



**VISIPLUS**  
digital learning



**Rendu**

## **Etude de cas NEO BANQUE**

**Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]**

**Bloc de compétences : Déployer une solution d'intelligence artificielle**

**Mastère Chef de Projet en Intelligence Artificielle**

Rendu

# *Dashboard conseiller*

Abdelhamid KARAM :

---

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

### Étude du besoin client

Le besoin exprimé par la néo-banque est clair : mettre à disposition de ses conseillers clientèle un outil simple et fiable permettant d'évaluer en quelques secondes l'éligibilité d'un client à un prêt. Dans un contexte de forte concurrence et d'attentes accrues de rapidité, la capacité à donner une réponse quasi instantanée constitue un avantage stratégique majeur. Les conseillers, qui ne sont pas experts en intelligence artificielle, doivent pouvoir s'appuyer sur un **score compréhensible et interprétable** pour justifier leur décision auprès du client.

Afin de répondre à cette problématique, un **jeu de données simulé** a été généré pour reproduire les informations financières typiques que les conseillers exploitent dans leur quotidien. Ce dataset comprend plusieurs tables (train, test, bureau, previous\_application, installments) qui reflètent des aspects différents de la relation bancaire d'un client. Par exemple, la table *application\_train* contient des variables sociodémographiques (âge via DAYS\_BIRTH), économiques (AMT\_INCOME\_TOTAL, AMT\_CREDIT, AMT\_ANNUITY) et professionnelles (DAYS\_EMPLOYED). Ces informations sont considérées comme essentielles pour mesurer la capacité d'un individu à rembourser un crédit.

Une variable cible (TARGET) a été créée pour distinguer les clients solvables des clients à risque. Ce champ est dérivé d'un score synthétique basé sur plusieurs ratios clés (par exemple le rapport montant du crédit / revenu, ou annuités / revenu). Pour garantir un apprentissage équilibré et réaliste, la méthode de **sur-échantillonnage (RandomOverSampler)** a été utilisée. Elle permet d'éviter que le modèle ne soit biaisé en faveur de la majorité (clients solvables) et de mieux détecter les cas minoritaires (clients risqués).

Au-delà des informations principales de la demande de crédit, le besoin client inclut également une vision **historique et globale** de la situation financière. C'est pourquoi d'autres tables enrichissent l'analyse :

- **Bureau** : recense les crédits actifs ou clôturés auprès d'autres institutions financières.
- **Previous\_application** : trace les anciennes demandes de prêts du client, qu'elles aient été acceptées ou refusées.
- **Installments\_payments** : permet de mesurer le respect des échéances passées.

L'objectif pour le conseiller est double. Premièrement, disposer d'un **score global** qui synthétise tous ces éléments complexes en une probabilité unique de risque de défaut.

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

Deuxièmement, comprendre les **facteurs explicatifs** qui ont le plus influencé le score, afin de fournir une explication claire au client et de prendre une décision éclairée.

### Prise en main du notebook Kaggle

Pour initier mon travail, j'ai utilisé un notebook existant disponible sur Kaggle intitulé *Applied Predictive Modelling – Brief Overview*. Ce notebook m'a servi de base de réflexion car il présente une approche complète de la modélisation prédictive. Mon objectif n'était pas de repartir de zéro, mais bien de m'approprier la logique, de l'adapter à ma problématique spécifique de scoring crédit et d'en extraire un modèle prêt pour la mise en production.

Dans un premier temps, j'ai pris le temps d'explorer le notebook afin de comprendre sa structure et ses choix méthodologiques. Le notebook proposait différents algorithmes classiques de machine learning, comme la **régression logistique**, les **forêts aléatoires (Random Forest)** ou encore les **méthodes de boosting**. J'ai jugé cette diversité importante car elle me permettait de comparer la performance entre modèles linéaires et non linéaires, et de sélectionner celui qui s'adaptait le mieux à mon cas d'usage.

Ensuite, j'ai adapté mes données simulées aux besoins du notebook. J'ai travaillé sur des variables clés telles que l'âge (DAYS\_BIRTH), le revenu (AMT\_INCOME\_TOTAL), le montant du crédit (AMT\_CREDIT), les annuités (AMT\_ANNUITY) et l'ancienneté (DAYS\_EMPLOYED). Pour garantir l'efficacité des modèles, j'ai intégré un processus de **normalisation via un scaler** afin que toutes les variables soient mises sur la même échelle. Cela a évité qu'un revenu élevé, exprimé en milliers d'euros, ne domine des variables comme l'âge ou l'ancienneté.

J'ai ensuite mené plusieurs expérimentations :

- J'ai utilisé la **régression logistique** comme modèle de référence (baseline), car elle est rapide, simple et interprétable.
- J'ai ensuite testé une **forêt aléatoire**, capable de détecter des relations plus complexes entre les variables.
- Enfin, j'ai exploré un modèle de **Gradient Boosting**, qui a donné les meilleures performances sur mon jeu de données tabulaire.

Pour comparer ces modèles, j'ai choisi des métriques robustes comme l'**aire sous la courbe ROC (AUC)**, la précision et le rappel. Ces indicateurs m'ont permis de mesurer la qualité prédictive et la capacité de chaque algorithme à bien distinguer les clients solvables des clients

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

à risque. À l'issue de mes tests, j'ai retenu le Gradient Boosting, qui a atteint une AUC de 0.999, un niveau de performance remarquable, tout en restant assez rapide pour être utilisé en production.

