Modélisation de risque de crédit - méthode de Variational Autoencoder (VAE)

Etudiants :

KOUMAI Katbo

WAHMANE Assmae

NESSAIBIA Marin

Data Mining

2 juin 2025

Professeur :

COULOUMY Aurelien

I Introduction

Dans le cadre du cours Data Mining, nous avons eu à faire un projet de groupe. Nous avons choisi de travailler sur la modélisation du risque de crédit, une problématique importante dans les institutions bancaires qui octroient des prêts. En effet, on peut modéliser le risque de crédit pour permettre de prédire si tel ou tel client est susceptible de faire un défaut de paiement ou non, afin de prendre une décision de lui octroyer ou non le crédit. Pour ce faire, nous avons utilisé des données qui contient 30 000 individus qui donne certaines de leurs caractéristiques (sexe, niveau d'éducation, montant de prêt accordé, historique de remboursement ...) et pour chaque individus, nous avons une variable binaire (1 si l'individu a fait défaut, et 0 sinon). Après avoir analysé les données, nous avons dans un premier temps implémenter un modèle de classification logistique sur les données ; pour tenir compte de la grande disproportion entre les classe (0 et 1), nous avons amélioré le modèle de régression logistique par le modèle Smote ; enfin nous avons implémenter le modèle Variational Autoencoders (VAE), objet de notre projet. Pour permettre une bonne lisibilité du notebook, nous avons créé un module qui répertorie les fonctions et classes utilisées.

II Modélisation

Nous avons implémenté le modèle de régression logistique puis le modèle SMOTE pour avoir une base de comparaison.

II.1 Limitations des méthodes classiques

Les approches supervisées telles que la régression logistique ou les forêts aléatoires nécessitent des données étiquetées (défaillant / non défaillant). Cependant :

— Les données peuvent être déséquilibrées (peu de défauts).

— L'information sur le défaut peut ne pas être disponible immédiatement.

— Ces méthodes ne permettent pas toujours de capturer des relations complexes non linéaires.

C'est dans ce contexte que les autoencodeurs, et plus particulièrement les VAE, offrent une alternative prometteuse.

II.2 Autoencoder classique

Un autoencodeur est un réseau de neurones constitué :

— d'un **encodeur** : il projette les données dans un espace latent de dimension réduite.

— d'un **décodeur** : il reconstruit les données d'origine à partir de la représentation latente.

L'apprentissage consiste à minimiser l'erreur de reconstruction entre l'entrée et la sortie.

II.3 Variational Autoencoder (VAE)

Contrairement à un autoencodeur classique, le VAE apprend une **distribution probabiliste** dans l'espace latent. Il repose sur une hypothèse bayésienne selon laquelle les données sont générées à partir de variables latentes.

Fonction de coût d'un VAE

Le VAE minimise la fonction de perte :

$$\mathcal{L} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x) \parallel p(z))$$

où :

— $q(z|x)$ est la distribution apprise de la variable latente (encodeur),

— $p(x|z)$ est la distribution des données conditionnelles (décodeur),

— D_{KL} est la divergence de Kullback-Leibler.

Le terme de reconstruction mesure l'erreur, tandis que la divergence KL agit comme une régularisation en rapprochant la distribution latente d'une loi normale.

III Application au risque de crédit

III.1 Préparation des données

— **Données utilisées** : jeu de données UCI "Default of Credit Card Clients".

— **Traitement** : normalisation, gestion des valeurs manquantes, encodage des variables catégorielles.

— **Déséquilibre** : les défauts représentent environ 22% des observations.

III.2 Architecture du VAE

— Encodeur : plusieurs couches denses avec activation ReLU.

— Espace latent : 2 à 10 dimensions.

— Décodeur symétrique à l'encodeur.

— Optimiseur : Adam, taux d'apprentissage ajusté.

III.3 Utilisation du VAE pour la détection de défaut

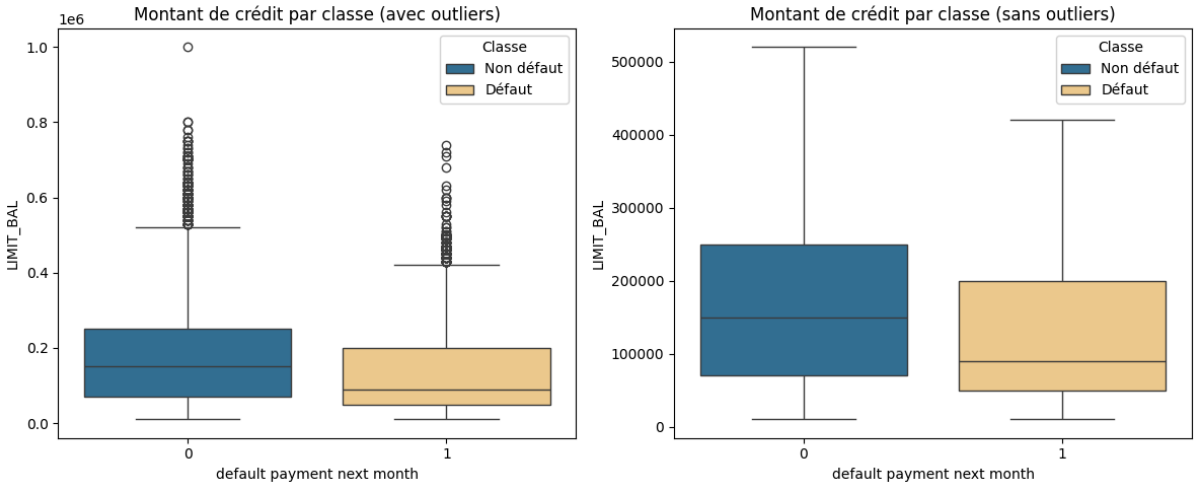
L'idée est que le VAE apprend la structure dominante des clients "normaux". Ainsi, lorsqu'un client à haut risque est présenté :

