

Université abdelmalek essaâdi

Faculté des Sciences et Techniques Tanger

Département Génie Informatique

**MASTER SCIENCES ET TECHNIQUES EN INTELLIGENCE
ARTIFICIELLE ET SCIENCES DE DONNEES (IASD)**

Rapport de Mini-Projet du Module Programmation Avancée avec Python

**Développement d'une Application Web pour l'Analyse
de Données et l'Apprentissage Automatique**

Réalisé Par :

Wafik Reda

Najjout Ayoub

Sous la direction de :

Pr. Sanae KHALI ISSA



Année Universitaire : 2024-2025

Sommaire

Introduction	5
Chapitre 1 : Contexte Général Du Projet	6
I. Introduction :	7
II. Contexte et Problématique :	7
1. Contexte du Projet :	7
2. Problématique :	7
III. Objectifs du Projet :	7
1. Objectif Principal :	7
2. Objectifs Spécifiques :	7
IV. Conclusion :	8
Chapitre 2 : Analyse et Conception.....	9
I. Introduction :	10
II. Besoins Fonctionnels et Besoins Non Fonctionnels :	10
1. Besoins fonctionnels :	10
2. Besoins non fonctionnels :	10
III. Conception Adoptée :	10
1. Langage UML :	10
2. Modélisation UML avec Draw.io :	10
IV. Diagrammes de Cas d'Utilisation :	11
V. Diagramme de Classes :	11
VI. Conclusion :	13
Chapitre 3 : Outils Techniques et Visualisation des Résultats.....	14
I. Introduction :	15
II. Outils et Technologies Utilisés :	15
1. Langage de Programmation :	15
2. Bibliothèques Python :	15
a. Streamlit	15
b. Scikit-learn :	16
c. Pandas :	16
d. Numpy :	16
e. Matplotlib :	16

3.	Environnement de Développement Intégré (IDE) :	17
4.	Gestion de Version :	17
III.	Présentation des Interfaces :	18
1.	Interface de Bienvenue :	18
2.	Création de Dataset :	21
3.	Interface d'importation du dataset :	23
4.	Interface De Préparation des Données :	29
5.	Interface d'entraînement du modèle :	32
6.	Interface de prédictions :	35
7.	Interface D'importation et utilisation d'un modèle :	36
IV.	Conclusion.....	37
	Conclusion Générale	38
	Bibliographie	40

Table De Figures

Figure 1 : Diagramme de cas d'utilisation d'un utilisateur	11
Figure 2 : Diagramme de classes	12
Figure 3 : Page De Bienvenu Screen N°1	19
Figure 4 : Page De Bienvenu Screen N°2	19
Figure 5 : Page De Bienvenu Screen N°3	20
Figure 6 : Page De Bienvenu Screen N°4	20
Figure 7 : Page De Bienvenu Screen N°5	21
Figure 8 : Interface De Creation De Dataset	22
Figure 9 : Téléchargement De Dataset Créé.....	22
Figure 10 : Exemple De Dataset téléchargé.....	23
Figure 11 : Imoprtation De Dataset Titanic	24
Figure 12 : Aperçu des données.....	24
Figure 13 : Informations Générales Du Dataset	25
Figure 14 : Statistiques Descriptives Du Dataset	25
Figure 15 : Valeurs manquantes Du Dataset.....	25
Figure 16 : Visualisation Des Données	26
Figure 17 : Heatmap et Matrice de Corrélation	26
Figure 18 : Visualisation 2D - Nuage de points.....	27
Figure 19 : Visualisation 3D - Nuage de points.....	27
Figure 20 : Détection des Outliers Pour Une Colonne Spécifique	28
Figure 21 : Détection des Outliers dans toutes les colonnes numériques	28
Figure 22 : Vérification de l'équilibre des colonnes catégoriques	29
Figure 23 : Interface De Préparation des Données	31
Figure 24 : Prétraitement automatique et Aperçu des données	33
Figure 25 : Sélectionner La variable cible cas « supervised »	33
Figure 26 : Sélection De Taille Pour les donnees dut test et Choix De Modèle.....	34
Figure 27 : Matrice De Confusion	34
Figure 28 : Rapport De Classification.....	34
Figure 29 : Interface De Prédictions	35
Figure 30 : Interface pour utiliser un modèle existant	36

Introduction

Dans le cadre de notre formation en programmation avancée avec Python, nous avons entrepris ce projet pour mettre en pratique nos compétences en développement et en apprentissage automatique. Ce projet représente une opportunité unique de consolider nos connaissances théoriques tout en réalisant une application concrète répondant à des problématiques réelles d'analyse et de modélisation de données.

L'objectif principal de notre projet est de développer une application informatique polyvalente et interactive qui permet d'importer, de préparer, d'analyser et de modéliser des ensembles de données qualitatives ou quantitatives. Grâce à cette application, nous offrons une interface conviviale intégrant plusieurs algorithmes d'apprentissage automatique, notamment : la régression linéaire (simple, multiple, polynomiale, logistique), les arbres de décision, les forêts aléatoires, les machines à vecteurs de support (SVM), les réseaux de neurones, le plus proche voisin (k-NN), le clustering k-means, et le naïf bayésien.

En plus des algorithmes, nous avons mis l'accent sur des fonctionnalités clés comme la gestion et la préparation des données (nettoyage, gestion des valeurs manquantes, normalisation), la visualisation des résultats sous forme de graphiques et tableaux explicites, ainsi que l'évaluation des modèles pour garantir des performances fiables. Nous avons également prévu une option pour exporter les résultats et les modèles entraînés, permettant une réutilisation ou un partage simple avec d'autres utilisateurs.

À travers ce projet, nous avons non seulement approfondi nos compétences en programmation avec Python, mais également exploré des concepts essentiels de machine learning et d'ingénierie logicielle. Ce travail nous a permis de relever des défis techniques tout en développant une application utile et pédagogique.

Chapitre 1 : Contexte Général Du Projet

I. Introduction :

Ce chapitre vise à introduire le cadre général du projet, en exposant les motivations qui ont conduit à sa réalisation ainsi que les enjeux associés.

II. Contexte et Problématique :

1. Contexte du Projet :

Dans un monde de plus en plus axé sur les données, les entreprises, les institutions académiques et les chercheurs cherchent constamment des moyens efficaces d'exploiter ces données pour en tirer des connaissances utiles. L'apprentissage automatique joue un rôle clé dans ce processus, en permettant d'automatiser des tâches complexes comme la prédiction, la classification et l'analyse des tendances.

Cependant, la complexité des algorithmes et la diversité des approches disponibles rendent souvent difficile leur adoption, notamment pour les débutants.

2. Problématique :

Face à la richesse des algorithmes d'apprentissage automatique et aux divers types de données disponibles, une question se pose : comment concevoir une application simple et intuitive qui permet d'appliquer ces algorithmes de manière efficace tout en répondant aux besoins variés des utilisateurs ?

III. Objectifs du Projet :

1. Objectif Principal :

L'objectif principal de ce projet est de développer une application informatique polyvalente qui facilite l'analyse de données et l'application d'algorithmes de machine learning pour des utilisateurs ayant des niveaux variés de compétence technique.

2. Objectifs Spécifiques :

- Proposer une interface conviviale pour l'importation et la gestion des données.
- Implémenter plusieurs algorithmes de machine learning permettant de répondre à différentes problématiques (prédiction, classification, clustering).

- Intégrer des outils de visualisation pour une interprétation claire des résultats.
- Fournir des mécanismes pour évaluer la performance des modèles.

IV. Conclusion :

Ce chapitre a permis de poser le cadre du projet en présentant les enjeux liés à l'analyse des données à travers l'apprentissage automatique et en définissant les objectifs principaux du projet. Il s'agit de développer une application intuitive permettant d'appliquer divers algorithmes de machine learning et de faciliter l'analyse des données pour les utilisateurs.

Dans les chapitres suivants, nous détaillerons la méthodologie suivie pour la conception de l'application, les outils et technologies utilisés pour sa réalisation, ainsi que les résultats obtenus à l'issue de son déploiement et de son évaluation.