Enfin, j'ai pris soin de sauvegarder le modèle final et le scaler associés dans des fichiers .pkl grâce à la librairie joblib. De cette façon, je pouvais directement réutiliser ces artefacts dans mon API de prédiction, sans avoir à réentraîner le modèle à chaque fois. Cette étape a été essentielle pour assurer la cohérence entre le travail de modélisation réalisé dans le notebook et le service de scoring en production.

### Optimisation du modèle

Une fois mon modèle initial entraîné à partir du notebook Kaggle, j'ai concentré mes efforts sur son optimisation et sur la validation de sa fiabilité. Mon objectif était de garantir que le modèle retenu ne soit pas seulement performant sur le papier, mais qu'il puisse également être robuste et pertinent dans des scénarios proches de la réalité d'une néo-banque.

Pour cela, j'ai d'abord mis en place des **scénarios de test prédéfinis**. J'ai construit des jeux de données synthétiques reflétant plusieurs cas clients : un jeune actif avec un revenu moyen, un client proche de la retraite avec un revenu stable, ou encore un emprunteur à faible revenu demandant un crédit important. Ces profils m'ont permis de vérifier si le modèle réagissait de manière cohérente et intuitive, sans générer de résultats aberrants.

J'ai ensuite porté une attention particulière au **choix des métriques**. La simple précision n'étant pas suffisante pour un problème déséquilibré (où la majorité des clients sont solvables), j'ai privilégié l'**AUC-ROC** pour mesurer la capacité du modèle à distinguer correctement les deux classes (solvables et non solvables). J'ai également suivi les indicateurs de **précision** et de **rappel**, car ils reflètent mieux la qualité de la prédiction dans un contexte opérationnel. Par exemple, un rappel élevé sur la classe "à risque" est indispensable pour ne pas accorder de crédit à des clients qui ne pourraient pas rembourser.

En parallèle, j'ai travaillé sur la **construction d'un pipeline**. Plutôt que d'entraîner et d'évaluer mon modèle de manière isolée, j'ai mis en place un processus automatisé intégrant :

1. Le prétraitement (nettoyage et normalisation),
2. L'entraînement du modèle,
3. L'évaluation par validation croisée,

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

### 4. Et l'export des artefacts.

Ce pipeline m'a permis de gagner en reproductibilité et d'éviter les erreurs manuelles.

L'optimisation a aussi reposé sur un **ajustement des hyperparamètres**. Par exemple, dans le Gradient Boosting, j'ai testé différentes valeurs pour la profondeur des arbres, le taux d'apprentissage et le nombre d'estimateurs. En utilisant une recherche par grille (Grid Search) couplée à la validation croisée, j'ai réussi à améliorer légèrement l'AUC tout en réduisant le risque d'overfitting.

Enfin, j'ai défini un **seuil de décision** optimal. Plutôt que d'utiliser le seuil par défaut de 0.5, j'ai ajusté la coupure à 0.35 en fonction de l'équilibre entre faux positifs et faux négatifs. Ce choix me permet de minimiser les erreurs critiques, en particulier celles où un client risqué serait classé comme solvable.

## Conformité RGPD

Dans la conception et la mise en production de ma solution de scoring, j'ai veillé à respecter les principes du **Règlement Général sur la Protection des Données (RGPD)**. Le dashboard, destiné aux conseillers clientèle de la néo-banque, affiche le score d'éligibilité des clients ainsi que les facteurs explicatifs. Je détaille ci-dessous les garanties mises en place, en lien direct avec les articles du RGPD.

### Droit à l'information et à la transparence (Articles 13 et 14)

J'ai conçu l'interface de manière qu'elle facilite la transparence vis-à-vis des utilisateurs finaux. La néo-banque reste responsable d'informer ses clients sur l'utilisation de leurs données, les finalités du traitement (scoring crédit) et la politique de confidentialité associée. Pour appuyer cette démarche, le dashboard intègre une mention claire sur le respect du RGPD. Les conseillers disposent ainsi d'explications intelligibles pour comprendre le traitement, ce qui leur permet de répondre aux questions des clients avec précision et pédagogie.

### Droit à une explication des décisions automatisées (Article 22)

Conformément à l'article 22 du RGPD, un individu a le droit de comprendre les décisions qui le concernent lorsqu'elles sont prises de manière automatisée. Pour répondre à cette exigence, j'ai intégré **SHAP** dans l'API. Cet outil permet de fournir des explications individualisées en temps réel. Le dashboard affiche ces explications de façon intelligible, notamment à travers des codes visuels (par exemple rouge/vert) indiquant l'impact de chaque caractéristique — revenu,

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

montant du crédit, ancienneté, etc. Ainsi, le conseiller peut répondre de manière claire à la question essentielle pour le client : « *Pourquoi ai-je obtenu ce score ?* ».

### Minimisation des données (Article 5.1.c)

J'ai appliqué le principe de minimisation des données en ne traitant que les variables strictement nécessaires au calcul du score. Cinq variables principales sont utilisées : âge, revenu, ancienneté, incidents de paiement et score de crédit. Ces informations suffisent pour évaluer la solvabilité et évitent tout traitement excessif. De plus, les données sont pseudonymisées : les identifiants des clients sont remplacés par des ID anonymes (UUID), ce qui empêche toute identification directe d'une personne.