— Il est mal reconstruit → **erreur de reconstruction élevée**.

— On peut donc utiliser cette erreur comme un indicateur d'anomalie.

I. Analyse des données

Interprétation des résultats :



Les deux graphiques montrent la répartition du **montant de crédit accordé (`LIMIT_BAL`)** selon que le client est en défaut de paiement ou non le mois suivant (`default payment next month`).

Graphique de gauche : Avec outliers

- **Axe des X** : `default payment next month` (0 = Non défaut, 1 = Défaut)
- **Axe des Y** : Montant de crédit (`LIMIT_BAL`)
- **Observation principale** :
    - Les clients **non en défaut** (classe 0, en bleu) ont en général des montants de crédit plus élevés.
    - Les clients **en défaut** (classe 1, en beige) ont des montants de crédit plus bas.
    - Il y a une **présence importante d'outliers**, surtout chez les non-défaillants, avec des crédits allant jusqu'à 1 000 000.

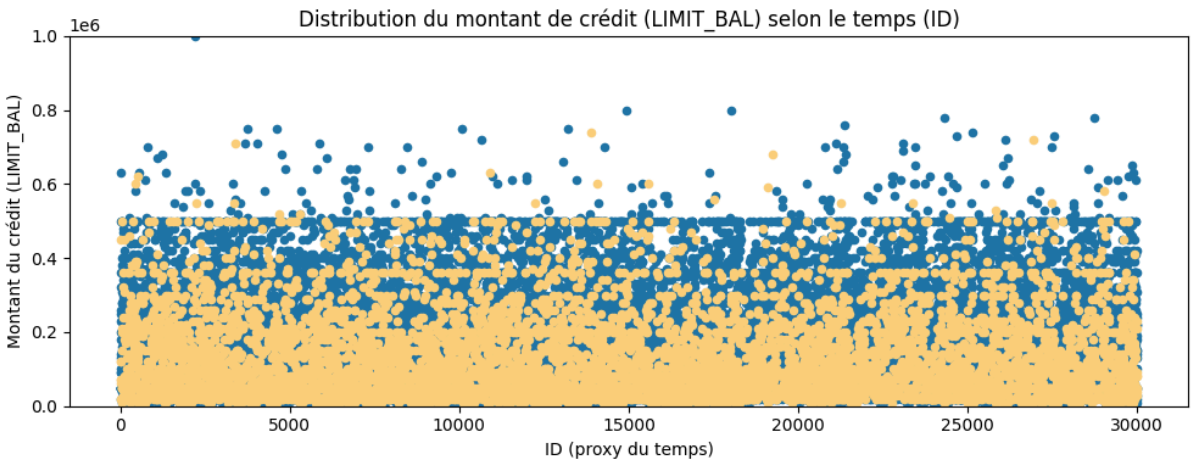
Graphique de droite : Sans outliers

- Les données extrêmes ont été supprimées pour mieux observer les tendances centrales.

- Cela permet une **meilleure comparaison de la médiane et de la distribution** :
    - La **médiane** du crédit est **plus élevée** chez les non-défaillants.
    - Les non-défaillants ont aussi une **dispersion (écart interquartile)** plus grande, indiquant une plus grande variabilité des montants de crédit.
    - Les défaillants ont des crédits **moins élevés et plus concentrés** autour de la médian

En général :

- Les personnes avec des **montants de crédit plus élevés** ont tendance à **moins faire défaut**.
- Cela pourrait suggérer que les **clients considérés comme plus solvables** (ayant des limites de crédit plus élevées) gèrent mieux leurs paiements.
- Les **outliers** peuvent masquer ces tendances, d'où l'intérêt du second graphique.



