Projet - Intelligence Artificielle Explicable (XAI)

Van Duy Le / Radja Djihane Elmiri

10 février 2025

1 Introduction

L'Intelligence Artificielle Explicable (XAI) est un domaine en pleine expansion qui vise à rendre les modèles d'apprentissage automatique plus transparents et interprétables. Alors que les modèles d'IA deviennent de plus en plus complexes, il est essentiel de comprendre comment ils prennent des décisions, notamment dans des domaines critiques comme la finance, la santé, ou les télécommunications. Ce projet explore plusieurs méthodes d'explication des modèles d'apprentissage automatique, en se concentrant sur des techniques telles que SHAP (SHapley Additive exPlanations), DiCE (Diverse Counterfactual Explanations), Grad-CAM (Gradient-weighted Class Activation Mapping), et LRP (Layer-wise Relevance Propagation). Ces méthodes permettent non seulement de comprendre les prédictions des modèles, mais aussi de détecter les biais potentiels et d'optimiser les performances.

Dans ce projet, nous appliquons ces techniques à des cas concrets, notamment l'analyse du churn (taux d'attrition) dans l'industrie des télécommunications. Nous utilisons des modèles comme XGBoost et RandomForestClassifier pour prédire les clients à risque de résiliation, et nous expliquons les décisions de ces modèles à l'aide de SHAP et DiCE. Enfin, nous explorons des techniques d'interprétabilité pour les réseaux de neurones profonds (Grad-CAM) et les modèles de traitement du langage naturel (BERT), ainsi que des méthodes de propagation de pertinence pour les réseaux de neurones graphiques (GNN-LRP).

2 SHAP : Comprendre comment fonctionne le modèle prédictif basé sur l'apprentissage automatique pour classer les clients potentiels en perte d'abonnement (XGBoost) - Projet 1

► Qu'est-ce que SHAP ?

SHAP (SHapley Additive exPlanations) est une méthode basée sur la théorie des jeux coopératifs, utilisée pour interpréter les modèles d'apprentissage automatique. Elle attribue une contribution spécifique à chaque caractéristique pour expliquer son importance dans la prédiction.

SHAP est particulièrement efficace pour comprendre les modèles complexes comme XGBoost et les réseaux de neurones en fournissant un score d'importance pour chaque caractéristique.

► Objectif de SHAP ?

L'objectif principal de SHAP est d'offrir une explication cohérente et transparente des décisions des modèles. Il permet de :

— Quantifier l'impact de chaque caractéristique sur la prédiction,
— Faciliter l'interprétation des modèles complexes en les rendant plus lisibles,
— Détecter et éviter d'éventuels biais algorithmiques,
— Optimiser le modèle en identifiant les caractéristiques les plus influentes.

2.1 Valeur de Shapley : Définition et Calcul

SHAP repose sur une fonction linéaire des caractéristiques binaires :

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

avec M le nombre de caractéristiques et ϕ_i leur contribution respective.

2.2 Formule de calcul des valeurs de Shapley

La valeur de Shapley est calculée comme suit :

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \tag{2}$$

avec f la fonction du modèle et S un sous-ensemble des caractéristiques F.

2.3 Calcul des prédictions partielles $f_S(x_S)$

Lorsqu'un sous-ensemble S de caractéristiques est utilisé :

$$f_S(x_S) = \mathbb{E}[f(x)|x_S] \tag{3}$$

Les caractéristiques absentes sont remplacées par leur moyenne dans l'ensemble d'entraînement.

2.4 Propriétés fondamentales des valeurs de Shapley

Les valeurs de Shapley respectent :

— **Efficacité** : La somme des contributions est égale à la différence entre la prédiction et la moyenne globale.

— **Symétrie** : Deux caractéristiques ayant le même impact ont la même valeur SHAP.

— **Additivité** : Pour les modèles basés sur bagging, la somme des valeurs SHAP des sous-modèles donne celle du modèle global.

— **Nullité** : Une caractéristique sans influence obtient une valeur SHAP de 0.

2.5 XGBoost : Définition et Fonctionnement

XGBoost (eXtreme Gradient Boosting) est un algorithme de boosting performant utilisé pour les tâches de classification et de régression. Il repose sur des arbres de décision successifs optimisés pour minimiser l'erreur et améliorer la précision.

Ses principales caractéristiques sont :

— Utilisation de la descente de gradient pour réduire l'erreur de prédiction,

— Régularisation pour éviter le surapprentissage,

— Traitement efficace des données manquantes et des caractéristiques catégoriques,

— Capacité à gérer de grands volumes de données avec des performances élevées.

2.6 Application de XGBoost et SHAP sur l'analyse du churn

Dans l'industrie des télécommunications, l'analyse du churn est essentielle pour comprendre pourquoi certains clients cessent leur abonnement. L'objectif est de classifier les clients à risque de résiliation en fonction de leurs comportements et de leurs caractéristiques démographiques.

Le churn (taux d'attrition) représente le nombre de clients ayant annulé leur abonnement. Une analyse approfondie permet aux entreprises d'identifier les causes du churn et d'adopter des stratégies adaptées pour le réduire.

Objectif : Classifier les clients à risque en se basant sur des variables numériques et catégoriques.

Le jeu de données comprend des informations sur les clients telles que :

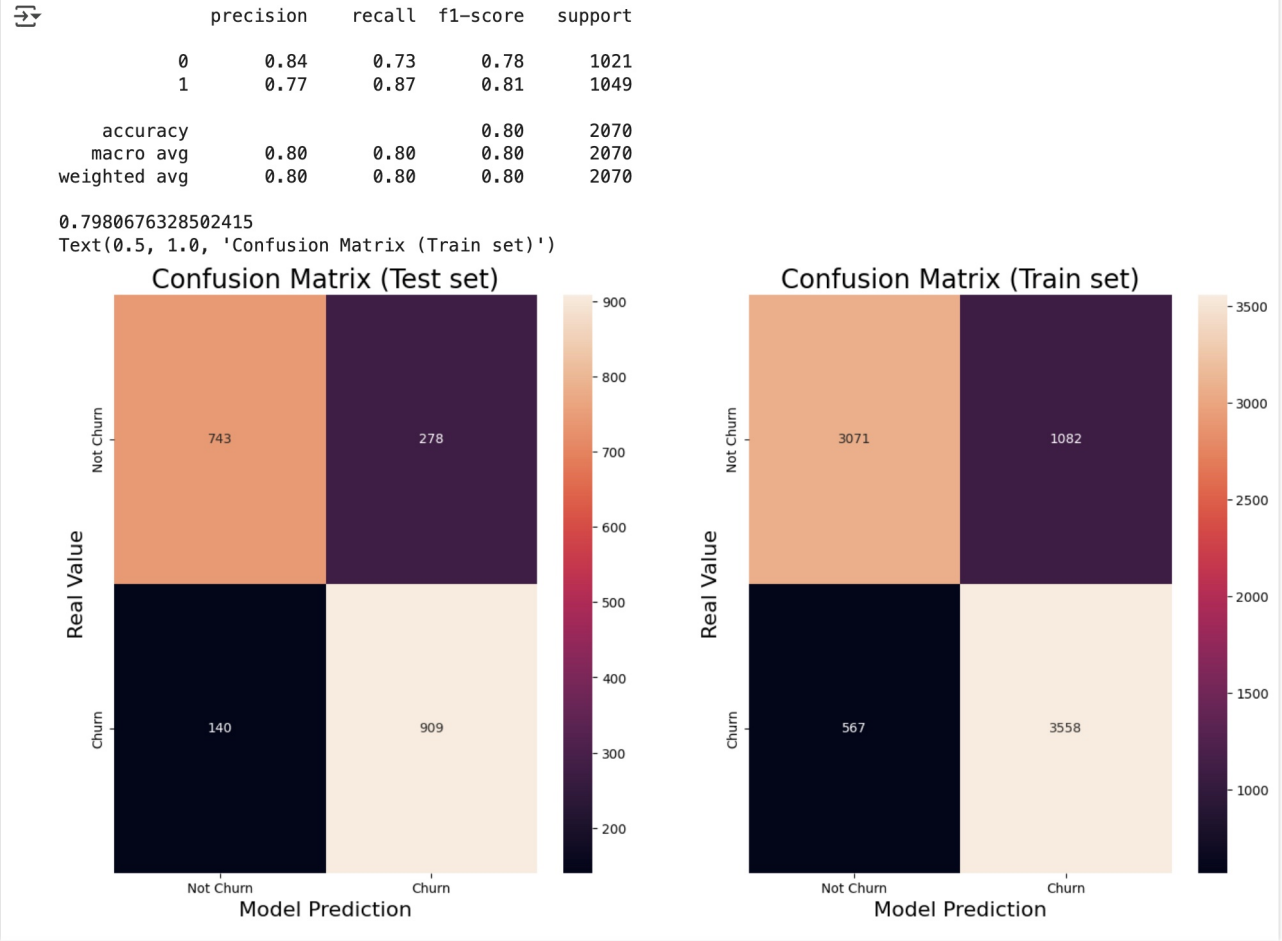
— customerID : Customer ID
— gender : Si le client est un homme ou une femme
— SeniorCitizen : Si le client est un senior ou non (1, 0)
— Partner : Si le client a un partenaire ou non (Yes, No)
— Dependents : Si le client a des personnes à charge ou non (Yes, No)
— tenure : Nombre de mois pendant lesquels le client est resté avec l'entreprise
— PhoneService : Si le client dispose d'un service téléphonique ou non (Yes, No)
— MultipleLines : Si le client possède plusieurs lignes ou non (Yes, No, No phone service)
— InternetService : Fournisseur de services Internet du client (DSL, Fiber optic, No)
— OnlineSecurity : Si le client dispose d'une sécurité en ligne ou non (Yes, No, No internet service)
— OnlineBackup : Si le client dispose d'une sauvegarde en ligne ou non (Yes, No, No internet service)
— DeviceProtection : Si le client bénéficie d'une protection d'appareil ou non (Yes, No, No internet service)
— TechSupport : Si le client dispose d'une assistance technique ou non (Yes, No, No internet service)
— StreamingTV : Si le client a accès à la télévision en streaming ou non (Yes, No, No internet service)
— StreamingMovies : Si le client a accès aux films en streaming ou non (Yes, No, No internet service)
— Contract : Type de contrat du client (Month-to-month, One year, Two year)
— PaperlessBilling : Si le client utilise la facturation électronique ou non (Yes, No)
— PaymentMethod : Mode de paiement du client (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
— MonthlyCharges : Montant facturé au client mensuellement
— TotalCharges : Montant total facturé au client
— Churn : Si le client s'est désabonné ou non (Yes = Churn, No = Not Churn)