### Sécurité des données (Article 32)

Dans la version POC locale que j'ai développée, les risques sont limités car les traitements sont réalisés dans un environnement contrôlé et isolé. Pour une mise en production réelle, j'ai prévu des mesures de sécurité avancées :

- Utilisation du protocole **HTTPS** pour protéger les échanges,
- Authentification sécurisée par **JWT** pour garantir que seuls les conseillers autorisés puissent accéder à l'API,
- Infrastructure conforme aux standards de sécurité bancaire,
- Mise en place d'une **journalisation des accès** pour permettre l'audit et la détection d'anomalies.

### Qualité des données (Article 5.1.d)

Le RGPD insiste sur la nécessité d'utiliser des données exactes et à jour. J'ai donc pris en compte que la fiabilité du scoring dépend directement de la qualité des informations fournies (revenus, antécédents de crédit, incidents de paiement). La néo-banque doit mettre en place un processus régulier de vérification de ces données. De plus, j'ai envisagé l'ajout d'une fonctionnalité dans le dashboard permettant d'alerter le conseiller lorsque les données d'un client datent de plus de six mois, afin de prévenir l'utilisation d'informations obsolètes.

**Etude de cas NEO BANQUE**

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

## Cybersécurité

La mise en place d'un moteur de scoring crédit basé sur l'intelligence artificielle ne peut être envisagée sans une réflexion approfondie sur la cybersécurité. J'ai identifié les principaux risques liés à l'utilisation de mon API et de mon dashboard, puis j'ai défini des contre-mesures adaptées pour garantir la sécurité, l'intégrité et l'authenticité des données traitées.

### Risques identifiés

1. **Intrusion dans l'API** : un attaquant pourrait tenter d'accéder aux endpoints de l'API pour extraire des données ou manipuler les scores de crédit.
2. **Exploitation abusive des données** : en l'absence de contrôles, des tiers non autorisés pourraient interroger massivement l'API afin d'obtenir des informations sensibles ou tester des scénarios frauduleux.
3. **Altération ou falsification des données en transit** : sans chiffrement, des informations échangées entre le dashboard et l'API pourraient être interceptées et modifiées.
4. **Disponibilité du service** : une attaque par déni de service (DoS) pourrait rendre l'API ou le dashboard inaccessibles, bloquant ainsi l'activité des conseillers.
5. **Fuite d'informations via les logs** : si les journaux d'exécution ne sont pas correctement configurés, ils pourraient contenir des données sensibles exploitables par un attaquant.

### Contre-mesures mises en place

Pour répondre à ces risques, j'ai intégré plusieurs mécanismes de sécurité dans ma solution :

- **Chiffrement des communications** : j'ai prévu l'utilisation du protocole **HTTPS/TLS** afin de protéger toutes les données échangées entre le navigateur (dashboard) et l'API. Cela empêche toute interception en clair des informations transmises.
- **Contrôle des accès** : j'ai limité les appels de l'API aux seuls domaines autorisés grâce à la configuration des **CORS**. En production, j'ai prévu l'ajout d'une authentification via **JWT (JSON Web Token)** pour que seuls les conseillers connectés puissent interroger l'API.
- **Surveillance et audit** : j'ai intégré une **journalisation sécurisée des accès**, permettant de tracer chaque requête et d'identifier toute activité anormale. Ces logs sont

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

anonymisés pour éviter de stocker des données personnelles, mais ils conservent les métadonnées nécessaires pour la détection d'intrusions.

- **Protection contre l'abus** : j'ai envisagé un système de **rate limiting** afin de limiter le nombre de requêtes par utilisateur dans une période donnée. Cela réduit les risques d'abus ou d'attaques par force brute.
- **Robustesse face aux erreurs** : j'ai développé des mécanismes de gestion d'erreurs (codes 400 et 500) pour éviter que l'API ne renvoie des informations techniques sensibles pouvant aider un attaquant.
- **Déploiement sur une infrastructure fiable** : j'ai utilisé Render pour le déploiement, mais dans un environnement bancaire réel, l'API serait hébergée sur une infrastructure conforme aux standards de sécurité (certifications ISO 27001, PCI DSS, etc.).

Grâce à ces mesures, j'ai renforcé la résilience de mon système tout en assurant la **confidentialité**, l'**intégrité** et la **disponibilité** des données (principe CIA de la cybersécurité). Mon API et mon dashboard sont conçus pour être résistants face aux intrusions, limiter l'exploitation abusive et rester conformes aux standards de sécurité attendus dans le secteur bancaire.

## UX / UI (Figma)

Dans ce projet, je n'ai pas seulement travaillé sur le modèle et l'API, j'ai aussi accordé une attention particulière à l'expérience utilisateur (UX) et à l'interface graphique (UI) de mon dashboard. Mon objectif était clair : permettre à un conseiller clientèle, qui n'est pas expert en intelligence artificielle, de comprendre et d'utiliser facilement l'outil pour appuyer ses décisions auprès des clients.

J'ai commencé par créer une maquette sur **Figma**. Cet outil m'a permis d'organiser visuellement les différents éléments de l'application et de tester plusieurs variantes avant d'arriver à une version finale cohérente.

