

# IAditMoi

Delemeasure Gabriel - Dufour Benoît - Beauconsin Enzo - Dubouloy Grégory

## Rapport de projet technique

Github Repository

<https://github.com/Bioz/IAditMoi>

Vidéo de démonstration

[Google Drive vidéo](#)

### **I. Rappel des objectifs du projet**

#### Explication de modèle de machine learning et mesures de qualités

1er objectif: Comprendre a minima comment marche un modèle simple d'apprentissage et comment il est possible de l'implémenter en partant de bibliothèques Python connues comme scikit-learn. On considérera pour cela des données simples issues du UCI Machine Learning repository (<https://archive.ics.uci.edu>).

2ème objectif: Découvrir les principales méthodes pour expliquer un modèle. On s'intéressera dans ce projet exclusivement aux méthodes dites post-hoc et agnostiques du modèle comme SHAP (<https://shap.readthedocs.io/en/latest/>) et LIME. Il existe de nombreux notebooks Python autour de la thématique de l'explicabilité (<https://github.com/kyaiooiayk/Explainable-AI-xAI-Notes>).

3ème objectif: Faire une liste de plusieurs mesures de qualité d'une explication et de chercher éventuellement des implémentations qui pourraient déjà exister.

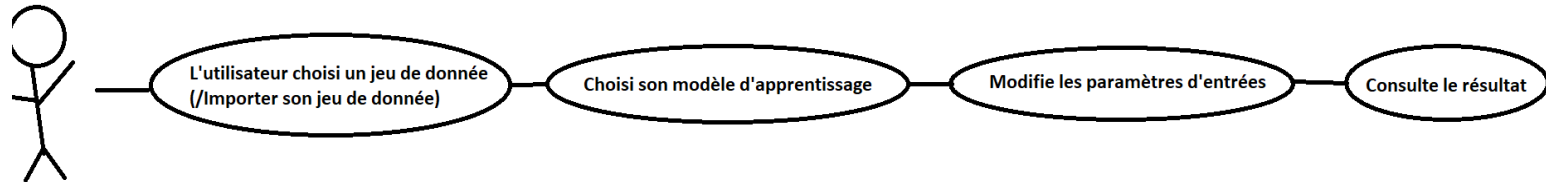
4ème objectifs: définir une interface web permettant de visualiser pour un modèle et un jeu de données les explications et leur qualité au sens des mesures identifiées précédemment. L'interface doit être interactive et très visuelle pour permettre de faire des démonstrations. Le code doit être modulaire de façon à permettre de faire évoluer les fonctionnalités avec d'autres modèles, d'autres outils d'explicabilités et d'autres mesures de qualité ou visualisations.

### **Au début ?**

Tout d'abord il a fallu commencer par définir les cas d'utilisations et choisir les technos que l'on allait utiliser (en sachant que python possède des librairies pour les model de machine learning). Ensuite essayer de définir une architecture selon notre cas d'utilisation, cette réflexion nous à amener à une structure avec 5 composants principaux, la gestion des

datas, la gestion des modèles (entraînement et prédiction), l'affichage/création des graphs d'explicabilités puis l'interface.

## II. Scénario d'utilisation



2 -L'utilisateur choisi parmi la liste des modèles d'apprentissage celui qu'il veut tester  
Une fois le modèle et les données choisi, le modèle s'entraîne avec les données sélectionnée

3- Une fois l'entraînement terminé l'utilisateur est face à une liste de paramètres qu'il peut modifier, ces modifications feront office de nouvelle donnée et c'est avec ces données que le modèle fera une prédiction.

4-Sous forme de graphique l'utilisateur consulte la prédiction, les explications ainsi que les mesures de qualités.

### Une prédiction c'est quoi ?

On veut poser une question à notre IA, est-ce qu' avec toutes les données que je te donne sur un individu, ce dernier pourra devenir joueur professionnel de volley.

En s'entraînant sur des données concrètes (taille poids de joueur pro) l'IA pourra faire une prédiction et dire si oui ou non l'individu pourra devenir joueur professionnel de volley

## III. Technologies utilisées

Le produit final est une application web codée en python. Nous avons choisi ce langage pour les librairies shap et scikit-learn qui sont très pratiques et complètes pour le travail qui nous était demandé. Désirant de ne pas mélanger les technos nous voulions trouver un moyen de créer une interface web avec python, c'est ainsi que nous somme tombé sur la lib streamlit (<https://docs.streamlit.io>) qui s'est avéré très intuitif et facil à utiliser pour un résultat convainquant visuellement.