Chapitre 2 : Analyse et Conception

I. Introduction :

Dans ce chapitre, nous allons explorer l'analyse et la conception de notre application en mettant l'accent sur les besoins fonctionnels et non fonctionnels, ainsi que sur la modélisation UML adoptée pour représenter la structure et le fonctionnement de notre projet. Des diagrammes clés, tels que les diagrammes de cas d'utilisation et de classes, seront présentés pour mieux illustrer l'architecture et les interactions au sein du système.

II. Besoins Fonctionnels et Besoins Non Fonctionnels :

1. Besoins fonctionnels :

Les besoins fonctionnels décrivent les fonctionnalités essentielles que l'application doit fournir. Ils incluent :

- Importer des ensembles de données (qualitatives ou quantitatives).
- Préparer et nettoyer les données (traitement des valeurs manquantes, normalisation).
- Appliquer des algorithmes de machine learning.
- Visualiser les résultats sous forme de graphiques et de tableaux.
- Exporter les modèles entraînés et les résultats obtenus.

2. Besoins non fonctionnels :

Les besoins non fonctionnels concernent les aspects de performance, d'ergonomie et de maintenance, notamment :

- Interface utilisateur simple et intuitive.
- Réponse rapide lors du traitement des données et de l'entraînement des modèles.
- Structure modulaire pour faciliter l'ajout de nouvelles fonctionnalités.

III. Conception Adoptée :

1. Langage UML :

Le langage UML (Unified Modeling Language) a été choisi pour modéliser les différents aspects du système, car il permet une visualisation claire et normalisée des interactions et des structures au sein de l'application.[1]

2. Modélisation UML avec Draw.io :



Draw.io a été utilisé comme outil de création de diagrammes UML. Cet outil offre une interface conviviale pour concevoir des diagrammes précis et bien organisés, essentiels pour la documentation et la communication entre les membres du projet.[2]

IV. Diagrammes de Cas d'Utilisation :

Un diagramme de cas d'utilisation est une représentation graphique dans le domaine de l'ingénierie logicielle qui illustre les interactions entre les acteurs externes (utilisateurs ou systèmes externes) et un système logiciel donné. Il met en évidence les différentes actions ou fonctionnalités offertes par le système du point de vue de l'utilisateur, en se concentrant sur ce que le système fait plutôt que sur comment il le fait. En résumé, un diagramme de cas d'utilisation décrit les interactions entre les acteurs et le système, ainsi que les fonctionnalités que le système propose pour répondre aux besoins des utilisateurs. [1]

Le diagramme ci-dessous décrit les interactions entre l'utilisateur et le système :

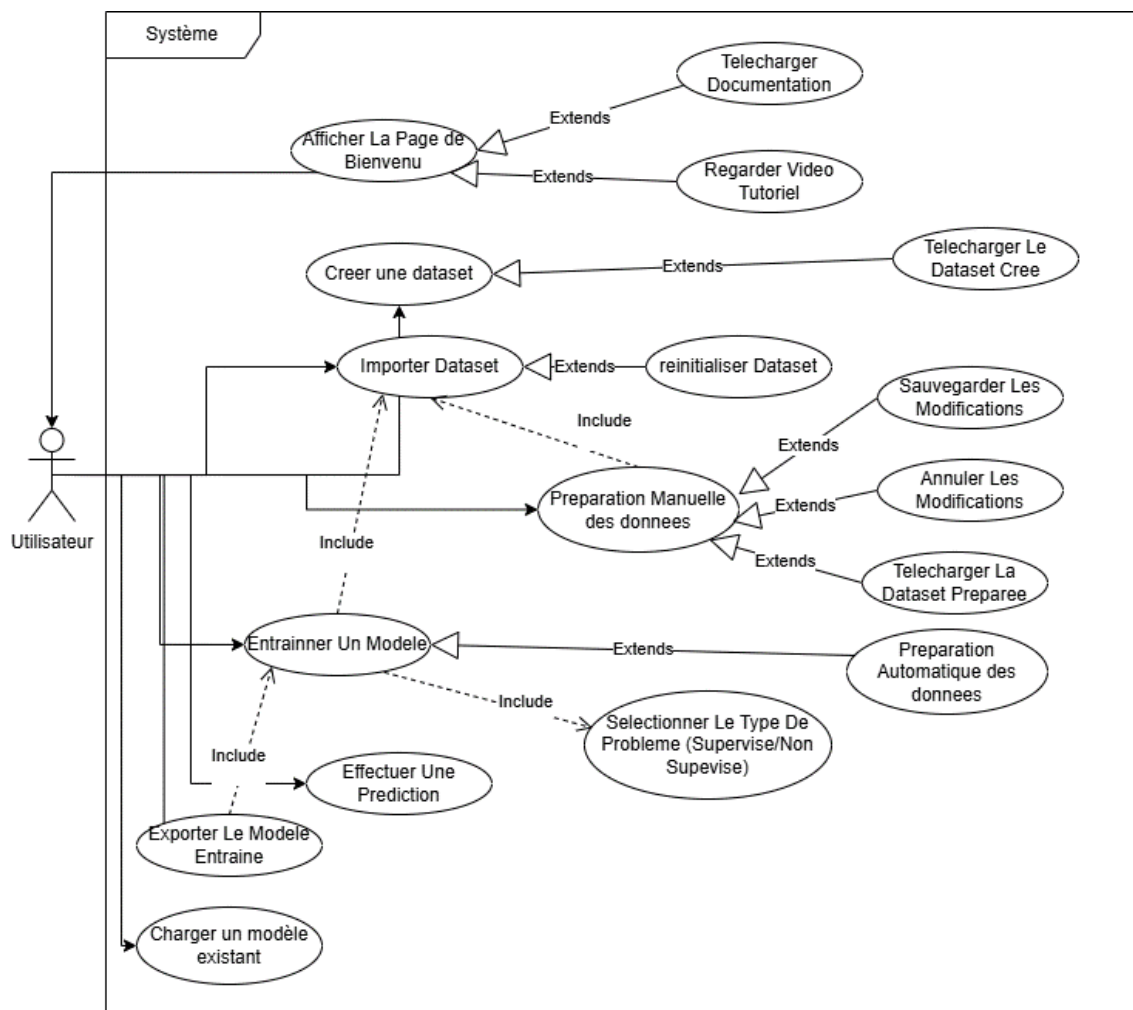


Figure 1 : Diagramme de cas d'utilisation d'un utilisateur

V. Diagramme de Classes :

Un diagramme de classes est une représentation visuelle utilisée en génie logiciel pour modéliser la structure d'un système. Il montre les classes du système, leurs attributs, leurs

méthodes, ainsi que les relations entre elles. Les diagrammes de classes sont couramment utilisés dans la programmation orientée objet pour décrire les types d'objets dans un système et leurs interactions. Le diagramme représente un ensemble de classes et leurs relations. Chaque classe a un nom, une description et une description de ses attributs. Les classes sont toutes liées entre elles, et elles le sont toutes d'une manière ou d'une autre. Les classes sont toutes liées entre elles d'une manière ou d'une autre.[1]

La figure ci -dessous représente le diagramme de classes de notre application :

