



République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Tunis El Manar
École Nationale d'Ingénieurs de Tunis



Projet Big Data

Banker : Système de Prédition et Monitoring des Prêts

Réalisé et présenté par :

Trigui Hatem
Hassouna Malek
Ben Abdallah Rania
Mzid Mortadha

Supervisé par :
M. Moez Ben Haj
Hmida



01 Introduction

Exploration et Préparation
des Données

Traitement Distribué
avec Spark

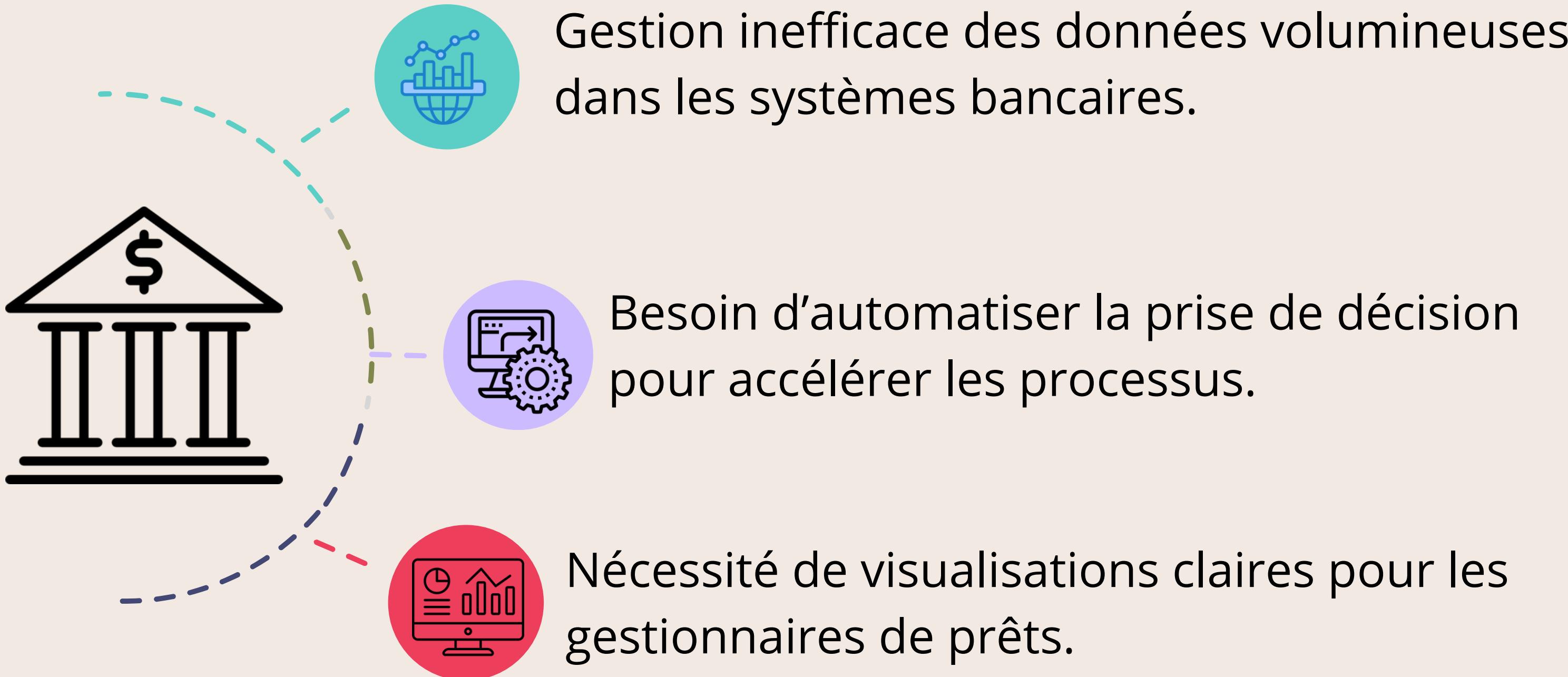
Notre Application “Banker”

Stockage et Visualisation
avec l'ELK Stack

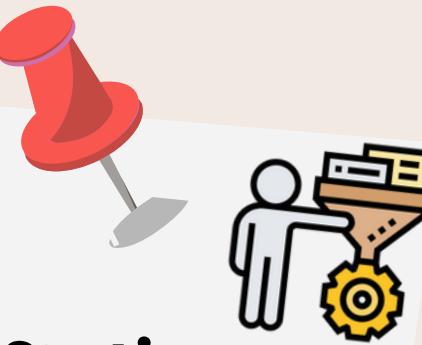
Conclusion

Introduction

Problématique



Solution

Exploration et préparation des données.

Traitement distribué avec Spark.



Développement d'une application de prédiction.

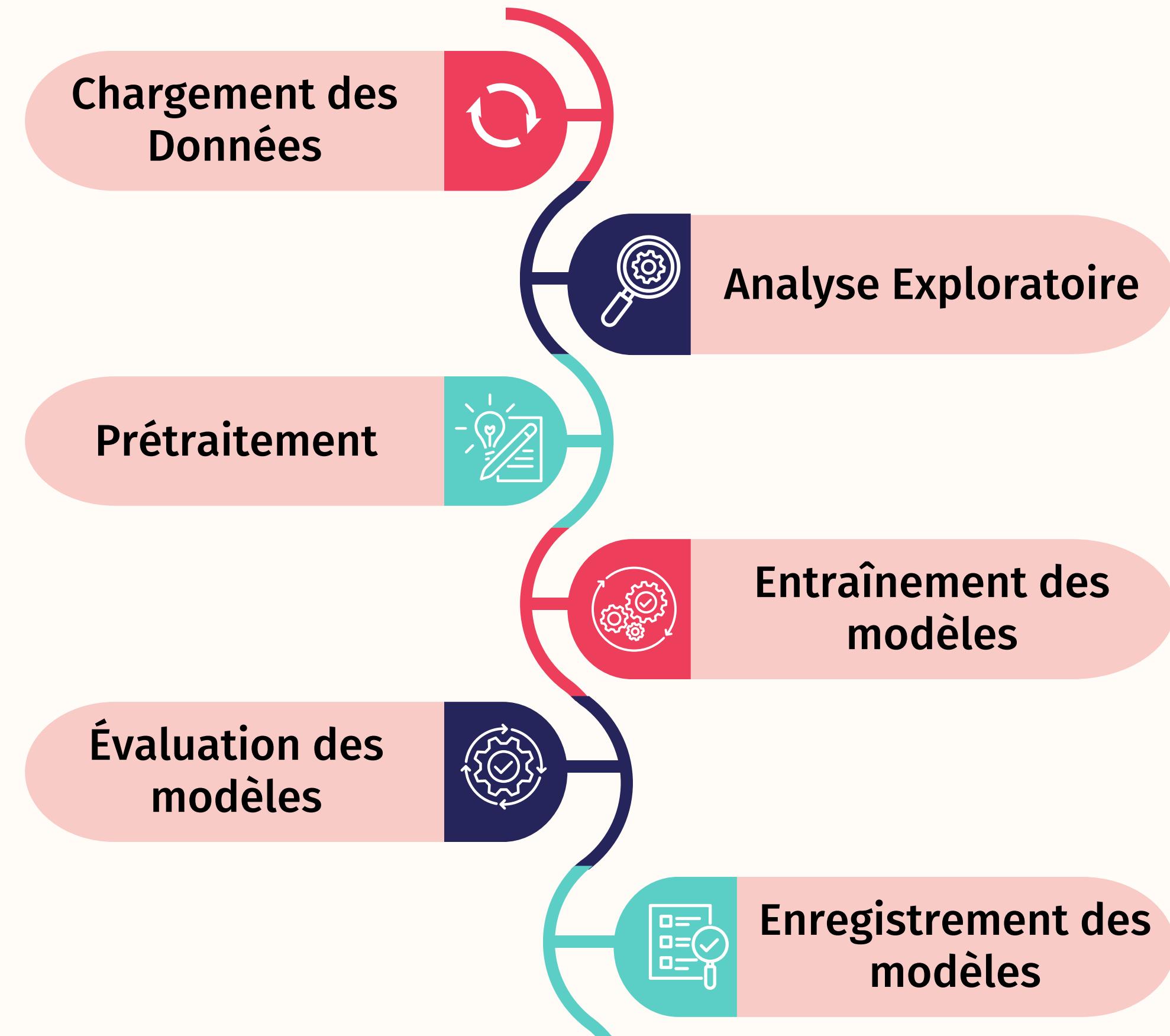
Stockage et visualisation dans ELK.



Simplifier et automatiser le traitement des prêts bancaires grâce aux données massives et à l'IA.

Exploration et Préparation des Données

PROCESSUS



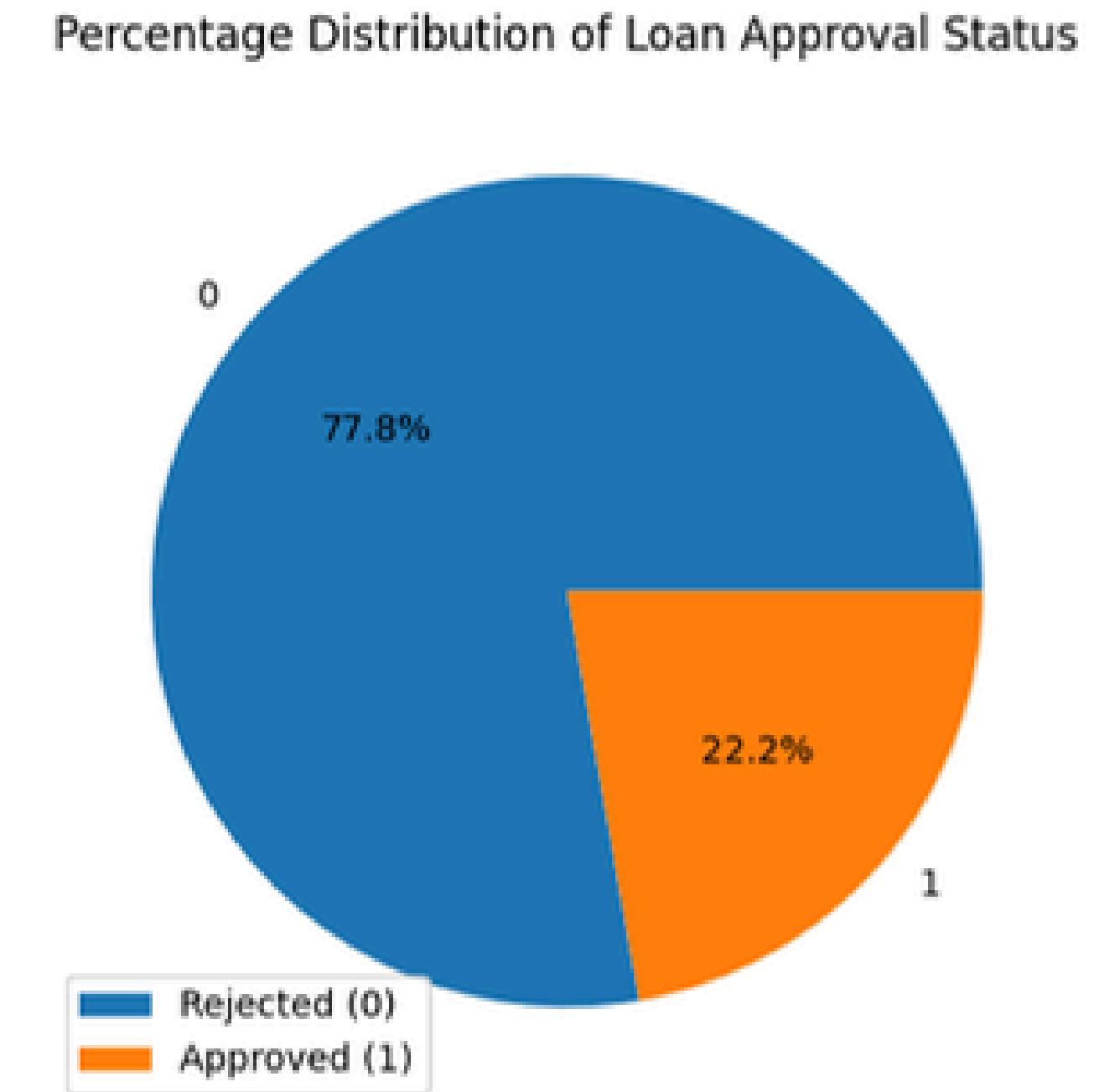
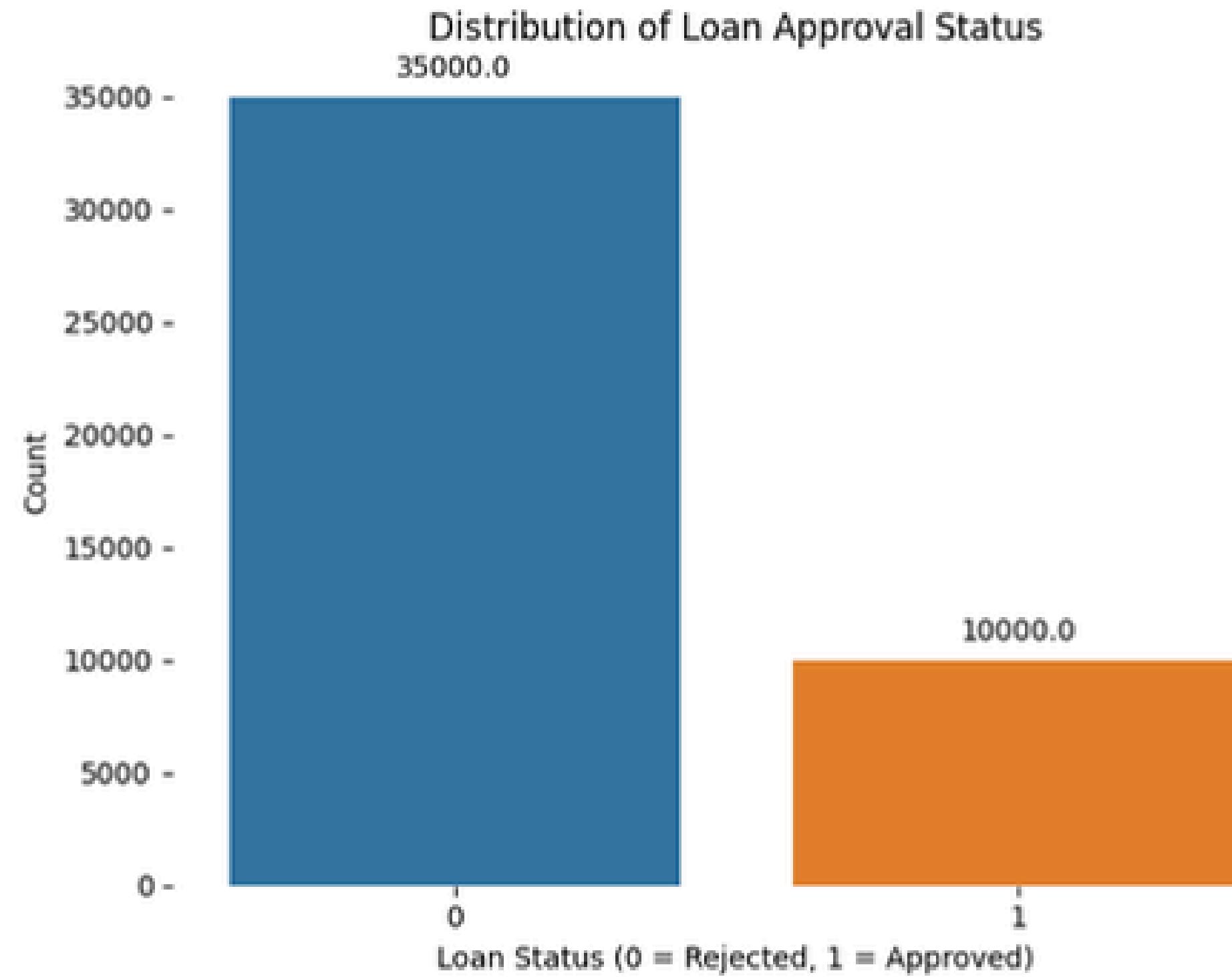
Aperçu de la base de données

Target

Shape of the dataset: (45000, 14)

	person_age	person_gender	person_education	person_income	person_emp_exp	person_home_ownership	loan_amnt	loan_intent	loan_int_rate	loan_percent_income	cb_person_cred_hist_length	credit_score	previous_loan_defaults_on_file	loan_status
0	22.0	female	Master	71948.0	0	RENT	35000.0	PERSONAL	16.02	0.49	3.0	561	No	1
1	21.0	female	High School	12282.0	0	OWN	1000.0	EDUCATION	11.14	0.08	2.0	504	Yes	0
2	25.0	female	High School	12438.0	3	MORTGAGE	5500.0	MEDICAL	12.87	0.44	3.0	635	No	1
3	23.0	female	Bachelor	79753.0	0	RENT	35000.0	MEDICAL	15.23	0.44	2.0	675	No	1
4	24.0	male	Master	66135.0	1	RENT	35000.0	MEDICAL	14.27	0.53	4.0	586	No	1

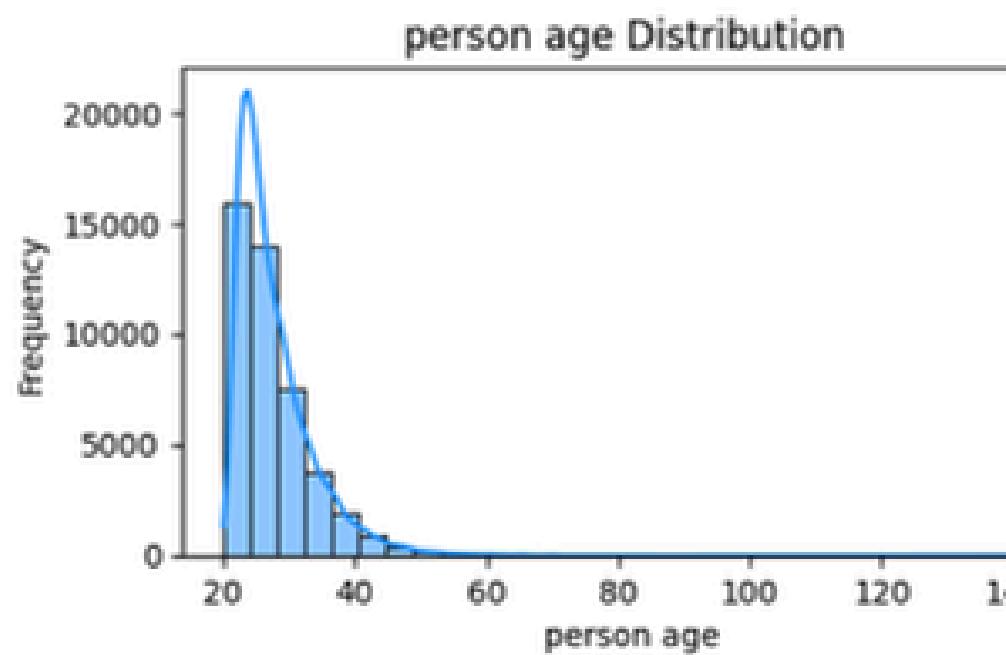
Analyse exploratoire des données (1)



Analyse exploratoire des données (2)