Utilisation du framework StreamLit, considéré comme un framework léger pour développer des applications web interactive orienté donnée. Transforme du code python en interface utilisateur.

Site : [StreamLit](https://streamlit.io)

Utilisation de la bibliothèque python Scikit-Learn, bibliothèque utilisé pour le Machine Learning offrant des outils pour construire entraîner et évaluer des modèles prédictifs  
Site : [Scikit-learn](https://scikit-learn.org/)

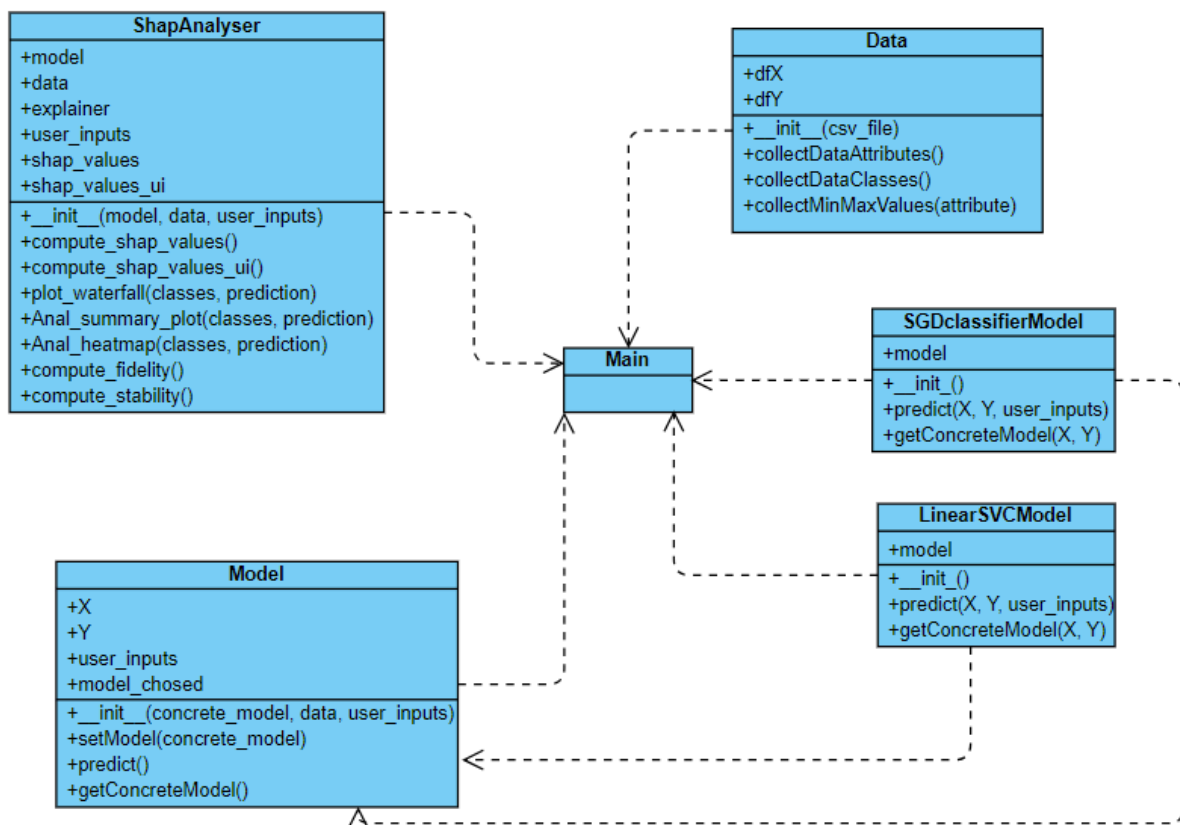
Utilisation de la bibliothèque Pandas pour manipuler et analyser des données, en particulier des données structurées  
Site : [Pandas](https://pandas.pydata.org/)

Utilisation de la bibliothèque SHAP pour expliquer les prédictions des modèles de Machine Learning avec différents plots obtenues grâce au SHAP values.  
Site : [SHAP](https://shap.readthedocs.io/)

Utilisation de la bibliothèque Matplotlib (nécessaire pour les plots SHAP)  
Site : [Matplotlib](https://matplotlib.org/)

## IV. Comment marche l'app

### Diagramme de Classe





La classe ShapAnalyzer facilite l'analyse et la visualisation des explications des prédictions d'un modèle grâce à SHAP. On crée un objet SHAPAnalyzer dans le main, qui prend en entrée un modèle entraîné, les données d'analyse et des données utilisateur. Elle initialise un explainer SHAP dont on se sert pour calculer puis stocker les valeurs SHAP. On utilise ensuite ces valeurs calculées pour mettre en place les graphes.