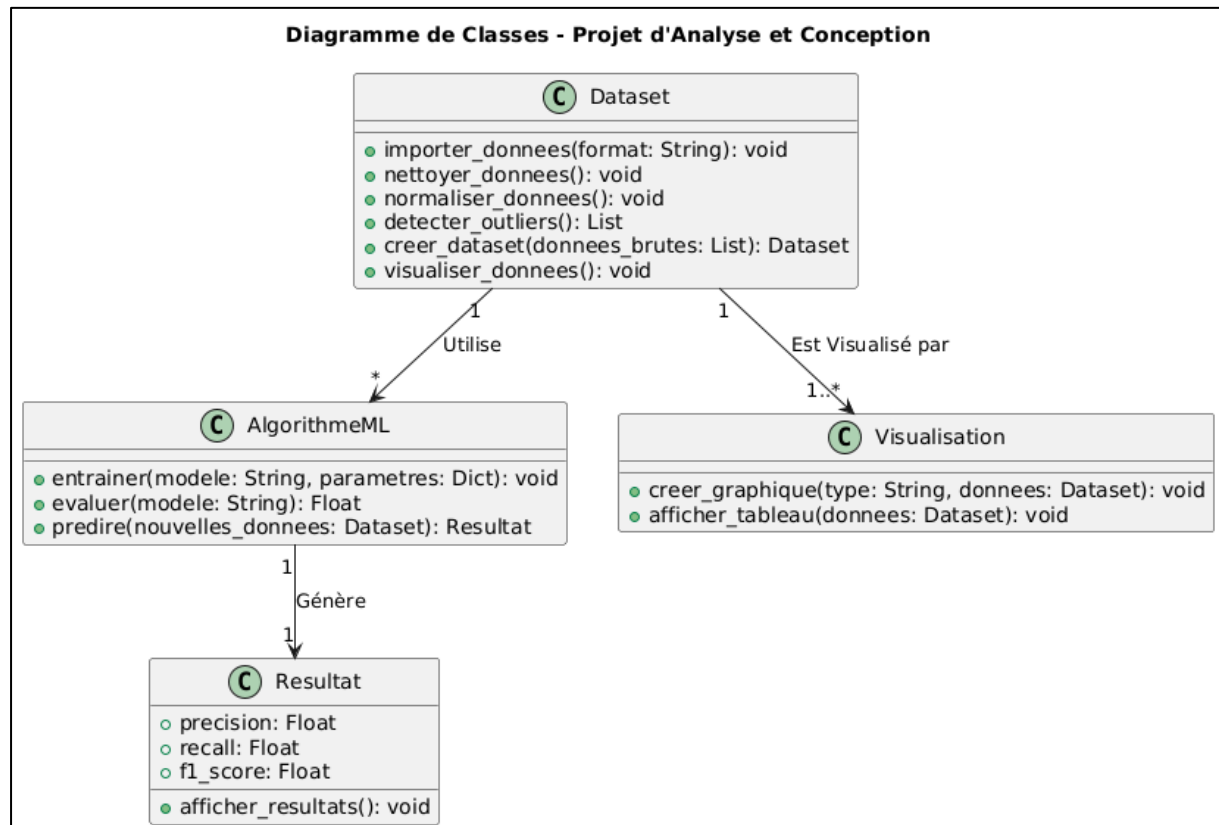


Figure 2 : Diagramme de classes

Voici les classes du diagramme ainsi qu'une description brève de chacune d'elles et leurs relations :

1. Dataset

Cette classe représente l'ensemble de données utilisé dans le projet. Elle fournit des fonctionnalités pour :

- Importer des données depuis différents formats (importer_donnees).
- Nettoyer et normaliser les données (nettoyer_donnees, normaliser_donnees).
- Détecter les valeurs aberrantes dans les données (detecter_outliers).
- Créer un dataset à partir de données brutes (creer_dataset).

- Visualiser les données sous forme graphique ou tabulaire (visualiser_donnees).

2. **AlgorithmeML**

Cette classe encapsule les algorithmes de Machine Learning. Elle permet de :

- Entraîner un modèle avec des paramètres spécifiques (entraîner).
- Évaluer les performances du modèle entraîné (evaluer).
- Faire des prédictions sur de nouvelles données (predire).

3. **Resultat**

Cette classe regroupe les résultats des évaluations des modèles. Elle contient :

- Des métriques de performance comme la précision (precision), le rappel (recall), et le F1-score (f1_score).
- Une méthode pour afficher les résultats des évaluations (afficher_resultats).

4. **Visualisation**

Cette classe est dédiée à la représentation graphique des données et des résultats. Elle permet de :

- Créer des graphiques personnalisés (creer_graphique).
- Afficher des tableaux détaillés des données ou des résultats (afficher_tableau).

Relations entre les classes :

1. **Dataset et AlgorithmeML**

- La classe Dataset est utilisée comme source de données pour entraîner et évaluer les modèles dans la classe AlgorithmeML.

2. **AlgorithmeML et Resultat**

- La classe AlgorithmeML génère un objet de type Resultat après l'évaluation d'un modèle.

3. **Dataset et Visualisation**

- Les données contenues dans la classe Dataset peuvent être visualisées sous forme de graphiques ou de tableaux à l'aide de la classe Visualisation.

VI. Conclusion :

Ce chapitre a permis de poser les bases du projet en identifiant clairement les besoins et en concevant une architecture robuste à travers les outils et diagrammes UML. Cette analyse approfondie guide le développement de l'application et garantit une structure cohérente et efficace pour atteindre les objectifs du projet.

Chapitre 3 : Outils Techniques et Visualisation des Résultats

I. Introduction :

Dans ce chapitre, nous allons présenter les outils techniques qui ont été utilisés pour développer et mettre en œuvre notre projet, ainsi que les résultats obtenus à travers les différentes étapes de son exécution. Nous expliquerons le rôle des environnements de développement et des bibliothèques sélectionnées dans la réalisation des objectifs du projet.

Par la suite, une démonstration détaillée de l'application sera effectuée pour mettre en avant ses principales fonctionnalités, notamment l'importation et la préparation des données, l'entraînement des modèles, la détection des valeurs aberrantes, et la visualisation interactive des résultats.

Enfin, nous concluons avec une discussion des performances des algorithmes implémentés et une analyse des résultats visualisés à travers l'interface utilisateur.

II. Outils et Technologies Utilisés :

Dans cette section, nous décrivons les principaux outils et technologies utilisés pour le développement du projet :

1. Langage de Programmation :



Python : Le langage principal utilisé pour le développement de l'application grâce à sa simplicité, sa richesse en bibliothèques pour le machine learning et la manipulation des données.

2. Bibliothèques Python :

Les bibliothèques Python utilisées dans ce projet jouent un rôle clé dans la manipulation des données, l'implémentation des algorithmes de machine learning, et la création d'une interface utilisateur conviviale. Voici une description détaillée de chaque bibliothèque :

a. Streamlit



Streamlit est un framework open source permettant de créer rapidement des applications web interactives pour le machine learning et la visualisation des données. Dans ce projet, Streamlit a été utilisé pour :

- Concevoir une interface utilisateur intuitive, avec des sections claires pour importer les données, appliquer les modèles, et visualiser les résultats.
- Offrir une expérience utilisateur interactive grâce à des widgets comme des boutons, des menus déroulants et des graphiques dynamiques.

b. Scikit-learn :



Scikit-learn est une bibliothèque dédiée au machine learning, fournissant des outils pour l'entraînement et l'évaluation des modèles. Elle a été utilisée pour :

- Importer les algorithmes de machine learning comme la régression linéaire, les arbres de décision, les forêts aléatoires, et les machines à vecteurs de support (SVM).
- Diviser les données en ensembles d'entraînement et de test.
- Calculer des métriques de performance telles que l'accuracy, le recall, et le F1-score pour évaluer les modèles.

c. Pandas :



Pandas est une bibliothèque puissante pour la manipulation et l'analyse des données. Elle a permis :

- De charger les données sous forme de DataFrame, une structure flexible pour manipuler les données tabulaires.
- De nettoyer les données en gérant les valeurs manquantes ou incohérentes.
- De transformer les colonnes ou d'extraire des sous-ensembles pertinents pour l'analyse.

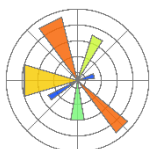
d. Numpy :



Numpy est une bibliothèque fondamentale pour les calculs scientifiques en Python. Elle a été utilisée pour :

- Gérer les tableaux multi-dimensionnels (arrays) et effectuer des calculs mathématiques rapides.
- Appliquer des transformations mathématiques lors de la normalisation des données ou dans les algorithmes de machine learning.

e. Matplotlib :



Matplotlib est une bibliothèque de visualisation des données. Dans ce projet, elle a servi à :

- Générer des graphiques tels que des histogrammes, des courbes et des diagrammes pour représenter les résultats des algorithmes.
- Créer des visualisations claires des performances des modèles et des caractéristiques des datasets, facilitant leur interprétation.