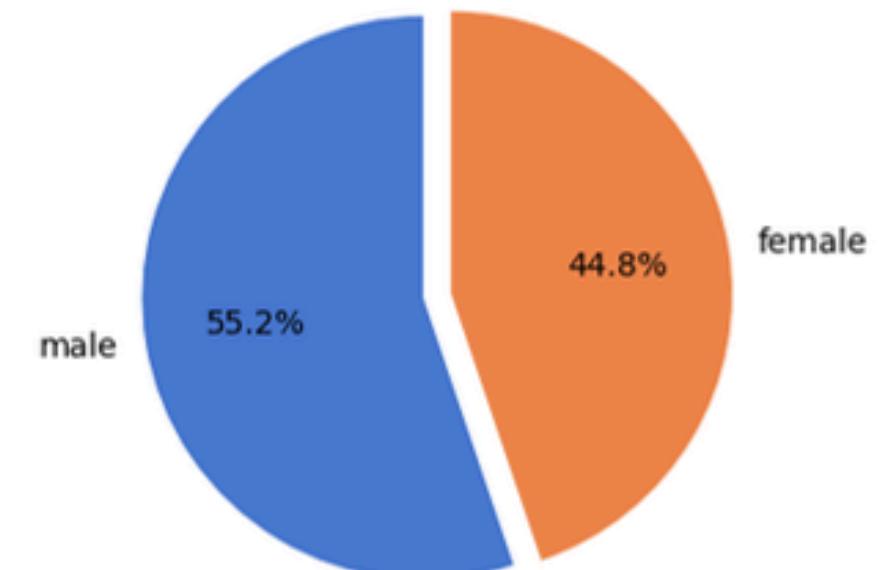
Distribution des Variables

distribution d'age



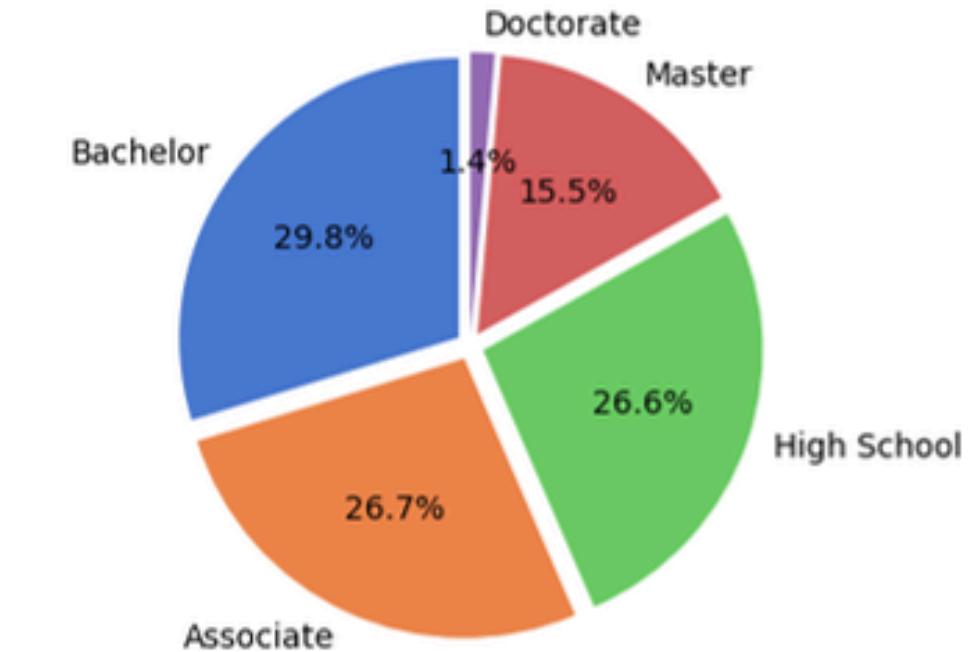
distribution du sexe

Percentage Distribution of person_gender



distribution du niveau d'éducation

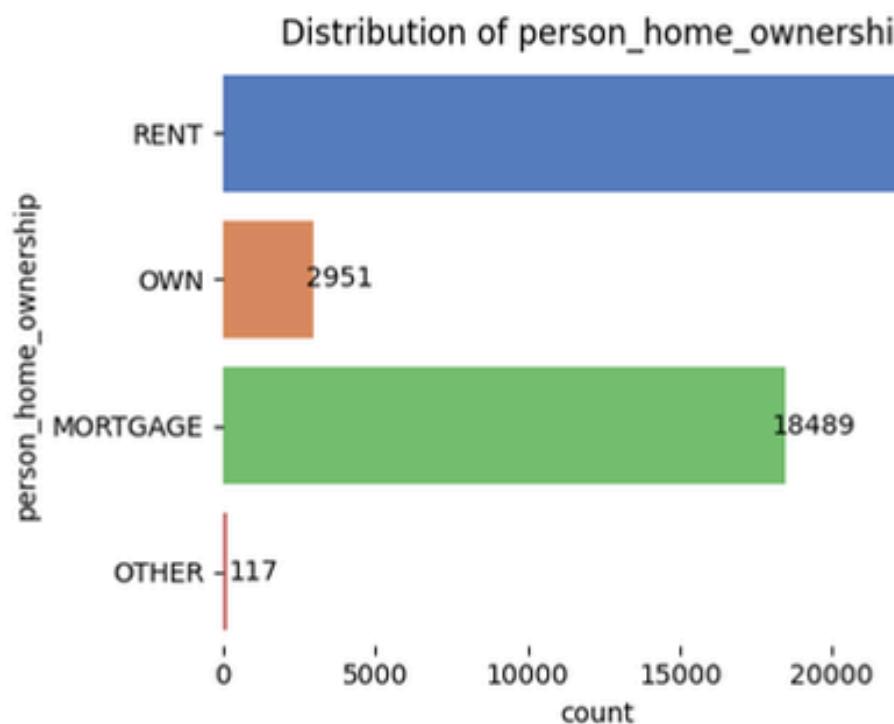
Percentage Distribution of person_education



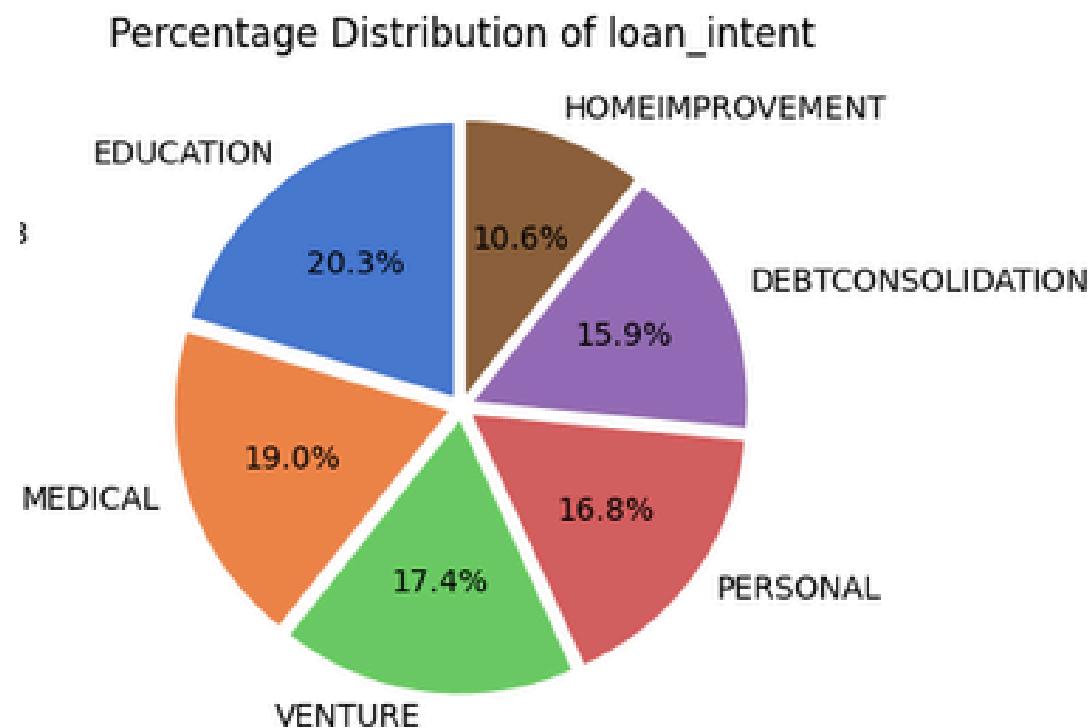
Analyse exploratoire des données (3)

Distribution des Variables

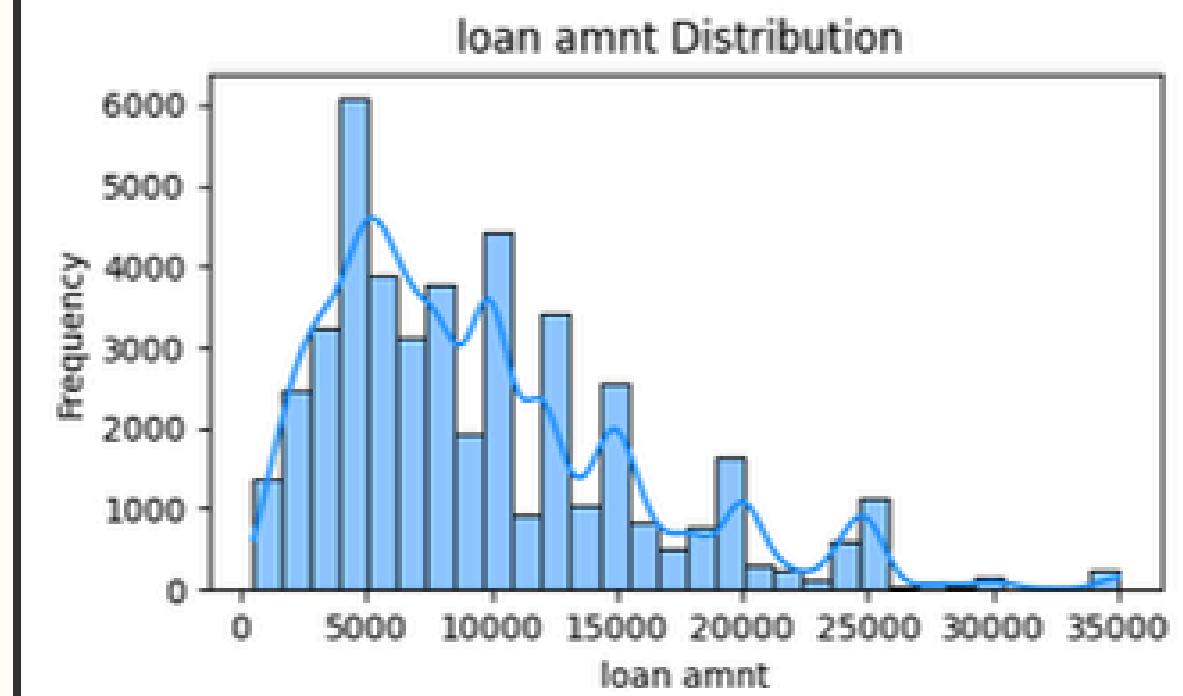
distribution du type du foyer



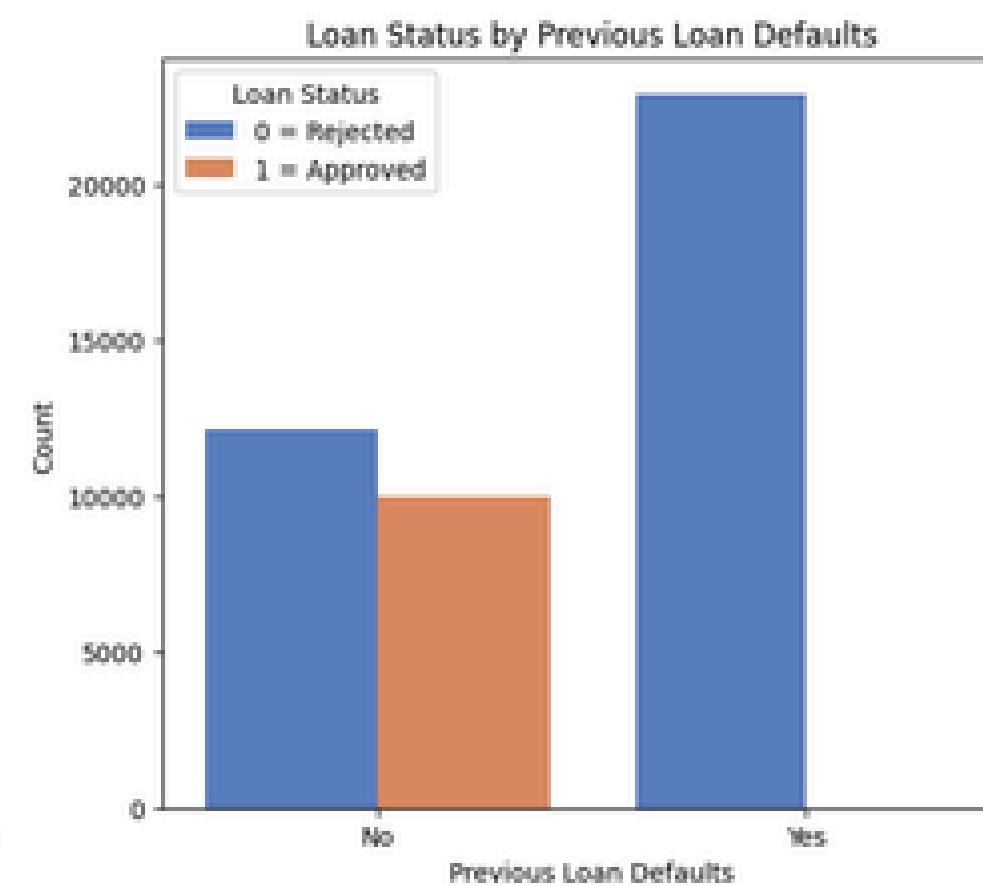
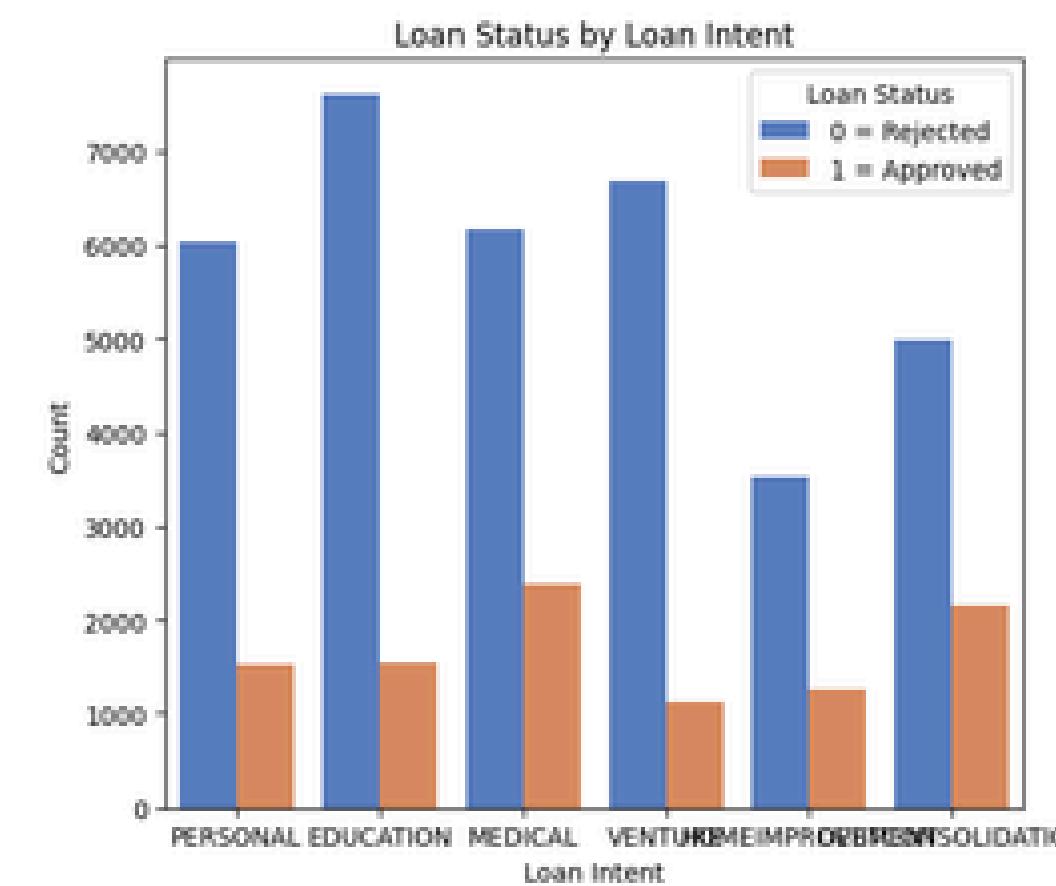
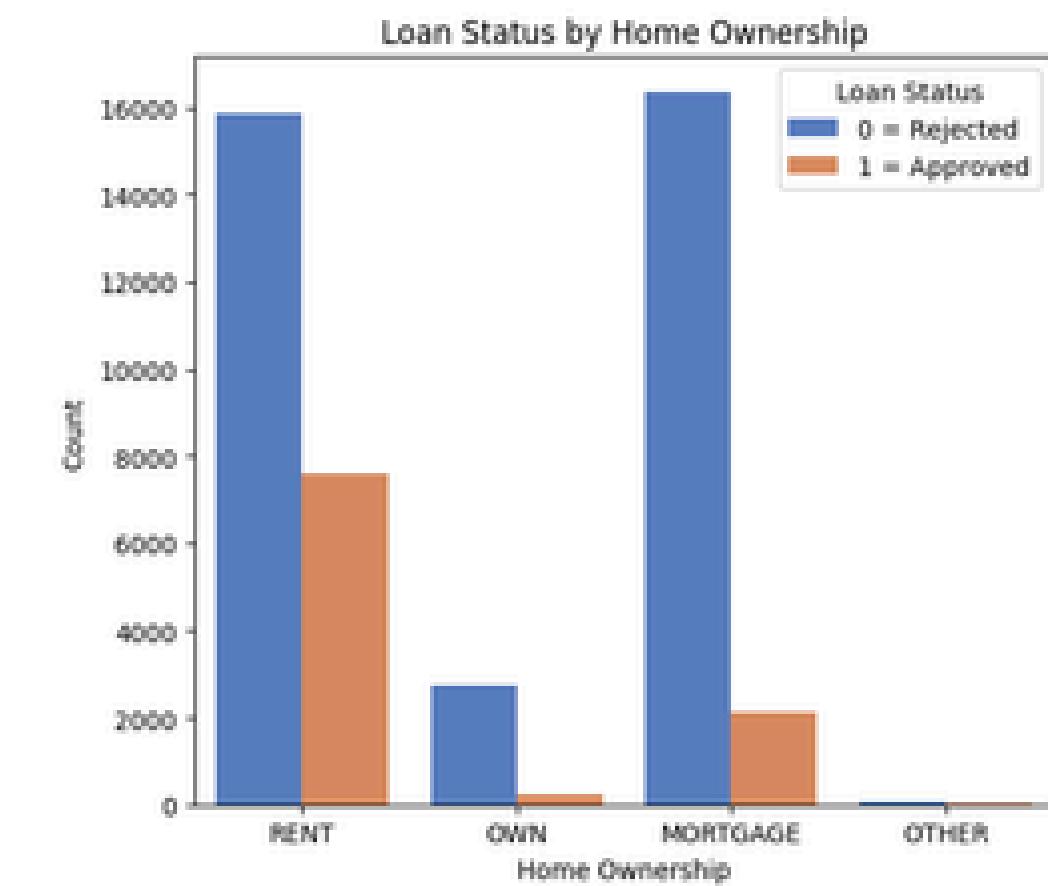
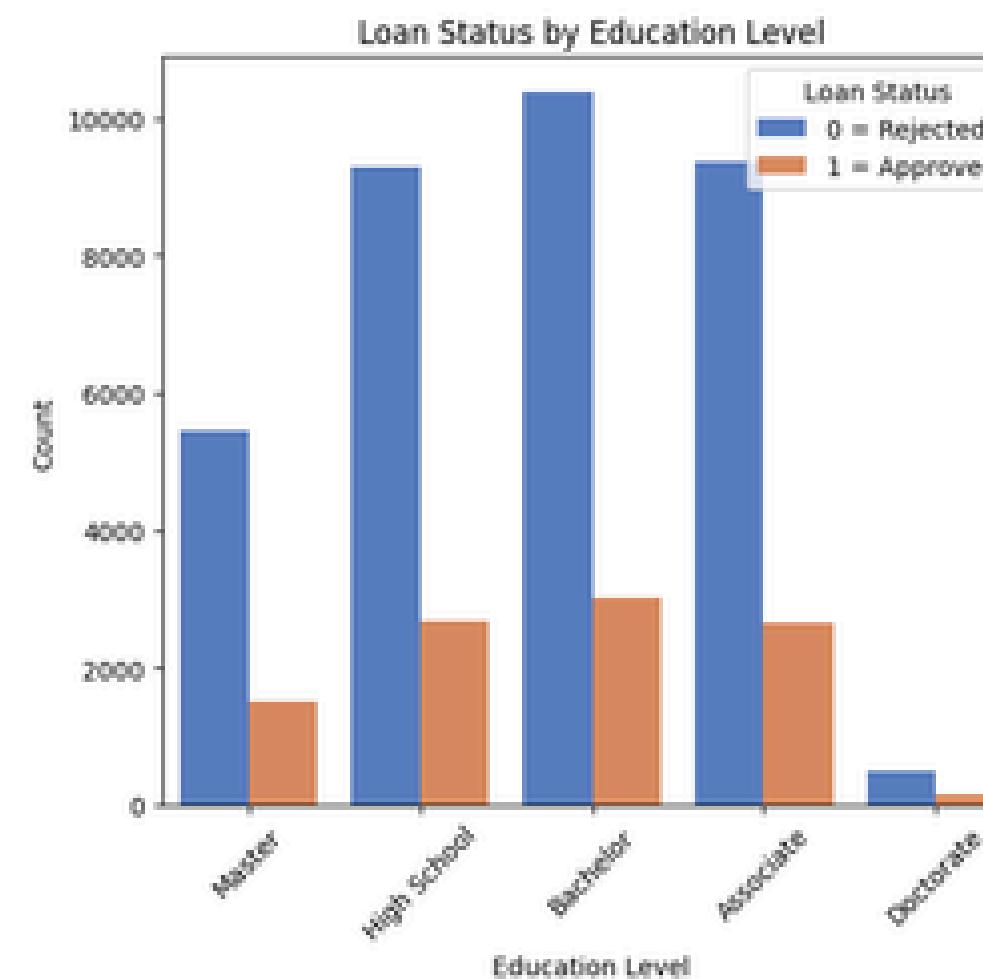
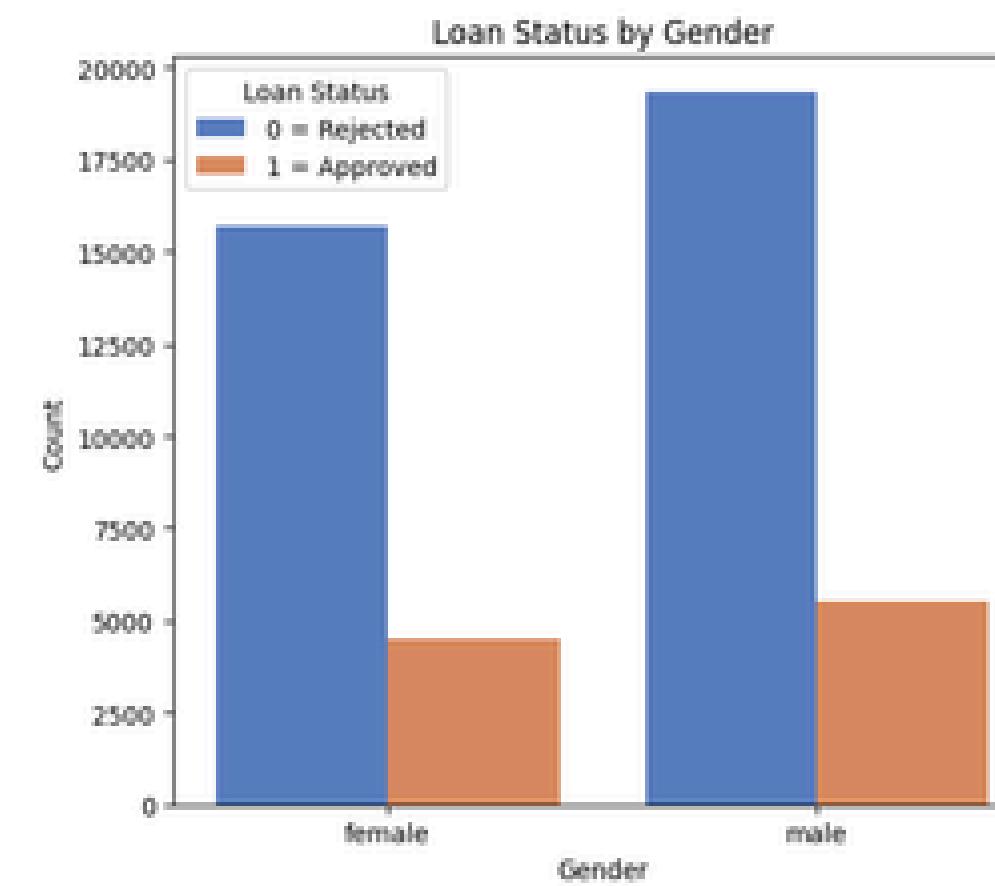
distribution de l'objectif du prêt