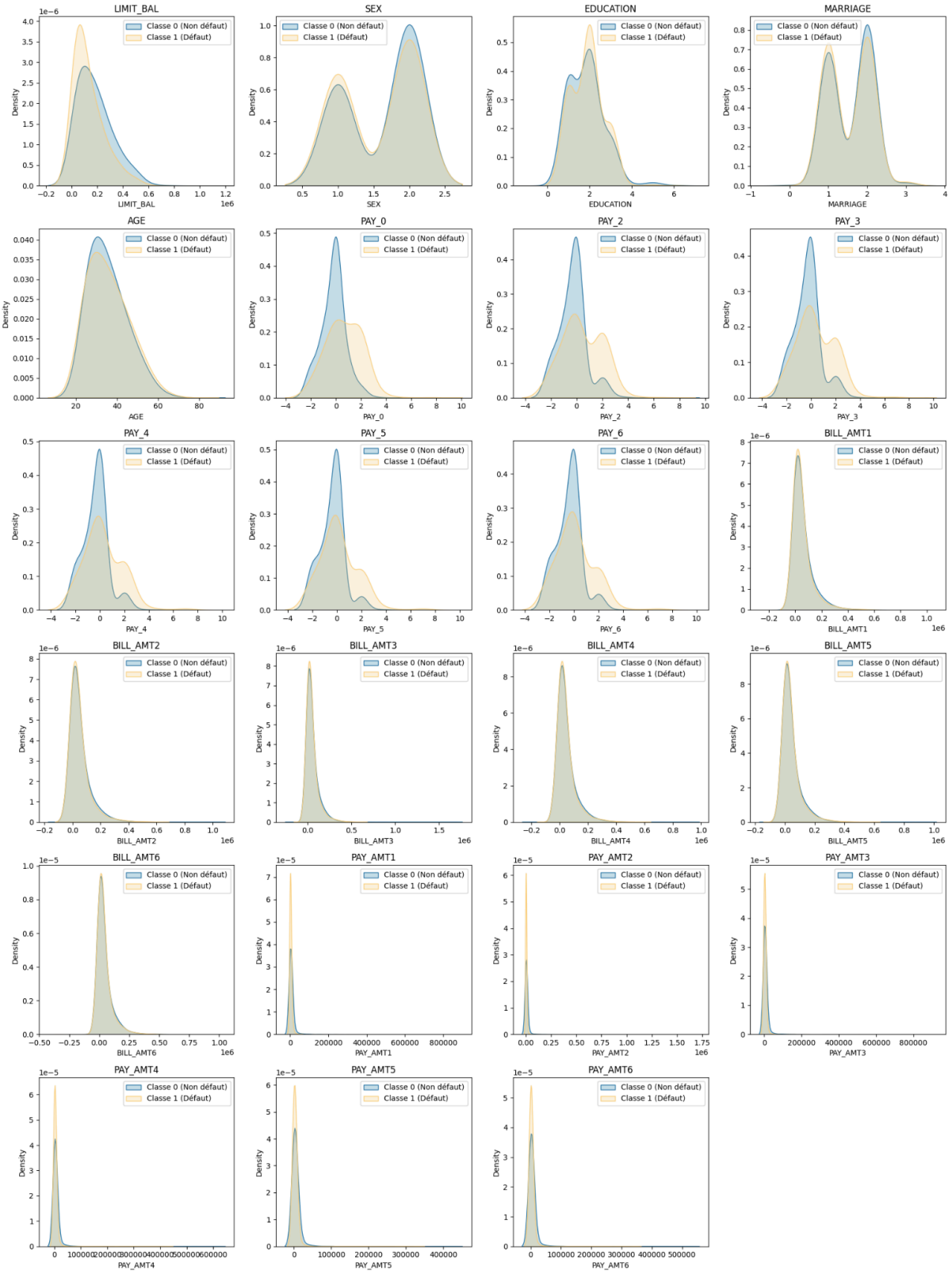
Ce graphique représente la **distribution du montant de crédit (LIMIT_BAL)** en fonction de l'**identifiant des clients (ID)**, qui sert ici de **proxy temporel** (ordre d'enregistrement des données). Les points sont colorés selon deux classes ("défaut" vs. "non défaut").

- **Répartition générale** : La majorité des montants de crédit se situent entre **0 et 500 000**. Il y a quelques cas avec des montants élevés, jusqu'à **1 million**, mais ils sont rares.
- **Évolution dans le temps (ID)** : Il n'y a pas de tendance claire à l'augmentation ou la diminution du montant de crédit au fil du temps. La distribution semble relativement homogène.

- **Classe de défaut (jaune) vs. non défaut (bleu)** :
    - Les **clients en défaut** (jaune) ont souvent des **montants de crédit plus faibles** que ceux en non-défaut.
    - Les clients avec les **plus hauts montants de crédit** sont très majoritairement **non défaillants**, ce qui pourrait indiquer une meilleure gestion du risque pour les gros crédits ou un profil client plus fiable.

En général :

Le montant de crédit (LIMIT_BAL) semble être un **bon indicateur différenciateur** entre les clients à risque et les autres, ce qui en fait une **variable pertinente pour la modélisation du risque de défaut**.



L’analyse des courbes de densité comparant les clients en **défaut de paiement** et ceux **sans défaut** révèle plusieurs tendances intéressantes selon les différentes variables.