Ces bibliothèques ont été intégrées de manière cohérente pour couvrir l'ensemble du flux de travail, depuis la préparation des données jusqu'à l'évaluation et la visualisation des résultats.

3. Environnement de Développement Intégré (IDE) :



Le projet a été développé en utilisant Visual Studio Code comme environnement de développement intégré (IDE). Ce choix a été motivé par sa légèreté, sa flexibilité et ses fonctionnalités adaptées au développement Python. Visual Studio Code offre une interface claire, une gestion facile des fichiers et plusieurs extensions utiles, telles que celles pour Python et Streamlit, permettant un développement rapide et efficace. Il offre également un terminal intégré, facilitant l'exécution des scripts et la gestion des dépendances, ce qui a grandement contribué à la productivité pendant le développement du projet.

4. Gestion de Version :



Git : Git est un système de contrôle de version décentralisé qui nous a permis de suivre les changements apportés au code tout au long du projet. Chaque modification a été enregistrée dans un historique, facilitant ainsi le retour à des versions antérieures si nécessaire.



GitHub : GitHub a servi de plateforme pour héberger notre code, offrir un accès à l'équipe, et faciliter la collaboration. Nous avons utilisé ses fonctionnalités de branchement et de fusion pour gérer les différentes versions du projet et éviter les conflits entre les contributions.

Ces outils ont joué un rôle clé dans la gestion et l'organisation du projet, en assurant un suivi rigoureux et une collaboration efficace entre les membres de l'équipe.

III. Présentation des Interfaces :

La plateforme développée est structurée en sept sections principales, chacune représentant une étape clé du flux de travail en apprentissage automatique. Elle offre une interface conviviale et guidée, permettant à l'utilisateur d'effectuer des tâches variées, allant de la préparation des données jusqu'à l'utilisation de modèles préexistants.

Voici un aperçu des sections principales :

1. **Bienvenue** : Une introduction à la plateforme avec des explications sur son utilisation.
2. **Création de Dataset** : Permet à l'utilisateur de créer et structurer un nouveau Dataset selon ses besoins.
3. **Importer les données** : Une section pour charger des données externes depuis divers formats (CSV, Excel, etc.).
4. **Préparation des données** : Offrant des outils pour nettoyer, normaliser et détecter les *outliers* avec une option de préparation automatique des données.
5. **Entraînement** : Permet à l'utilisateur de choisir des modèles d'apprentissage automatique, de les configurer, et de les entraîner.
6. **Prédiction et exportation** : Une interface pour utiliser un modèle entraîné sur de nouvelles données, avec la possibilité d'exporter le modèle sous format pkl pour une utilisation ultérieure.
7. **Utiliser un modèle existant** : Cette section permet à l'utilisateur de charger un modèle existant (au format pkl) et de l'appliquer sur de nouvelles données.

Grâce à cette structure modulaire, la plateforme garantit une flexibilité maximale tout en restant accessible aux débutants et utile pour des cas d'usage avancés. Chaque section sera décrite en détail avec les interfaces correspondantes.

1. Interface de Bienvenue :

L'interface *Bienvenue* est la première page de la plateforme et a pour but d'introduire l'utilisateur aux fonctionnalités et aux objectifs de l'application. Elle offre une vue d'ensemble des étapes nécessaires pour exploiter efficacement les outils proposés. Cette section inclut également des ressources d'accompagnement, telles qu'une vidéo explicative

et une documentation téléchargeable, pour faciliter la prise en main, même pour les utilisateurs novices.

Les captures d'écran suivantes illustrent le contenu et la présentation de cette interface :



Figure 3 : Page De Bienvenu Screen N°1



Figure 4 : Page De Bienvenu Screen N°2

Les figures 3 et 4 présentent une introduction à la plateforme ainsi que ses objectifs principaux. La figure 3 met en avant l'interface d'accueil, tandis que la figure 4 détaille les objectifs de la plateforme, illustrant ainsi les principales fonctionnalités offertes aux utilisateurs.



Figure 5 : Page De Bienvenu Screen N°3

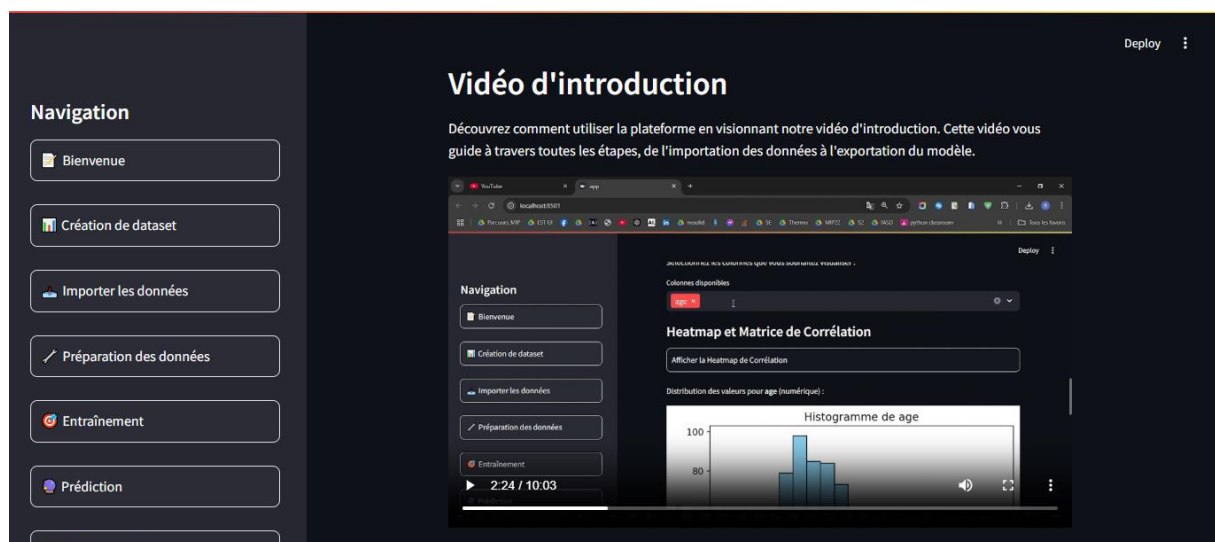


Figure 6 : Page De Bienvenu Screen N°4

Les figures 5 et 6 illustrent les étapes clés pour démarrer avec la plateforme et présentent une vidéo tutoriel.

La figure 5 décrit les différentes étapes à suivre, tandis que la figure 6 montre une vidéo explicative qui guide l'utilisateur à travers un exemple d'utilisation de la plateforme.