2.7 Méthodologie

1. Charger et traiter les données
2. Entraînement du modèle XGBoost sur les données des clients.
3. Utilisation de SHAP pour analyser l'impact de chaque caractéristique sur les prédictions.

2.8 Le Resultat :

La Prediction du Model XGBoost :



	precision	recall	f1-score	support
0	0.84	0.73	0.78	1021
1	0.77	0.87	0.81	1049
accuracy			0.80	2070
macro avg	0.80	0.80	0.80	2070
weighted avg	0.80	0.80	0.80	2070

Analyse générale du modèle XGBoost :

— Le modèle XGBoost fonctionne bien pour classifier les clients entre **Churn** (attrition) et **Not Churn** (non-attrition) avec un **taux élevé de prédictions correctes** sur les ensembles d'entraînement et de test.

— Le modèle présente une **grande précision**, mais il est important de prêter attention à des métriques détaillées comme la **Précision**, le **Rappel** et le **F1-Score** afin d'assurer un équilibre entre les classes.

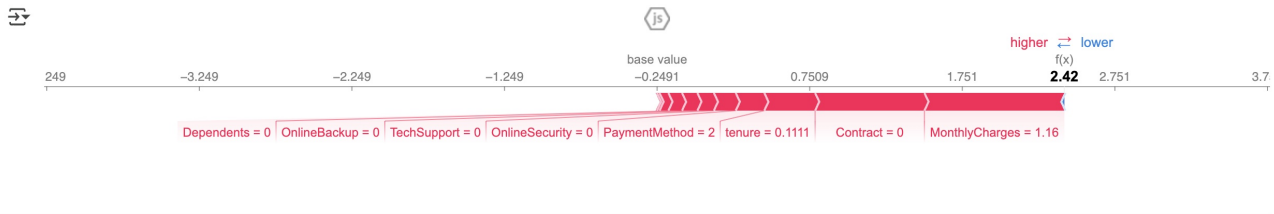
Points forts :

— **Bonne détection des clients susceptibles de résilier (Churn) :**
  — Le **taux de faux négatifs (FN)** est faible, ce qui signifie que le modèle omet rarement des clients à risque.
  — Le **rappel** est élevé, ce qui montre que le modèle est efficace pour détecter les clients à risque.
— **Pas de surapprentissage (overfitting) :**
  — Les performances sur les ensembles d'entraînement et de test sont relativement similaires.

Limites :

— **Nombre relativement élevé de faux positifs (FP) :**
  — Un nombre significatif de clients est faussement classé comme **Churn**, ce qui peut entraîner un gaspillage de ressources en ciblant des clients qui ne résilient pas réellement.

Le Resulat du SHAP Force Plot :



— **Base value :** -0.2491
— Cette valeur représente la moyenne des *log-odds* sur l'ensemble des données. Elle constitue le point de départ avant l'ajout des contributions des différentes caractéristiques.
— **Valeur prédite (f(x)) :** 2.42
— La probabilité que le client soit dans la classe **Churn** est calculée comme suit :

$$\text{Probability} = \frac{1}{1 + e^{-f(x)}} = \frac{1}{1 + e^{-2.42}} \approx 0.918$$

— **Conclusion :** Le modèle prédit avec une probabilité de 91.8% que ce client est dans la classe *Churn*.

Caractéristiques influentes (Higher = Churn)

*** Caractéristiques augmentant la probabilité de Churn (en rouge) :**

1. **MonthlyCharges = 1.16 :** Les frais mensuels élevés contribuent le plus à l'augmentation de la probabilité de Churn.

2. **Contract = 0 :** Le client possède un contrat *Month-to-month*, ce qui est fortement associé à une probabilité plus élevée de résiliation.
3. **Tenure = 0.1111 :** Le client utilise le service depuis peu de temps (environ 1.33 mois), ce qui augmente le risque de résiliation.
4. **PaymentMethod = 2 :** Ce mode de paiement (par exemple, paiement électronique) semble être lié à une probabilité plus élevée de Churn.
5. **OnlineSecurity = 0, TechSupport = 0, OnlineBackup = 0 :**
   — Le client n'utilise pas ces services à valeur ajoutée, ce qui augmente la probabilité de Churn.
6. **Dependents = 0 :** L'absence de personnes à charge peut rendre le client moins engagé dans le service.

*** Absence de caractéristiques réduisant la probabilité de Churn**

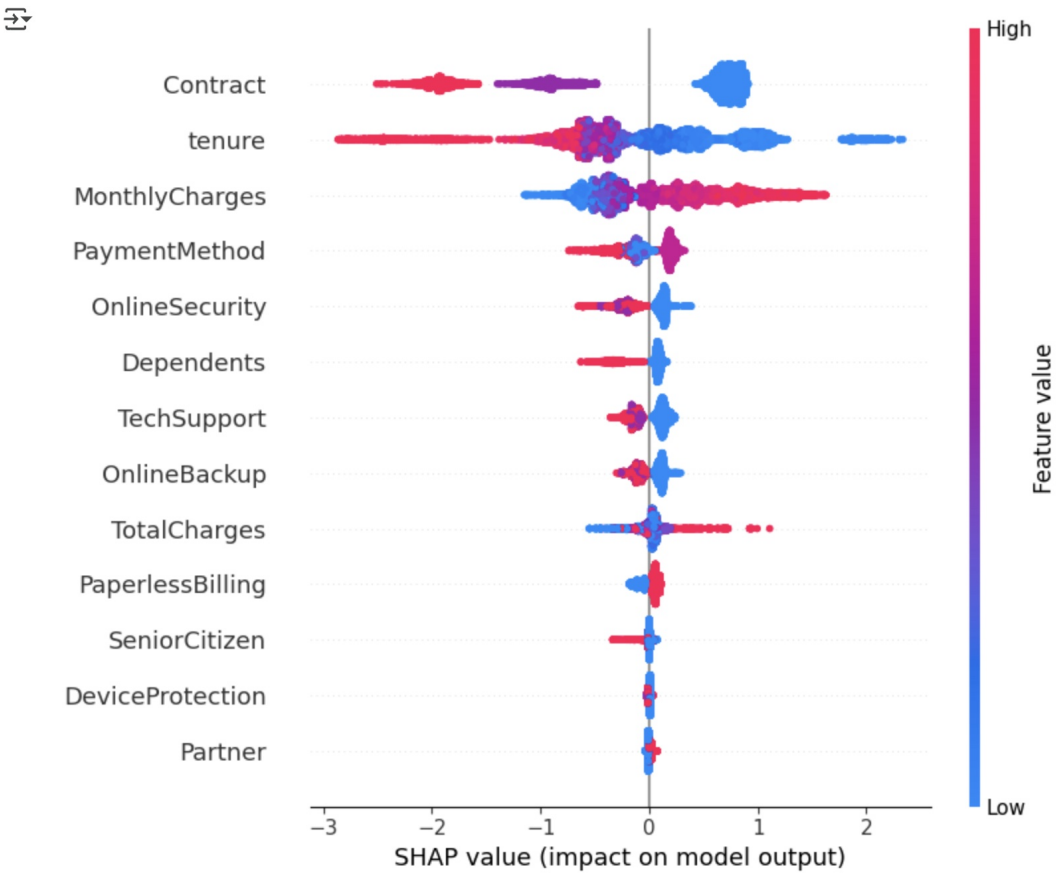
— Dans ce graphique, aucune caractéristique (en bleu) ne contribue à réduire la probabilité de Churn.

