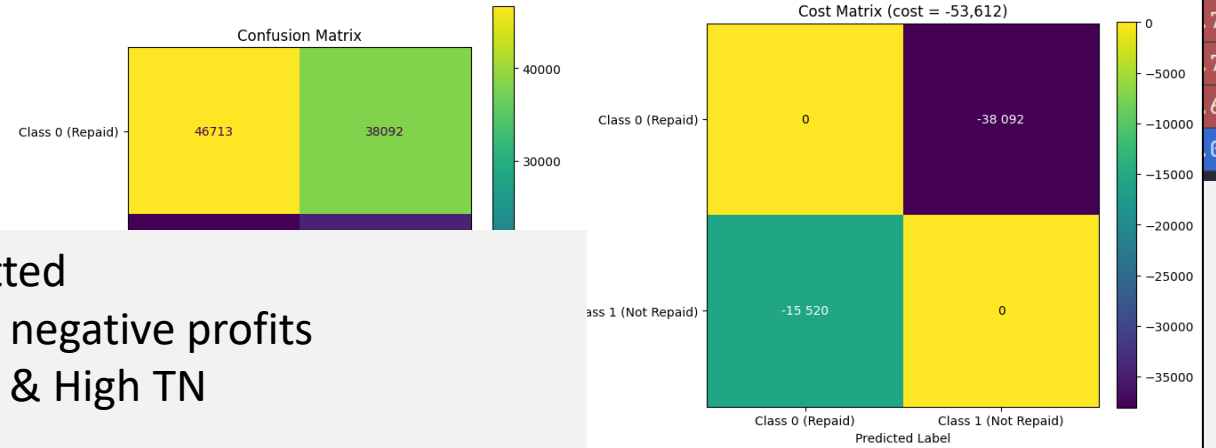
Summary



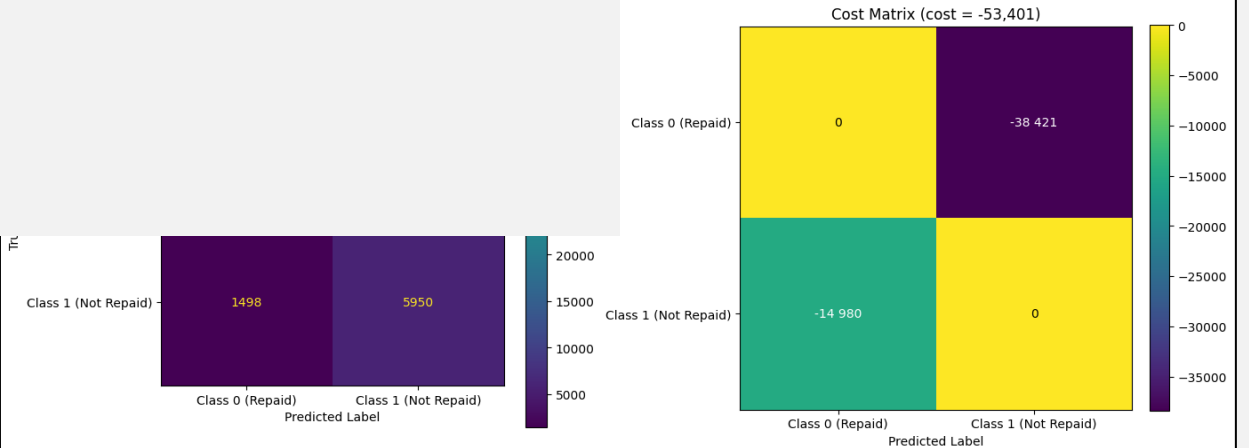
Confusion-Cost Matrices for DummyClassifier (Scorer: roc_auc)



Confusion-Cost Matrices for XGBClassifier (Scorer: business)