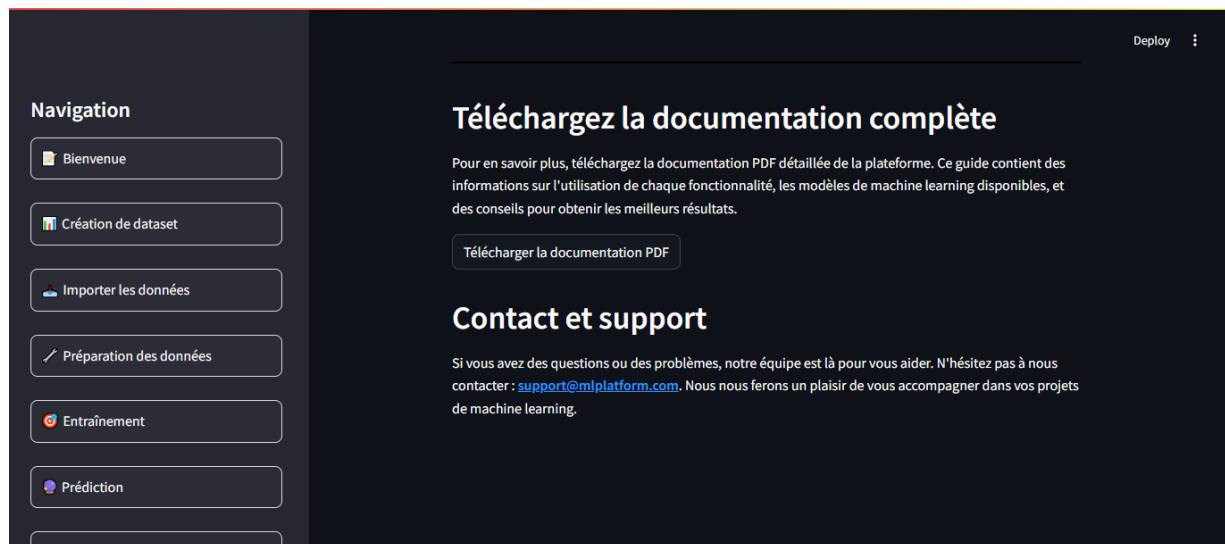


Figure 7 : Page De Bienvenu Screen N°5

La figure 7 présente un bouton permettant de télécharger un guide PDF détaillant l'utilisation de la plateforme. Juste en dessous, un email de contact est fourni pour que les utilisateurs puissent facilement joindre le support en cas de questions ou de problèmes.

2. Création de Dataset :

La section *Création de Dataset* permet aux utilisateurs de créer facilement un jeu de données personnalisé en définissant le nombre de lignes et de colonnes, ainsi que les noms de ces colonnes. Une fois le dataset créé et édité à l'aide d'un tableau interactif, l'utilisateur peut visualiser le dataset directement sur la plateforme.

De plus, la plateforme offre la possibilité d'exporter le dataset sous différents formats, tels que CSV, Excel, ou JSON, pour une utilisation ultérieure. Cela permet une grande flexibilité dans l'exportation et la gestion des données créées.

Les figures 8 et 9 illustrent respectivement la création du dataset et le processus d'exportation du dataset dans le format choisi. La dernière capture montre un exemple de dataset téléchargé, prêt à être utilisé dans les étapes suivantes du processus de machine learning.

Deploy

Création de Dataset

Nombre de lignes

3

Nombre de colonnes

4

Nom de la colonne 1

Age

Nom de la colonne 2

Sexe

Nom de la colonne 3

Country

Nom de la colonne 4

Happy

	Age	Sexe	Country	Happy
0	22	male	morocco	No
1	19	female	morocco	No
2	80	male	Norway	Yes

Dataset édité:

	Age	Sexe	Country	Happy
0	22	male	morocco	No
1	19	female	morocco	No
2	80	male	Norway	Yes

Exporter les données

Format d'export

Excel

Télécharger Excel

Figure 8 : Interface De Creation De Dataset

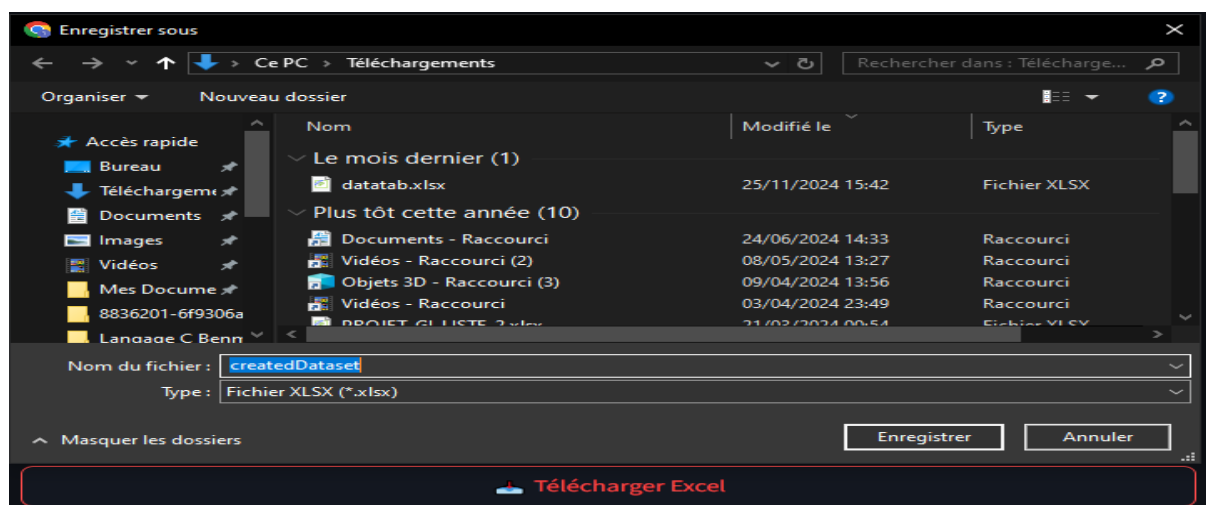


Figure 9 : Téléchargement De Dataset Créé

	A	B	C	D	E
1	Age	Sexe	Country	Happy	
2	22	male	morocco	No	
3	19	female	morocco	No	
4	80	male	Norway	Yes	

Figure 10 : Exemple De Dataset téléchargé

3. Interface d'importation du dataset :

L'interface proposée est conçue pour simplifier l'importation, l'exploration et l'analyse initiale des données dans une application web interactive. À titre d'exemple, le célèbre dataset Titanic sera utilisé pour illustrer les fonctionnalités. Les utilisateurs peuvent importer des fichiers aux formats CSV, Excel ou JSON tout en spécifiant des options comme l'inclusion ou l'exclusion de l'en-tête. Une fois les données importées, l'interface offre une vue d'ensemble complète des dimensions, des types de colonnes et un résumé statistique des données numériques.

L'exploration avancée inclut des fonctionnalités telles que la vérification de l'équilibre d'une colonne spécifique (utile pour détecter les déséquilibres dans des classes ou catégories). De plus, elle intègre une analyse des valeurs aberrantes : l'utilisateur peut obtenir un compte détaillé des valeurs aberrantes présentes dans chaque colonne, avec la possibilité de définir un seuil de détection adapté. Pour une colonne donnée, l'interface permet également de visualiser les lignes spécifiques contenant des valeurs aberrantes, facilitant ainsi l'inspection et la prise de décision.

Côté visualisation, l'interface propose des outils comme une carte de chaleur pour visualiser les corrélations entre les variables, des histogrammes pour analyser les distributions, des

graphiques en barres pour les colonnes catégoriques, ainsi que des nuages de points interactifs (2D ou 3D) pour explorer les relations entre des variables sélectionnées.

Cette interface riche en fonctionnalités est conçue pour aider les utilisateurs à effectuer un prétraitement et une analyse exploratoire efficaces, tout en mettant en lumière des informations essentielles telles que les déséquilibres et les anomalies dans les données.

Les captures d'écran suivantes illustrent le contenu et la présentation de cette interface :



Figure 11 : Imoprtation De Dataset Titanic

The screenshot shows the 'Aperçu des données' (Data Preview) interface. The navigation sidebar is the same as in Figure 11. The main area is titled 'Aperçu des données' and displays a table of the first 10 rows of the Titanic dataset. Below the table, it states 'Dimensions du dataset : 891 lignes, 15 colonnes'. The table has the following columns: index, survived, pclass, sex, age, sibsp, parch, fare, embarked, class, who, and adult_male.