distribution de la valeur du prêt

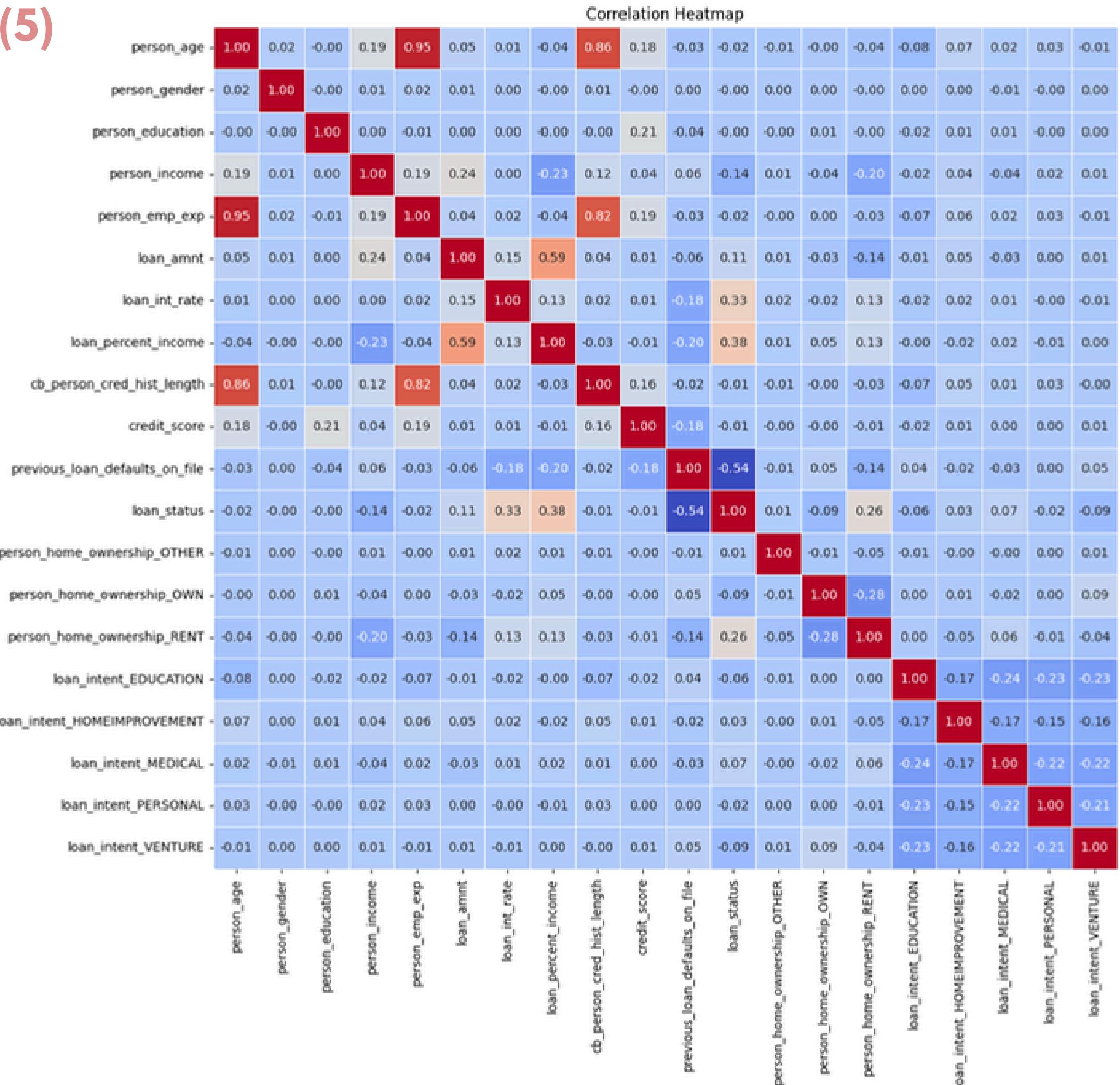


Analyse exploratoire des données (4)

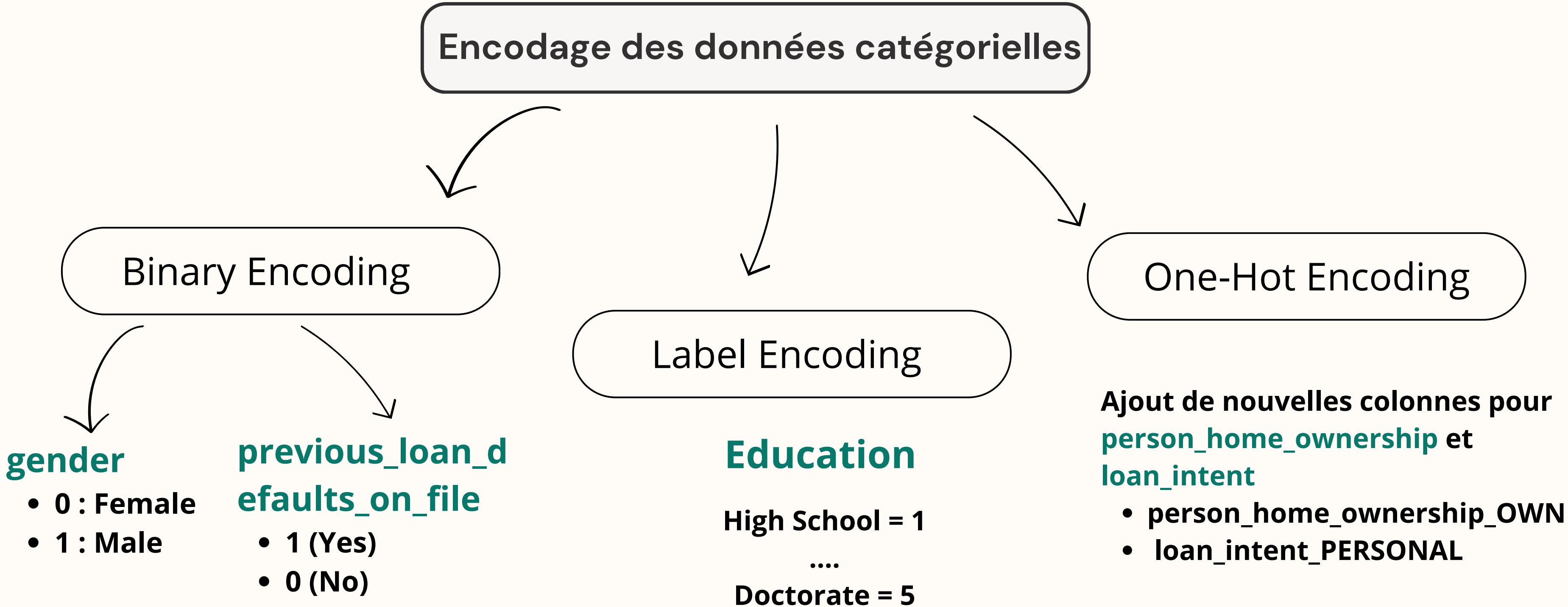


Analyse exploratoire des données (5)

Correlation Map



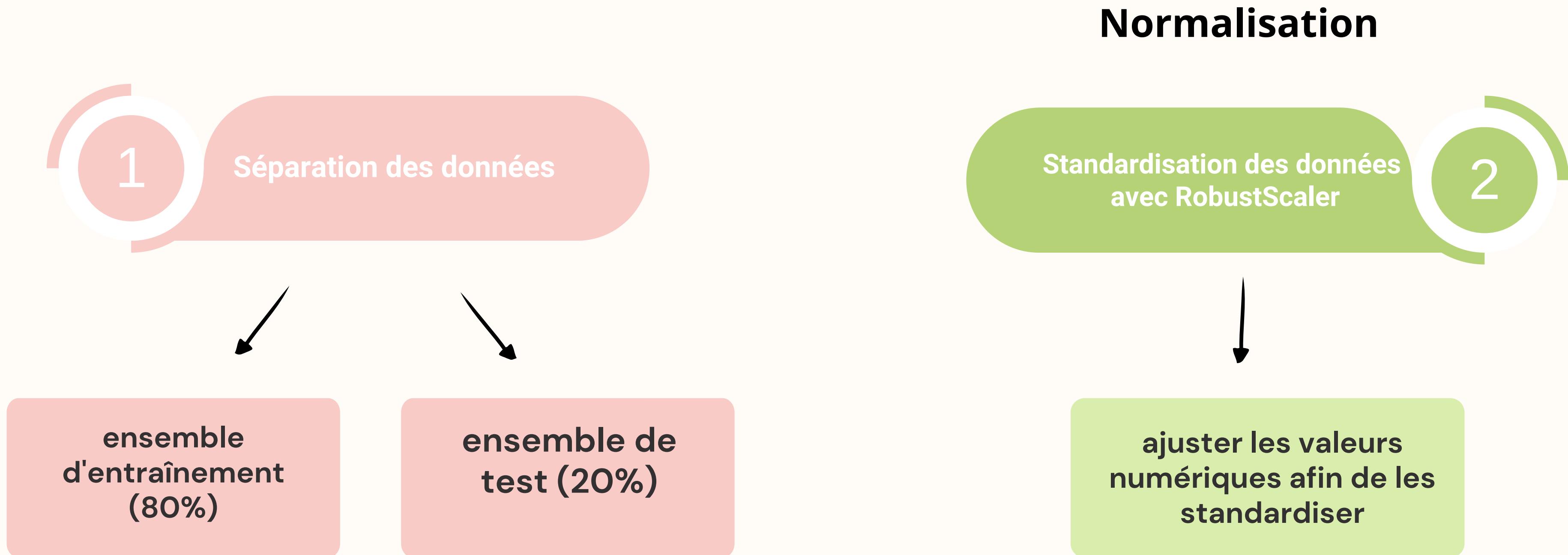
Prétraitement des données (1)



Prétraitement des données (2)

person_gender	person_education	person_home_ownership_OTHER	person_home_ownership_OWN	person_home_ownership_RENT	loan_inten_EDUCATION	loan_intent_HOMEIMPROVEMENT	loan_intent_MEDICAL	loan_intent_PERSONAL	loan_intent_VENTURE
0	4	False	False	True	False	False	False	True	False
0	1	False	True	False	True	False	False	False	False
0	1	False	False	False	False	False	True	False	False
0	3	False	False	True	False	False	True	False	False
1	4	False	False	True	False	False	True	False	False

Prétraitement des données (3)

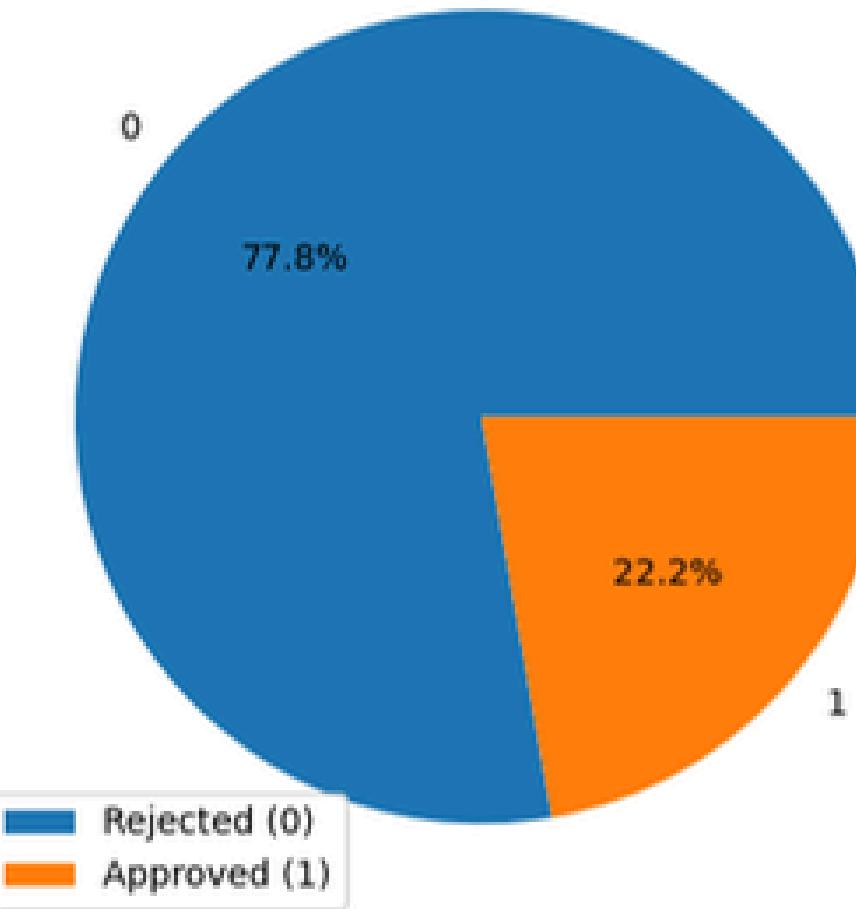


Entraînement des modèles (1)



Données déséquilibrées

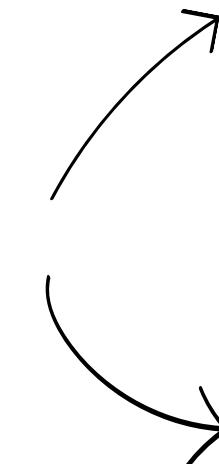
Percentage Distribution of Loan Approval Status