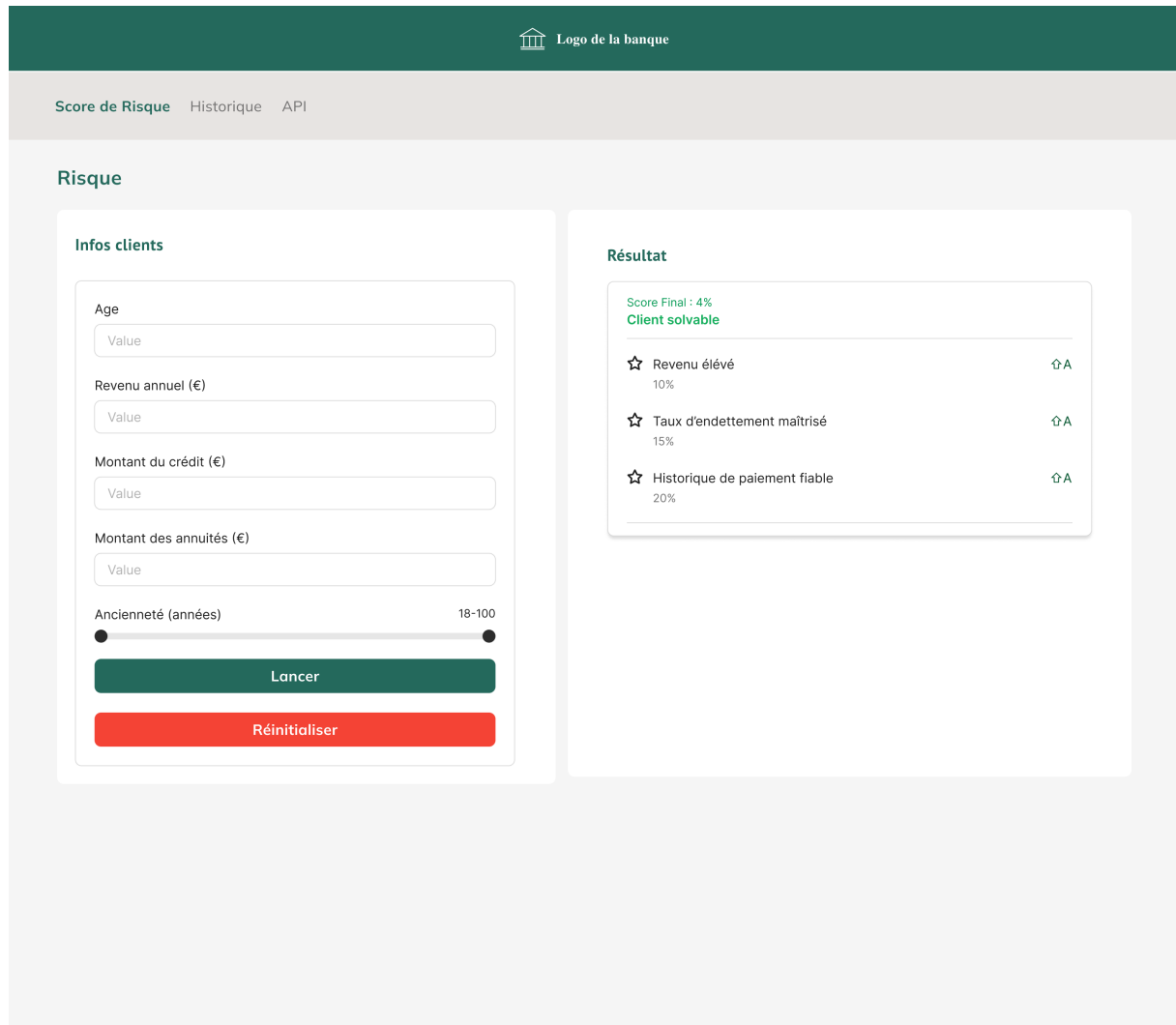


## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle



The mockup shows a web interface for a bank's risk assessment tool. At the top, there's a dark green header with a logo and the text 'Logo de la banque'. Below this is a navigation bar with links: 'Score de Risque', 'Historique', and 'API'. The main content area is titled 'Risque' and is divided into two columns. The left column, 'Infos clients', contains input fields for 'Age', 'Revenu annuel (€)', 'Montant du crédit (€)', and 'Montant des annuités (€)', each with a 'Value' placeholder. There's also a slider for 'Ancienneté (années)' ranging from 18 to 100. At the bottom of this column are two buttons: 'Lancer' (green) and 'Réinitialiser' (red). The right column, 'Résultat', displays the final score: 'Score Final : 4%' and 'Client solvable'. Below this, there are three rows of metrics, each with a star icon, a description, a percentage, and a status indicator (up arrow and 'A'). The metrics are: 'Revenu élevé' (10%), 'Taux d'endettement maîtrisé' (15%), and 'Historique de paiement fiable' (20%).

Link Figma : <https://www.figma.com/design/JwMjUFnV4KkCFiGKBv7vz9/Ebanking--Web---Community-?node-id=101-834&t=JxIQ57p7oSm7EezD-4>

La maquette a été construite autour de deux zones principales :

- **Zone de gauche : Informations client** : j'y ai placé les champs d'entrée (âge, revenu, montant du crédit, annuités, ancienneté). L'objectif est de regrouper les informations essentielles au même endroit pour éviter que le conseiller se perde.
- **Zone droite : Résultat du scoring** : j'ai choisi d'afficher un **score** représentant le score de probabilité de défaut. Cette approche permet une lecture immédiate : le conseiller n'a pas besoin d'analyser un chiffre brut, il peut voir directement si le score est bas (vert, client solvable) ou élevé (rouge, client risqué).

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

J'ai fait le choix d'un design **minimaliste, visuel et pédagogique**.