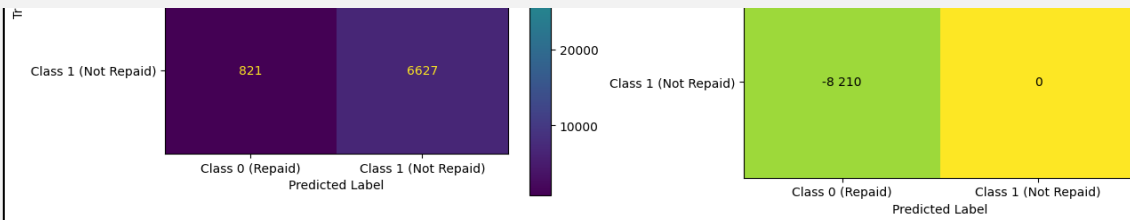


Confusion-Cost Matrices for XGBClassifier (Scorer: business) at Threshold 0.48

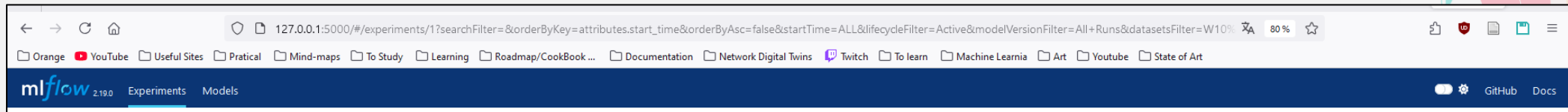
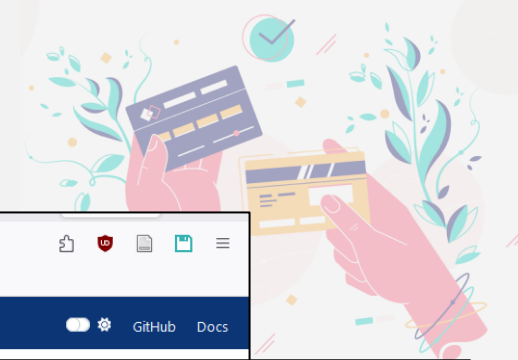


- Logistic Regression → faces convergence issues & cannot be fitted
- RandomForest → is the best in terms of pure FN reduction but negative profits
- LightGBM & XGBoost → Decent balance between FN reduction & High TN

Our choice → LightGBM



ML Flow



mlflow 2.19.0 Experiments Models GitHub Docs

Registered Models

Share and manage machine learning models. [Learn more](#)

Filter registered models by name or tags Q

Name ↕	Latest version	Aliased versions	Created by	Last modified	Tags
DummyClassifier - roc_auc	Version 2			2025-01-16 20:34:03	—
LGBMClassifier - business	Version 2	@ champion : Version 2 +1		2025-01-17 01:08:23	—
LGBMClassifier - roc_auc	Version 1			2025-01-16 21:28:35	—
RandomForestClassifier - business	Version 1	@ candidate : Version 1		2025-01-16 23:55:05	—
RandomForestClassifier - roc_auc	Version 1			2025-01-16 23:39:00	—
XGBClassifier - business	Version 1	@ candidate : Version 1		2025-01-16 23:24:22	—
XGBClassifier - roc_auc	Version 1			2025-01-16 23:09:51	—

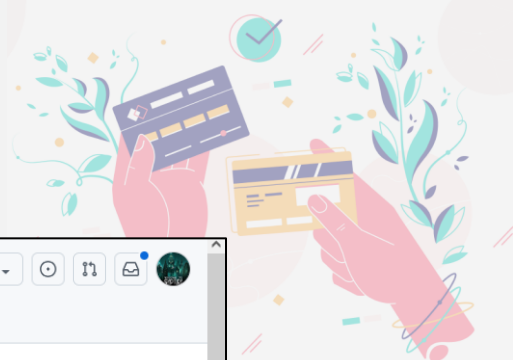
<input type="checkbox"/>	DummyClassifier (ro...	1 day ago	3.7min	Ovidiu PAS...	C:\Users...	DummyClassifier - r...	0.4950	{ "model": "DummyClassif...	DummyClassifier	-
<input type="checkbox"/>	DummyClassifier (ro...	1 day ago	4.5min	Ovidiu PAS...	C:\Users...	DummyClassifier - r...	0.4956	{ "model": "DummyClassif...	DummyClassifier	-
<input type="checkbox"/>	DummyClassifier (ro...	4 days ago	1.9min	Ovidiu PAS...	C:\Users...	sklearn	0.5019	{ "model": "DummyClassif...	DummyClassifier	-
<input type="checkbox"/>	DummyClassifier (ro...	4 days ago	2.1min	Ovidiu PAS...	C:\Users...	sklearn	0.5019	{ "model": "DummyClassif...	DummyClassifier	-