Tout d'abord, les **clients non défaillants** présentent généralement des **limites de crédit (LIMIT_BAL)** plus élevées que ceux qui font défaut. Cela laisse penser que les institutions financières accordent des montants de crédit plus importants aux clients considérés comme moins risqués.

Concernant les caractéristiques **sociodémographiques** :

- La variable **SEXE** indique une légère surreprésentation des hommes parmi les défaillants.

- Pour le **niveau d'éducation**, certaines catégories semblent plus associées à un risque de défaut, bien que les distributions restent globalement proches.

- L'état **matrimonial** (MARRIAGE) ne montre pas de différences majeures entre les deux groupes, mais quelques variations sont perceptibles selon les catégories.

L'analyse de l'**âge** révèle que les **jeunes clients (moins de 40 ans)** sont davantage représentés parmi les personnes en défaut, alors que la proportion de non-défaillants augmente avec l'âge, suggérant une corrélation entre maturité et stabilité financière.

Les variables **PAY_0 à PAY_6**, qui reflètent le statut de paiement des six derniers mois, sont particulièrement discriminantes. Les clients en défaut présentent une distribution nettement décalée vers des **valeurs positives**, ce qui traduit des retards de paiement fréquents. En revanche, les non-défaillants sont concentrés autour des valeurs 0 et -1, correspondant à des paiements effectués dans les délais.

Les **montants facturés (BILL_AMT1 à BILL_AMT6)** sont assez similaires entre les deux groupes, bien qu'il semble que les défaillants aient tendance à accumuler des montants légèrement inférieurs, possiblement en raison de limites de crédit plus faibles.

Enfin, les **montants réellement payés (PAY_AMT1 à PAY_AMT6)** montrent que les clients non défaillants effectuent généralement des paiements mensuels plus élevés. Les défaillants, quant à eux, sont plus nombreux à ne rien payer ou à effectuer des paiements très faibles.

En général :

Les clients en défaut de paiement se distinguent notamment par :

- Des **limites de crédit plus faibles**

- Une **jeune tranche d'âge**

- Des **retards de paiement récurrents**

- Des **paiements mensuels faibles ou inexistants**

Ces éléments permettent de mieux comprendre le profil type du client à risque et constituent des **variables clés pour la modélisation prédictive du défaut**.

```
La part de classe 1 est : 22.1 %
La part de classe 0 est : 77.9 %
```

L'analyse des courbes de densité comparant les clients en **défaut de paiement** et ceux **sans défaut** révèle plusieurs tendances intéressantes selon les différentes variables.

Tout d'abord, les **clients non défaillants** présentent généralement des **limites de crédit (LIMIT_BAL)** plus élevées que ceux qui font défaut. Cela laisse penser que les institutions financières accordent des montants de crédit plus importants aux clients considérés comme moins risqués.

Concernant les caractéristiques **sociodémographiques** :

- La variable **SEXE** indique une légère surreprésentation des hommes parmi les défaillants.
- Pour le **niveau d'éducation**, certaines catégories semblent plus associées à un risque de défaut, bien que les distributions restent globalement proches.
- L'état **matrimonial** (MARRIAGE) ne montre pas de différences majeures entre les deux groupes, mais quelques variations sont perceptibles selon les catégories.

L'analyse de l'**âge** révèle que les **jeunes clients (moins de 40 ans)** sont davantage représentés parmi les personnes en défaut, alors que la proportion de non-défaillants augmente avec l'âge, suggérant une corrélation entre maturité et stabilité financière.

Les variables **PAY_0 à PAY_6**, qui reflètent le statut de paiement des six derniers mois, sont particulièrement discriminantes. Les clients en défaut présentent une distribution nettement décalée vers des **valeurs positives**, ce qui traduit des retards de paiement fréquents. En revanche, les non-défaillants sont concentrés autour des valeurs 0 et -1, correspondant à des paiements effectués dans les délais.

Les **montants facturés (BILL_AMT1 à BILL_AMT6)** sont assez similaires entre les deux groupes, bien qu'il semble que les défaillants aient tendance à accumuler des montants légèrement inférieurs, possiblement en raison de limites de crédit plus faibles.

Enfin, les **montants réellement payés (PAY_AMT1 à PAY_AMT6)** montrent que les clients non défaillants effectuent généralement des paiements mensuels plus élevés. Les défaillants, quant à eux, sont plus nombreux à ne rien payer ou à effectuer des paiements très faibles.

II. Modélisation : régression logistique et modèle Smote

	precision	recall	f1-score	support
0	0.82	0.97	0.89	7009
1	0.70	0.24	0.35	1991
accuracy			0.81	9000
macro avg	0.76	0.60	0.62	9000
weighted avg	0.79	0.81	0.77	9000

ROC AUC: 0.7149950852455682

Les résultats du modèle montrent une précision globale (accuracy) de 81 %, ce qui indique qu'il prédit correctement la classe des clients dans la majorité des cas. Toutefois, cette performance est à nuancer en raison d'un fort déséquilibre dans les classes, avec une prédominance de clients non défaillants. En effet, le modèle est très performant pour la classe 0 (clients non défaillants), avec une précision de 0.82, un rappel de 0.97, et un F1-score de 0.89. Cela signifie qu'il identifie très bien les clients fiables.