Les techniques de rééchantillonnage

Suréchantillonnage (Oversampling)

Surapprentissage (overfitting)
Augmentation du temps d'entraînement



Sous-échantillonnage (Undersampling)

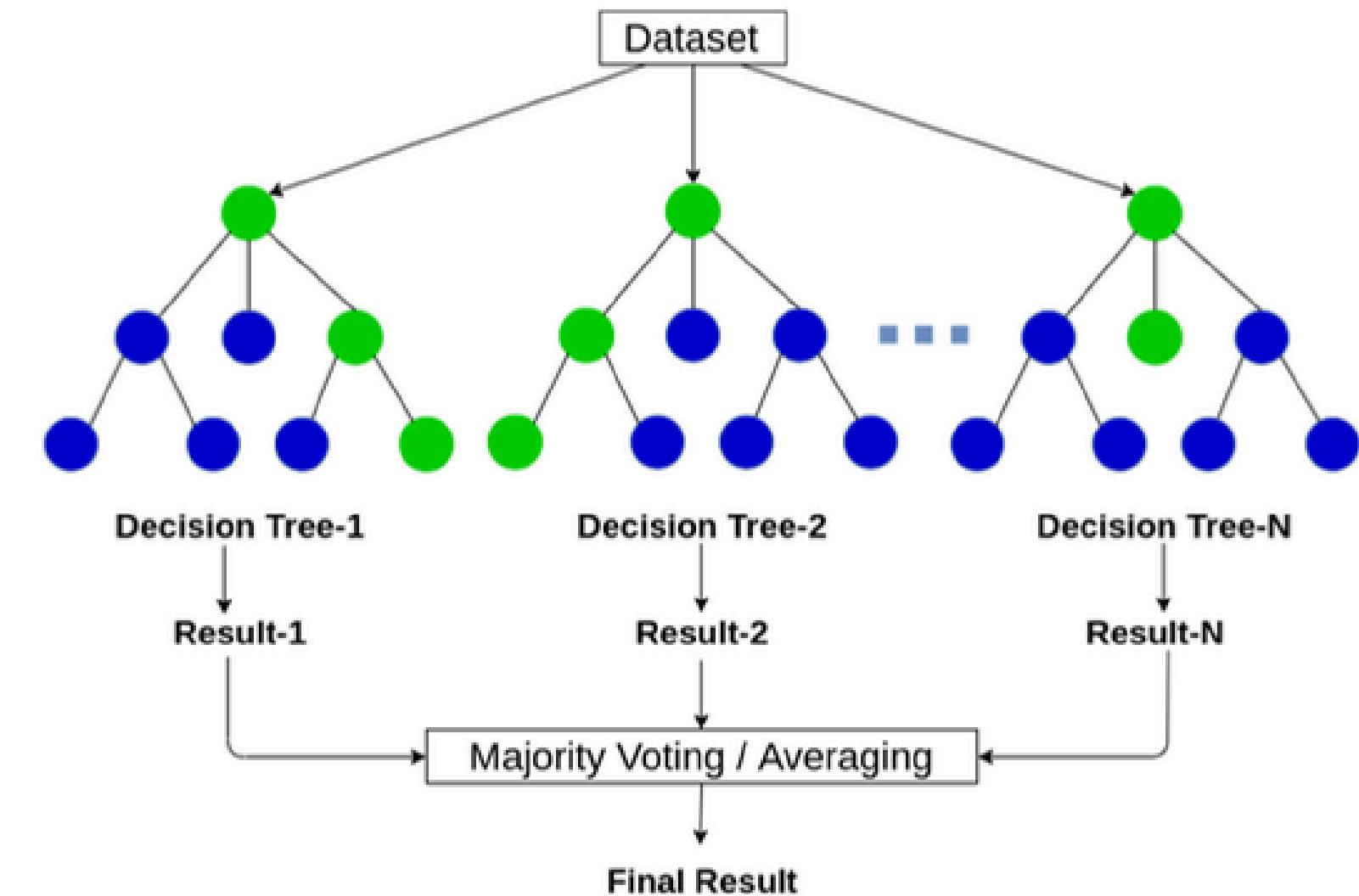
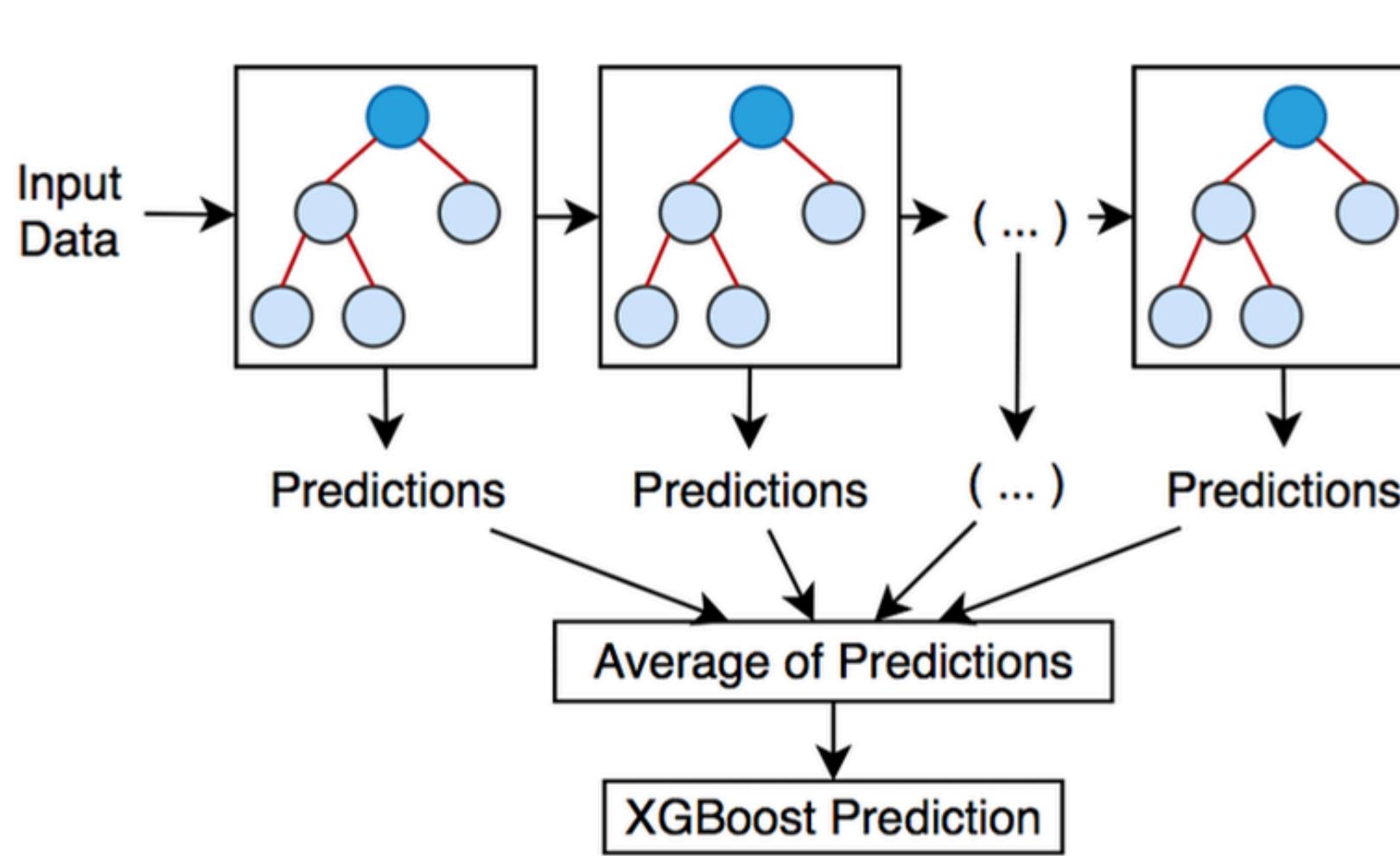


Perte d'informations importantes
Baisse de performance

Entraînement des modèles (2)



Utilisation des algorithmes tels que **XGBoost**, **CatBoost**, **Random Forest** et **Logistic Regression** qui permettent de gérer des classes pondérées et sont particulièrement **adaptés** aux données déséquilibrées.



Evaluation des modèles

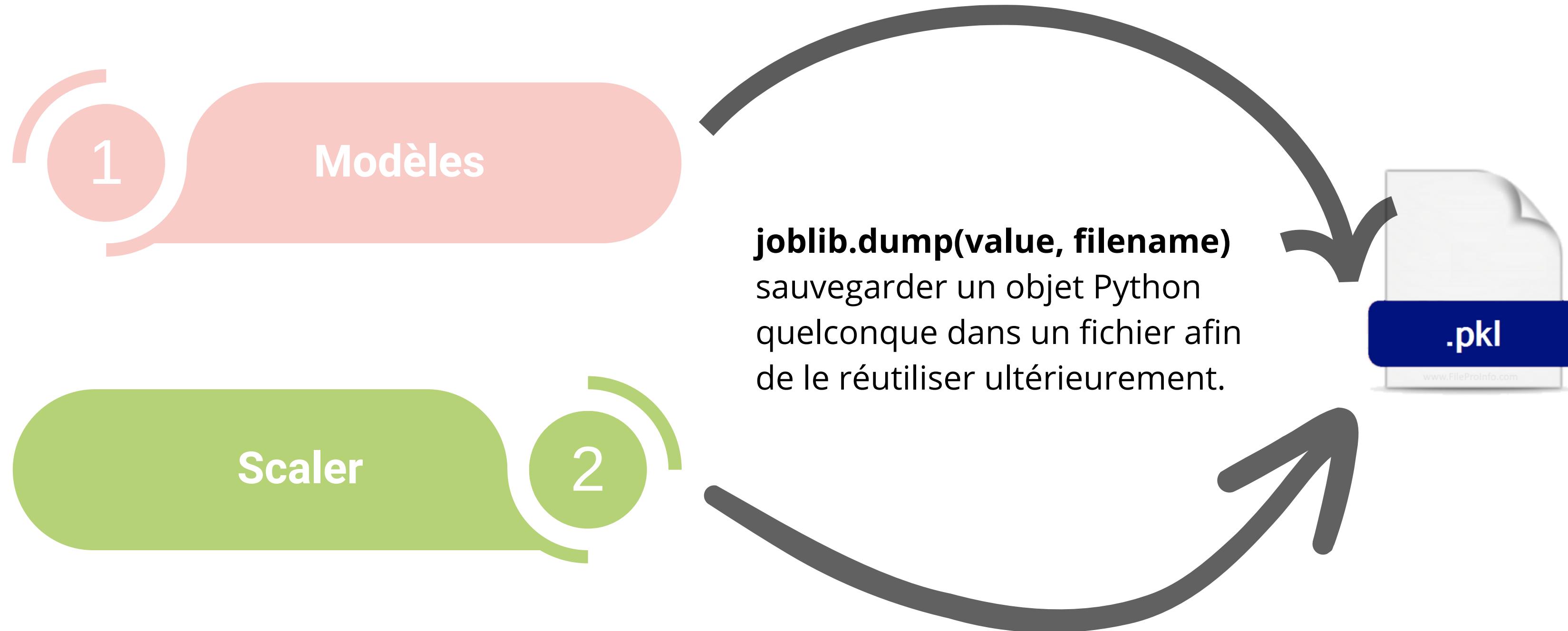
Classification Report for Logistic Regression			
	precision	recall	f1-score
0	0.93	0.94	0.93
1	0.77	0.74	0.76

Classification Report for XGBoost			
	precision	recall	f1-score
0	0.95	0.97	0.96
1	0.89	0.80	0.84

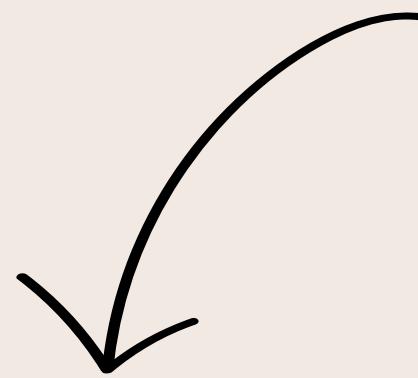
Classification Report for Random Forest			
	precision	recall	f1-score
0	0.94	0.98	0.96
1	0.90	0.78	0.83

Classification Report for CatBoost			
	precision	recall	f1-score
0	0.94	0.97	0.96
1	0.89	0.80	0.84

Enregistrement des modèles

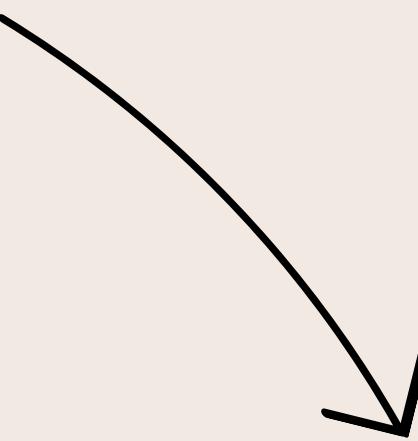


Traitement Distribué avec **Spark**



Qu'est-ce que c'est ?

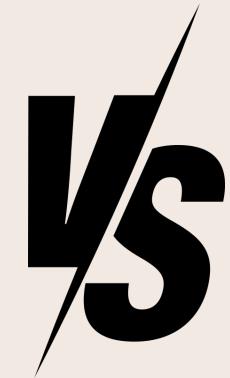
Un framework open-source conçu pour le traitement distribué de données massives



Que permet Spark ?

Il permet d'effectuer des analyses rapides et distribuées

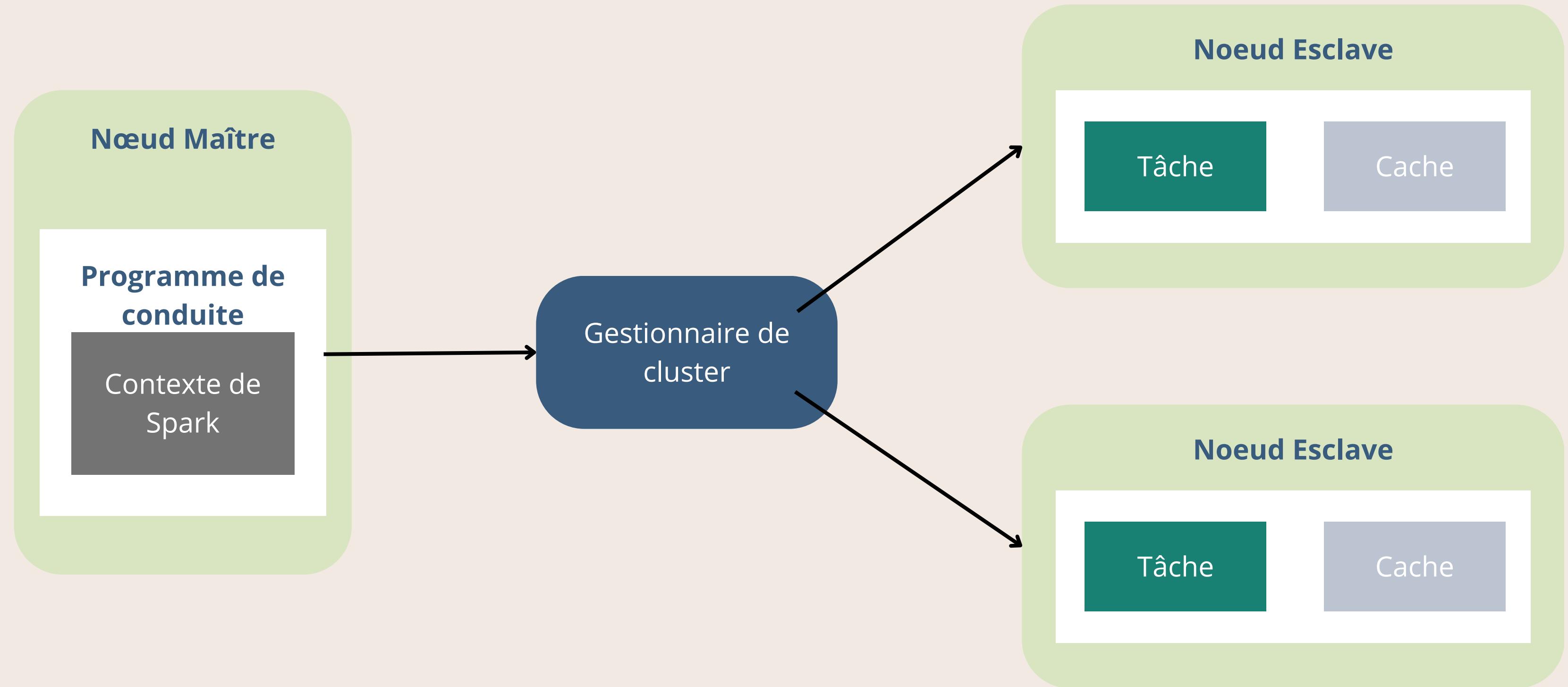
Comparaison entre traitement simple et traitement avec Spark



- Python est conçu pour un traitement local des données.
- Traite les données sur une seule machine.
- Limité par la mémoire de la machine sur laquelle il s'exécute.
- Rapide pour les petites quantités de données.
- Lent pour les grandes quantités de données.

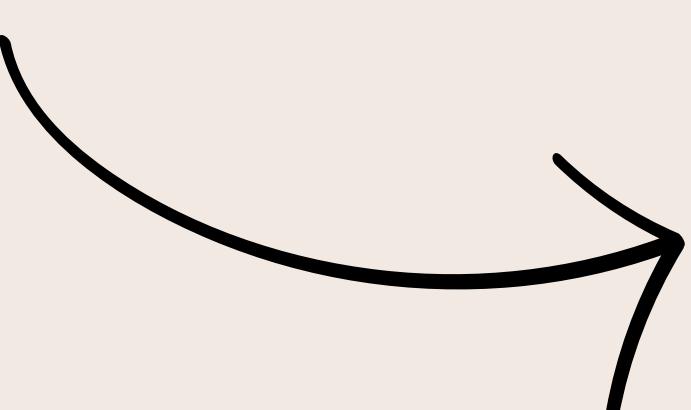
- Permet un traitement distribué et parallèle.
- Hautement scalable. Il peut traiter de très grands ensembles de données en les répartissant sur un cluster de machines.
- Rapide pour les petites comme pour les grandes quantités de données.