Résumé des plots SHAP :

Les outils de visualisation SHAP sont essentiels pour comprendre les prédictions des modèles d'apprentissage automatique en expliquant l'impact des caractéristiques sur les résultats.

Parmi ces outils, le *waterfall plot* est particulièrement utile pour analyser les contributions des caractéristiques pour une observation individuelle. Il permet de comprendre comment chaque caractéristique influe sur la transition entre la valeur de base (baseline) et la valeur prédite. Par exemple, dans un modèle de crédit, on peut expliquer pourquoi un client spécifique présente une probabilité élevée de défaut de paiement. Le graphique montre les caractéristiques qui contribuent le plus à cette prédiction, comme un revenu faible, un score de crédit bas ou un jeune âge.

Le *summary plot*, quant à lui, offre une vue d'ensemble globale des caractéristiques importantes sur l'ensemble des observations. Ce graphique montre non seulement l'importance des caractéristiques mais aussi la distribution des contributions, permettant de détecter des schémas globaux dans les données. Par exemple, dans un modèle de satisfaction client, ce type de visualisation peut révéler que les caractéristiques les plus influentes sont le temps d'attente, la qualité du produit et le prix, avec des temps d'attente élevés ayant un impact négatif sur la satisfaction.

Enfin, le *heatmap plot* permet de visualiser les interactions entre caractéristiques et leurs contributions globales. Il est particulièrement utile pour repérer les dépendances complexes entre les variables. Par exemple, dans un modèle de prédiction de la valeur immobilière, une heatmap peut révéler qu'une grande superficie combinée à une localisation prisée a une forte influence positive sur la valeur prédite.

Ces visualisations se complètent en répondant à des objectifs distincts : le *waterfall plot* permet une analyse approfondie au niveau individuel, le *summary plot* met en évidence les tendances globales, et le *heatmap plot* explore les interactions complexes entre les caractéristiques. Elles permettent la création de modèles compréhensibles qui renforcent la confiance que peuvent avoir les utilisateurs et contribuent à l'identification d'éventuels biais

#### D. Mesures de qualités

Fidélité (Faithfulness) Évalue si l'explication reflète fidèlement le fonctionnement réel du modèle. Pour ce faire on calcul des prédictions approximées avec les shap values puis on récupère les prédictions de base. On s'assure de remplacer les champs textes par des valeurs numériques puis on calcule un coefficient de fidélité (coefficient de corrélation) grâce à la matrice de corrélation. Si coef proche de 1 on explique bien les prédictions du modèle, si proche de 0 on arrive pas à bien

expliquer. Pour vulgariser on fait la prédiction puis on calcule les shap values puis à partir de ses shap values on recalcule des prédictions et on les compare aux originales (Méthode `compute_fidelity()` dans `shapAnalyser`).

[source fidélité](#)

La stabilité : c'est la méthode `compute_stability()` dans `ShapAnalyser`

C'est la qualité qui mesure la stabilité d'un modèle de machine learning

Les modèles doivent rester stables face aux variations, pour mesurer cela on regarde la différence entre nos données et des données légèrement altérées.

[source stabilité](#)

## E. Interface

Finalement, nous avons donc une interface qui nous permet d'instrumentaliser concrètement toutes les choses précédentes avec comme mot d'ordre la modularité. On a tout d'abord une page d'accueil où on demande la sélection d'un fichier csv conforme contenant un jeu de données, csv par ailleurs modifiable par la suite si besoin.

Une fois cela fait on arrive donc sur la page principale qui va contenir tout ce qui nous intéresse : on aura à gauche dans la sidebar tous les attributs du jeu de donnée qui sont générés de manière modulaire en fonction du csv en paramètre, pour chaque attribut on aura un slider permettant de modifier leur valeur, minimum et maximum définis par les min et max respectifs de chaque attribut du JDD, que l'utilisateur peut donc manipuler pour définir ses propres paramètres.