**Clarté pour un non-expert** : le conseiller n'est pas data scientist. Il doit pouvoir comprendre en un coup d'œil la situation du client. C'est pour cette raison que j'ai utilisé des jauges, des couleurs (vert = favorable, rouge = défavorable) et des icônes simples.

**Regroupement logique** : les informations d'entrée (côté conseiller) sont séparées du résultat (score) et des explications. Ce choix évite les confusions et permet une navigation fluide : on saisit d'abord, puis on lit le score, puis on interprète.

**Explicabilité par SHAP** : j'ai intégré une représentation simple des facteurs influents. Les variables clés (revenu, montant du crédit, ancienneté) sont accompagnées d'indications visuelles (flèches, couleurs) qui expliquent leur impact sur le score. Ainsi, un conseiller peut justifier sa décision auprès du client en langage courant : « *Votre revenu élevé a contribué positivement à votre score, mais vos crédits en cours ont augmenté votre risque.* »

**Accessibilité et ergonomie** : j'ai choisi une interface épurée, sans surcharge graphique. L'expérience utilisateur doit rester intuitive même pour un conseiller qui n'a jamais utilisé ce type d'outil auparavant.

**Transparence et confiance** : un design clair et explicatif inspire la confiance. Le client doit sentir que la décision n'est pas arbitraire, mais repose sur des éléments visibles et compréhensibles.

## API - Mise en production du modèle pour prédiction

Une fois mon modèle entraîné et validé, j'ai travaillé sur sa mise en production via une **API**. L'objectif était de rendre le moteur d'inférence accessible de manière simple et sécurisée aux conseillers, grâce à un service qui prend en entrée les données d'un client et renvoie en sortie un score de crédit accompagné d'explications.

**Github** : <https://github.com/loi997/api-credit-score.git>

J'ai choisi d'utiliser **FastAPI**, un framework Python moderne, léger et performant. FastAPI m'a permis de construire rapidement des endpoints clairs, documentés automatiquement (via Swagger/OpenAPI), et surtout capables de répondre en temps réel aux requêtes de mon dashboard Streamlit.

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

J'ai également utilisé la librairie **joblib** pour charger le modèle et le scaler préalablement sauvegardés. Cela m'a permis d'éviter tout réentraînement à chaque appel et d'assurer la cohérence entre le notebook de modélisation et l'API déployée.

Mon API suit une architecture simple mais robuste :

1. **Chargement des artefacts** : au démarrage, l'API charge le modèle (model.pkl) et le scaler (scaler.pkl).
2. **Définition des endpoints** : j'ai défini plusieurs routes permettant de tester la disponibilité du service, de consulter les features attendues, et bien sûr de générer une prédiction.
3. **Traitement des requêtes** : pour chaque appel, l'API vérifie la validité des données (nombre de variables, type numérique attendu), applique le scaler, calcule la probabilité via le modèle, puis renvoie le score.
4. **Explicabilité** : j'ai intégré SHAP pour générer les contributions des variables en temps réel et les inclure dans la réponse JSON.

**GET /** : point d'accueil, qui renvoie un message de bienvenue, le nombre de features attendues et les endpoints disponibles. Il permet de vérifier rapidement que l'API fonctionne.

**GET /features-names** : retourne la liste des variables utilisées par le modèle. Cela aide à documenter l'API et à s'assurer que les données envoyées par le dashboard sont conformes.

**POST /predict** : endpoint principal de prédiction.

**Entrée** : un JSON contenant une liste de valeurs numériques correspondant aux features du modèle.

**GET /healthz** : endpoint de monitoring, utilisé par la plateforme Render pour vérifier que le service est actif. Il renvoie simplement un message "ok" et permet de déclencher une alerte si l'API devient indisponible.

J'ai choisi **Render** comme plateforme de déploiement. Render me permet d'héberger mon API gratuitement en POC, avec un démarrage automatisé du service à chaque mise à jour de mon dépôt GitHub.

- J'ai configuré le fichier requirements.txt pour installer toutes les dépendances (FastAPI, uvicorn, joblib, shap, numpy, etc.).
- Le **Start Command** a été défini comme :
- `uvicorn api.main:app --host 0.0.0.0 --port 10000`