Fonctionnement de Spark



Problèmes d'Installation

Des incompatibilités entre les versions de Spark et du JDK



Solution avec Docker

Nous avons utilisé une image Docker préconfigurée pour Spark nommée **jupyter/pyspark-notebook**, incluant toutes les dépendances nécessaires ainsi qu'un notebook Jupyter intégré. Grâce à ce les conteneur de Docker, nous avons pu exécuter Spark dans un environnement standardisé

Comparaison du Chargement des Données

Sans PySpark

```
import pandas as pd

# Chargement des données depuis un fichier CSV
df = pd.read_csv('data.csv')

# Affichage des premières lignes
print(df.head())
```

Avec PySpark

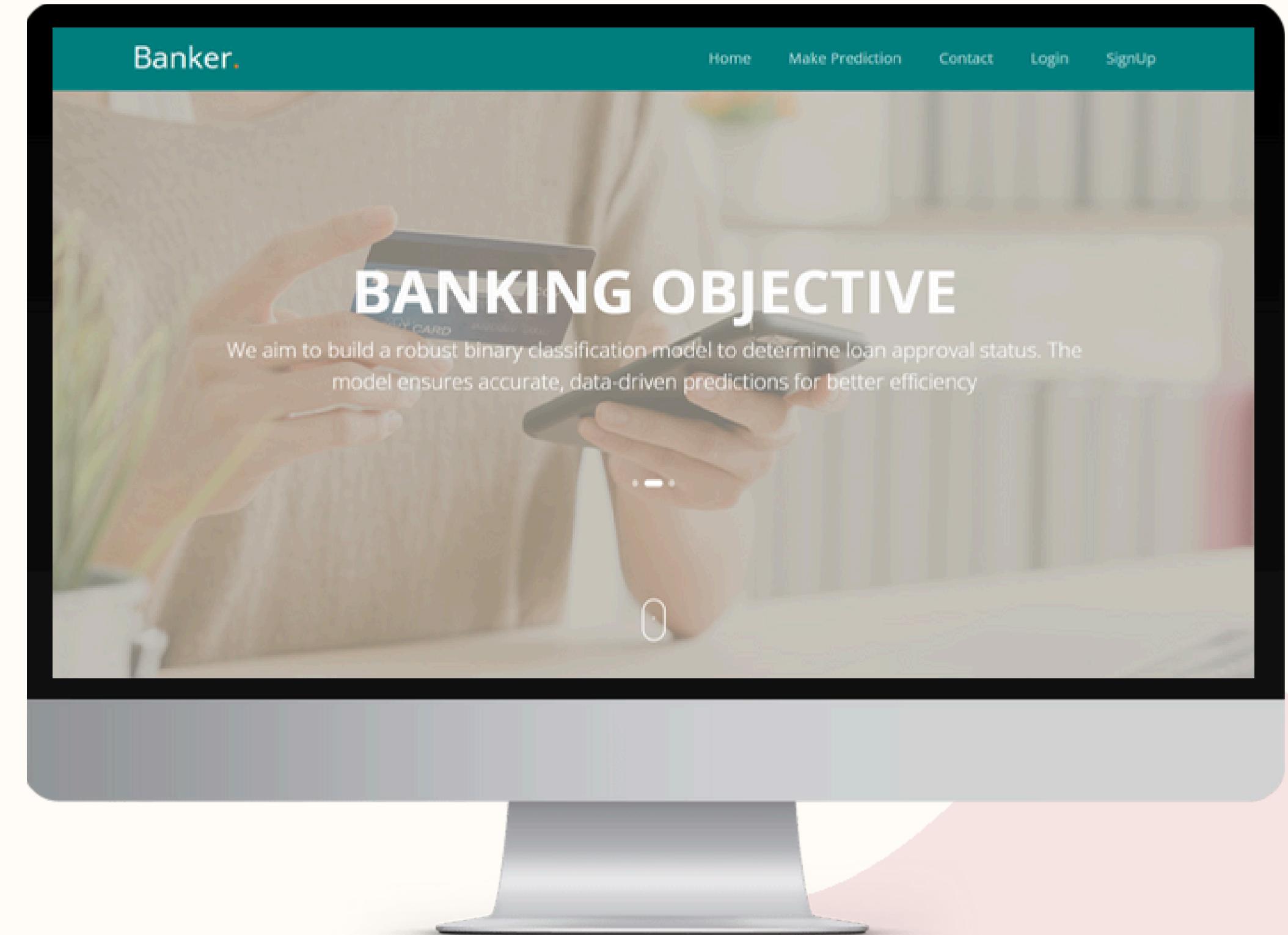
```
from pyspark.sql import SparkSession

# Création de la session Spark
spark = SparkSession.builder.appName("DataLoading").getOrCreate()

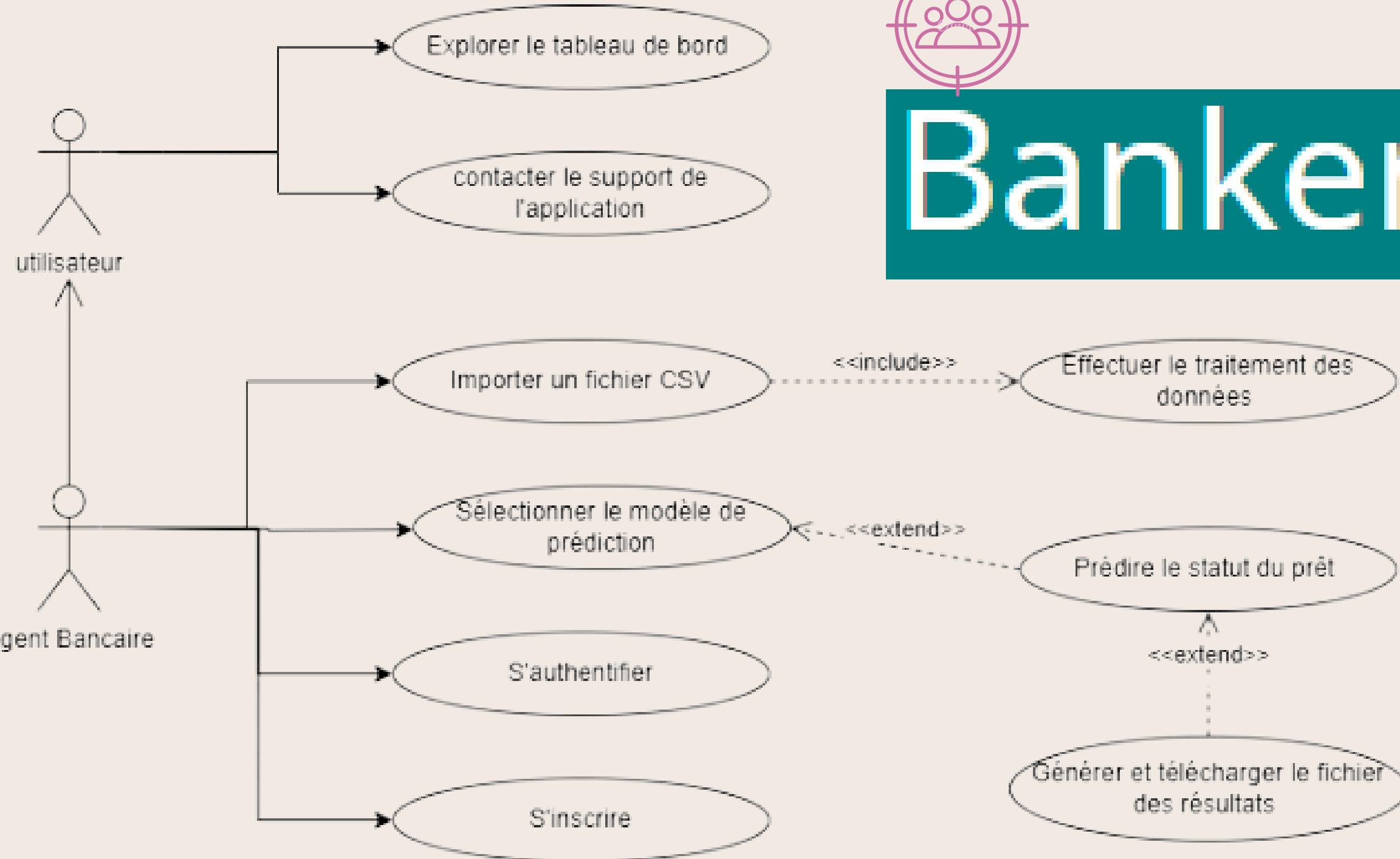
# Chargement des données depuis un fichier CSV
spark_df = spark.read.csv('data.csv', header=True, inferSchema=True)

# Affichage des premières lignes
spark_df.show()
```

Notre Application “Banker”

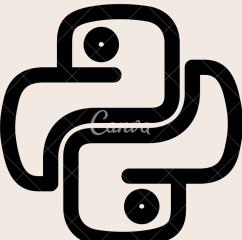


Public Cible

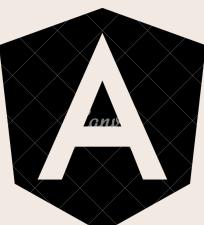


Développement

Partie Backend : Flask



Partie FrontEnd : Angular

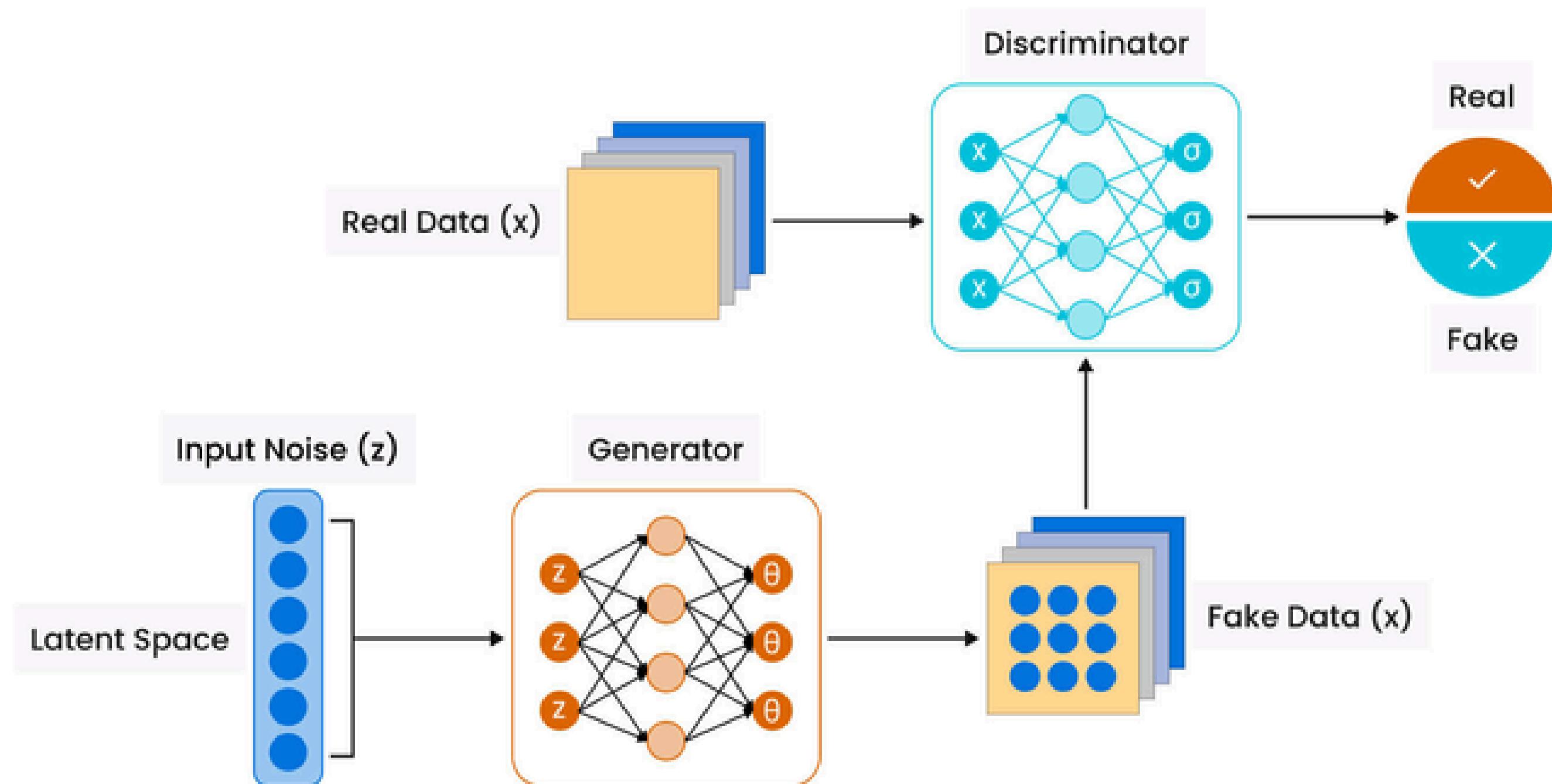


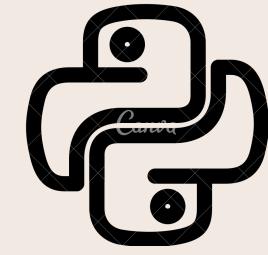
Objectifs

- **Prédiction de prêt :** Offrir une plateforme permettant aux utilisateurs de prédire le statut des prêts en utilisant des modèles de machine learning.
1/ Importer des données clients stockées dans un CSV
2/ Fournir une fonctionnalité permettant aux utilisateurs et agents bancaires de générer et télécharger un fichier CSV contenant les résultats des prédictions.

Génération des données avec les Réseaux Antagonistes Génératifs (GANs)

Generative Adversarial Network (GAN)





Partie Backend : Flask

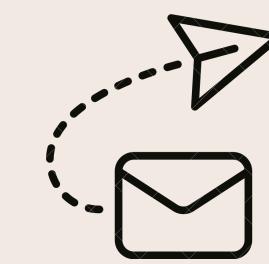
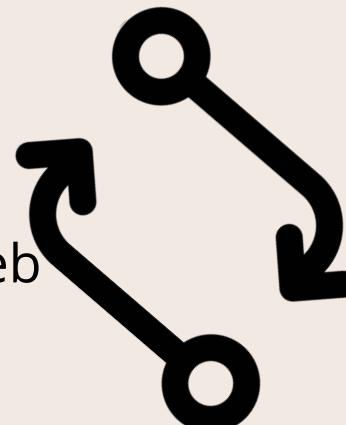
- **Flask:** Le framework principal pour créer l'API web
- **CORS:** Permet d'autoriser les requêtes entre domaines différents
- Les modèles sont stockés dans un dictionnaire, où chaque clé correspond à un type de modèle. Ce chargement est effectué une seule fois lors du démarrage de l'application.

• Fonction de prétraitement des données:

Binary Encoding /Ordinal Encoding/One-Hot Encoding/
Ajout de colonnes manquantes/Réorganisation des colonnes

• Route pour la prédiction : /predict:

1. Vérification du fichier
2. Chargement du fichier CSV
3. Vérification du modèle choisi
4. Prétraitement des données
5. Normalisation des données
6. Prédiction
7. Enregistrement des résultats
8. Envoi du fichier de résultats

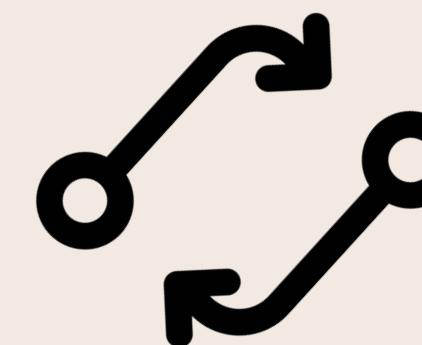


API (service):

PredictionService:

Service créé pour gérer l'envoi du fichier et du modèle à l'API backend

HttpClient

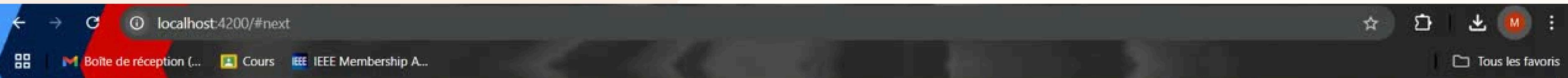


Partie FrontEnd : Angular

- **Génération des Composants:**
 1. Home
 2. Prediction
 3. login
 4. signup
 5. contact
- **déclaration des routes pour chaque composant:**
Dans le fichier "app-routing.module.ts"
- **définir le user interface pour chaque composant:**
Dans le fichier "Nom du composant.HTML"
- **définir les méthodes nécessaires pour chaque composant:**
Dans le fichier "Nom du composant.ts"
- **Appel du service PredictionService dans le fichier "prediction.ts":**
Pour assurer l'envoie des requêtes et les réponses entre le BackEnd et le FrontEnd



Démo



Banker.