Prêt à
dépendre



CI/CD Pipeline

Code Versioning



Rapture244 / OC-Projet-7-

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Search: Type / to search

Commits

main

All users All time

Commits on Jan 17, 2025

- Error in making the .slugignore, Heroku app crash 98f2345
ovidiu.pascal committed 5 days ago · ✓ 2 / 2
- Small changes: notes, script, README, .gitignore 56b5cdc
ovidiu.pascal committed 5 days ago · ✓ 2 / 2
- Last API Changes 40a5163
ovidiu.pascal committed 5 days ago
- Added the final notebooks and their outputs a044b81
ovidiu.pascal committed 5 days ago
- Small typo in saving xgboost Model + added 2 main title to threshold 4ee3c74
ovidiu.pascal committed 5 days ago

Commits on Jan 16, 2025

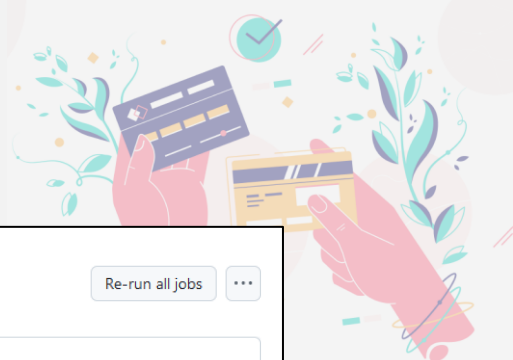
- Moved local streamlit & created a cloud streamlit to request api a2aad14
ovidiu.pascal committed last week
- Integrating CI/CD & Heroku deployment after 13 Errors/trials + poetry 31b234e
ovidiu.pascal committed last week
- Modifying imports after the changes in local packages/modules 5b3dcf9
ovidiu.pascal committed last week
- Moving failed attempt at ML Flow API

Tests made with PyTest

```
1 """
2 This test suite validates the behavior and robustness of the '/predict' endpoint in the Flask API.
3
4 Key Features:
5 1. Utilizes 'pytest' fixtures to create a reusable Flask test client for API requests.
6 2. Covers multiple scenarios to ensure comprehensive testing of the '/predict' endpoint.
7
8 Test Cases:
9 - 'test_predict_valid_id': Verifies correct handling of valid IDs that exist in the dataset.
10 - 'test_predict_invalid_id': Ensures appropriate error handling for IDs that do not exist in the dataset.
11 - 'test_predict_missing_id': Tests the API's response to payloads missing the required 'id' field.
12 - 'test_predict_invalid_payload': Validates error handling for invalid, non-JSON payloads.
13 - 'test_api_availability': Confirms the API responds correctly to undefined endpoints.
14
15 Dependencies:
16 - Pytest: Framework for writing and running tests.
17 - Flask: Framework providing the API to test.
18 - JSON: Used for crafting request payloads and interpreting responses.
19
20 Notes:
21 - Ensure the 'api.local_main' module and the Flask app are correctly configured and imported.
22 - Adjust 'payload' values in test cases as needed to match the data in the actual dataset.
23 - This test suite assumes '/predict' is the only defined route; update the 'test_api_availability' case if other endpoints exist.
24
25 """
```

```
1
2 # ===== IMPORTS ===== #
3 import pytest
4 from api.local_main import app
5
6
7 # ===== TEST ===== #
8 #
9 # =====
10 @pytest.fixture
11 def client():
12     """
13     Creates a test client for the Flask app.
14     """
15     with app.test_client() as client:
16         yield client
17
18 def test_predict_valid_id(client):
19     """
20     Test case for a valid ID that exists in the dataset.
21     """
22     payload = {"id": 100001} # Ensure this ID exists in the dataset as an integer
23     response = client.post("/predict", json=payload)
24     assert response.status_code == 200
25     data = response.get_json()
26     assert data is not None
27     assert "SK_ID_CURR" in data
28     assert data["SK_ID_CURR"] == payload["id"]
29     assert "predicted_proba" in data
30     assert "predicted_target" in data
31     assert "status" in data
32
33 def test_predict_invalid_id(client):
34     """
35     Test case for an invalid ID that does not exist in the dataset.
36     """
37     payload = {"id": 999999} # Non-existent ID as an integer
38     response = client.post("/predict", json=payload)
39     assert response.status_code == 404
40     data = response.get_json()
41     assert data is not None
42     assert "error" in data
43     assert "not found" in data["error"].lower()
44
45 def test_predict_missing_id(client):
46     """
47     Test case for a missing 'id' field in the payload.
48     """
49     payload = {} # Missing 'id'
50     response = client.post("/predict", json=payload)
51     assert response.status_code == 400
52     data = response.get_json()
53     assert data is not None
54     assert "error" in data
55     assert "missing or invalid 'id'" in data["error"].lower()
56
57 def test_predict_invalid_payload(client):
58     """
59     Test case for an invalid payload that is not JSON.
60     """
61     response = client.post("/predict", data="not a json payload")
62     assert response.status_code == 400
63     data = response.get_json()
64     assert data is not None
65     assert "error" in data
66     assert "invalid json payload" in data["error"].lower()
67
68 def test_api_availability(client):
69     """
70     Test case for accessing an undefined endpoint.
71     """
72     response = client.get("/") # Assuming no endpoint at "/"
73     assert response.status_code == 404
74
```