*** Synthèse**

— **Risque élevé de Churn (91.8%) :** Le client présente plusieurs facteurs de risque, tels qu'un contrat court, des frais mensuels élevés et un faible engagement.

— **Caractéristiques principales :**
   — *MonthlyCharges*, *Contract*, et *Tenure* sont les facteurs les plus importants augmentant la probabilité de Churn.

Le Resulat du SHAP Global Plot :



Analyse globale du modèle à l'aide de SHAP Summary Plot

*** Signification des axes**

— **Axe horizontal (SHAP value) :** La valeur SHAP indique l'impact de chaque caractéristique sur la sortie du modèle.

  — **SHAP > 0 :** La caractéristique augmente la probabilité que le client soit classé dans la classe **Churn**.

  — **SHAP < 0 :** La caractéristique réduit la probabilité que le client soit classé dans la classe **Churn**.

— **Couleurs (Feature Value) :**

  — **Bleu :** Valeurs faibles de la caractéristique.

  — **Rouge :** Valeurs élevées de la caractéristique.

*** Caractéristiques les plus importantes**

— Les caractéristiques sont triées par importance (de haut en bas).

— **Les trois caractéristiques les plus importantes :**

1. **Contract :**

   — Les clients avec des contrats longs (*Two year*, en rouge) ont une probabilité

plus faible de résilier (SHAP négatif).

— Les clients avec des contrats courts (*Month-to-month*, en bleu) ont une probabilité élevée de résilier (SHAP positif).

2. **Tenure :**
   — Les clients ayant une longue durée d'engagement (en rouge) sont moins susceptibles de résilier (SHAP négatif).
   — Les nouveaux clients (en bleu) ont un risque plus élevé de résiliation (SHAP positif).
3. **MonthlyCharges :**
   — Les clients avec des frais mensuels élevés (en rouge) sont plus susceptibles de résilier (SHAP positif).
   — Les clients avec des frais mensuels bas (en bleu) sont moins susceptibles de résilier.

*** Caractéristiques avec un impact moyen**

— Les caractéristiques telles que **PaymentMethod**, **OnlineSecurity**, et **Dependents** ont un impact modéré :

1. **PaymentMethod :** Certains modes de paiement (ex. manuel ou instable) augmentent le risque de résiliation.
2. **OnlineSecurity :** Ne pas utiliser de sécurité en ligne (en bleu) augmente la probabilité de résiliation.
3. **Dependents :** Les clients sans personnes à charge (en bleu) sont plus susceptibles de résilier.

*** Caractéristiques avec un faible impact**

— Les caractéristiques comme **PaperlessBilling**, **SeniorCitizen**, et **DeviceProtection** ont un impact moindre :

— **PaperlessBilling :** Les clients utilisant la facturation électronique ont une probabilité légèrement plus élevée de résiliation.

— **SeniorCitizen :** L'impact est peu significatif par rapport aux autres caractéristiques.

*** Synthèse globale**

— **Caractéristiques clés :** Les caractéristiques *Contract*, *tenure*, et *MonthlyCharges* ont le plus grand impact global sur les prédictions.

— **Tendances générales :**

— Les contrats longs et la durée d'engagement réduisent considérablement la probabilité de résiliation.

— Les frais mensuels élevés augmentent significativement le risque de résiliation.

*** Recommandations stratégiques**

1. **Cibler les clients à haut risque (*Contract = Month-to-month, tenure faible*) :**
   — Proposer des offres pour encourager la migration vers des contrats longs.
   — Offrir des incitations pour prolonger l'engagement.
2. **Améliorer la satisfaction des clients avec des frais élevés (*MonthlyCharges*) :**
   — Introduire des remises ou des services supplémentaires pour augmenter la perception de valeur.
3. **Renforcer les services à valeur ajoutée (*OnlineSecurity, TechSupport*) :**
   — Inciter les clients à souscrire à ces services pour réduire leur risque de résiliation.

*** Conclusion**

— Ce graphique SHAP montre que le modèle XGBoost a bien appris les caractéristiques importantes et les tendances de prédiction.

— Se concentrer sur des caractéristiques comme *Contract*, *tenure*, et *MonthlyCharges* peut aider l'entreprise à réduire efficacement le taux de résiliation (*Churn*).

3 DiCE : Comprendre comment fonctionne le modèle pré- dictif basé sur l'apprentissage automatique pour classer les clients potentiels en perte d'abonnement (RandomForestClassifier) - Projet 2

Qu'est-ce que DiCE ?

DiCE (Diverse Counterfactual Explanations) est une méthode basée sur l'apprentissage automatique qui génère des explications contrefactuelles pour un modèle prédictif. Elle propose des scénarios alternatifs montrant les modifications nécessaires pour obtenir une prédiction différente, tout en maximisant la diversité des solutions proposées.

DiCE est particulièrement utile pour rendre les modèles comme **RandomForestClassifier** plus interprétables en fournissant des exemples concrets et exploitables.

Objectif de DiCE ?

L'objectif principal de DiCE est de fournir des explications contrefactuelles compréhensibles et exploitables pour :

— Identifier les changements nécessaires pour modifier une prédiction,

— Augmenter la transparence des modèles complexes en rendant leurs décisions plus claires,

— Améliorer la confiance des utilisateurs dans les systèmes de décision,

— Détecter et atténuer les biais potentiels des modèles.

3.1 Explications contrefactuelles : Définition et Calcul

DiCE génère un ensemble de k exemples contrefactuels $\{c_1, c_2, \ldots, c_k\}$ en modifiant les valeurs des caractéristiques d'une instance originale x. Ces exemples doivent respecter deux contraintes fondamentales :

— **Proximité :** Les exemples contrefactuels doivent être proches de l'instance originale x pour garantir leur utilité.

— **Diversité :** Les exemples contrefactuels doivent être variés pour explorer différentes alternatives possibles.

3.2 Diversité via le Processus de Points Déterminantal (DPP)

La diversité est mesurée en utilisant un Processus de Points Déterminantal (DPP), défini comme suit :

$$\text{dpp\_diversity} = \det(K)$$

où K est une matrice définie par :

$$K_{i,j} = \frac{1}{1 + \text{dist}(c_i, c_j)}$$

avec $\text{dist}(c_i, c_j)$ représentant une métrique de distance entre les exemples contrefactuels c_i et c_j.

3.3 Proximité

La proximité est mesurée par la moyenne des distances entre chaque exemple contrefactuel c_i et l'instance originale x :

$$\text{Proximity} := -\frac{1}{k} \sum_{i=1}^{k} \text{dist}(c_i, x)$$

où $\text{dist}(c_i, x)$ est une métrique de distance (par exemple, la distance euclidienne).

3.4 Contraintes supplémentaires liées au monde réel

DiCE permet d'incorporer des contraintes spécifiques aux problèmes réels, comme :

— Des limites supérieures ou inférieures sur certaines caractéristiques (par exemple, un âge maximal ou minimal),

— Des restrictions sur les valeurs catégoriques (par exemple, les valeurs doivent appartenir à des catégories valides),

— La garantie de maintenir des relations logiques entre les caractéristiques (par exemple, un salaire élevé implique un âge supérieur à un seuil).

3.5 Fonction de perte de DiCE

Pour générer des explications contrefactuelles, DiCE optimise la fonction de perte suivante :

$$C(x) = \arg \min_{c_1,c_2,\ldots,c_k} \frac{1}{k} \sum_{i=1}^{k} \text{yloss}(f(c_i), y) + \lambda_1 \frac{1}{k} \sum_{i=1}^{k} \text{dist}(c_i, x) - \lambda_2 \text{dpp\_diversity}(c_1, c_2, \ldots, c_k)$$

— $\text{yloss}(f(c_i), y)$: Mesure la différence entre la prédiction du modèle $f(c_i)$ pour l'exemple contrefactuel c_i et le résultat souhaité y.

— λ_1 : Hyperparamètre qui pondère l'importance de la proximité.

— λ_2 : Hyperparamètre qui pondère l'importance de la diversité.

— $\text{dpp\_diversity}(c_1, c_2, \ldots, c_k)$: Mesure la diversité entre les exemples contrefactuels générés.

3.6 RandomForestClassifier : Définition et Fonctionnement

RandomForestClassifier est un algorithme d'ensemble basé sur des forêts aléatoires, utilisé pour les tâches de classification. Il combine plusieurs arbres de décision pour améliorer la précision et la robustesse des prédictions.

Ses principales caractéristiques sont :

— Utilisation d'un échantillonnage aléatoire des données et des caractéristiques pour réduire la variance,

— Combinaison des prédictions des arbres via le vote majoritaire pour la classification,

— Résistance aux données bruitées et réduction du risque de surapprentissage,

— Capacité à gérer des ensembles de données complexes avec de nombreuses caractéristiques.