Home

Make Prediction

Contact

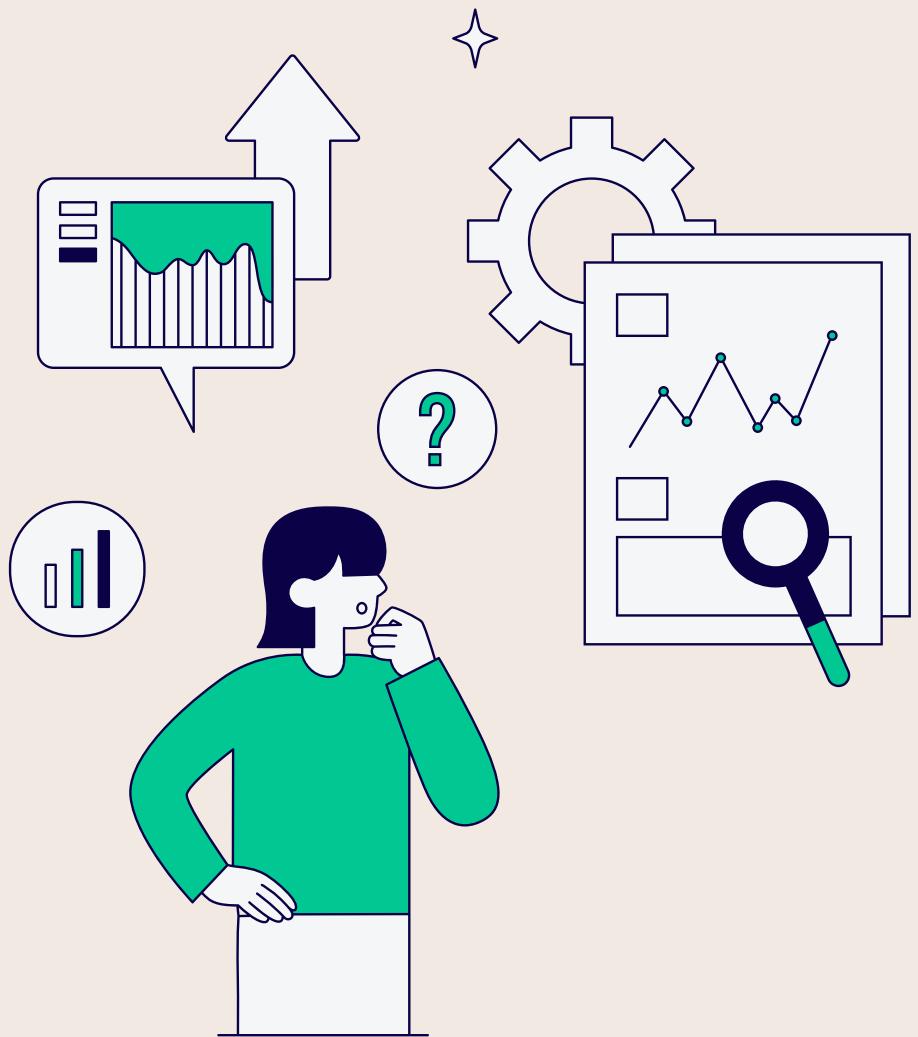
Login

SignUp

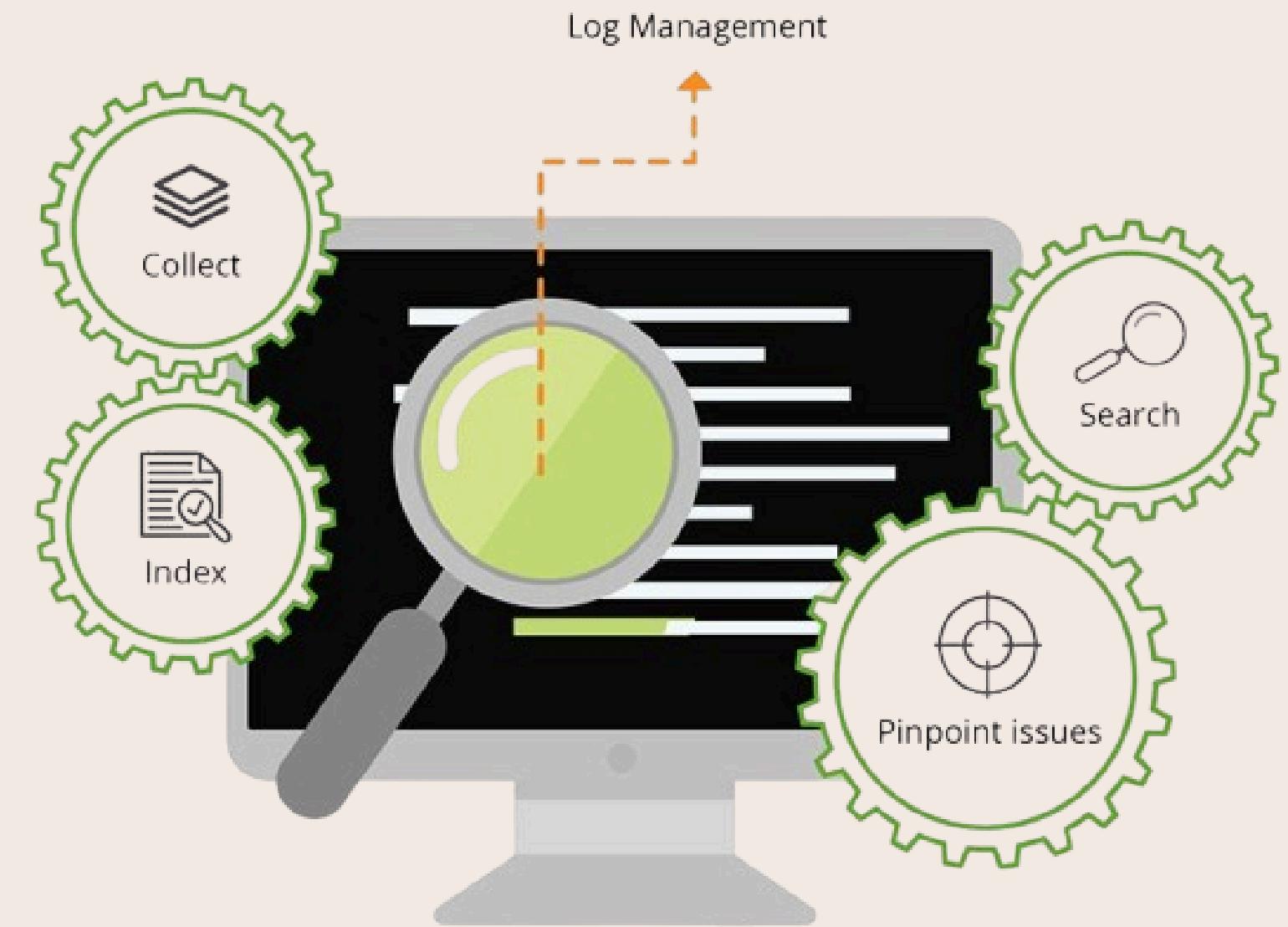
DATA DRIVEN LENDING
CARD
Our mission is to harness big data for smarter financial systems. By enabling transparent and fair loan decisions, we drive innovation in the lending process.

Stockage et Visualisation avec l'ELK Stack

Problématique



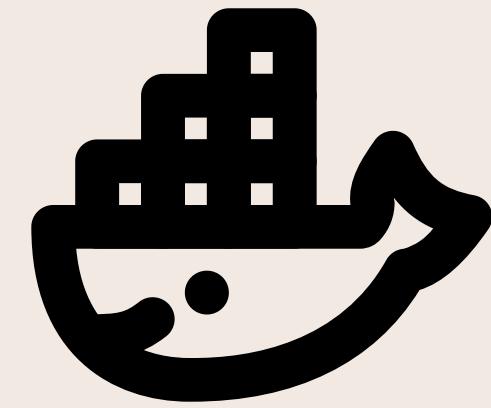
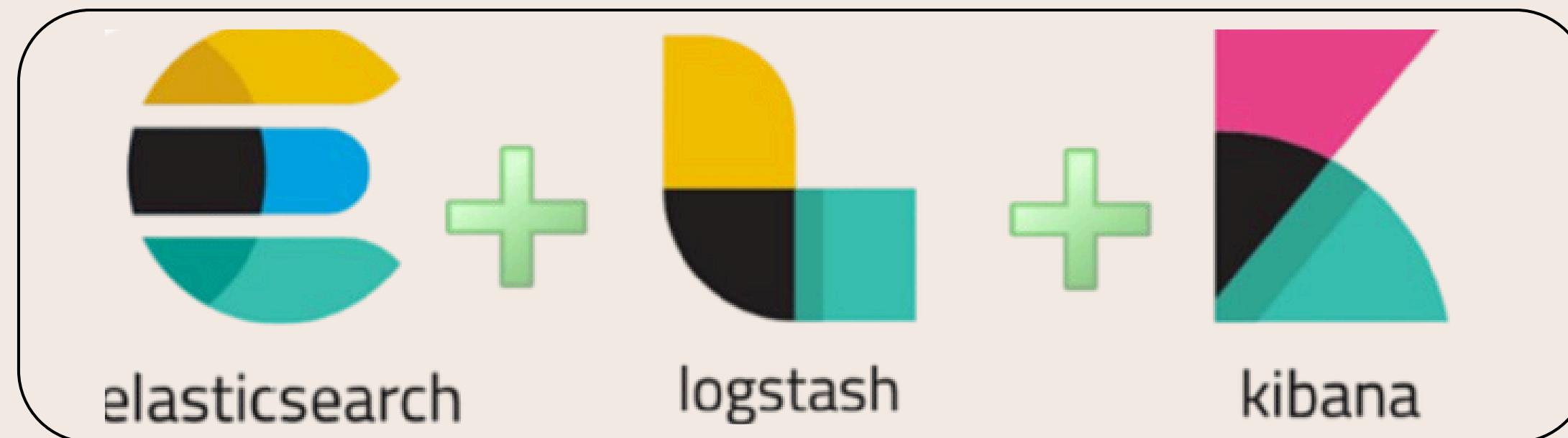
la complexité de la surveillance et de la gestion des prêts de manière claire.



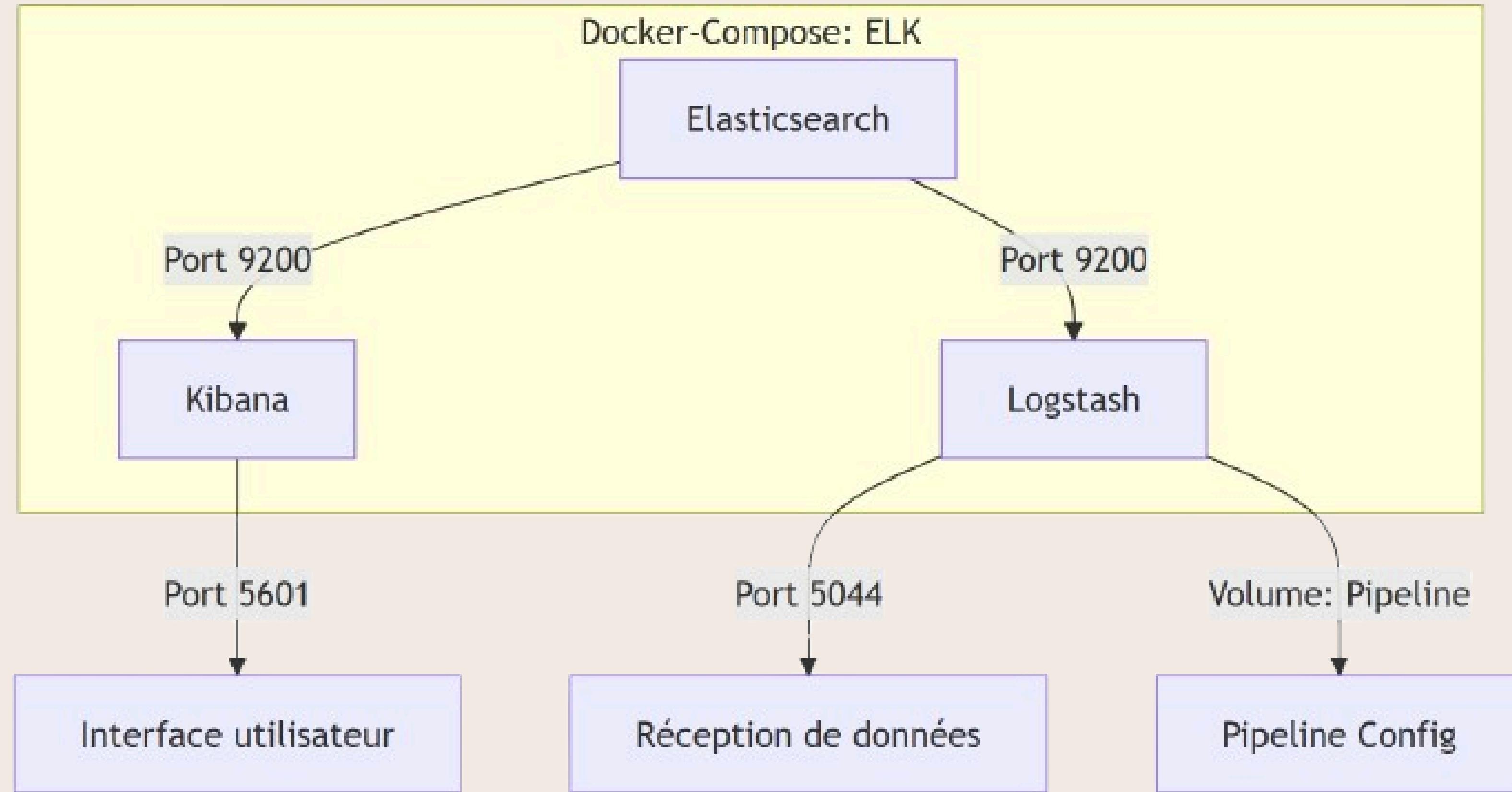
une solution qui garantisse une visualisation claire pour les gestionnaires de prêts.

Solution

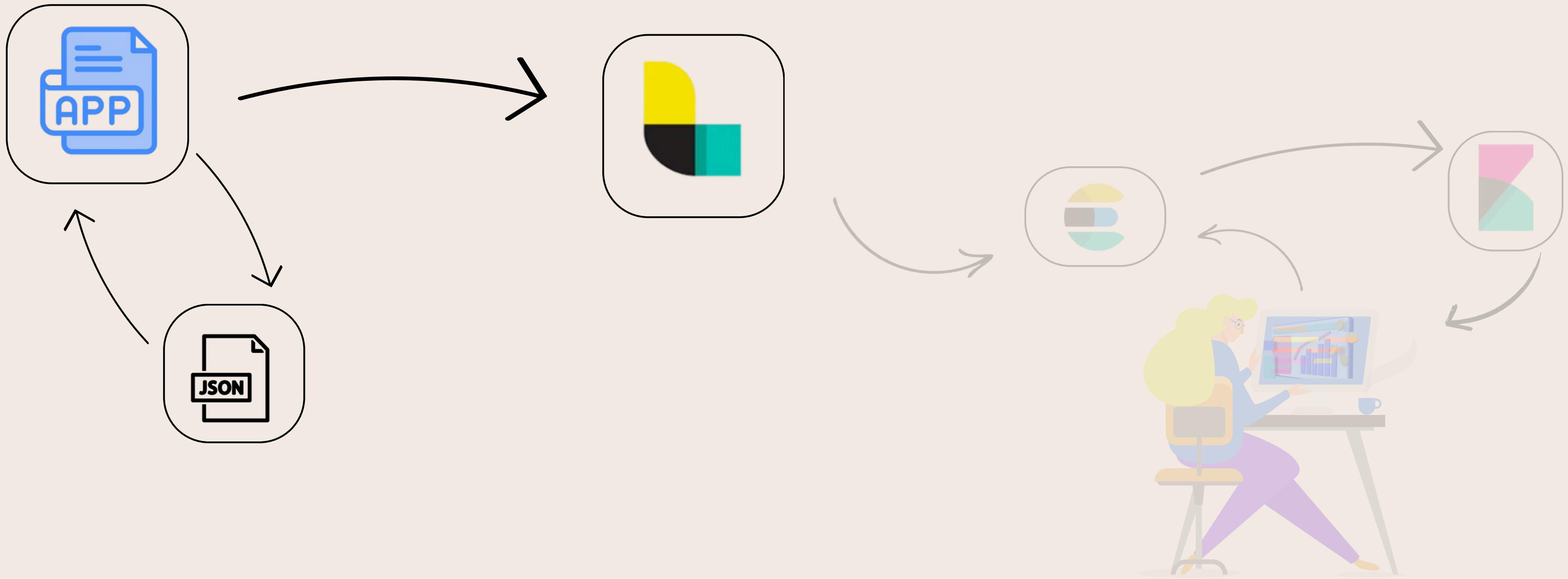
Solution ELK



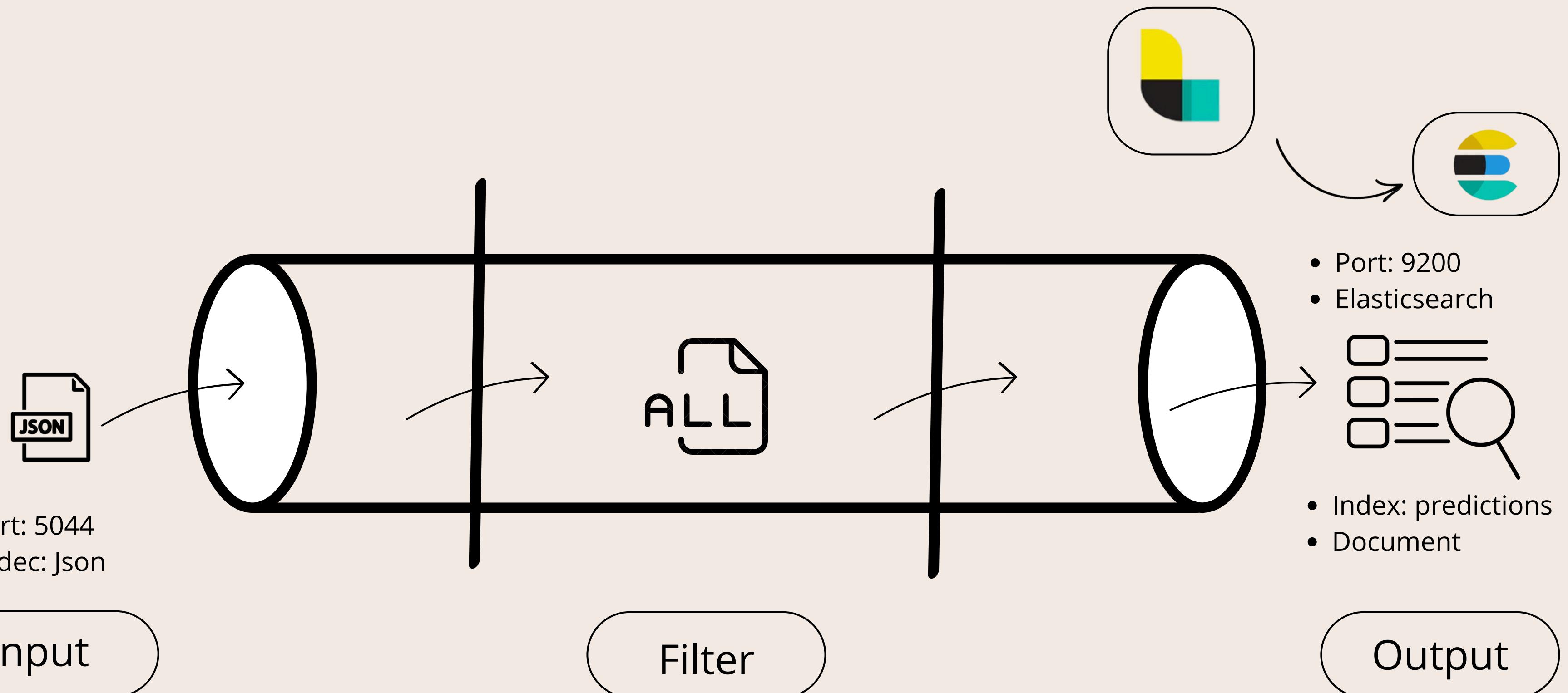
Configuration du stack ELK avec Docker Compose