GitHub Actions to build, test & deploy the API



Rapture244

<> Code Issues

Actions

All workflows

CI/CD Heroku Deployment

Management

Caches

Deployments

Attestations

Runners

Usage metrics

Performance metrics

← CI/CD Heroku Deployment

✓ Error in making the .slugignore, Heroku app crash #27

Re-run all jobs ...

Summary

Jobs

✓ build-and-test

✓ deploy

Run details

Usage

Workflow file

Annotations

1 warning

⚠ ubuntu-latest pipelines will use ubuntu-24.04 soon. For more details, see <https://github.com/actions/runner-images/issues/10636>

deploy

succeeded 5 days ago in 2m 11s

Search logs

🔄 ⚙️

> ✓ Set up job

1s

> ✓ Check out code

3s

> ✓ Set up Python

0s

> ✓ Install Poetry

6s

> ✓ Install dependencies (only main)

9s

> ✓ Install Heroku CLI

5s

> ✓ Deploy to Heroku

1m 47s

> ✓ Post Set up Python

0s

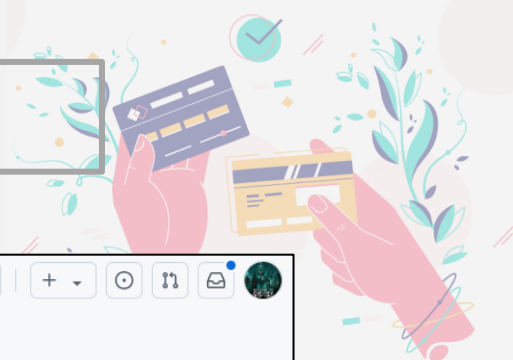
> ✓ Post Check out code

0s

> ✓ Complete job

0s

Used GitHub Secrets To Secure the Connection to Heroku



Screenshot of the GitHub repository settings page for 'Rapture244 / OC-Projet-7-'. The 'Settings' tab is selected, and the 'Actions secrets and variables' section is active. The left sidebar shows the repository structure, with 'Actions' selected under 'Code and automation'. The main content area displays 'Actions secrets and variables' with a description of secrets and variables. Below this, there are tabs for 'Secrets' and 'Variables'. The 'Environment secrets' section shows a message 'This environment has no secrets.' and a 'Manage environment secrets' button. The 'Repository secrets' section shows a table with one secret named 'HEROKU_API_KEY' last updated 'last week'. A 'New repository secret' button is visible in the top right of the repository secrets section.

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Actions

Codespaces

Dependabot

Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. [Learn more about encrypted secrets](#). Variables are shown as plain text and are used for **non-sensitive** data. [Learn more about variables](#).

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Secrets Variables

Environment secrets

This environment has no secrets.