3.7 Avantages de DiCE

— **Transparence :** Les explications contrefactuelles fournissent des exemples concrets qui aident les utilisateurs à comprendre les décisions du modèle.
— **Exploration multiple :** En générant des exemples variés, DiCE permet d'explorer différentes solutions possibles.
— **Flexibilité :** DiCE s'adapte aux contraintes spécifiques aux problèmes réels, rendant les explications plus pertinentes.

3.8 Application de RandomForestClassifier et DiCE sur l'annlyse du churn

Nous utilisons toujours l'ensemble de données du Projet 1.

3.9 Méthodologie

1. Charger et traiter les données
2. Entraînement du modèle RandomForestClassifier sur les données des clients.
3. Utilisation de DiCE pour génerer les explications contrefactuelles

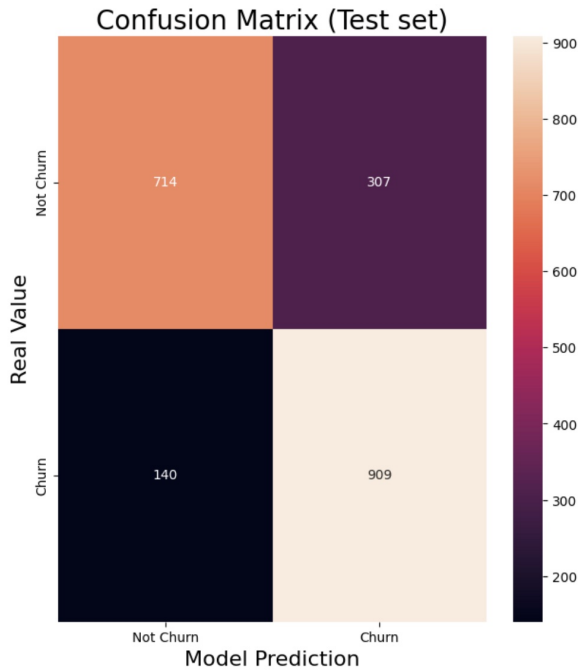
3.10 Le Resultat :

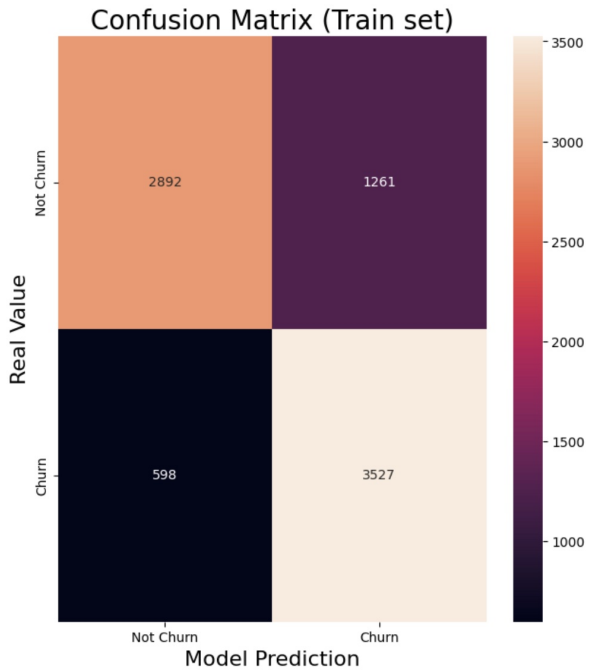
```
              precision    recall  f1-score   support

           0       0.84      0.70      0.76      1021
           1       0.75      0.87      0.80      1049

    accuracy                           0.78      2070
   macro avg       0.79      0.78      0.78      2070
weighted avg       0.79      0.78      0.78      2070

0.7840579710144927
Text(0.5, 1.0, 'Confusion Matrix (Train set)')
```





Résumé général du modèle RandomForestClassifier

*** Performances globales**

— **Précision (Accuracy) :** 78.4% sur le jeu de test.

— **F1-Score :** 76% pour la classe *Not Churn* et 80% pour la classe *Churn*.

— **Observation :** Le modèle fonctionne particulièrement bien pour détecter les clients susceptibles de résilier (*Churn*).

*** Points forts**

— **Rappel élevé pour la classe *Churn* (87%) :** Le modèle identifie efficacement la majorité des clients à risque de résiliation.

— **Bonne généralisation :** Les performances entre le jeu d'entraînement et le jeu de test sont relativement similaires.

*** Limitations**

— **Nombre élevé de Faux Positifs (307 cas) :** Une part importante des clients *Not Churn* est incorrectement prédite comme *Churn*, ce qui peut entraîner une utilisation inefficace des ressources.

— **Rappel faible pour la classe *Not Churn* (70%) :** Le modèle manque de capturer certains clients qui ne présentent pas de risque de résiliation.

*** Suggestions d'amélioration**

— Ajuster les hyperparamètres du modèle RandomForest pour réduire le nombre de Faux Positifs.

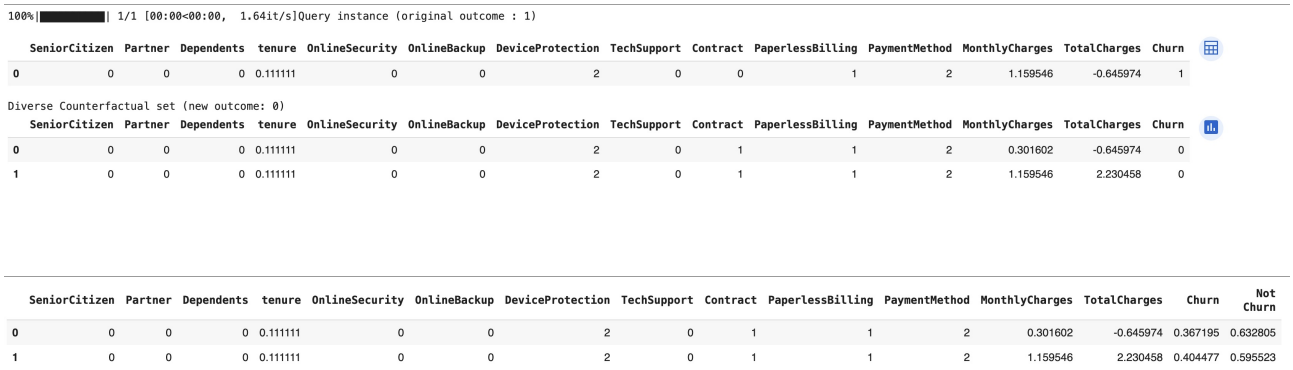
— Améliorer les caractéristiques utilisées ou affiner le seuil de décision afin d'augmenter la précision globale.

*** Conclusion**

— Le modèle **RandomForestClassifier** montre une forte capacité à identifier les clients à risque de résiliation (*Churn*).

— Toutefois, il est nécessaire de travailler sur la réduction des Faux Positifs pour diminuer le gaspillage de ressources et améliorer les performances globales.

Visualisation



100%|| 1/1 [00:00<00:00, 1.64it/s]Query instance (original outcome : 1)

	SeniorCitizen	Partner	Dependents	tenure	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	0	0	0	0.111111	0	0	2	0	0	1	2	1.159546	-0.645974	1

Diverse Counterfactual set (new outcome: 0)

	SeniorCitizen	Partner	Dependents	tenure	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	0	0	0	0.111111	0	0	2	0	1	1	2	0.301602	-0.645974	0
1	0	0	0	0.111111	0	0	2	0	1	1	2	1.159546	2.230458	0

	SeniorCitizen	Partner	Dependents	tenure	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn	Not Churn
0	0	0	0	0.111111	0	0	2	0	1	1	2	0.301602	-0.645974	0.367195	0.632805
1	0	0	0	0.111111	0	0	2	0	1	1	2	1.159546	2.230458	0.404477	0.595523

Analyse des résultats de DiCE

*** Instance originale**

— **Outcome initial :** 1 (*Churn*).

— **Valeurs des caractéristiques de l'instance originale :**

  — **SeniorCitizen :** 0

  — **Partner :** 0

  — **Dependents :** 0

  — **Contract :** 0 (*Month-to-month*)

  — **MonthlyCharges :** 1.159546

  — **TotalCharges :** -0.645974

— **Observation :** L'instance originale est prédite comme appartenant à la classe **Churn**.

*** Ensemble contrefactuel généré par DiCE**

— **Outcome modifié :** 0 (*Not Churn*).

— **Instances contrefactuelles générées :**

1. **Instance 1 :**
   — **Contract :** 1 (*One-year*)
   — **MonthlyCharges :** 0.301602
   — **TotalCharges :** -0.645974
2. **Instance 2 :**
   — **Contract :** 1 (*One-year*)
   — **MonthlyCharges :** 1.159546
   — **TotalCharges :** 2.230458

*** Analyse des résultats contrefactuels**

— **Contract :** Les deux instances contrefactuelles modifient la valeur de *Contract* de 0 (*Month-to-month*) à 1 (*One-year*). Cela montre que les contrats plus longs réduisent significativement le risque de résiliation.