Transmission des données avec Logstash



Configuration de la pipeline Logstash

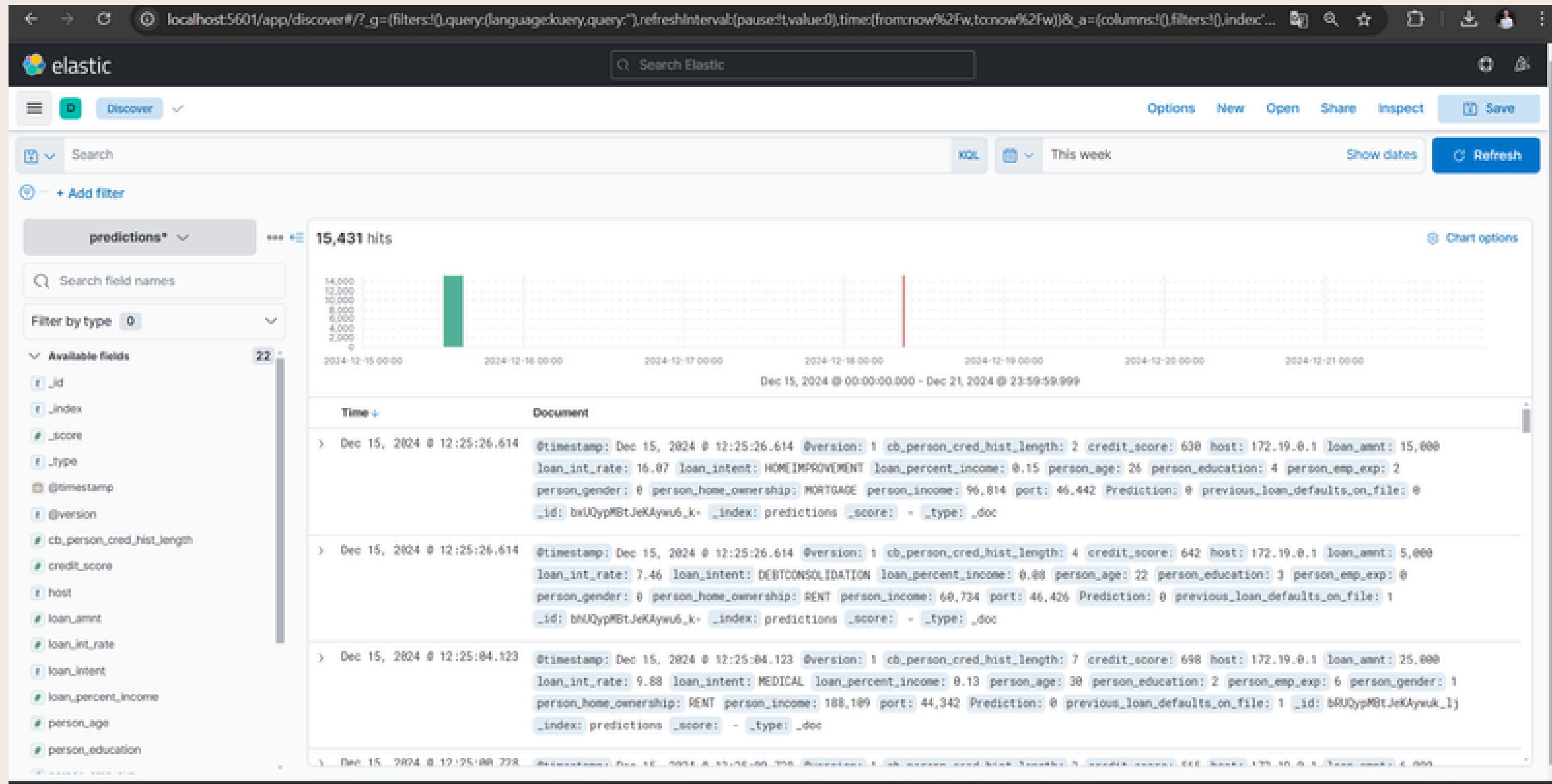


Input

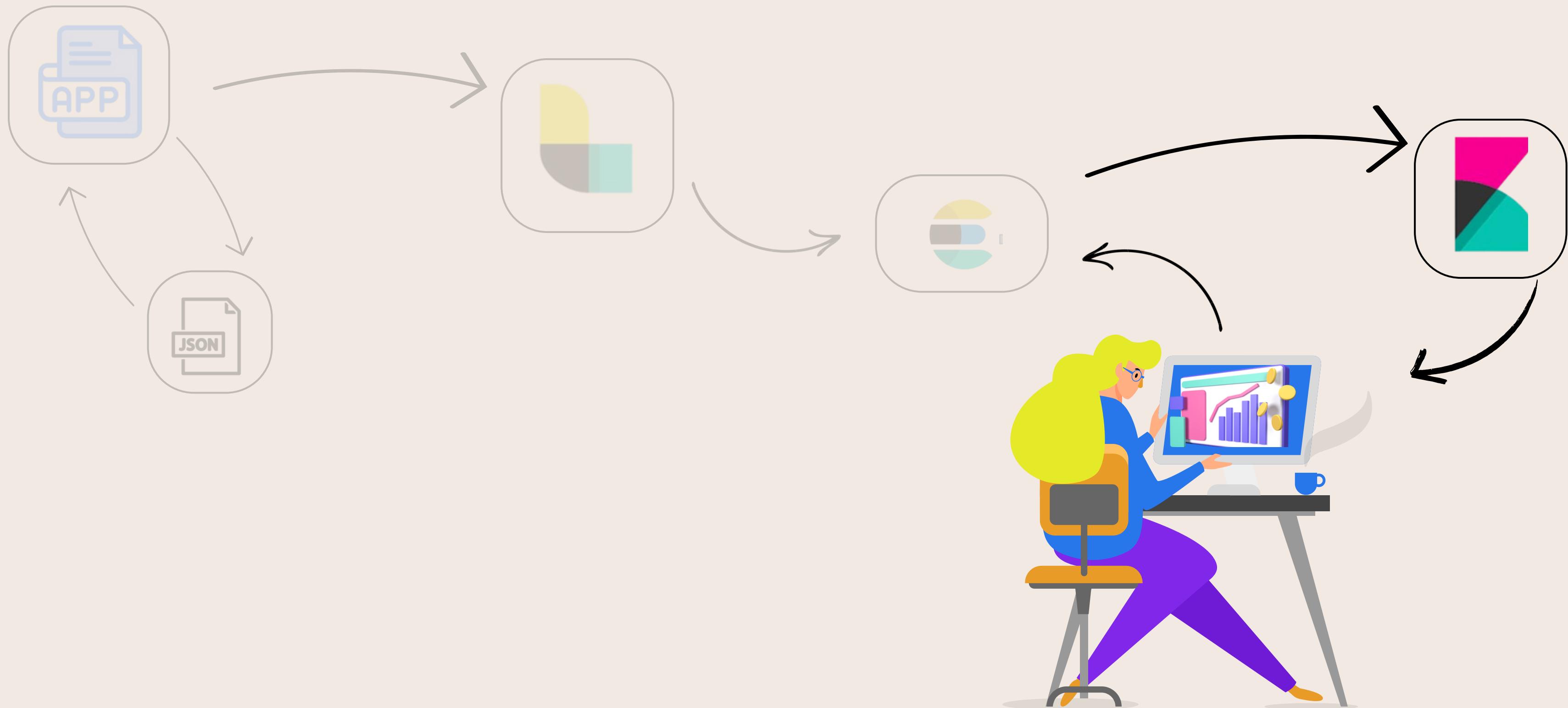
Filter

Output

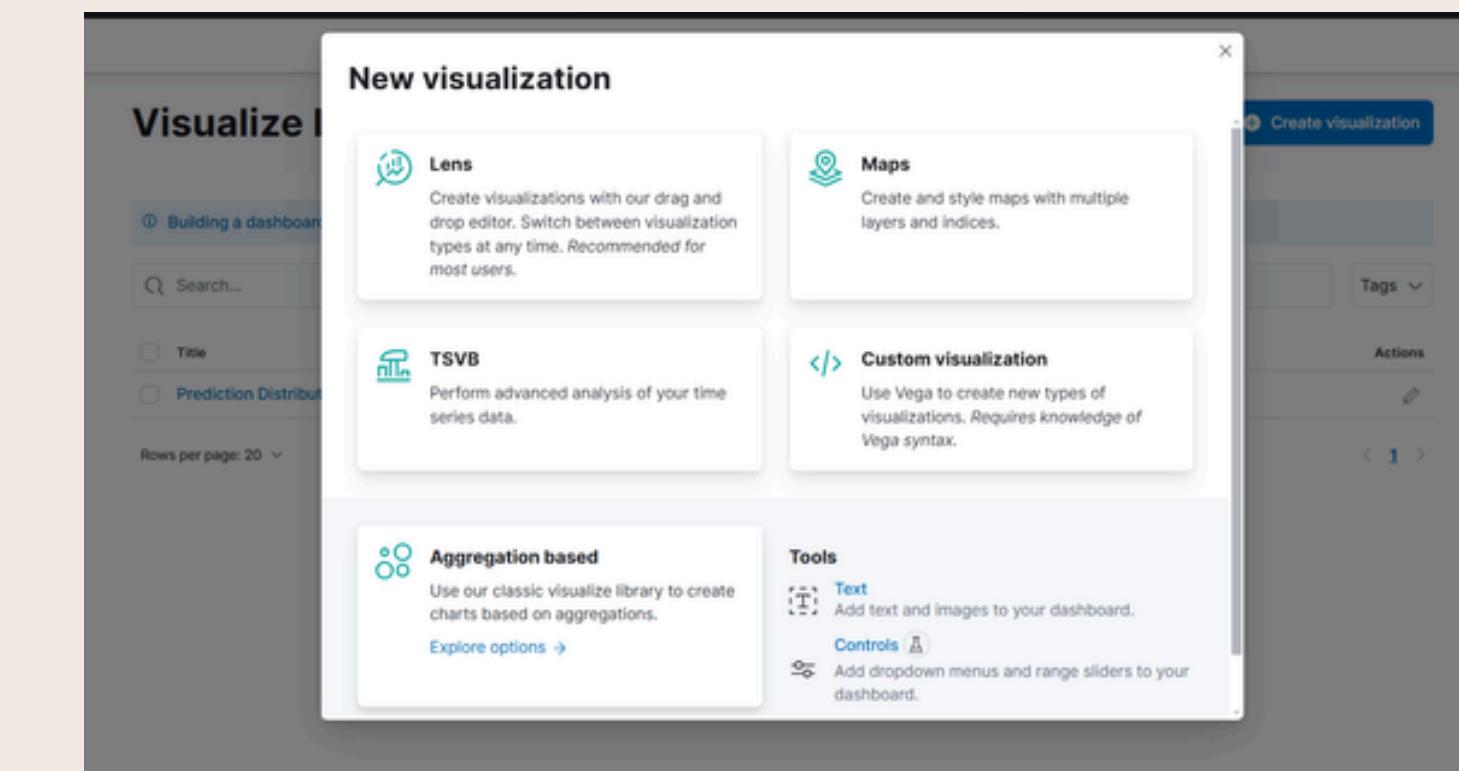
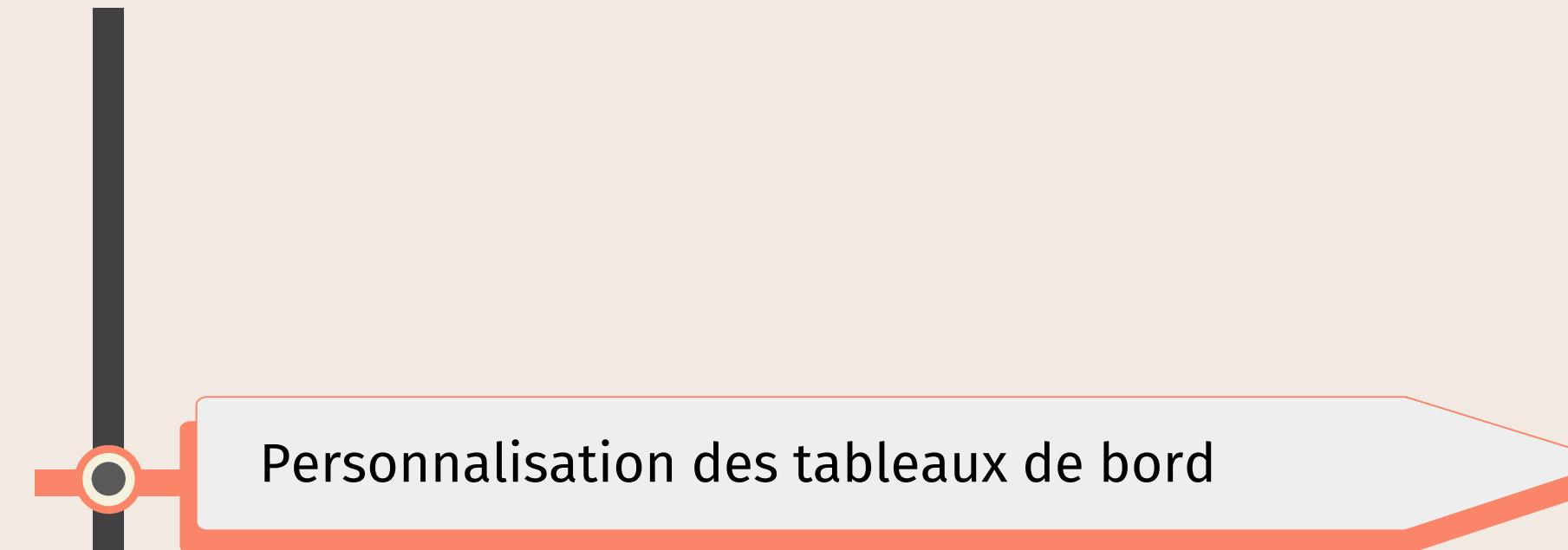
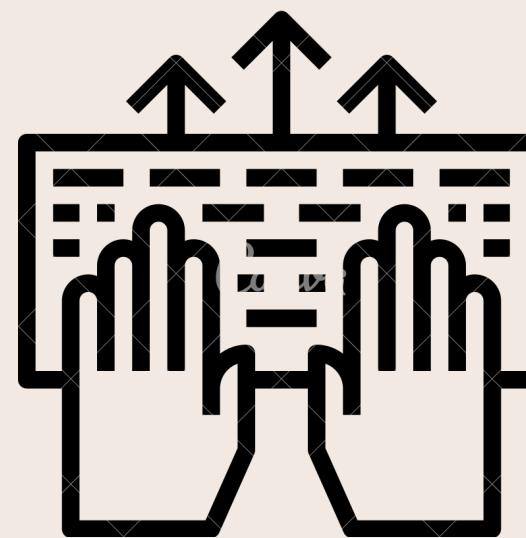
Visualisation des documents avec (l'indice prédition)