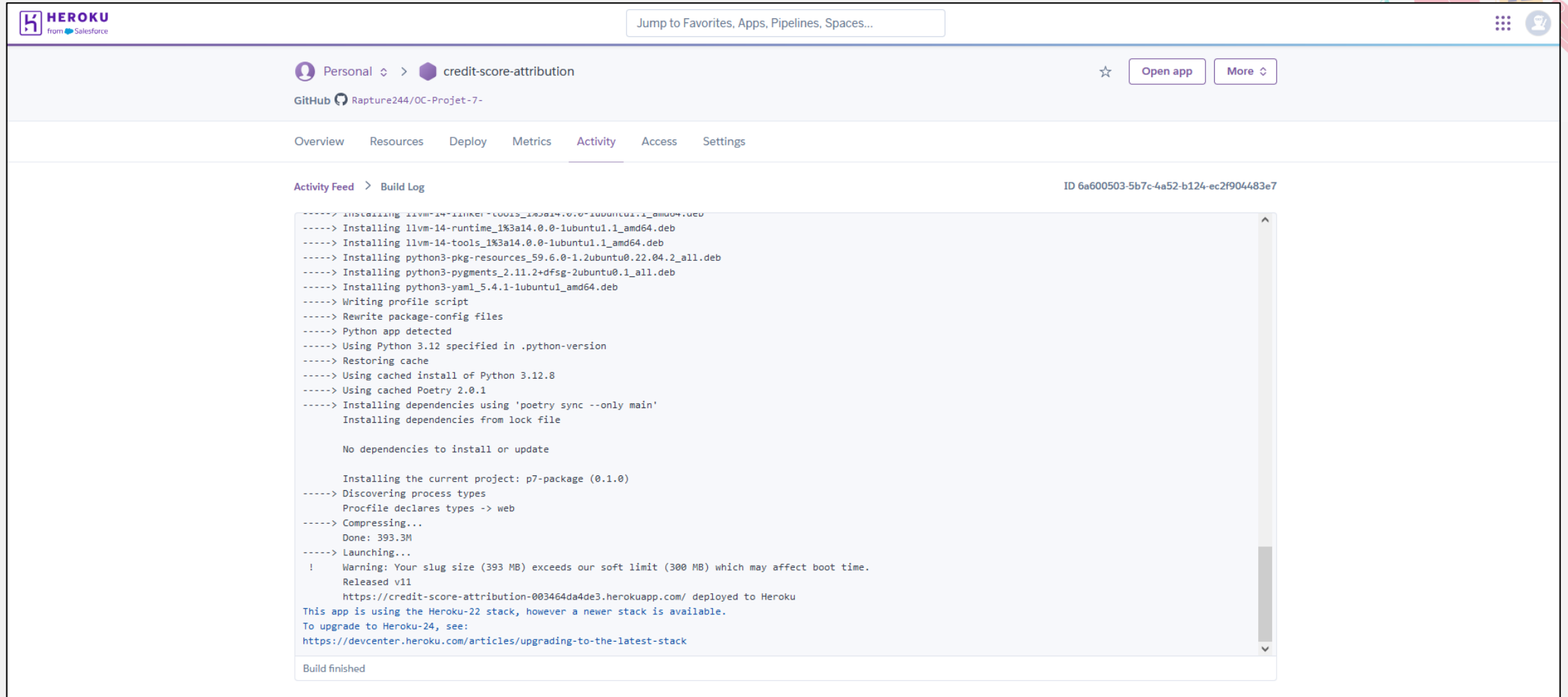
Manage environment secrets

Repository secrets

New repository secret

Name ↕	Last updated
HEROKU_API_KEY	last week

API Deployment on Heroku



The screenshot displays the Heroku dashboard for a personal application named 'credit-score-attribution'. The interface includes a top navigation bar with the Heroku logo, a search bar, and user profile information. Below this, a secondary navigation bar shows the app's name and a link to the GitHub repository 'Rapture244/OC-Projet-7-'. The main content area features a tabbed interface with 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The 'Activity' tab is selected, showing an 'Activity Feed' and a 'Build Log' for a specific build (ID: 6a600503-5b7c-4a52-b124-ec2f904483e7). The build log details the installation of system dependencies (llvm-14, python3-pkg-resources, python3-pygments, python3-yaml), the setup of the Python environment (Python 3.12, Poetry 2.0.1), and the deployment of the application (p7-package 0.1.0). A warning message indicates that the slug size (393 MB) exceeds the soft limit (300 MB), which may affect boot time. The log concludes with the deployment URL and a note about upgrading to the latest Heroku stack.