— **MonthlyCharges :**
   — L'**Instance 1** réduit la valeur de *MonthlyCharges* de 1.159546 à 0.301602, indiquant qu'une réduction des frais mensuels peut diminuer le risque de résiliation.
   — L'**Instance 2** conserve la même valeur pour *MonthlyCharges*, mais modifie d'autres caractéristiques pour atteindre le résultat souhaité.

— **TotalCharges :**
   — L'**Instance 2** augmente la valeur de *TotalCharges* de -0.645974 à 2.230458. Cela suggère que des dépenses totales plus élevées sont associées à une probabilité plus faible de résiliation.

*** Conclusion**

— Le modèle **RandomForestClassifier** montre une forte capacité à identifier les clients à risque de résiliation (*Churn*), avec un rappel élevé pour cette classe.

— Les explications contrefactuelles générées par DiCE mettent en évidence les caractéristiques principales comme **Contract** et **TotalCharges**, qui influencent significativement les prédictions.

— Bien que les performances globales soient satisfaisantes, il est nécessaire de réduire le nombre de **Faux Positifs**, car cela peut entraîner une utilisation inefficace des ressources pour des clients qui ne sont pas réellement à risque.

— En optimisant les hyperparamètres du modèle et en affinant les caractéristiques, il serait possible d'améliorer davantage la précision et de réduire les erreurs.

— Enfin, les résultats contrefactuels fournis par DiCE offrent des pistes exploitables pour élaborer des stratégies d'affaires, telles que la promotion des contrats à long terme (*One-year*) ou l'encouragement à augmenter les dépenses totales (*TotalCharges*).

4 Grad-CAM : How to visualize class activation maps to bring Interpretability to Deep Learning - Projet 3

*** Qu'est-ce que Grad-CAM ? Objectif de Grad-CAM**

Grad-CAM (Gradient-weighted Class Activation Mapping) est une technique permettant de visualiser et d'expliquer comment les modèles d'apprentissage profond prennent des décisions, en particulier dans les tâches de reconnaissance d'images. Grad-CAM génère une carte thermique, identifiant les régions de l'image qui influencent le plus la décision du modèle.

L'objectif principal de Grad-CAM est :

— Comprendre la prise de décision du modèle d'apprentissage profond et vérifier s'il se base sur des caractéristiques pertinentes.

— Détecter et réduire les biais du modèle, en s'assurant qu'il ne s'appuie pas sur des caractéristiques indésirables.

— Aider à améliorer et ajuster les modèles, notamment dans les applications médicales, les véhicules autonomes et d'autres domaines.

*** Mécanisme de fonctionnement de Grad-CAM**

Grad-CAM se base sur le gradient de la prédiction finale par rapport à une couche convolutionnelle finale pour identifier les régions les plus importantes de l'image qui ont été utilisées pour prendre la décision. Les étapes principales sont les suivantes :

1. Calculer le gradient de la sortie de prédiction par rapport aux cartes de caractéristiques (feature maps) de la dernière couche convolutionnelle.
2. Calculer la moyenne des gradients pour chaque filtre afin de déterminer leur importance relative.
3. Combiner les cartes de caractéristiques après pondération.
4. Générer et afficher la carte thermique sur l'image d'origine.

*** Implémentation de Grad-CAM dans votre programme**

Notre programme utilise Pytorch pour implémenter Grad-CAM sur le modèle EfficientNet-B0 et Resnet50. Les étapes incluent :

1. Charger le modèle Xception avec des poids pré-entraînés sur ImageNet.
2. Identifier la dernière couche convolutionnelle du modèle.
3. Calculer le gradient de la sortie finale par rapport aux cartes de caractéristiques de la dernière couche convolutionnelle.
4. Calculer la moyenne des gradients, pondérer et combiner les cartes de caractéristiques.
5. Générer la carte thermique et la superposer sur l'image d'origine.

*** Le Resultat Image Entrée**

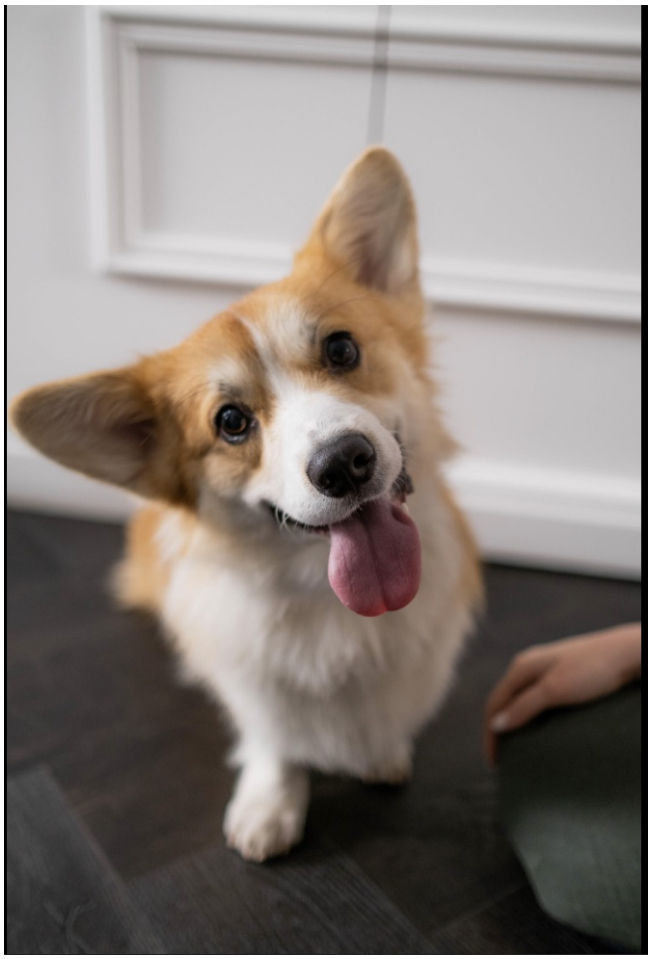
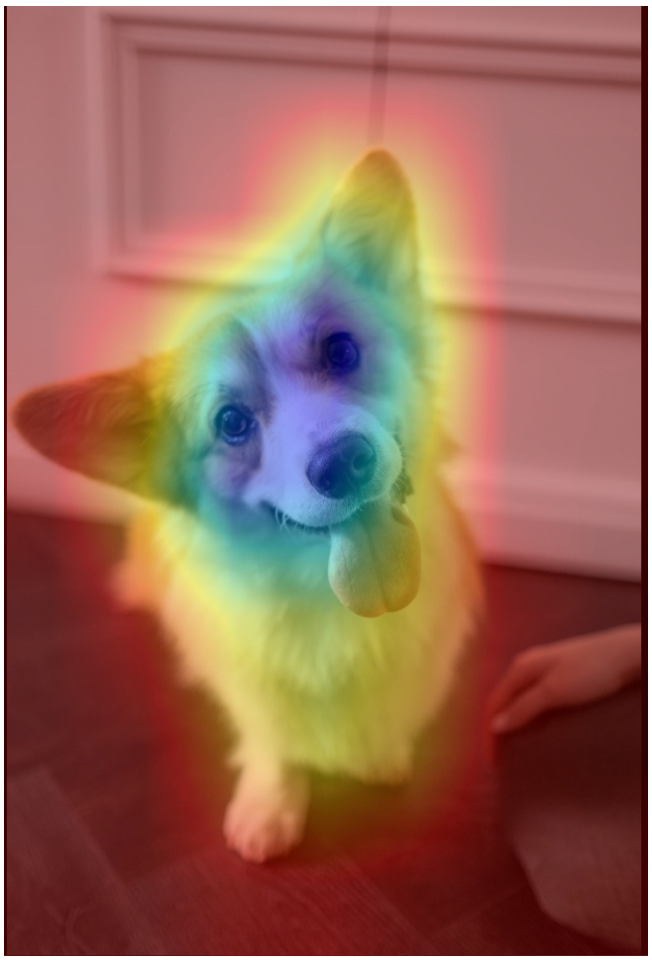


Image après la méthode Grad CAM



***Analyse des résultats**

Le modèle se concentre principalement sur le visage du chien Les zones **rouges/jaunes** (les plus importantes) apparaissent principalement autour des **yeux, du museau et de la bouche**. Ce sont des **caractéristiques clés** utilisées par de nombreux modèles de classification d'images pour reconnaître un chien.

Certaines zones activées sur les oreilles et l'arrière-plan Les **oreilles et le sommet de la tête** sont également en rouge, ce qui peut indiquer que les oreilles sont des **éléments distinctifs** pour la classification du chien. Une légère coloration rouge en haut de l'image peut être due à du **bruit** ou à l'influence de la **lumière de l'arrière-plan**.