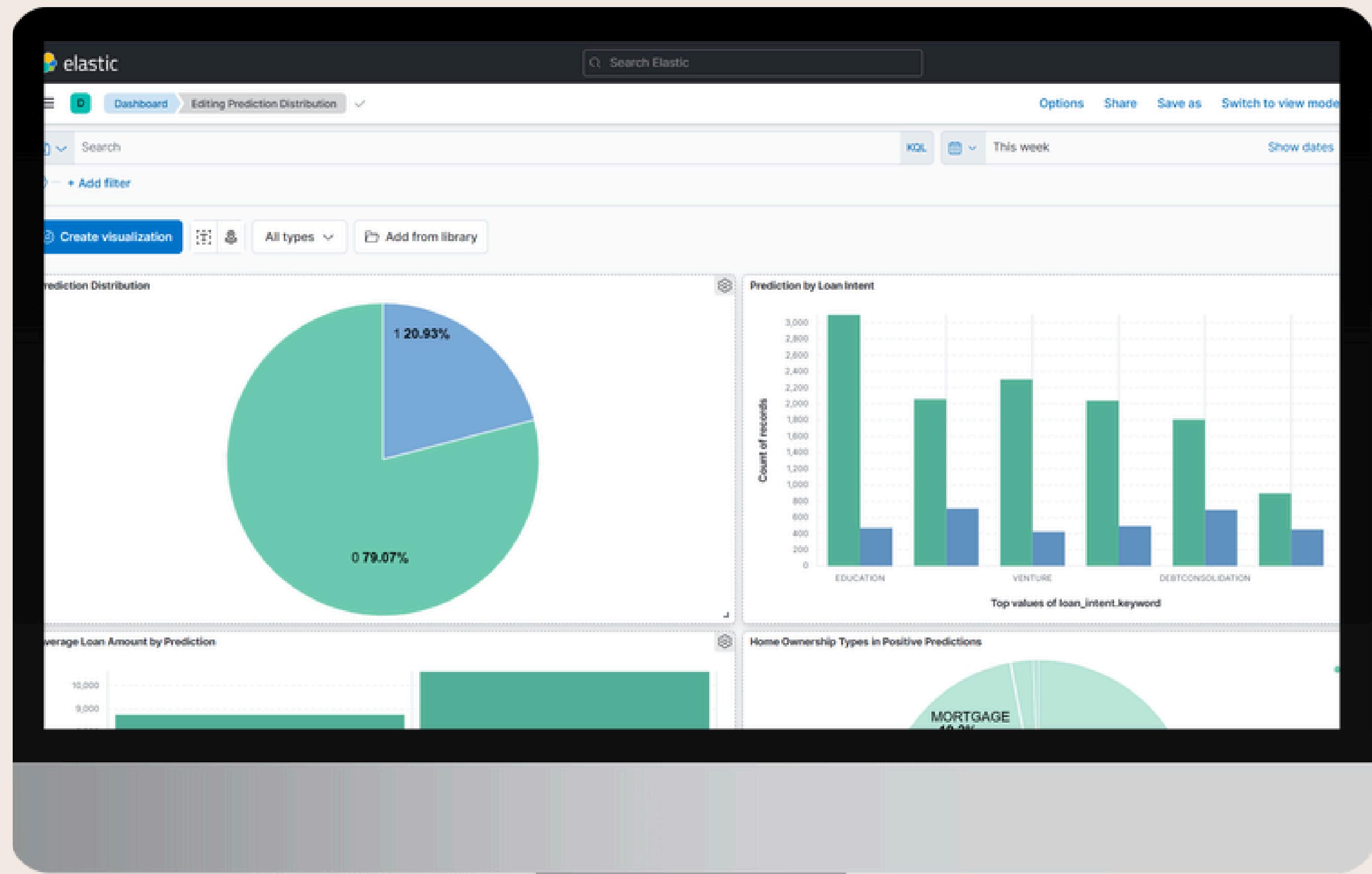
Visualisation avec Kibana



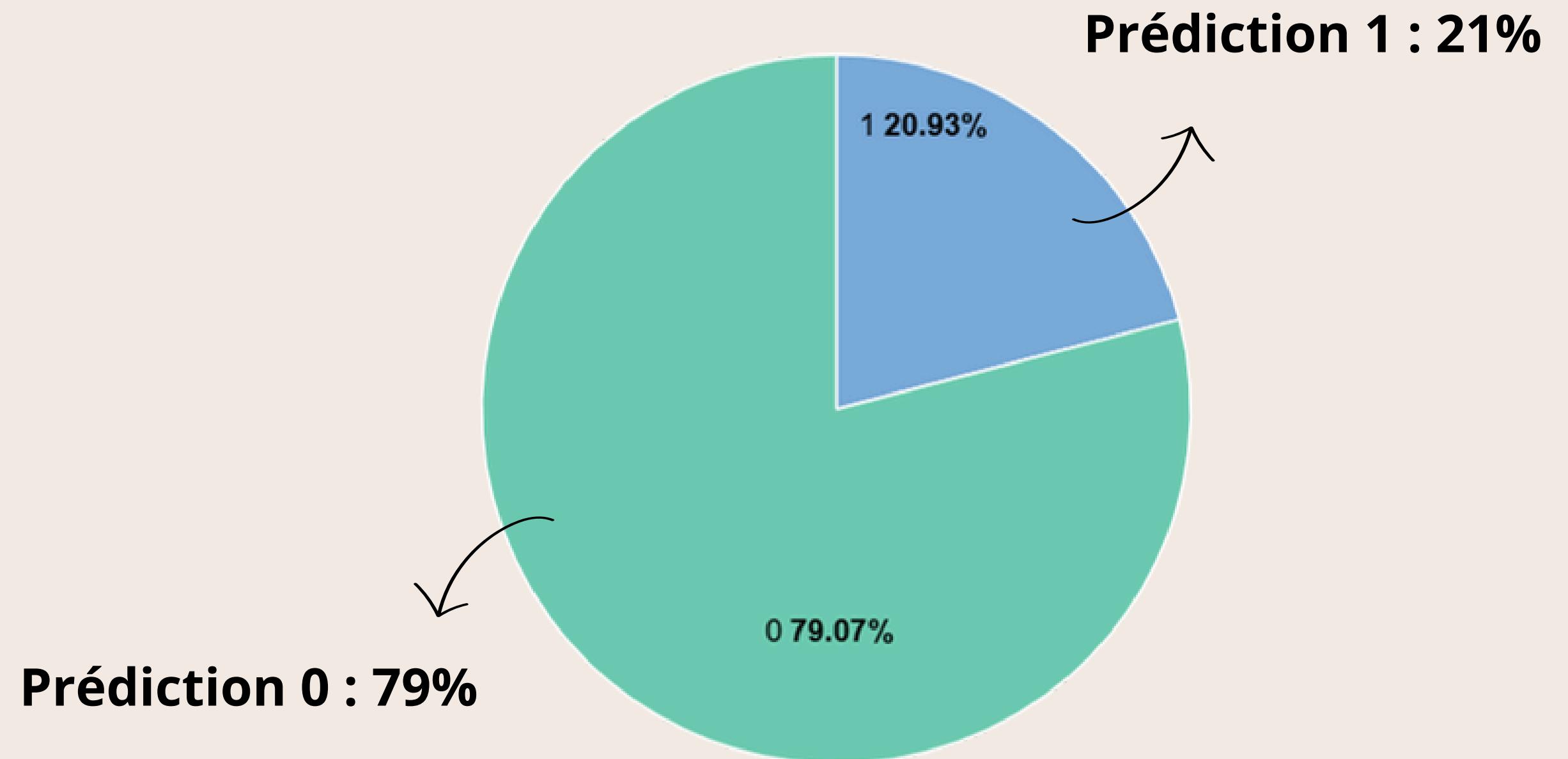
Création de tableaux de bord



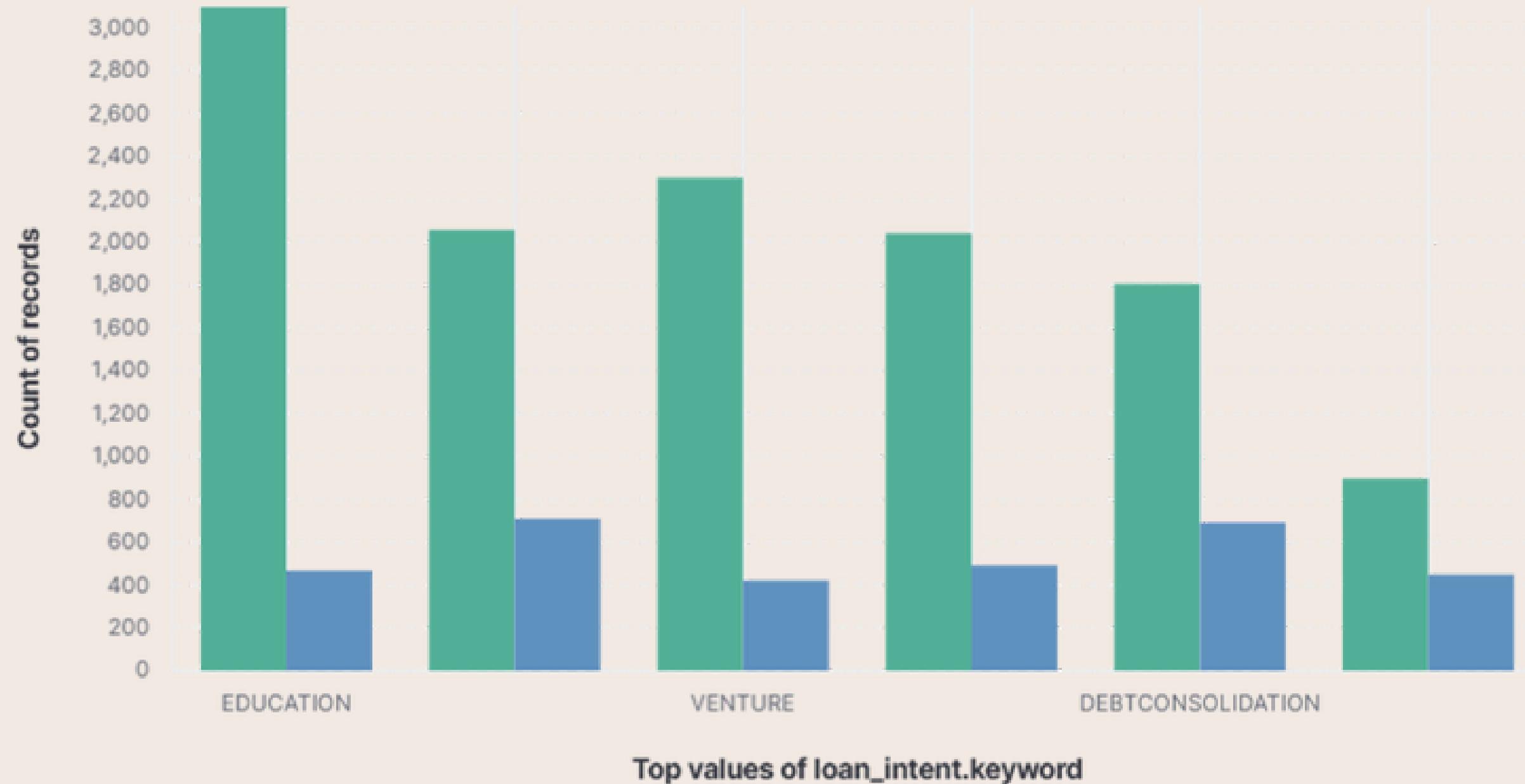
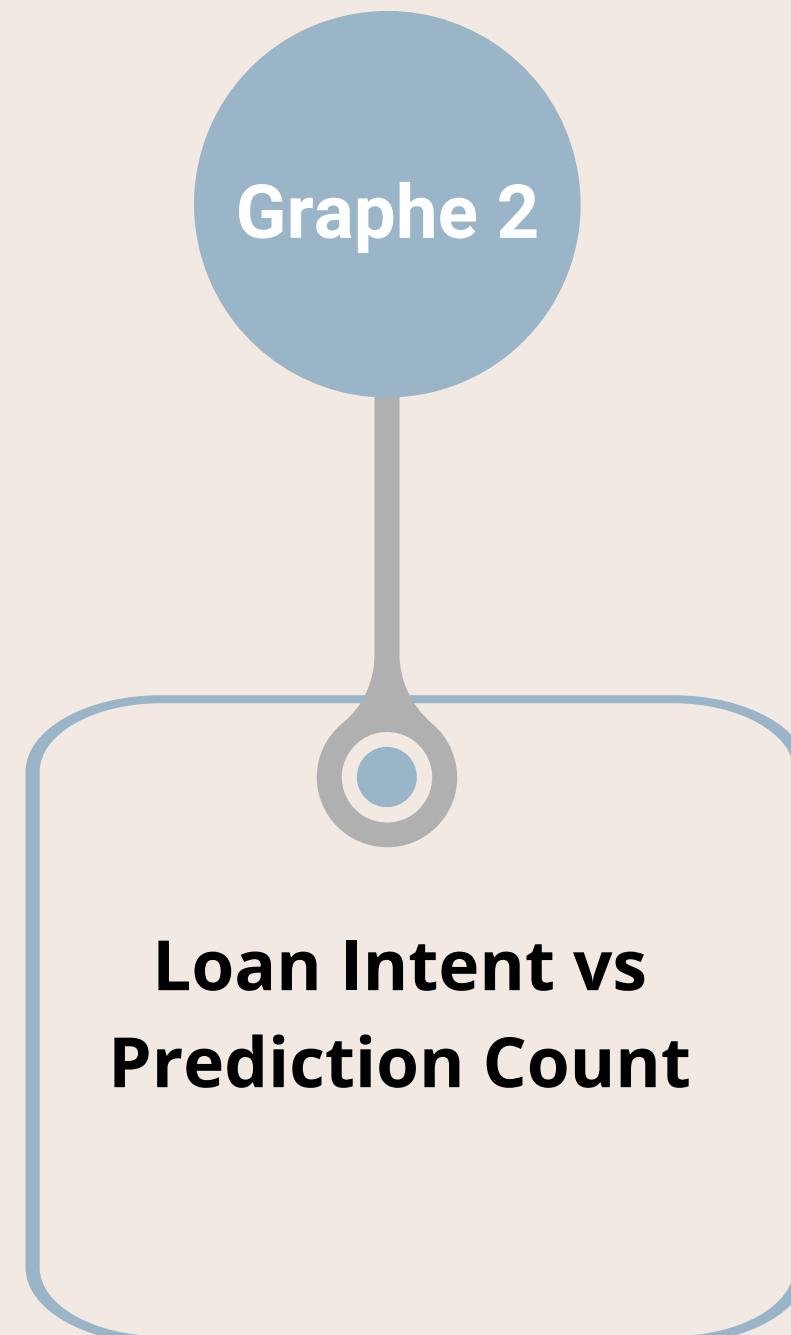
Visualisation avec Kibana



Comprendre la proportion des prédictions positives (1) et négatives (0).

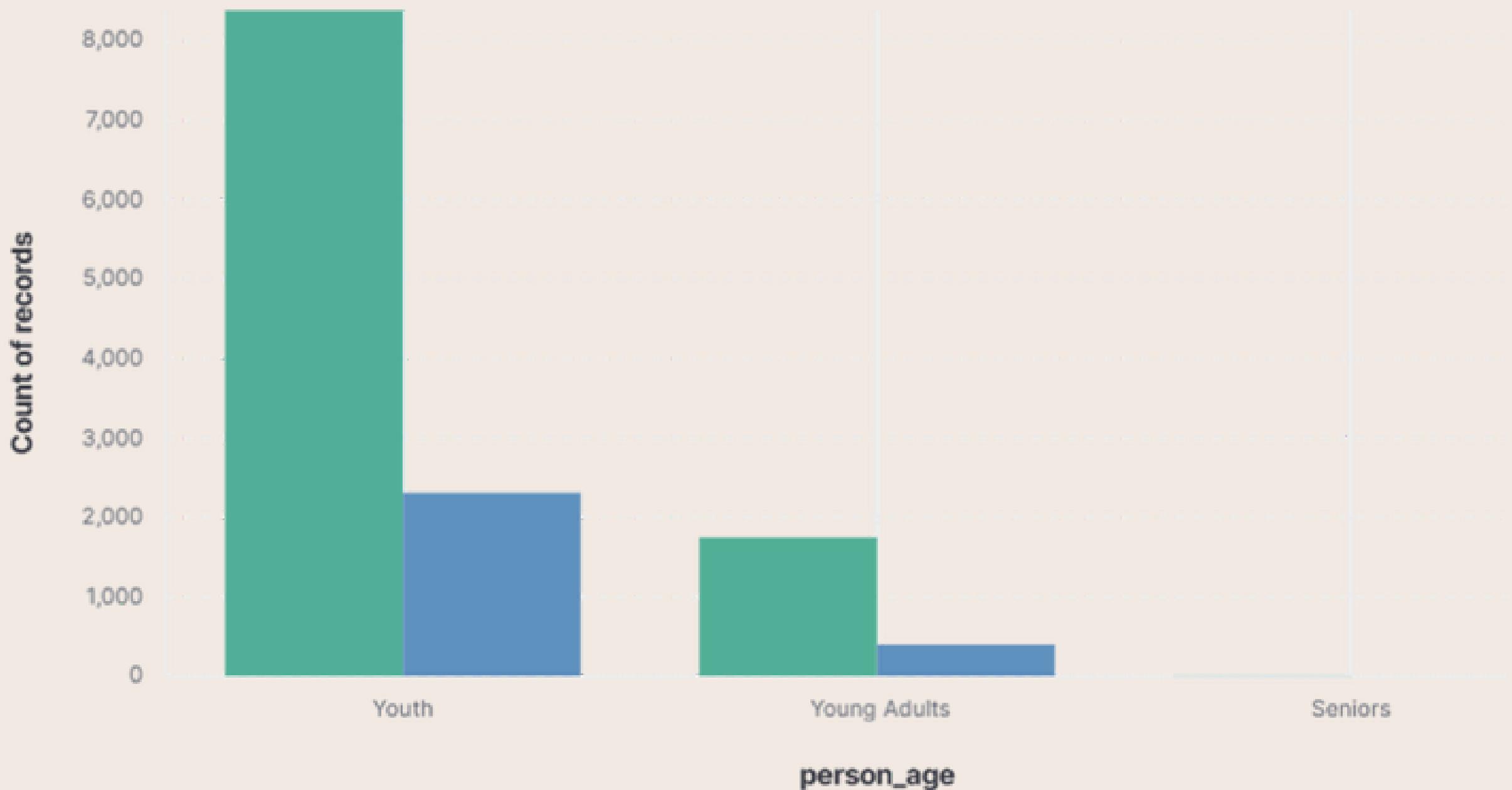


Analyser quelles intentions de prêt ont le plus grand nombre de prédictions positives (1).



Visualisation avec Kibana

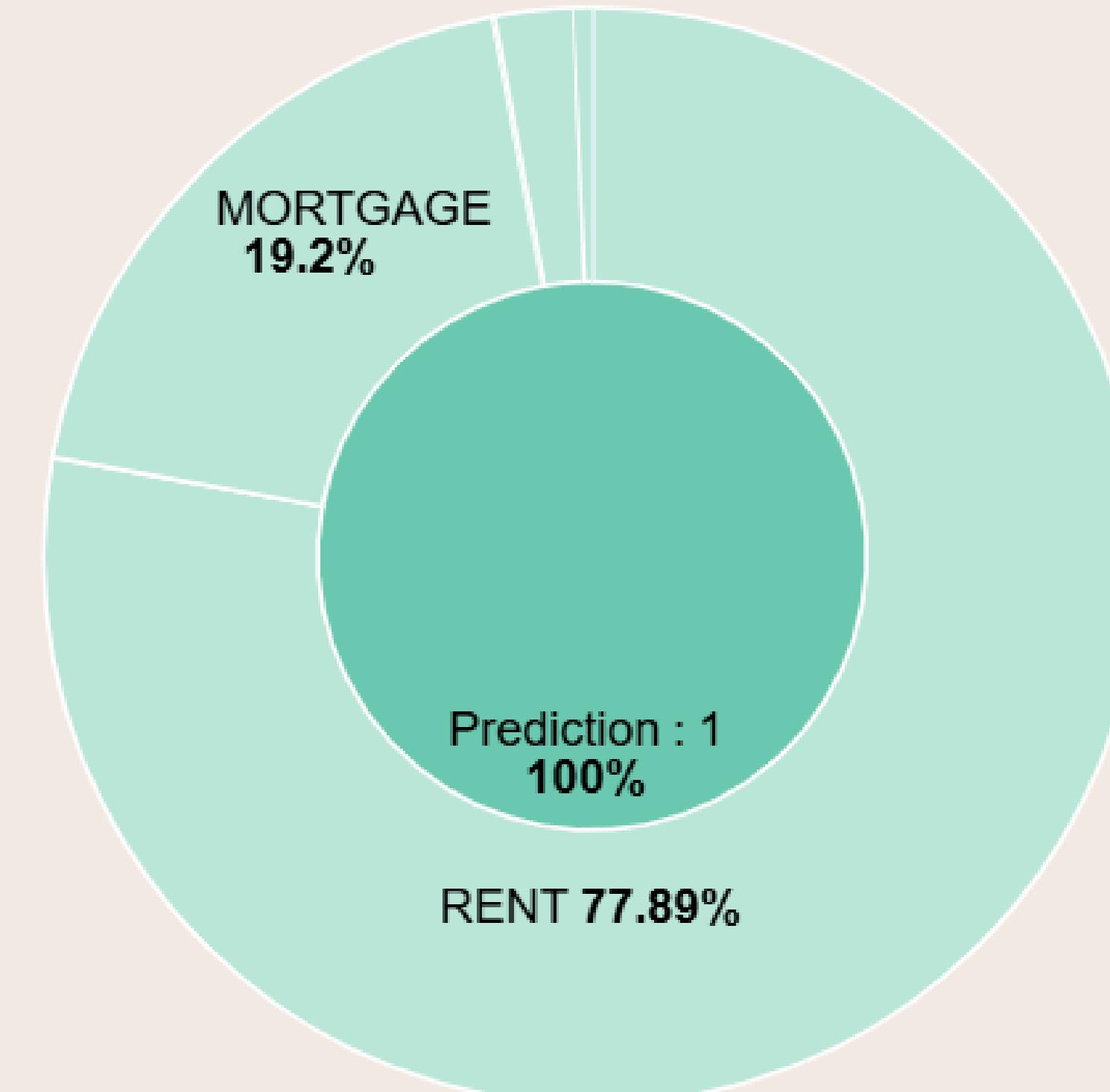
Comprendre comment les prédictions varient selon les groupes d'âge



Graphe 4

Home Ownership
Types in Positive
Predictions

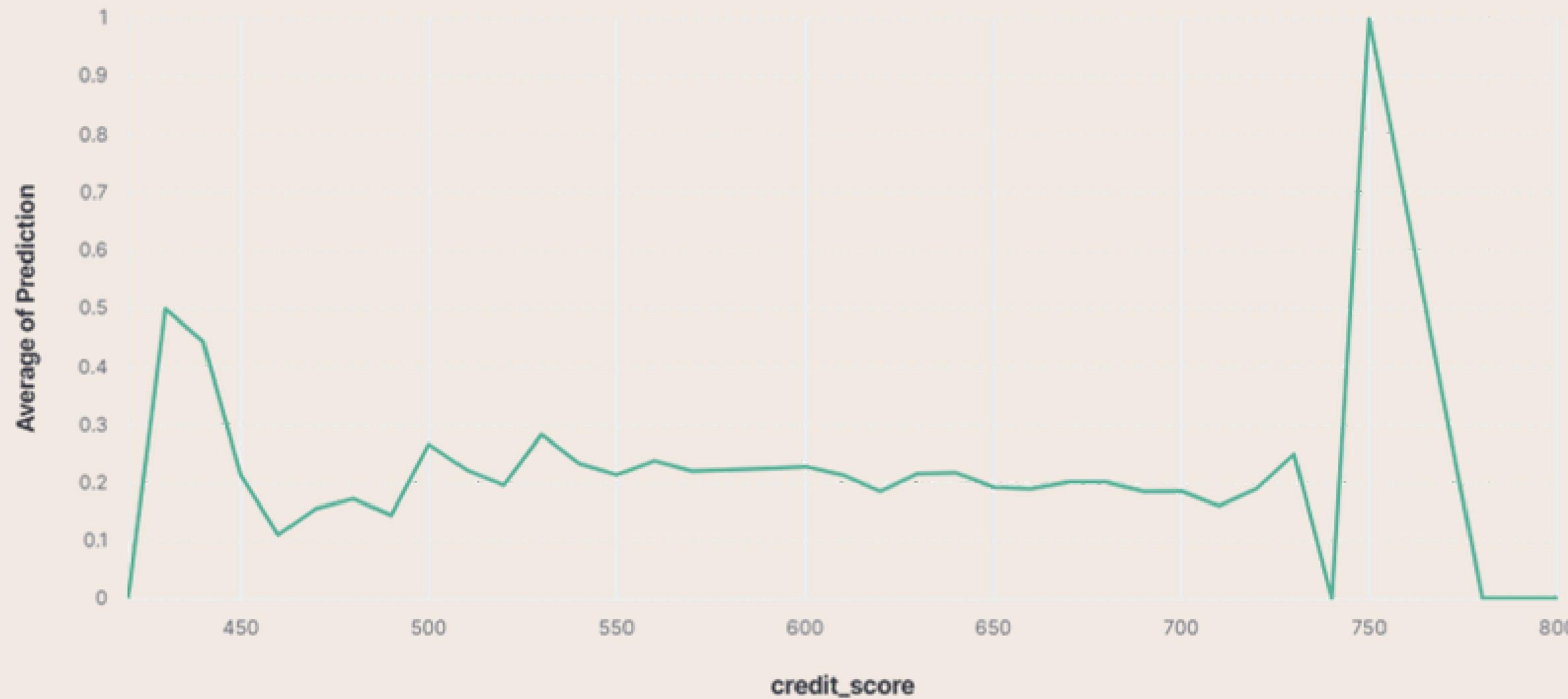
Identifier le type de logement dans les prédictions positives.



Graphe 5

Credit Score vs
Prediction

Analyser comment les prédictions varient en fonction du score de crédit.



Conclusion

Conclusion



- Automatisation de processus d'attribution des prêts.
- Visualisation et suivi en temps réel des données.



**MERCI DE VOTRE
ATTENTION**



**MERCI DE VOTRE
ATTENTION**