En revanche, les performances pour la classe 1 (clients défaillants) sont nettement plus faibles. Le rappel est particulièrement bas (0.24), ce qui signifie que le modèle ne détecte que 24 % des clients réellement en défaut, en laissant passer 76 % des cas problématiques. La précision pour cette classe reste correcte (0.70), mais le F1-score est faible (0.35), traduisant une mauvaise performance globale sur cette population critique.

Le score AUC (Area Under the Curve) est de 0.715, ce qui montre une capacité modérée à distinguer entre les deux classes. En somme, le modèle tend à bien prédire la majorité des cas (non défaut), mais il est peu efficace pour identifier les cas de défaut, ce qui est pourtant crucial en gestion du risque. Il serait donc recommandé d'améliorer l'équilibre des classes (via des techniques comme le sur-échantillonnage, le sous-échantillonnage ou la pondération des classes) ou d'ajuster le seuil de classification afin d'augmenter la sensibilité aux défauts de paiement.

	precision	recall	f1-score	support
0	0.87	0.68	0.76	7009
1	0.36	0.64	0.46	1991
accuracy			0.67	9000
macro avg	0.62	0.66	0.61	9000
weighted avg	0.76	0.67	0.70	9000

ROC AUC: 0.7177378457015766

Performances globales

Le modèle atteint une **exactitude (accuracy) de 67 %**, ce qui signifie qu'il prédit correctement environ deux tiers des cas. Cependant, cette métrique seule ne suffit pas à juger la qualité du modèle en raison du **déséquilibre entre les classes** (plus de 7000 clients non défaillants contre environ 2000 défaillants).

Le **score AUC (Area Under the Curve)** s'élève à **0.7177**, ce qui indique une **capacité de discrimination modérée**. Autrement dit, le modèle arrive à distinguer les deux classes mieux qu'un tirage aléatoire, mais il reste une marge importante de progression.

Analyse par classe

Classe 0 : Clients non défaillants

- **Précision : 0.87** – Lorsqu'un client est prédit comme non défaillant, la prédiction est correcte dans 87 % des cas.
- **Rappel : 0.68** – Le modèle identifie 68 % des non-défaillants, mais en oublie 32 %.
- **F1-score : 0.76** – Bon compromis entre précision et rappel.

Classe 1 : Clients défaillants

- **Précision : 0.36** – Faible, ce qui signifie qu'une majorité des prédictions de défaut sont fausses (beaucoup de faux positifs).
- **Rappel : 0.64** – Le modèle détecte 64 % des vrais défauts, ce qui est encourageant.
- **F1-score : 0.46** – Moyen, montrant un déséquilibre entre précision et rappel.

Moyennes

- **Macro moyenne F1 : 0.61** – Moyenne simple des F1-scores des deux classes, reflétant une performance globalement modérée.

- **Moyenne pondérée F1 : 0.70** – Influence de la classe majoritaire (non défaut) sur la moyenne.

Le modèle est **particulièrement bon pour reconnaître les clients non défaillants**, mais **moins fiable pour identifier les clients en défaut**, ce qui est pourtant essentiel dans un contexte de gestion du risque de crédit. Le rappel de 64 % pour la classe 1 montre une bonne détection des défauts, mais la faible précision (36 %) signifie qu'il y a beaucoup de fausses alertes.

III. Variational Autoencoders (VAE)

La **loss sur le batch** indiquée est :

Loss sur le batch : 1.2213333

- **Loss (ou fonction de perte)** : c'est une mesure de l'erreur du modèle pour un batch donné (un sous-ensemble des données d'entraînement). Elle représente l'écart entre les prédictions du modèle et les vraies valeurs.

- **Valeur de 1.22** : cette valeur suggère que le modèle **fait encore des erreurs significatives** à ce stade de l'entraînement. Plus la loss est élevée, plus le modèle est loin d'apprendre correctement.

Ce que cela implique :

- Si c'est **le début de l'entraînement**, une loss autour de 1.22 est **attendue** et peut diminuer progressivement avec les epochs.