La main présente dans l'image pourrait introduire du bruit On observe une **main** dans le coin droit de l'image, mais **elle n'est pas fortement mise en évidence par Grad-CAM**. Cela peut indiquer que **le modèle distingue bien le chien de la main**, ce qui est un bon signe de robustesse.

*** Conclusion**

Grad-CAM permet de mieux comprendre les décisions des modèles d'apprentissage

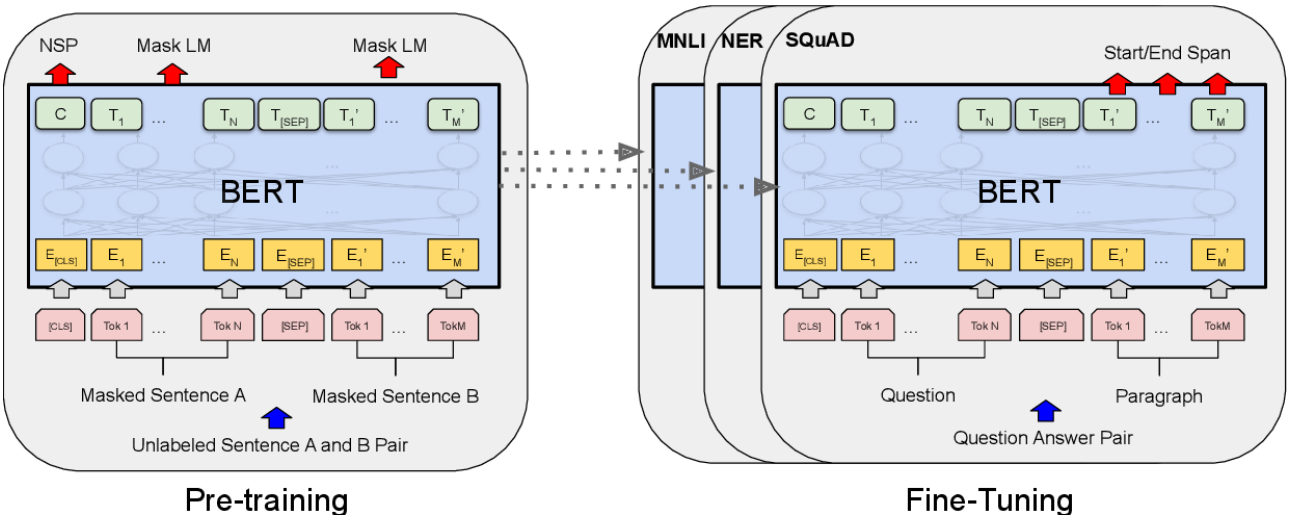
profond en mettant en évidence les régions clés d'une image. Votre programme implémente Grad-CAM avec le modèle Xception, aidant ainsi à identifier précisément les zones qui contribuent à la prédiction.

5 BERT et SQuAD pour l'interprétation et l'analyse des réponses à des questions en langage naturel - Projet 4

- Le modèle BERT

  (Azizah et al., 2023) Google a créé le modèle de traitement du langage naturel (NLP) appelé Bidirectional Encoder Representations from Transformers (BERT) en 2018. En utilisant un entraînement bidirectionnel, le modèle BERT dans le Transformer combine ensuite le contexte des couches gauche et droite. Le modèle BERT utilise l'architecture Transformer pour étudier la représentation des mots dans le contexte des phrases et des documents simultanément. BERT est un modèle d'apprentissage non supervisé, ce qui signifie qu'il est entraîné à partir de grandes quantités de données non étiquetées, afin que le modèle puisse apprendre automatiquement les motifs présents dans ces données. La capacité de BERT à comprendre le contexte du langage humain a conduit à des avancées majeures dans diverses tâches de traitement du langage naturel, telles que la compréhension du langage naturel, le traitement du langage naturel, et bien plus encore.

  Comme le montre la figure suivante, BERT utilise une stratégie novatrice de pré-entraînement et de fine-tuning. Le modèle BERT est d'abord entraîné sur des données massives non étiquetées lors de la phase de pré-entraînement, afin d'acquérir une connaissance générale du langage. Le modèle est ensuite ajusté pour des tâches spécifiques avec moins de données. Tous les paramètres sont ajustés pendant la phase de fine-tuning.



Pre-training Fine-Tuning

Chaque occurrence de l'entrée est précédée d'un symbole unique appelé [CLS], et [SEP] est un jeton séparateur spécial (comme une question et une réponse séparées). Grâce à cette méthode, BERT est capable d'appliquer les connaissances qu'il a acquises pendant la phase de pré-entraînement à certaines tâches. BERT est considéré comme l'état de l'art dans le domaine du NLP en partie grâce à cette méthode.

BERT utilise l'architecture Transformer, dont l'objectif principal est l'Attention. L'Attention détermine l'accent principal ou le contexte de la séquence ; ainsi, l'encodeur ajoutera également des mots-clés importants et des mots lors de la traduction entre les langues. L'Attention est une autre fonction utilisée pour mapper les requêtes, suivre les paires clé-valeur, et produire des données sous forme de vecteurs. L'équation de l'attention peut être vue dans l'Équation.

$$attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d_k}})V$$

Avec :

— Q est la matrice qui construit la requête (contient un vecteur pour chaque mot).

— K est la clé.

— V est la valeur elle-même.

— d_k est la dimension du vecteur de clé K.

— $\frac{QK^T}{\sqrt{d_k}}$ est l'étape pour calculer le poids d'attention, qui est le résultat du produit scalaire entre Q et K, divisé par la racine carrée de d_k.

— SoftMax est une des fonctions d'activation en apprentissage profond qui est utile pour les poids d'attention, car elle permet d'obtenir des valeurs comprises entre 0 et 1 (un type de probabilité).

► Méthode et code utilisés dans le projet

— Installation des Bibliothèques : transformers (pour utiliser BERT), captum (pour l'interprétabilité), pandas, matplotlib, seaborn, etc.
— Un modèle BERT pré-entraîné est utilisé pour la tâche de Question Answering avec le dataset SQuAD (transformers permet de charger et d'utiliser ce modèle).
— Prétraitement des Données (conversion du texte en tokens et passage des tokens dans le modèle pour obtenir des prédictions).
— captum est utilisé pour analyser l'importance des mots en fonction de leur contribution à la prédiction du modèle.
— Utilisation de seaborn et matplotlib pour représenter l'importance des mots dans la réponse.

▶ Les données testées

— Les données ont été mises à jour, le modèle analysera la question "Where is the Eiffel Tower located ?" en utilisant le texte "The Eiffel Tower is one of the most famous landmarks in the world, located in Paris, France.".
— Un autre exemple a été étudié avec la question "Who discovered gravity ?" et le texte "Gravity was discovered by Sir Isaac Newton when he observed an apple falling from a tree. This discovery led to the formulation of the laws of motion and universal gravitation.".

▶ Interprétation avec Captum

Cette méthode de Captum (LayerIntegratedGradients (LIG)) (Kokhlikyan et al., 2020) est utilisée pour attribuer des scores d'importance aux tokens du texte. Ces scores indiquent à quel point chaque token a contribué à la prédiction du modèle. Les scores d'importance sont visualisés sous forme de graphique à barres, où chaque barre représente un token du texte. La hauteur de la barre indique l'importance du token dans la prédiction.

Les tokens les plus importants pour la prédiction sont ceux qui sont directement liés à la réponse, c'est-à-dire "paris" et "france" pour le premier exemple. Ces tokens ont reçu les scores d'importance les plus élevés, ce qui signifie qu'ils ont joué un rôle crucial dans la décision du modèle.

Le modèle BERT a correctement identifié la réponse à la question en se basant sur les tokens pertinents du texte et l'utilisation de Captum a permis de comprendre comment le modèle a pris sa décision en attribuant des scores d'im-

portance aux tokens. Cela montre que le modèle se concentre sur les mots clés qui sont directement liés à la réponse.

6 La Propagation de Pertinence par Couche (Layer-wise Relevance Propagation,LRP) - Projet 5

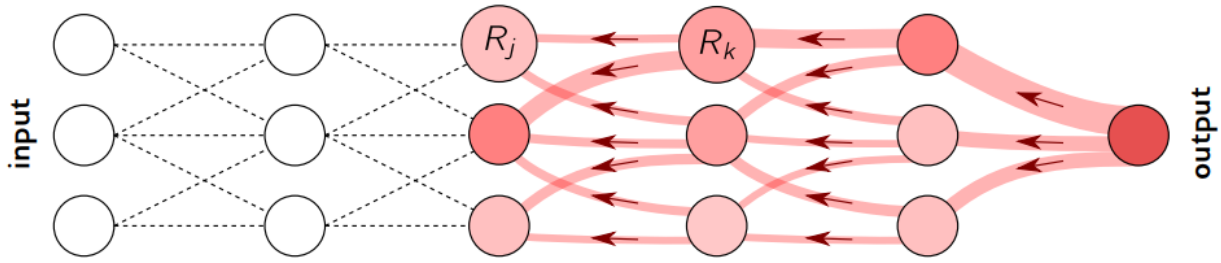
► Layer-wise Relevance Propagation

(Montavon et al.) Est une technique d'explication applicable aux modèles structurés en réseaux de neurones, où les entrées peuvent être, par exemple, des images, des vidéos ou du texte. LRP fonctionne en propageant la prédiction $f(x)$ à travers le réseau de neurones de manière rétrograde, en utilisant des règles de propagation locales spécialement conçues.