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

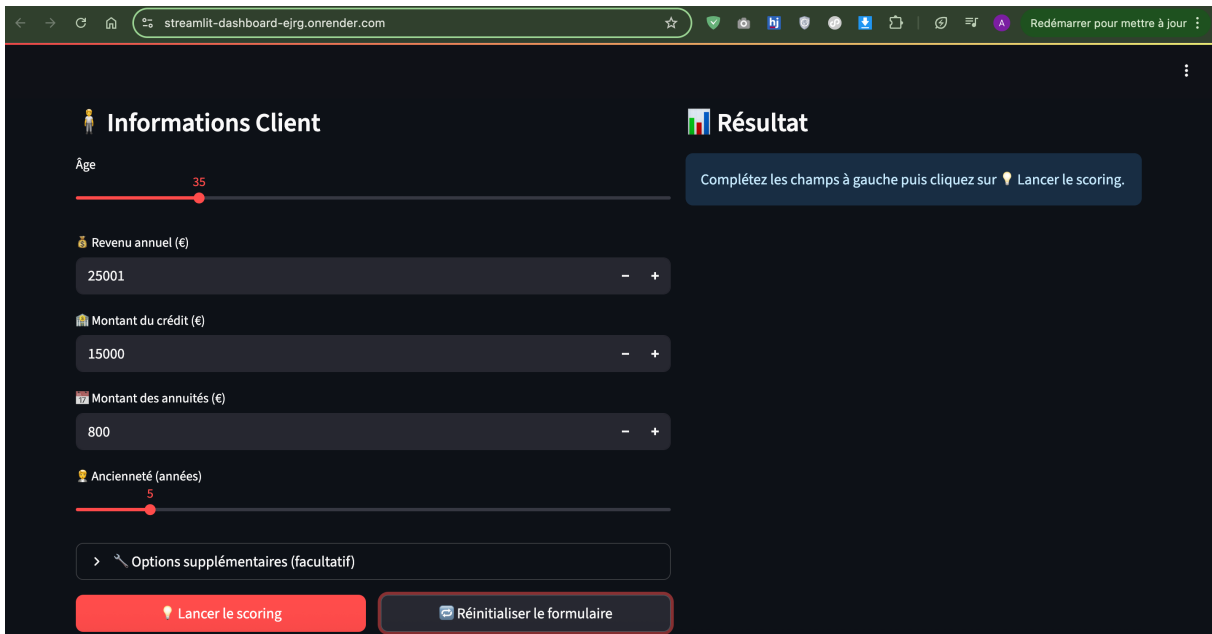
- L'API est désormais disponible à une URL publique, que le dashboard peut interroger directement : <https://streamlit-dashboard-ejrg.onrender.com/>

Grâce à cette mise en production, mon modèle est devenu un **moteur d'inférence en ligne**. Les conseillers peuvent saisir les informations d'un client dans le dashboard, qui envoie la requête à l'API et récupère instantanément un score accompagné d'explications. Cette architecture sépare clairement la logique de calcul (API) de la présentation visuelle (Streamlit), ce qui rend le système plus robuste, évolutif et maintenable.