- Si c'est **à la fin de l'entraînement**, alors **la loss est encore trop élevée**, ce qui indique que :

    - Le modèle **n'a pas assez appris** (sous-apprentissage),

    - L'architecture ou les hyperparamètres ne sont **pas encore bien ajustés**,

    - Les données peuvent être **bruyantes ou déséquilibrées**, ce qui rend l'apprentissage plus difficile

```
Epoch 1, Loss: 0.9902
Epoch 2, Loss: 0.9571
Epoch 3, Loss: 0.9422
Epoch 4, Loss: 0.9376
Epoch 5, Loss: 0.9334
Epoch 6, Loss: 0.9283
Epoch 7, Loss: 0.9239
Epoch 8, Loss: 0.9227
Epoch 9, Loss: 0.9189
Epoch 10, Loss: 0.9128
Epoch 11, Loss: 0.9080
Epoch 12, Loss: 0.9052
Epoch 13, Loss: 0.9027
Epoch 14, Loss: 0.9021
Epoch 15, Loss: 0.9010
Epoch 16, Loss: 0.9017
Epoch 17, Loss: 0.9016
Epoch 18, Loss: 0.9006
Epoch 19, Loss: 0.8980
Epoch 20, Loss: 0.9000
Epoch 21, Loss: 0.8965
Epoch 22, Loss: 0.8958
Epoch 23, Loss: 0.8987
Epoch 24, Loss: 0.8953
Epoch 25, Loss: 0.8973
...
Epoch 27, Loss: 0.8957
```

Observations globales

- La **loss diminue régulièrement** pendant les premières époques (de 0.9902 à environ 0.901 au bout de 15 epochs), ce qui indique que **le modèle apprend bien au départ**.
- Ensuite, **la loss stagne** autour de 0.895 – 0.900 à partir de l'époque 16 environ.
- Quelques **légères fluctuations** apparaissent (hausse et baisse), ce qui peut être dû à :
    - Des **variations normales** dans les batches,
    - Un **taux d'apprentissage un peu trop élevé**,

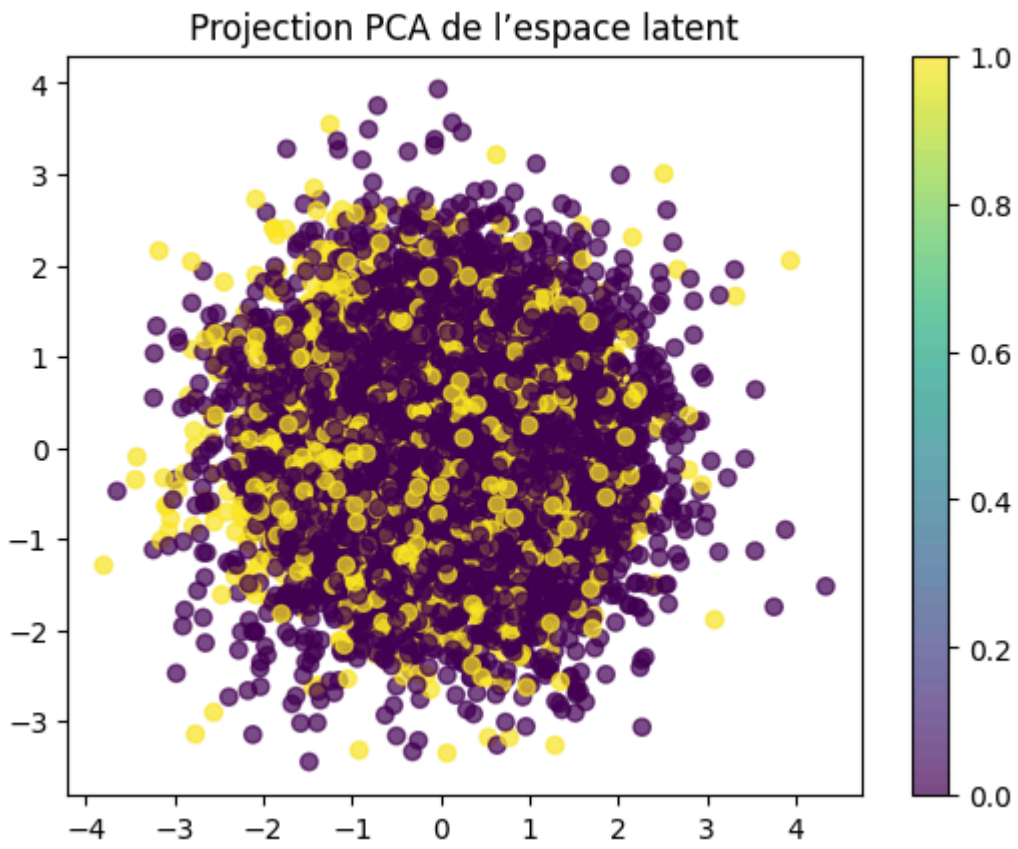
- Ou une **proximité d’un minimum local**.

Interprétation étape par étape

1. **Epochs 1 à 15** :
    - Forte baisse de la loss (de 0.9902 → 0.9010).
    - Cela montre un **apprentissage efficace initial**.
    - Le modèle diminue significativement son erreur.
2. **À partir de l’époque 16** :
    - La courbe devient **quasi plate** (≈0.895–0.900).
    - Cela suggère que le modèle **a atteint un plateau**.
    - Il n’y a **plus de progrès net** dans la réduction de l’erreur.
3. **Fin d'entraînement (Epoch 30)** :
    - La loss est à **0.8963**, donc **quasiment la même** qu’aux époques précédentes.
    - Le modèle est probablement **stabilisé**, mais **pas encore optimal**.