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22	1	0	7.25	S	Third	man	True
1	1	1	female	38	1	0	71.2833	C	First	woman	False
2	1	3	female	26	0	0	7.925	S	Third	woman	False
3	1	1	female	35	1	0	53.1	S	First	woman	False
4	0	3	male	35	0	0	8.05	S	Third	man	True
5	0	3	male	None	0	0	8.4583	Q	Third	man	True
6	0	1	male	54	0	0	51.8625	S	First	man	True
7	0	3	male	2	3	1	21.075	S	Third	child	False
8	1	3	female	27	0	2	11.1333	S	Third	woman	False
9	1	2	female	14	1	0	30.0708	C	Second	child	False

Figure 12 : Aperçu des données



Figure 13 : Informations Générales Du Dataset

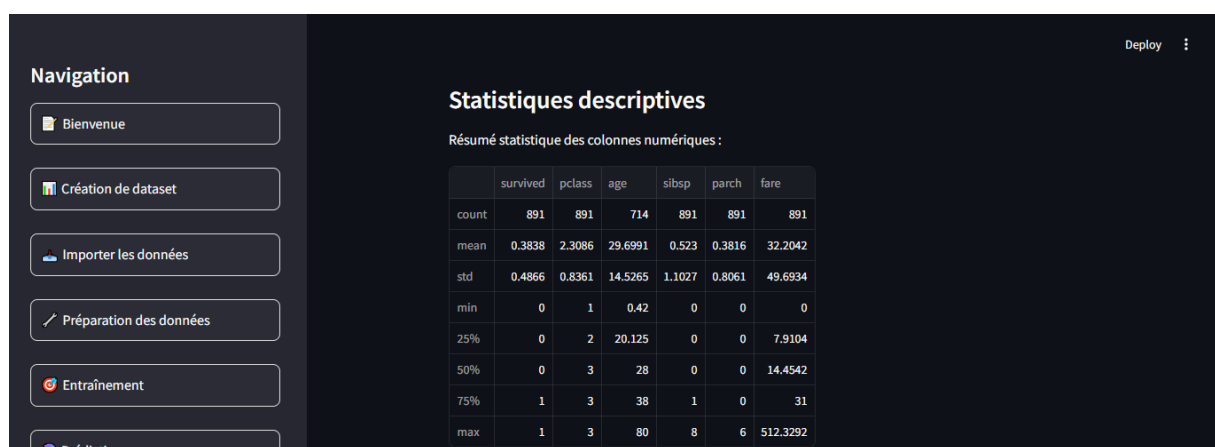


Figure 14 : Statistiques Descriptives Du Dataset

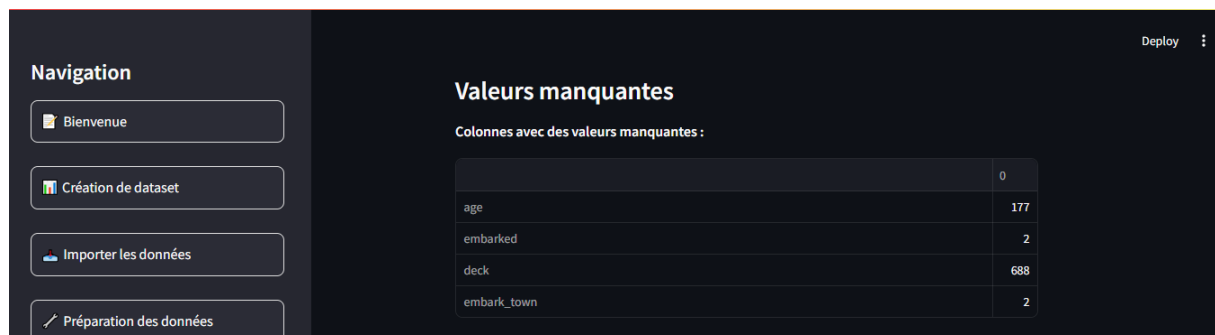


Figure 15 : Valeurs manquantes Du Dataset

Visualisation des données

Sélectionnez les colonnes que vous souhaitez visualiser :

Colonnes disponibles

age x



Heatmap et Matrice de Corrélation

Afficher la Heatmap de Corrélation

Distribution des valeurs pour age (numérique) :