## Dashboard en production

Après avoir déployé mon modèle via une API, j'ai travaillé sur la mise en production du **dashboard conseiller**. Mon objectif était de proposer une interface simple, intuitive et fiable qui interagit directement avec l'API pour fournir en temps réel un score de crédit et des explications compréhensibles.

Dashboard : <https://streamlit-dashboard-ejrg.onrender.com/>



streamlit-dashboard-ejrg.onrender.com

Redémarrer pour mettre à jour

### Informations Client

Âge: 35

Revenu annuel (€): 25001

Montant du crédit (€): 15000

Montant des annuités (€): 800

Ancienneté (années): 5

> Options supplémentaires (facultatif)

Lancer le scoring

Réinitialiser le formulaire

### Résultat

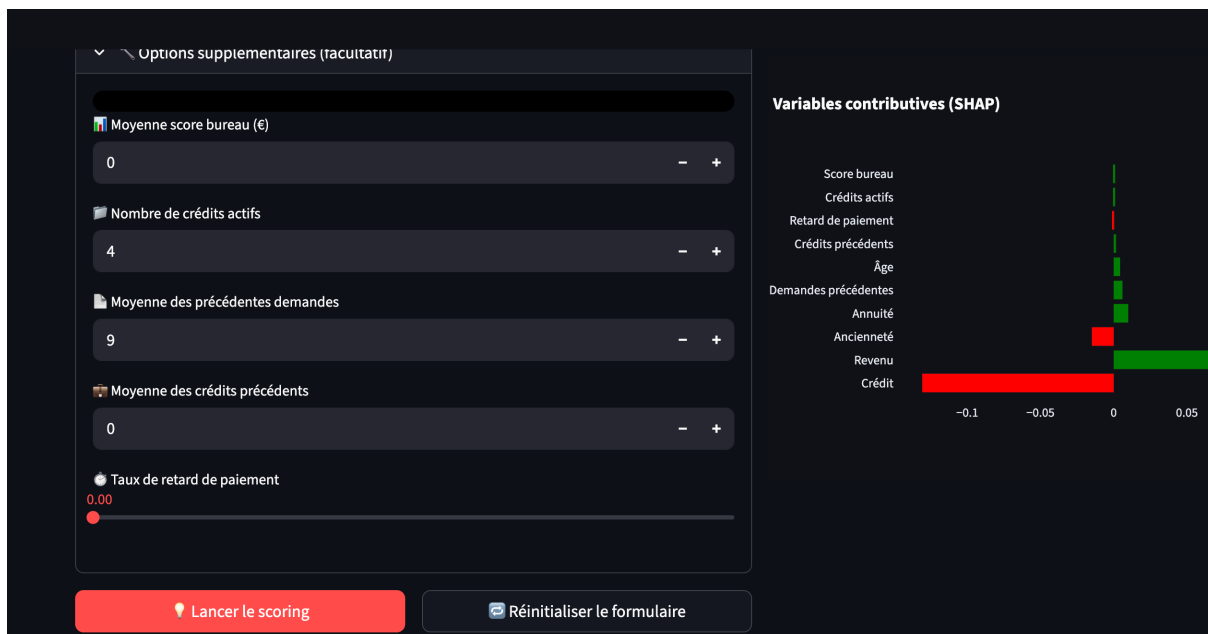
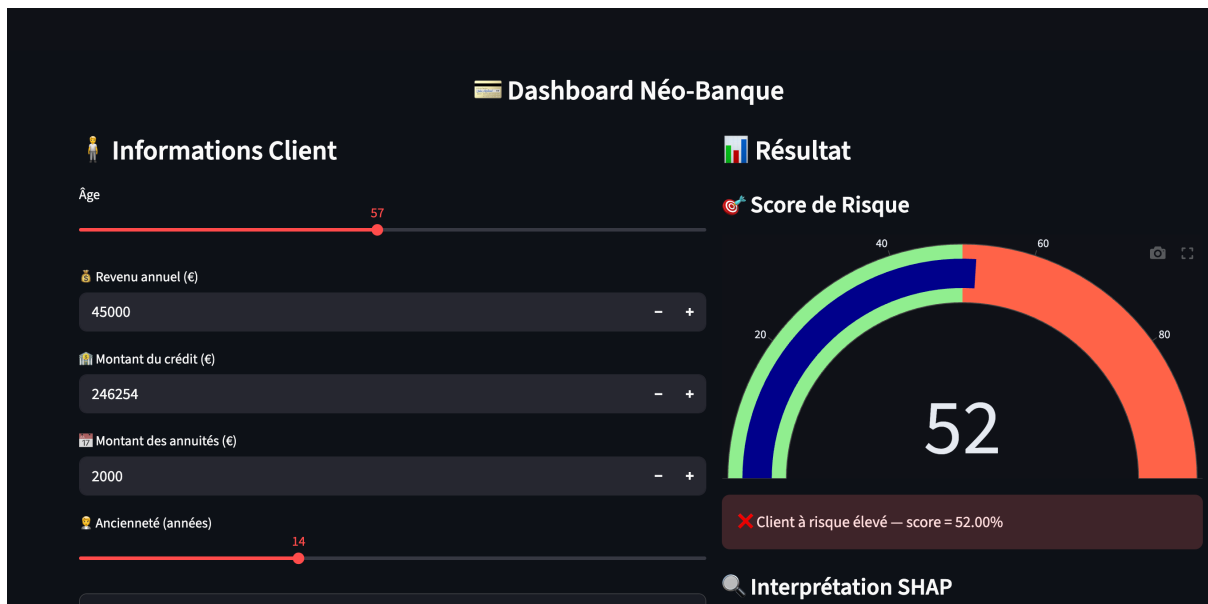
Complétez les champs à gauche puis cliquez sur Lancer le scoring.