On a ensuite le centre de la page qui va contenir ce que contenait déjà la page d'accueil afin de permettre de changer de jeu de données comme annoncé précédemment sans avoir à relancer la page.

On a plus bas un tableau généré encore une fois de manière modulaire en fonction des attributs contenus dans le JDD avec pour chacun d'entre eux la valeur paramétrée à l'aide des sliders de la sidebar. Suite à ça nous avons un nouveau tableau, toujours généré de manière modulaire selon le JDD qui va contenir les différentes possibilités de résultats pour les prédictions.

Maintenant on passe donc au choix du modèle de Machine Learning, certains seront plus précis que d'autres et inversement d'un JDD à l'autre donc il était important qu'on ajoute cette sélection au sein de l'interface qui agit de manière dynamique, encore une fois nul besoin de redémarrer quoi que ce soit, on a ajouté en plus de cela la précision de prédiction du modèle sélectionné sur le JDD actuel puis pour finir la prédiction du modèle par rapport aux valeurs choisies

Pour finir nous avons donc nos différents plots SHAP expliqués au point précédent.

Un dernier point important à mentionner est que nous avons pris la décision de faire l'entièreté de cette interface en anglais non seulement pour l'homogénéité avec le contenu des JDD qui sont souvent en anglais mais aussi pour permettre un accès plus global à notre application que si nous nous étions limité au français, à voir un jour éventuellement pour une traduction automatique selon la langue système de l'utilisateur ou autre.

### StreamLit, ses limites

Présenté comme la solution parfaite, streamlit est un framework pour concevoir des interfaces d'app-web, cependant nous nous sommes vite approchés de certaines limites. Si on se réfère à notre cas d'usage, la première chose que fait un utilisateur en arrivant sur l'app, c'est d'importer son jeu de données sous la forme CSV.

Dans notre réflexion, l'idéal aurait été de dédier une page à cet usage, ce qui est possible car on peut créer plusieurs pages avec streamlit. Cependant, à partir du moment où il existe au moins deux pages sur l'app, il devient impossible de cacher la barre de navigation à droite.

Par conséquent, sur notre page dédiée à l'import du jeu de données, il aurait été possible de directement naviguer sur la page des résultats sans avoir importé les données. Il aurait été possible de faire une vérification sur la page des résultats pour renvoyer sur la première page dans le cas où les données ne sont pas trouvés mais ce n'est pas le rendu que nous voulions obtenir.

Les deux seules méthodes présentes dans la doc concernant la sidebar

<b>position</b> ( <i>"sidebar" or "hidden"</i> ) keyword-only	The position of the navigation menu. If <code>position</code> is <code>"sidebar"</code> (default), the navigation widget appears at the top of the sidebar. If <code>position</code> is <code>"hidden"</code> , the navigation widget is not displayed.  If there is only one page in <code>pages</code> , the navigation will be hidden for any value of <code>position</code> .
<b>expanded</b> ( <i>bool</i> ) keyword-only	Whether the navigation menu should be expanded. If this is <code>False</code> (default), the navigation menu will be collapsed and will include a button to view more options when there are too many pages to display. If this is <code>True</code> , the navigation menu will always be expanded; no button to collapse the menu will be displayed.  If <code>st.navigation</code> changes from <code>expanded=True</code> to <code>expanded=False</code> on a rerun, the menu will stay expanded and a collapse button will be displayed.

Dans `position()`, on voit qu'il est possible de masquer la sidebar UNIQUEMENT si il n'existe qu'une page dans la structure du projet.

Pour que le projet puisse fonctionner avec plusieurs pages il faut que le projet respecte la structure suivante :

```
your-repository/  
├── pages/  
│   └── page_1.py
```

|   └─ page\_2.py  
 |   └─ your\_app.py  
 sources : [Page Doc StreamLit](#)