La procédure de propagation mise en œuvre par LRP est soumise à une propriété de conservation, selon laquelle ce qui a été reçu par un neurone doit être redistribué à la couche inférieure en quantité égale. Ce comportement est analogue aux lois de conservation de Kirchhoff dans les circuits électriques et est partagé par d'autres travaux sur les explications. Soient j et k des neurones situés à deux couches consécutives du réseau de neurones. La propagation des scores de pertinence $(R_k)_k$ à une couche donnée vers les neurones de la couche inférieure est réalisée en appliquant la règle suivante :

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}}$$

La quantité z_{jk} modélise la mesure dans laquelle le neurone j a contribué à rendre le neurone k pertinent. Le dénominateur sert à garantir la propriété de conservation. La procédure de propagation se termine une fois que les caractéristiques d'entrée ont été atteintes. Si l'on utilise la règle ci-dessus pour tous les neurones du réseau, il est facile de vérifier la propriété de conservation par couche $\sum_j R_j = \sum_k R_k$, et par extension la propriété de conservation globale $\sum_i R_i = f(x)$ La procédure globale de LRP est illustrée dans la figure suivante :



Bien que LRP diffère clairement de l'approche de décomposition de Taylor simple, nous observerons que chaque étape de la procédure de propagation peut être modélisée comme une décomposition de Taylor spécifique.

LRP a été appliqué pour découvrir des biais dans les modèles et ensembles de données couramment utilisés en apprentissage automatique (ML). Il a également été utilisé pour extraire de nouvelles informations à partir de modèles ML fonctionnant bien, par exemple dans la reconnaissance des expressions faciales. LRP a permis de trouver des caractéristiques pertinentes pour la localisation de sources audio, d'identifier des points d'intérêt dans des traces de canaux latéraux, et de détecter des motifs EEG expliquant les décisions dans les interfaces cerveau-ordinateur. Dans le domaine biomédical, LRP a été utilisé pour identifier des caractéristiques spécifiques aux sujets dans les modèles de marche, pour mettre en évidence des structures cellulaires pertinentes en microscopie, ainsi que pour expliquer des prédictions thérapeutiques. Enfin, une extension appelée CLRP a été appliquée pour mettre en lumière des sections moléculaires pertinentes dans le contexte de l'évaluation des interactions protéine-ligand (Hochuli et al., Sep 2018).

- Explication du Code

  Le code du projet met en œuvre la Propagation de Pertinence (LRP) sur un réseau de neurones entièrement connecté (FCNN) composé d'une couche d'entrée (i), une couche cachée (j) et une couche de sortie (k). Son but est de vérifier que LRP respecte les propriétés fondamentales : **Positivité** et **Conservativité**.

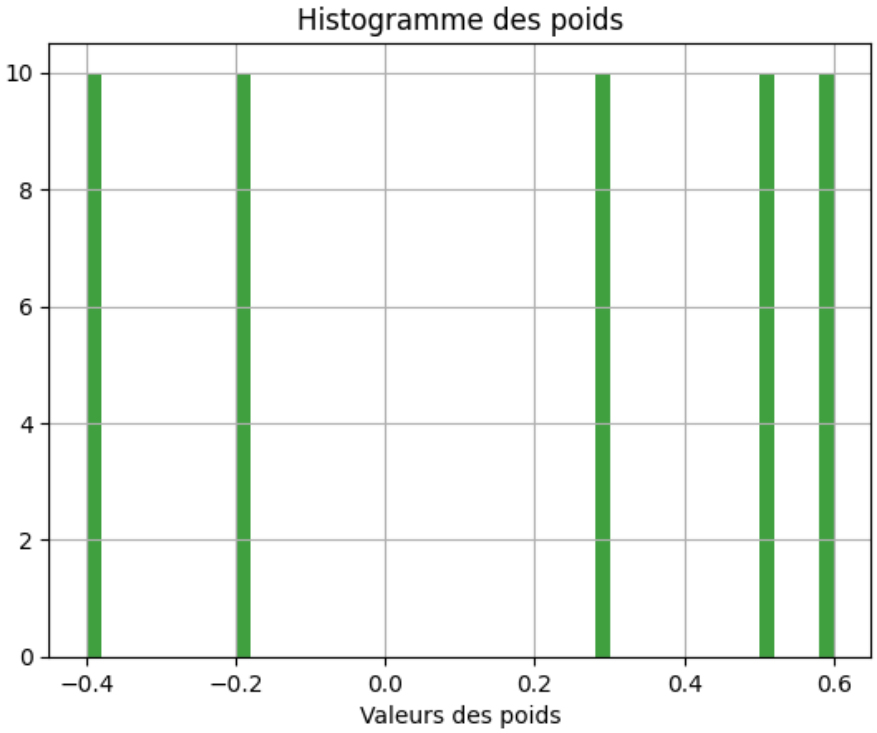
- Interprétation sur des nouvelles entrées

  Après un définition des nouveaux poids et entrées on a constaté que le test de positivité garantit que chaque pertinence calculée pour chaque neurone

reste positive. Cela est conforme à l'hypothèse selon laquelle LRP ne génère pas de pertinences négatives, ce qui signifie que chaque neurone contribue de manière additive à la sortie du réseau. Comme le test a été validé, cela prouve que les relevances sont bien distribuées sans valeurs incohérentes.

Le test de conservation des pertinences a réussi, la valeur de sortie du réseau (11.1) dans notre exemple est en cohérence avec la conservation de la pertinence.

L'histogramme montre la distribution des poids du réseau. On observe que certains poids sont positifs et d'autres négatifs, ce qui peut influencer l'activation des neurones cachés et, par conséquent, la répartition des pertinences.



Les résultats montrent que l'algorithme LRP est correctement implémenté et respecte les propriétés fondamentales de conservation et de positivité. Cela signifie que les relevances calculées peuvent être utilisées pour interpréter les contributions des entrées aux décisions du réseau de manière fiable.

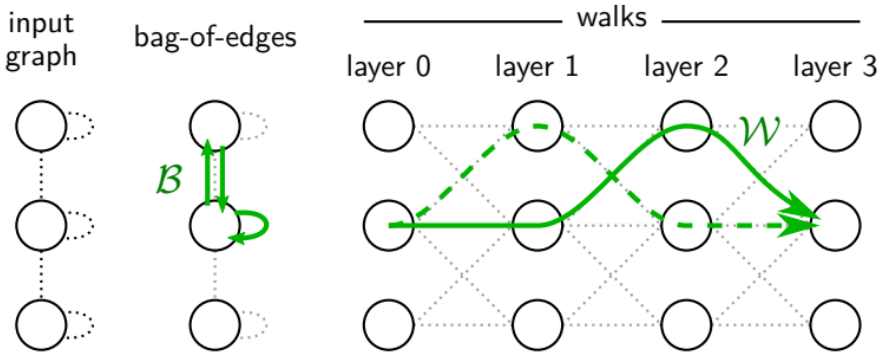
7 GNN-LRP - Projet 6

▶ Expliquer les GNN en pratique

Le développement de Taylor d'ordre supérieur est simple et mathéma-

tiquement fondé. Cependant, l'extraction systématique des dérivées d'ordre supérieur d'un réseau de neurones est difficile et ne s'adapte pas aux modèles complexes.

Pour répondre à cette première limitation, nous introduisons le concept de marche (walk) W, que nous définissons comme une séquence ordonnée d'arêtes connectant des nœuds dans des couches consécutives du réseau de neurones à graphes (GNN). La relation entre un ensemble d'arêtes (bag-of-edges) et les marches est illustrée pour un graphe simple.



Parce que chaque marche correspond à un ensemble spécifique d'arêtes (bag-of-edges), une explication basée sur les marches hérite de toutes les informations contenues dans une explication basée sur un ensemble d'arêtes. En particulier, il est toujours possible de retrouver l'explication par ensemble d'arêtes à partir d'une explication basée sur les marches. Cependant, l'utilisation des marches présente deux avantages supplémentaires :

Premièrement, une explication basée sur les marches fournit plus d'informations sur la manière dont les multiples couches du GNN ont été utilisées pour arriver à la prédiction. Par exemple, comme illustré dans la figure, elle peut révéler si la transmission de messages entre deux nœuds a eu lieu dans les premières ou les dernières couches du GNN.

Deuxièmement, le fait que les marches se connectent plus directement à la structure du GNN confère des avantages pratiques importants pour le calcul des explications.