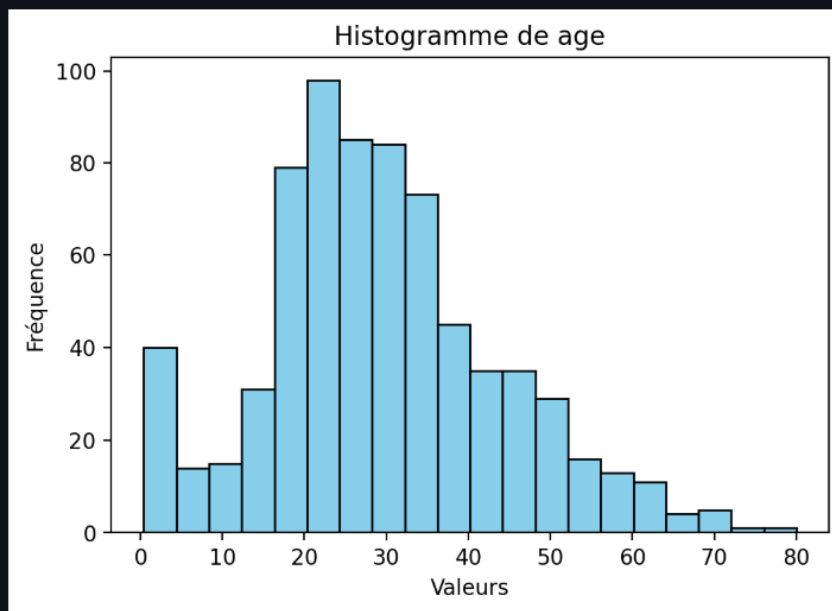


Figure 16 : Visualisation Des Données

Heatmap et Matrice de Corrélation

Afficher la Heatmap de Corrélation

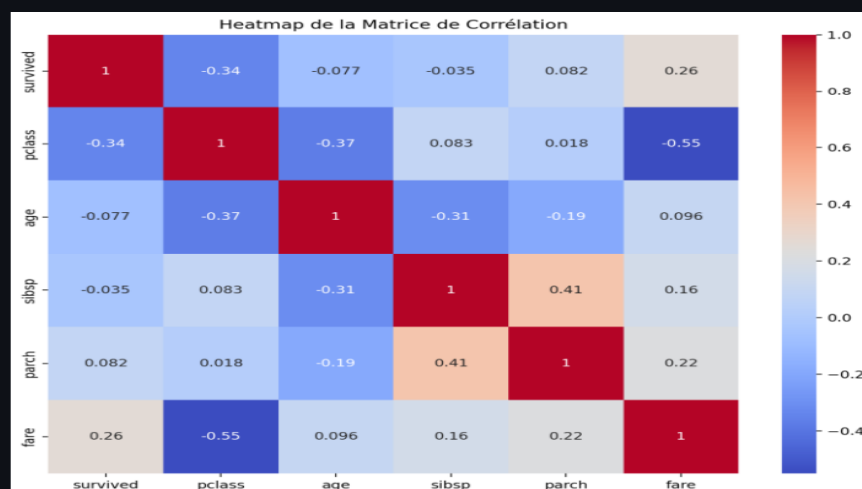


Figure 17 : Heatmap et Matrice de Corrélation

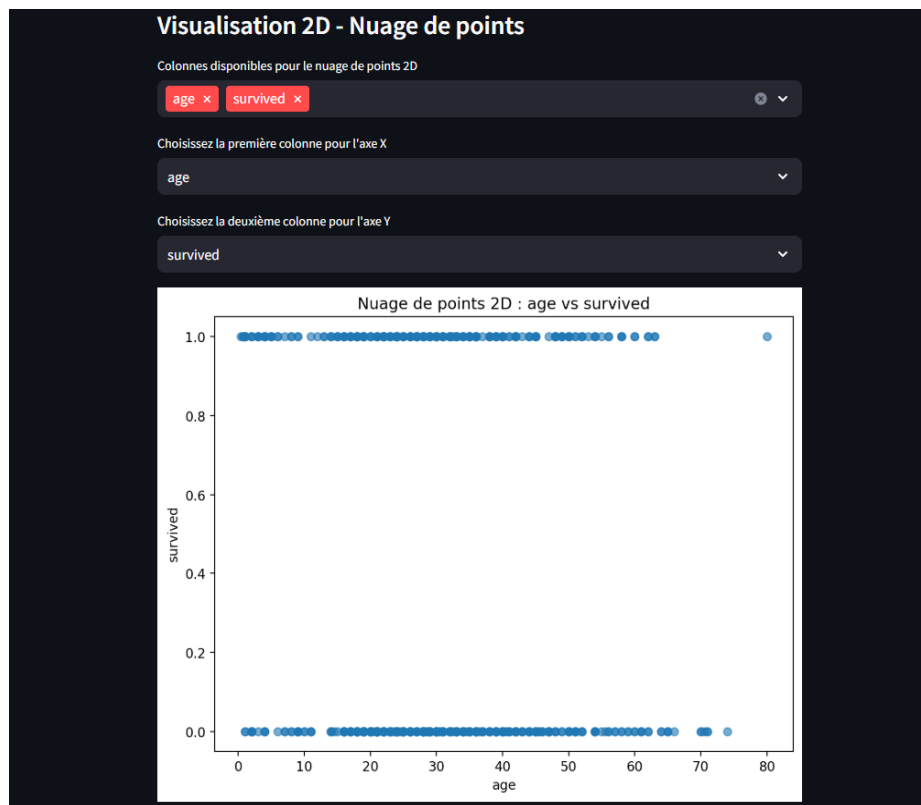


Figure 18 : Visualisation 2D - Nuage de points

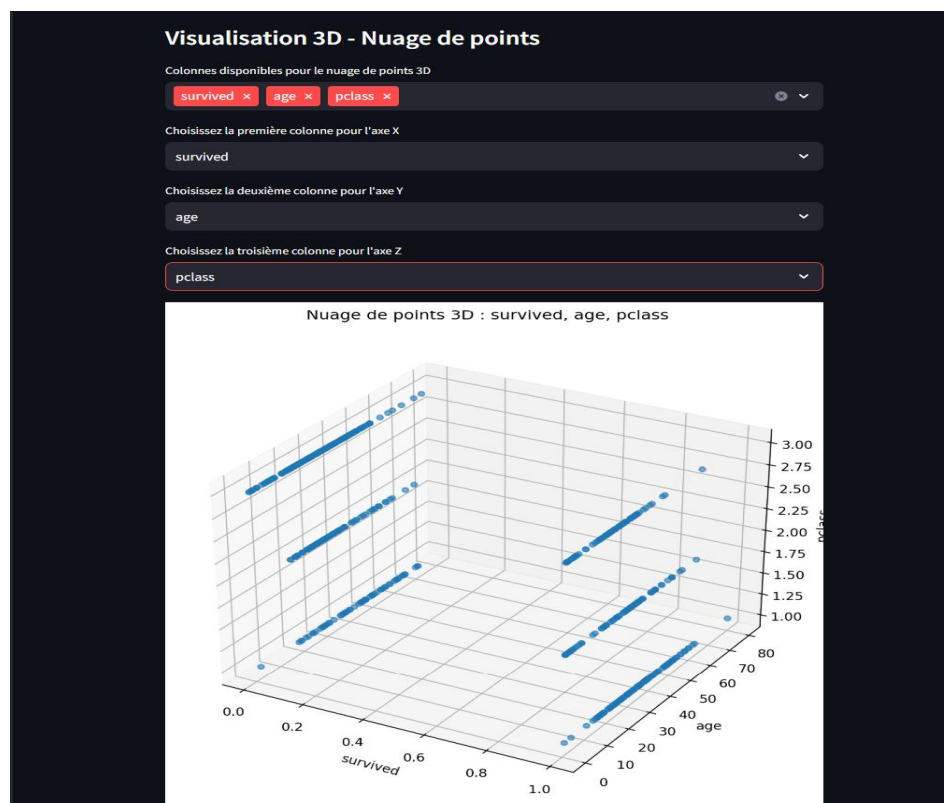


Figure 19 : Visualisation 3D - Nuage de points

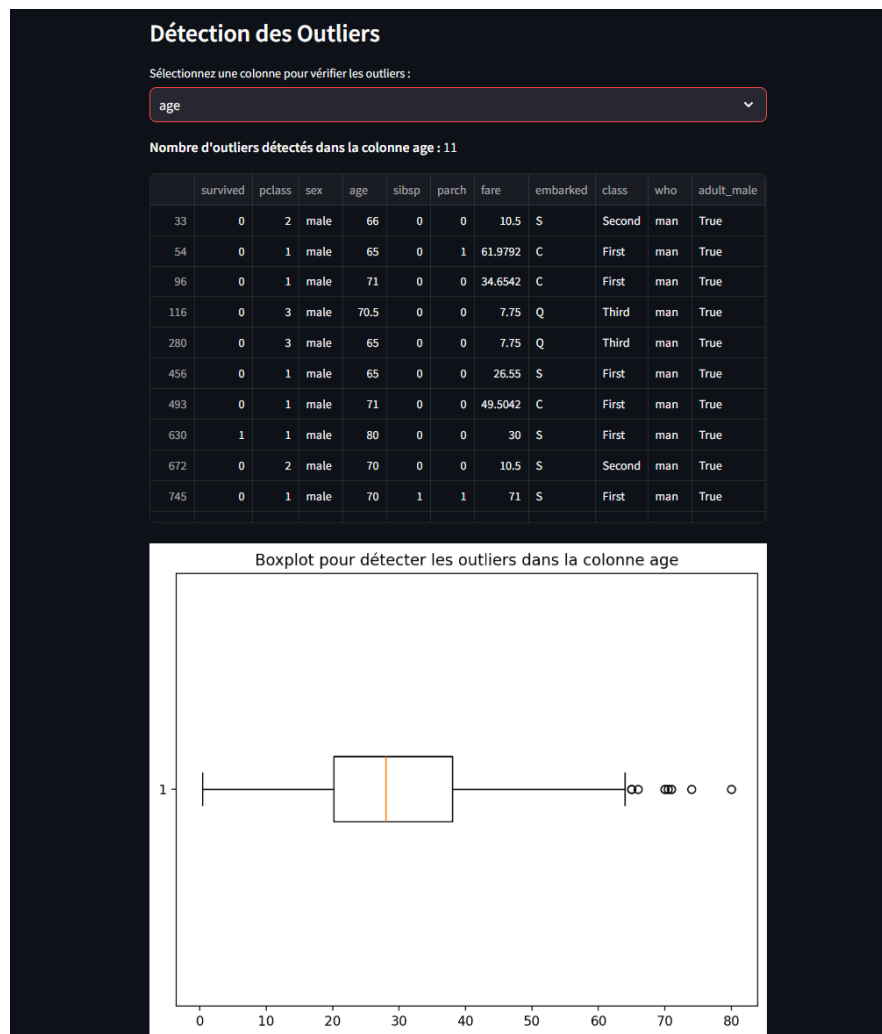


Figure 20 : Détection des Outliers Pour Une Colonne Spécifique

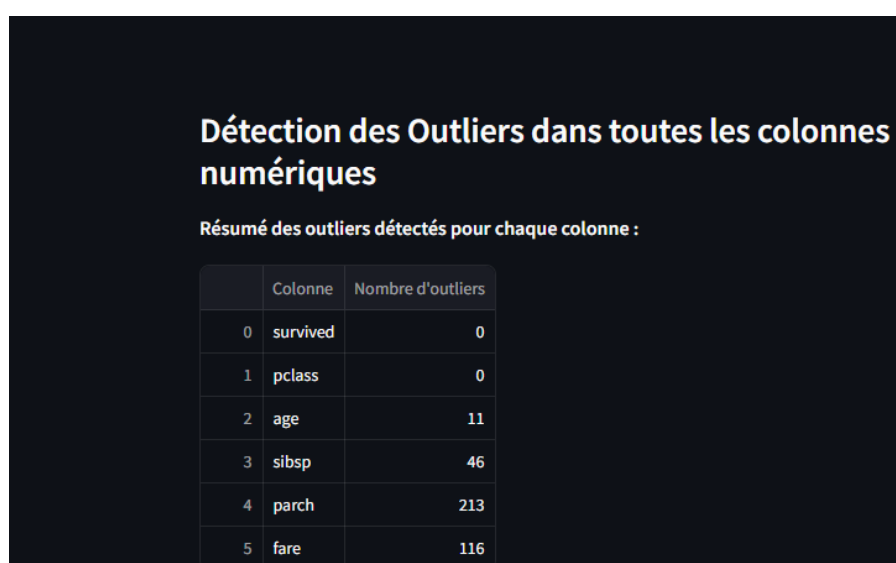


Figure 21 : Détection des Outliers dans toutes les colonnes numériques

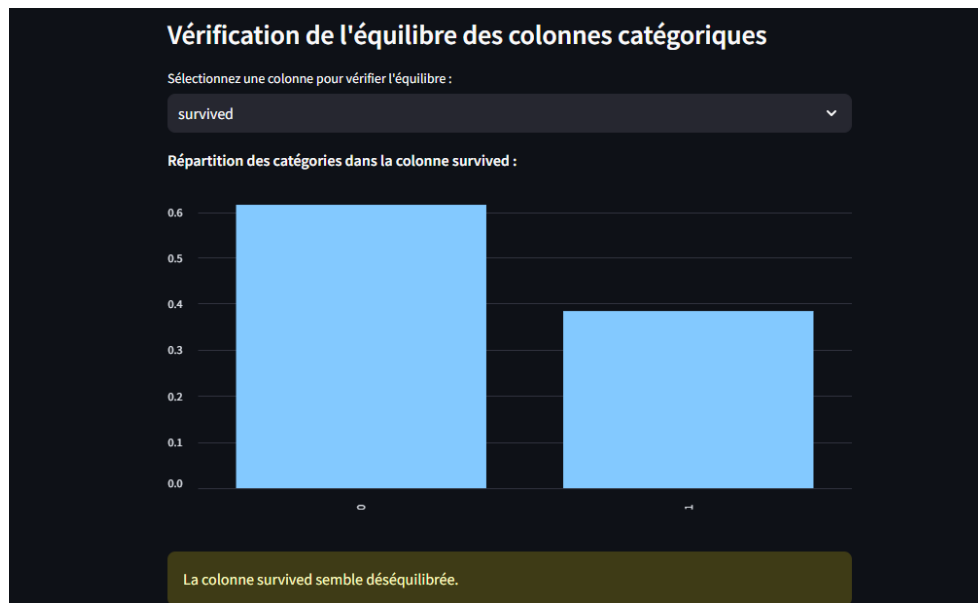











Figure 22 : Vérification de l'équilibre des colonnes catégoriques

4. Interface De Préparation des Données :

L'interface de préparation des données permet de manipuler et de transformer facilement les données pour les rendre prêtes à être utilisées dans des modèles de machine learning. Voici un aperçu des fonctionnalités disponibles dans cette interface :

1.  **Modification et suppression des lignes** : Cette fonctionnalité permet à l'utilisateur de modifier directement les valeurs d'une ligne ou de supprimer une ligne entière, offrant ainsi une flexibilité pour nettoyer et ajuster les données.
2.  **Suppression de colonnes** : L'utilisateur peut supprimer des colonnes qui sont jugées inutiles ou redondantes, ce qui permet de simplifier le jeu de données et de se concentrer sur les informations pertinentes.
3.  **Ajout de colonnes ou lignes** : Il est également possible d'ajouter des colonnes ou des lignes dans le jeu de données. Cela permet d'introduire de nouvelles variables ou d'ajouter des informations manquantes.
4.  **Gestion des doublons** : Cette fonctionnalité aide à détecter et supprimer les doublons dans les données, ce qui garantit que chaque ligne représente une donnée unique et valide.
5.  **Gestion des valeurs manquantes** :

- Pour les **colonnes numériques**, l'utilisateur peut choisir de remplir les valeurs manquantes par la **moyenne** ou la **médiane** de la colonne.
 - Pour les **colonnes non numériques**, on peut remplir les valeurs manquantes par le **mode** de la colonne.
 - L'utilisateur peut aussi décider de **supprimer** les lignes contenant des valeurs manquantes ou de ne rien faire.
6.  **Normalisation et standardisation** : Ces opérations sont utiles pour ajuster les échelles des données. L'utilisateur peut choisir d'effectuer la normalisation ou la standardisation pour **une seule colonne** ou pour **toutes les colonnes** du jeu de données, afin de garantir que les variables soient sur la même échelle.
 7.  **Encodage des colonnes catégoriques** : Les colonnes catégoriques peuvent être encodées pour être utilisées par les modèles. L'utilisateur peut choisir d'encoder **une seule colonne** ou **toutes les colonnes** catégoriques du jeu de données.
 8.  **Gestion des outliers** : Les **outliers** (valeurs aberrantes) peuvent être supprimés soit **globalement**, c'est-à-dire pour toutes les colonnes, soit spécifiquement pour **une seule colonne**. Cela permet de réduire l'impact des données extrêmes sur l'analyse et la modélisation.