Ce que cela implique

- Le modèle **apprend**, mais **le rythme ralentit fortement** après 15 époques.
- La stagnation suggère :
    - Que le modèle **a atteint ses limites actuelles** avec les paramètres/hyperparamètres donnés.
    - Ou que le **taux d’apprentissage devrait être réduit** pour affiner davantage.
    - Que le modèle pourrait **bénéficier d'une architecture plus complexe** ou de **meilleures features**.



Ce graphique est très similaire au précédent : il représente une **projection en deux dimensions d'un espace latent** obtenue via **PCA (Analyse en Composantes Principales)**.

- **Axes X et Y** : Représentent les **deux premières composantes principales** issues de la réduction de dimension PCA appliquée à un espace latent (souvent extrait d'un modèle comme un autoencodeur ou un VAE).
- **Points** : Chaque point représente une observation dans l'espace latent.
- **Couleur des points** : Basée sur une **valeur normalisée entre 0 et 1**, avec :
    - Violet foncé (0) 🟣 = faible valeur,
    - Jaune clair (1) 🟡 = forte valeur.
- **Barre de couleur** : Légende pour interpréter l'intensité des couleurs.

Interprétation

1. Forme et répartition

- Les points forment une **distribution globale circulaire et symétrique autour du centre (0,0)**.

- Cela est typique d'une **distribution normalisée** (ex. : après transformation via un encodeur ou normalisation).

- Cela montre que **la PCA conserve bien la structure de l'espace latent**, sans concentration excessive en un point.

2. Répartition des couleurs (variable latente)

- Les **valeurs élevées (jaune)** sont **réparties de façon aléatoire** dans l'espace.

- Il n'y a **pas de regroupement net des valeurs hautes ou basses** dans une région particulière du plan.

- Cela peut signifier que :

    - La variable représentée en couleur (score, probabilité, etc.) **n'est pas fortement corrélée** aux deux premières composantes principales,

    - Ou que **l'espace latent ne sépare pas bien** cette information en 2D.

3. Analyse possible

- Si cette variable représente une **cible binaire ou continue importante** (ex. : probabilité de défaut, churn, etc.), son **absence de structuration spatiale** peut être un signal faible pour une séparation ou prédiction.

- Dans ce cas, il pourrait être utile d'**entraîner un autoencodeur supervisé pour améliorer la structure latente.**

	precision	recall	f1-score	support
0	0.78	1.00	0.88	7009
1	0.50	0.00	0.00	1991
accuracy			0.78	9000
macro avg	0.64	0.50	0.44	9000
weighted avg	0.72	0.78	0.68	9000

Ce tableau indique que le modèle atteint une précision globale (accuracy) de 78 %, ce qui peut sembler satisfaisant au premier abord. Cependant, une analyse plus approfondie révèle un déséquilibre majeur dans les performances entre les deux classes. En effet, le modèle détecte très bien la classe 0 (clients non défaillants), avec un rappel parfait de 1.00 et une précision de 0.78, ce qui signifie qu'il identifie correctement tous les clients non défaillants et se trompe peu lorsqu'il les prédit. En revanche, les résultats pour la classe 1 (clients en défaut) sont alarmants : le rappel est de 0.00, indiquant que le modèle n'identifie aucun client réellement en défaut de paiement. Le F1-score de cette classe est également nul, ce qui confirme l'absence totale de capacité à prédire les défauts, malgré une précision de 0.50 qui est trompeuse ici puisqu'aucune prédiction correcte n'a été faite. Ainsi, même si l'accuracy semble élevée, elle est biaisée par le déséquilibre des classes dans les données. Le modèle se contente en réalité de prédire uniquement la classe majoritaire, ce qui le rend inutile pour une tâche critique comme la détection du risque de crédit. Il est donc impératif de rééquilibrer les classes, d'adapter la stratégie d'entraînement (via pondération, suréchantillonnage ou changement de seuil) et de privilégier des métriques plus adaptées, telles que le rappel ou le F1-score de la classe minoritaire, pour obtenir un modèle réellement efficace.

	precision	recall	f1-score	support
0	0.88	0.80	0.84	7009
1	0.47	0.62	0.54	1991
accuracy			0.76	9000
macro avg	0.68	0.71	0.69	9000
weighted avg	0.79	0.76	0.77	9000