Pour simplifier la présentation de ces algorithmes, nous introduisons une nouvelle variable $\hat{\Delta} \leftarrow (\Delta, ..., \Delta)$ qui distingue les arêtes apparaissant à différentes couches du GNN. Nous exprimons ensuite la sortie du GNN comme

une fonction de cette entrée étendue, c'est-à-dire $f(\hat{\Delta})$. Nous adoptons également une notation basée sur les nœuds, où les marches sont représentées par la séquence de nœuds qu'elles traversent de la première couche à la couche supérieure, par exemple $W = (..., J, K, L, ...)$. Les lettres J,K,L désignent des nœuds dans des couches consécutives, et les points de suspension $'...'$ servent de placeholder pour les nœuds situés au début et à la fin de la marche. Nous désignons en outre par $\hat{\lambda}_{JK}$ l'élément de $\hat{\Delta}$ représentant la connexion entre le nœud J et le nœud K. (Schnake et al., 2020)

► Explication du code

La première partie du code consiste à définir les fonctions :

— relevance_curves_xy_computation : calcule des courbes de pertinence selon les coordonnées des nœuds dans le graphe.

— compute_walks : génère des marches de longueur 3 en parcourant la matrice d'adjacence.

— set_graph_layout : crée un graphe et génère un layout pour la visualisation des nœuds.

La deuxième partie du code génère un graphe à échelle avec le modèle de Barabási-Albert (BA), une méthode bien connue pour la génération de graphes avec une distribution de degré qui suit une loi de puissance. Ce genre de graphe est utilisé pour modéliser des réseaux complexes comme les réseaux sociaux ou les réseaux de citation. Ensuite, il affiche ces graphes à l'aide de Matplotlib.

La 3ème partie du code définit un modèle de Graph Neural Network (GNN) basé sur les matrices d'adjacence, avec un passage avant (forward pass) et des méthodes de calcul de la LRP (Layer-wise Relevance Propagation).

La 4ème partie du code concerne l'entraînement du modèle GNN et la 5ème partie du code concerne le test du modèle GNN après l'entraînement.

La 6ème partie du code concerne l'explication des prédictions du modèle GNN à l'aide de la méthode LRP (Layer-wise Relevance Propagation), suivie de l'affichage des courbes de relevance pour les graphes testés.

► Interprétation des résultats après une modification des entrées

On a refait le code complet avec les modifications des données :

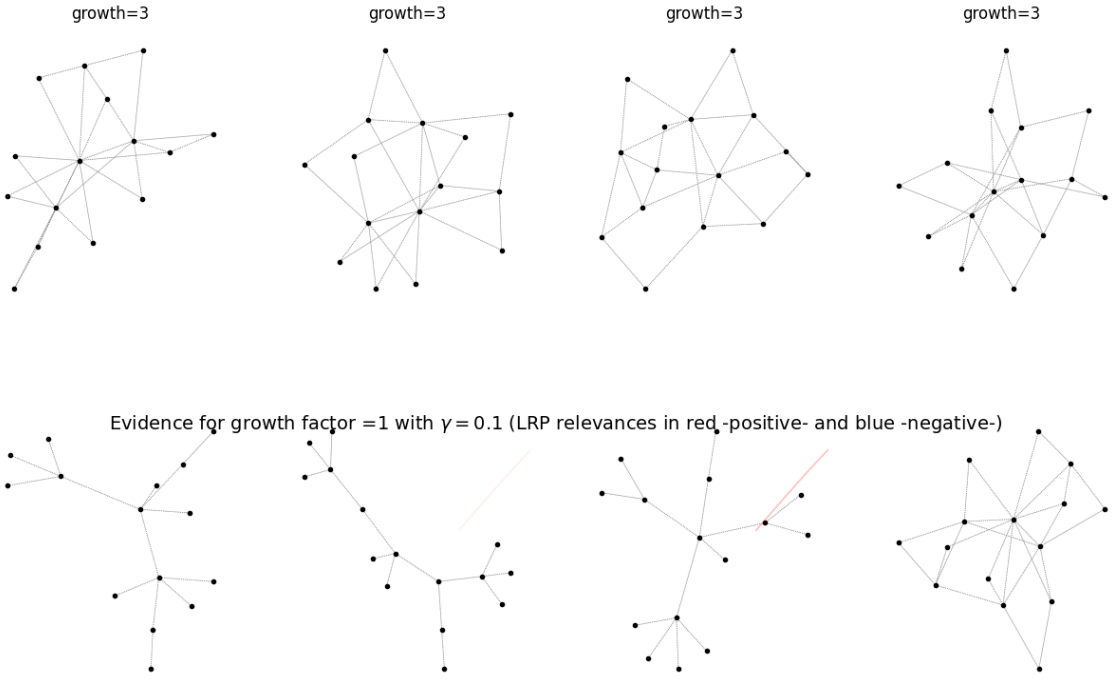
— Modification des coordonnées des nœuds pour les courbes de pertinence.

— Modification de la matrice d'adjacence pour les marches de longueur 3.
— Un autre layout pour la disposition des nœuds dans le graphe.

Les résultats des coordonnées x et y des courbes de pertinence montrent des valeurs qui varient de manière lissée, créant une courbe autour des nœuds de la marche et suivent un modèle sinusoïdal ou linéaire qui représente la pertinence des marches dans le graphe. Les valeurs convergent progressivement vers un point d'équilibre, montrant ainsi la continuité entre les trois nœuds de la marche.

Les marches calculées entre les nœuds du graphe sont correctement générées à partir de la matrice d'adjacence. Et Les coordonnées retournées pour la disposition des nœuds dans le graphe avec le layout Fruchterman-Reingold indiquent où chaque nœud du graphe est placé dans l'espace 2D. Ces positions sont calculées de manière à minimiser les croisements d'arêtes et à optimiser la clarté visuelle du graphe.

Chaque graphique généré montre le réseau échelle-free avec des relevances visuellement expliquées. Les arêtes entre les nœuds sont tracées en fonction de leur contribution à la prédiction, et cette représentation visuelle permet de comprendre facilement pourquoi le modèle a fait une prédiction donnée.



Ce projet montre comment utiliser un modèle GNN pour des tâches de prédiction sur des graphes, tout en offrant des moyens d'expliquer les décisions

du modèle à travers la méthode LRP.

8 Conclusion

Ce projet a permis d'explorer plusieurs méthodes d'Intelligence Artificielle Explicable (XAI) pour interpréter et expliquer les décisions des modèles d'apprentissage automatique. À travers des techniques comme SHAP, DiCE, Grad-CAM, et LRP, nous avons pu analyser en détail les prédictions de modèles complexes tels que XGBoost, RandomForestClassifier, et les réseaux de neurones profonds. Ces méthodes ont montré leur efficacité pour identifier les caractéristiques les plus influentes dans les prédictions, détecter les biais potentiels, et fournir des explications compréhensibles aux utilisateurs.

Dans le contexte de l'analyse du churn, nous avons constaté que des facteurs comme la durée du contrat, les frais mensuels, et la durée d'engagement des clients jouent un rôle crucial dans la prédiction du risque de résiliation. Les explications contrefactuelles générées par DiCE ont également permis de proposer des stratégies concrètes pour réduire le taux de churn, comme la promotion de contrats à long terme ou l'offre de services supplémentaires pour augmenter la satisfaction client.

Enfin, l'application de techniques d'interprétabilité comme Grad-CAM et LRP a permis de mieux comprendre les décisions des modèles d'apprentissage profond et des réseaux de neurones graphiques, en mettant en évidence les régions ou les nœuds les plus pertinents pour les prédictions. Ces résultats montrent que l'explicabilité est un outil puissant pour améliorer la transparence et la confiance dans les systèmes d'IA, tout en offrant des pistes pour optimiser les modèles et les stratégies d'affaires.

En conclusion, ce projet souligne l'importance de l'explicabilité dans l'IA moderne, en montrant comment elle peut être utilisée pour prendre des décisions plus éclairées, réduire les biais, et améliorer les performances des modèles dans des domaines variés.

Références

S. F. N. Azizah, H. D. Cahyono, S. W. Sihwi, and W. Widiarto. Performance analysis of transformer based models (bert, albert, and roberta) in fake news detection. *International Conference on Information and Communications Technology (ICOIACT)*, 2023.

Hochuli, J., Helbling, A., Skaist, T., Ragoza, M., Koes, and D.R. Visualizing convolutional neural network protein-ligand scoring. *Journal of Molecular Graphics and Modelling 84*, Sep 2018.

N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum : A unified and generic model interpretability library for pytorch. *Facebook AI*, 2020.

G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Muller. Layer-wise relevance propagation : An overview. *Université de Berlin.*

T. Schnake, O. Eberle, J. Lederer, S. N. K. T. Schutt, K.-R. Muller, and G. Montavon. Higher-order explanations of graph neural networks via relevant walks. 2020.