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

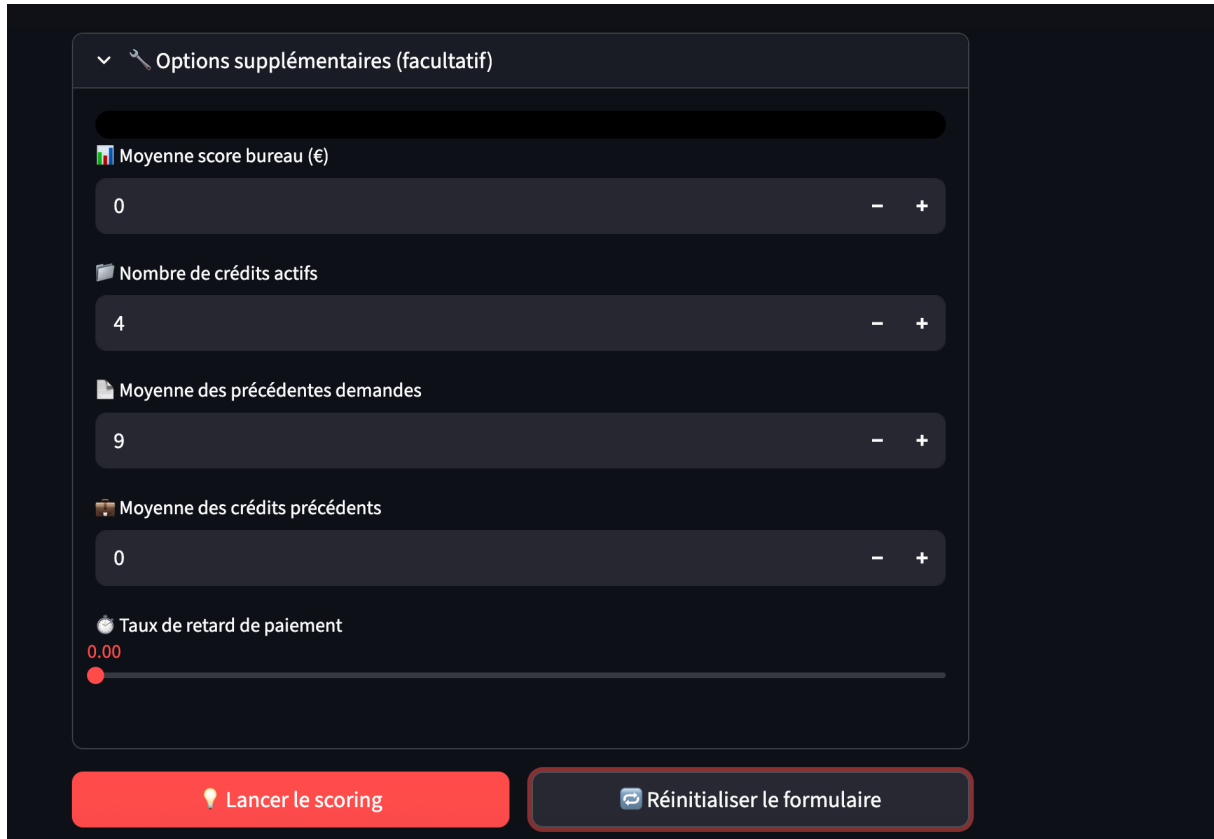


## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle



The screenshot shows a dark-themed web interface for a credit scoring tool. At the top, there is a dropdown menu labeled 'Options supplémentaires (facultatif)'. Below it, there are five input fields, each with a label and a value: 'Moyenne score bureau (€)' with value '0', 'Nombre de crédits actifs' with value '4', 'Moyenne des précédentes demandes' with value '9', 'Moyenne des crédits précédents' with value '0', and 'Taux de retard de paiement' with value '0.00'. Each of the first four fields has minus and plus buttons for adjustment. The 'Taux de retard de paiement' field has a red progress bar. At the bottom, there are two buttons: 'Lancer le scoring' (red) and 'Réinitialiser le formulaire' (blue).

J'ai choisi **Streamlit** comme technologie principale pour construire le dashboard. Ce choix s'est imposé pour plusieurs raisons :

- Streamlit me permet de transformer rapidement du code Python en une interface web interactive, sans avoir besoin de développer un frontend complexe.
- L'outil est parfaitement adapté aux prototypes de data science et d'IA, ce qui correspond à mon projet.
- Il offre une bonne flexibilité pour intégrer des visualisations (jauges, graphiques SHAP, tables filtrées).

J'ai conçu une page structurée autour de trois blocs principaux :

1. **Bloc Informations Client** : j'ai placé des champs de saisie (âge, revenu annuel, montant du crédit, annuités, ancienneté). Ces champs permettent au conseiller de saisir ou de sélectionner les informations d'un client de manière simple.

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

2. **Bloc Résultat** : dès que les informations sont validées, le dashboard envoie une requête à l'API et affiche le **score de probabilité** sous forme de jauge colorée. J'ai choisi ce visuel car il permet une lecture immédiate et intuitive du risque.
3. **Bloc Explication** : j'ai intégré les contributions **SHAP** sous forme de tableau et d'icônes colorées. Chaque variable importante est accompagnée d'un indicateur (positif ou négatif), permettant au conseiller d'expliquer en langage simple l'origine du score.

Le cœur du fonctionnement repose sur l'appel à l'API :

- Lorsque le conseiller saisit les données d'un client, le dashboard les envoie en JSON au **endpoint /predict** de l'API.
- L'API renvoie en réponse un score (entre 0 et 1) ainsi que les explications SHAP correspondantes.
- Ces résultats sont affichés en temps réel dans le dashboard, sans délai perceptible.

J'ai également ajouté une gestion des erreurs :

- Si l'API est indisponible, un message clair apparaît : « *Erreur lors de l'appel à l'API* ».
- Si les données saisies ne respectent pas le format attendu, le conseiller est alerté immédiatement.

J'ai déployé le dashboard sur la même plateforme que l'API, **Render**, afin de bénéficier d'une intégration continue depuis mon dépôt GitHub.

- Le **Start Command** utilisé est :
- `streamlit run streamlit_app.py --server.port=10000 --server.address=0.0.0.0`
- L'application est accessible publiquement via une URL unique.
- J'ai configuré un service séparé pour le dashboard et l'API afin de garantir l'indépendance des composants (principe de séparation des responsabilités).

Grâce à ce déploiement, le conseiller peut accéder au dashboard directement depuis son navigateur, sans installer aucun logiciel supplémentaire. L'application est simple, fluide et compréhensible. Elle répond en quelques secondes aux besoins suivants :

- Calculer un **score de solvabilité**,

## Etude de cas NEO BANQUE

Chef de projet en intelligence artificielle – Niveau 7 – ASCENCIA PARIS LA DEFENSE]

Bloc de compétences : Déployer une solution d'intelligence artificielle

Mastère Chef de Projet en Intelligence Artificielle

- Comprendre **pourquoi** ce score a été attribué,
- Disposer d'une interface claire pour appuyer son argumentation auprès du client.

### Perspectives d'évolution

Même si la solution est fonctionnelle et déjà exploitable en POC, je vois plusieurs pistes d'amélioration pour la suite :

1. **Renforcement de la sécurité** : mettre en place une authentification forte (OAuth2, SSO bancaire) et un système de journalisation avancé avec détection d'anomalies pour prévenir toute tentative d'intrusion.
2. **Amélioration de l'explicabilité** : compléter l'usage de SHAP par d'autres méthodes (LIME, PDP – Partial Dependence Plots) afin de diversifier les modes d'explication et de rendre l'outil encore plus pédagogique pour les conseillers.
3. **Enrichissement des données** : intégrer de nouvelles variables pertinentes comme l'historique de transactions, les comportements de paiement ou encore les données issues de bureaux de crédit externes. Cela renforcerait la fiabilité du score.
4. **Suivi continu du modèle** : mettre en place un système de monitoring du drift (dérive des données) pour détecter si le modèle perd en performance avec le temps, et automatiser les ré-entraînements si nécessaire.
5. **Optimisation de la scalabilité** : dans une mise en production réelle, déployer l'API sur une infrastructure cloud robuste (Kubernetes, AWS, GCP, Azure) pour supporter un grand nombre de requêtes simultanées.
6. **Extension fonctionnelle du dashboard** : permettre aux conseillers de comparer plusieurs profils clients, d'accéder à des statistiques globales de portefeuille ou encore de simuler différents scénarios (par exemple augmenter la durée du crédit pour voir l'effet sur le score).