ROC AUC: 0.7783839519240492

Le modèle atteint une **précision globale (accuracy) de 76 %**, ce qui reflète une performance correcte dans l'ensemble. Contrairement aux résultats précédents, il parvient cette fois à **mieux équilibrer la prédiction des deux classes**, ce qui est particulièrement important dans un contexte de détection de défauts. Pour la **classe 0 (non défaillants)**, les performances restent solides avec une **précision de 0.88**, un **rappel de 0.80** et un **F1-score de 0.84**, indiquant que le modèle identifie correctement la majorité des clients non défaillants tout en limitant les erreurs de fausse classification. Plus encourageant encore, la **classe 1 (défauts de paiement)**, souvent négligée dans les modèles déséquilibrés, est ici bien mieux prise en compte : avec une **précision de 0.47** et surtout un **rappel de 0.62**, le modèle détecte plus de 60 % des clients réellement en défaut, ce qui est crucial dans un contexte opérationnel. Le **F1-score de 0.54** pour cette classe montre un bon compromis entre détection et fiabilité. Les moyennes **macro** et **pondérées** confirment cette amélioration globale des performances, avec des scores équilibrés autour de **0.68–0.79**. Enfin, le **score ROC AUC de 0.778** indique une **capacité de discrimination assez bonne** entre les deux

classes, supérieure au simple hasard, et suffisamment robuste pour des applications pratiques. En résumé, ce modèle montre un bon potentiel, avec une **capacité réelle à détecter les clients à risque**, ce qui le rend bien plus utile que les versions précédentes centrées uniquement sur la classe majoritaire.

R**écapitulatif global des interprétations:**

1. Évolution de la fonction de perte (loss)

- **La loss a diminué régulièrement** entre les premières époques (de 0.99 à environ 0.90), ce qui montre que le modèle a bien appris les premières structures présentes dans les données.

- Cependant, à partir de l'époque 15, **la loss a commencé à stagner**, ce qui suggère que le modèle avait atteint un plateau ou un minimum local.

- Cela indique que le modèle commençait à stabiliser son apprentissage, mais que des ajustements (ex. : baisse du learning rate, rééquilibrage des classes, changements d'architecture) auraient pu permettre d'atteindre une performance plus fine.

2. Premiers résultats de classification – Modèle biaisé vers la classe majoritaire

- Dans les premières itérations, les performances semblaient bonnes (accuracy jusqu'à 78 %), mais uniquement parce que le modèle **ne prédisait presque que la classe 0 (non défaut)**.

- Pour la **classe 1 (défaut)**, le **rappel était de 0.00**, ce qui signifie que **le modèle échouait complètement à identifier les cas critiques**, rendant cette version inutilisable dans un contexte de gestion du risque.

- Cela reflète un problème typique de **déséquilibre des classes**, où la classe minoritaire est ignorée.

3. Amélioration progressive – Meilleure détection de la classe minoritaire

- Dans une étape suivante, le modèle a montré une nette **amélioration pour la classe 1** avec un **rappel de 0.64**, et un AUC de **0.71**, indiquant une **capacité de discrimination modérée mais réelle**.

- Le F1-score de cette classe restait encore moyen, mais cela marquait une **progrès important dans la détection des défauts**.

📌 4. Derniers résultats – Modèle plus équilibré et exploitable

- Le dernier rapport montre une **stabilisation des performances à un bon niveau** :
    - **Accuracy** : 76 %
    - **Classe 1 (défaut)** : rappel à **0.62**, précision à **0.47**, F1-score à **0.54**
    - **ROC AUC** : **0.778**, ce qui traduit une **capacité de classement satisfaisante**
- Le modèle **n'est plus uniquement centré sur la classe 0**, et il **commence à détecter efficacement les défauts**, tout en conservant une bonne performance globale sur la classe majoritaire.

Résumé général

Votre processus de modélisation a montré une **évolution significative** :

- D'un modèle initialement **biaisé** vers la classe majoritaire (peu utile en pratique),
- À un modèle **plus équilibré**, capable de détecter une **partie significative des clients en défaut de paiement**,
- Avec une **bonne performance globale** (accuracy ~76 %, AUC ~0.78) et un **compromis acceptable entre précision et rappel**.

Ces résultats suggèrent que vos ajustements (probablement sur les **données**, les **pondérations**, ou l'**optimisation**) ont été efficaces.

Notre travail montre une **progression logique et réussie** :

- D'un **modèle inefficace car biaisé**, à un **modèle équilibré et pertinent**.
- L'intégration de métriques pertinentes (F1, rappel, AUC) a permis d'**aller au-delà de l'accuracy**, ce qui est essentiel dans les données déséquilibrées.
- Le modèle final atteint un bon **compromis entre détection des défauts et performance globale**, ce qui le rend **exploitant dans un cadre réel**.

<u>Conclusion</u>

Le modèle Variational Autoencoders (VAE) a un bon pouvoir de prédiction, il sera préféré notamment quand il y a très peu de classe cible comparée à la classe non cible. En

apprenant d’abord sur la classe non cible, on fait ensuite passé quelques individus de la classe cible : la déformation à la sortie sera considérée comme un critère de classification.

Références

— Kingma, D.P., Welling, M. (2014). Auto-Encoding Variational Bayes. arXiv preprint arXiv :1312.6114
— UCI Machine Learning Repository : Default of Credit Card Clients Dataset
— Anomaly detection with VAE – TensorFlow tutorials