## V. Matrice RACI

Activité	DUBOULOY Grégory	DUFOUR Benoît	DELEMASURE Gabriel	BEAUCOUSIN Enzo
Rendez-vous Client	R	R	R	R
Synthèse Rendez-vous	R	A	A	A
Compte-Rendu Projet	R	R	R	R
Recherche Technologies	A	R	A	A
Scénario d'utilisation	A	R	R	A
Prototype Interface	A	R	A	A
Logique de prédiction, implémentation des modèles classe Model, LinearSVCModel et SGDclassifierModel	A	R	A	A
Recherche sur les JDD et conversion des données en CSV pour incorporation dans l'app	A	A	R	A
Mise en place de la classe Data	R	A	R	A
Modularité des JDD dans l'interface	R	A	R	A
Modularité des Modèles dans l'interface	R	A	A	A
Gestion de l'Interface dans la class Main	R	A	A	A
Implémentation schéma d'explication et classe SHAP analyser	A	A	A	R
Mesures de qualités	A	R	R	A

R : Responsable | A : Accountable



## ***VI. Lancer l'application***

Voici la fiche de configuration pour l'application lAduitMoi :


Environnement :

-Python 3.12 (Certaines bibliothèques ne fonctionnent pas avec les versions suivantes)

Bibliothèques utilisées et à installer :

- pandas
- streamlit
- shap
- numpy
- matplotlib
- scikit-learn

Format requis des jeux de données avant leur conversion en .csv :  
exemple de format requis (attention sous forme txt ici)

 dataset\_37\_diabetes.arff - Bloc-notes

Fichier Edition Format Affichage Aide

```
preg,plas,pres,skin,insu,mass,pedi,age,class
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
8,125,96,0,0,0,0.232,54,tested_positive
4,110,92,0,0,37.6,0.191,30,tested_negative
10,168,74,0,0,38,0.537,34,tested_positive
10,139,80,0,0,27.1,1.441,57,tested_negative
1,189,60,23,846,30.1,0.398,59,tested_positive
5,166,72,19,175,25.8,0.587,51,tested_positive
7,100,0,0,0,30,0.484,32,tested_positive
0,118,84,47,230,45.8,0.551,31,tested_positive
7,107,74,0,0,29.6,0.254,31,tested_positive
1,103,30,38,83,43.3,0.183,33,tested_negative
1,115,70,30,96,34.6,0.529,32,tested_positive
3,126,88,41,235,39.3,0.704,27,tested_negative
8,99,84,0,0,35.4,0.388,50,tested_negative
7,196,90,0,0,39.8,0.451,41,tested_positive
9,119,80,35,0,29,0.263,29,tested_positive
11,143,94,33,146,36.6,0.254,51,tested_positive
10,125,70,26,115,31.1,0.205,41,tested_positive
7,147,76,0,0,39.4,0.257,43,tested_positive
1,97,66,15,140,23.2,0.487,22,tested_negative
13,145,82,19,110,22.2,0.245,57,tested_negative
5,117,92,0,0,34.1,0.337,38,tested_negative
5,109,75,26,0,36,0.546,60,tested_negative
3,158,76,36,245,31.6,0.851,28,tested_positive
3,88,58,11,54,24.8,0.267,22,tested_negative
6,92,92,0,0,19.9,0.188,28,tested_negative
10,122,78,31,0,27.6,0.512,45,tested_negative
4,103,60,33,192,24,0.966,33,tested_negative
11,138,76,0,0,33.2,0.42,35,tested_negative
9,102,76,37,0,32.9,0.665,46,tested_positive
2,90,68,42,0,38.2,0.503,27,tested_positive
4,111,72,47,207,37.1,1.39,56,tested_positive
3,180,64,25,70,34,0.271,26,tested_negative
7,133,84,0,0,40.2,0.696,37,tested_negative
7,106,92,18,0,22.7,0.235,48,tested_negative
9,171,110,24,240,45.4,0.721,54,tested_positive
7,159,64,0,0,27.4,0.294,40,tested_negative
0,180,66,39,0,42,1.893,25,tested_positive
1,146,56,0,0,29.7,0.564,29,tested_negative
2,71,70,27,0,28,0.586,22,tested_negative
7,103,66,32,0,39.1,0.344,31,tested_positive
7,105,0,0,0,0,0.305,24,tested_negative
1,103,80,11,82,19.4,0.491,22,tested_negative
1,101,50,15,36,24.2,0.526,26,tested_negative
```

Pour ce qui concerne l'utilisation globale de l'application, se référer à la vidéo de démonstration fournie intitulée 'Démonstration.mp4'

pour lancer le code, dans un terminal se rendre dans le dossier du projet puis taper cette commande pour lancer l'app: streamlit run main.py