HEROKU
from Salesforce

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > credit-score-attribution

GitHub Rapture244/OC-Projet-7-

Overview Resources Deploy Metrics Activity Access Settings

Activity Feed > Build Log ID 6a600503-5b7c-4a52-b124-ec2f904483e7

```
-----> Installing llvm-14-compiler-tools_1%3a14.0.0-1ubuntu1.1_amd64.deb
-----> Installing llvm-14-runtime_1%3a14.0.0-1ubuntu1.1_amd64.deb
-----> Installing llvm-14-tools_1%3a14.0.0-1ubuntu1.1_amd64.deb
-----> Installing python3-pkg-resources_59.6.0-1.2ubuntu0.22.04.2_all.deb
-----> Installing python3-pygments_2.11.2+dfsg-2ubuntu0.1_all.deb
-----> Installing python3-yaml_5.4.1-1ubuntu1_amd64.deb
-----> Writing profile script
-----> Rewrite package-config files
-----> Python app detected
-----> Using Python 3.12 specified in .python-version
-----> Restoring cache
-----> Using cached install of Python 3.12.8
-----> Using cached Poetry 2.0.1
-----> Installing dependencies using 'poetry sync --only main'
      Installing dependencies from lock file

      No dependencies to install or update

      Installing the current project: p7-package (0.1.0)
-----> Discovering process types
      Procfile declares types -> web
-----> Compressing...
      Done: 393.3M
-----> Launching...
      !   Warning: Your slug size (393 MB) exceeds our soft limit (300 MB) which may affect boot time.
      Released v11
      https://credit-score-attribution-003464da4de3.herokuapp.com/ deployed to Heroku
This app is using the Heroku-22 stack, however a newer stack is available.
To upgrade to Heroku-24, see:
https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack

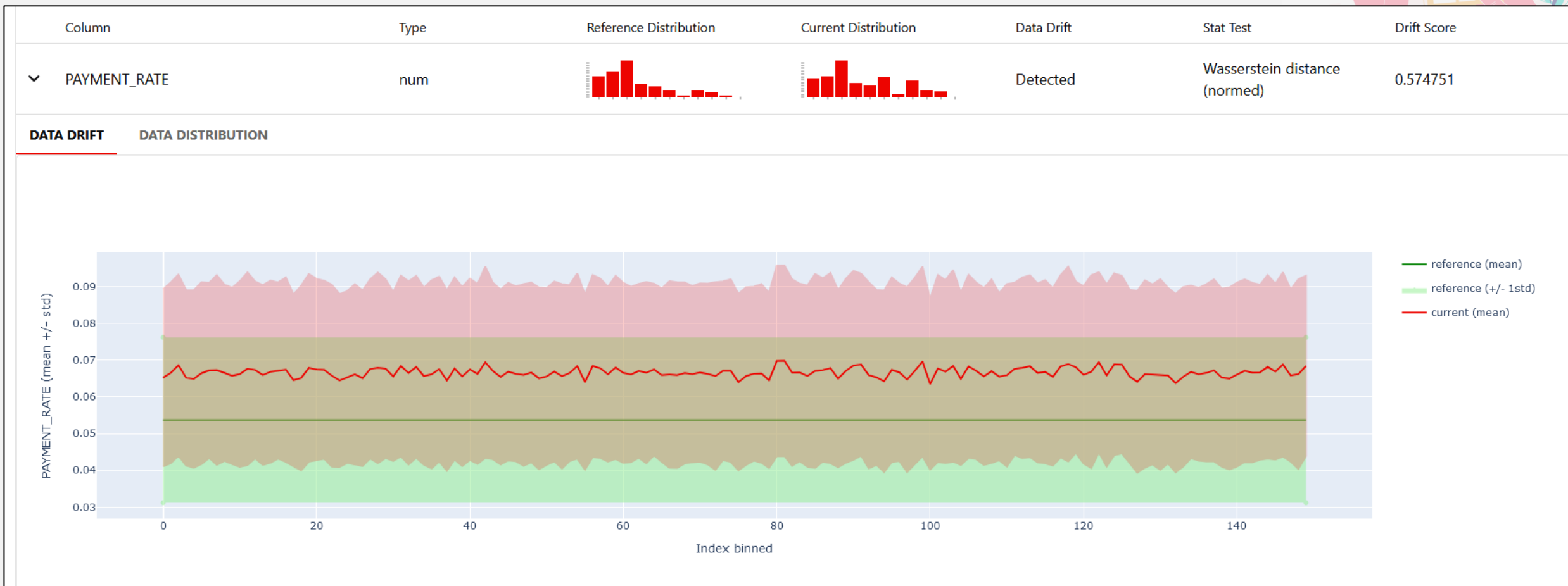
Build finished
```

Prêt à
dépendre



Data Drift

Data Drift





Live Demonstration

- ML Flow UI
- Streamlit test of the API + Heroku Log of the app

Prêt à
dépendre



Q&A