 9.  **Équilibrage des classes** : Cette fonctionnalité permet d'équilibrer les classes d'une colonne cible spécifique, ce qui est particulièrement utile lorsque le jeu de données présente un déséquilibre entre les classes. L'équilibrage peut être effectué en utilisant des techniques comme le **sur-échantillonnage** ou le **sous-échantillonnage**.

L'utilisateur peut également **sauvegarder** les modifications apportées au jeu de données préparé, ce qui lui permet de conserver une version propre et transformée de ses données. Après la sauvegarde, il est possible de **télécharger** le fichier modifié pour une utilisation ultérieure. En revanche, si l'utilisateur ne souhaite pas conserver les changements effectués, il a également la possibilité d'**annuler** les modifications et revenir à l'état initial du jeu de données.

Ces outils permettent de préparer efficacement les données avant de passer à l'étape de modélisation, en offrant un contrôle précis sur les transformations et le nettoyage des données.

La figure suivante montre l'interface de préparation des données, où l'utilisateur peut effectuer diverses opérations de nettoyage et de transformation sur le jeu de données :

Préparation des Données : Gestion des Valeurs Manquantes, Normalisation, Encodage et Suppression des Outliers

- Modification et suppression des lignes
- Suppression de colonnes
- Ajout de colonnes ou lignes
- Gestion des doublons
- Gestion des valeurs manquantes
- Normalisation et standardisation
- Encodage des colonnes catégoriques
- Gestion des outliers
- Équilibrage des classes

Aperçu du dataset mis à jour :

	sp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
23		0	35.5	S	First	man	True	A	Southampton	yes	True
24		1	21.075	S	Third	child	False	None	Southampton	no	False
25		5	31.3875	S	Third	woman	False	None	Southampton	yes	False
26		0	7.225	C	Third	man	True	None	Cherbourg	no	True
27		2	263.0	S	First	man	True	C	Southampton	no	False

Exporter les données

Format d'export

CSV

Télécharger CSV

Annuler les modifications

Sauvegarder les modifications

Figure 23 : Interface De Préparation des Données

5. Interface d'entraînement du modèle :

Dans la section dédiée à l'entraînement des modèles, l'interface a été conçue pour guider l'utilisateur tout au long du processus de manière intuitive et fluide. L'un des objectifs majeurs de cette interface est de permettre à l'utilisateur de préparer les données automatiquement si ce n'est pas déjà fait manuellement dans la section de préparation des données. Pour cela, un bouton permettant de lancer cette préparation automatique est mis à disposition, offrant ainsi une gestion simplifiée des étapes de prétraitement.

Une fois les données prêtes, l'utilisateur peut visualiser un aperçu des données prétraitées, afin de vérifier que les transformations appliquées sont conformes à ses attentes. Cette étape inclut une vue détaillée des premières lignes des données, ainsi qu'une visualisation des types de chaque colonne.

L'utilisateur peut ensuite sélectionner le type de problème qu'il souhaite traiter : supervisé ou non supervisé. Si l'option supervisée est choisie, il devra spécifier la colonne cible. Selon la colonne cible sélectionnée, le type de problème (classification ou régression) sera automatiquement détecté, ce qui permet de simplifier l'expérience utilisateur en éliminant le besoin de faire un choix manuel à ce stade.

Pour les problèmes de classification, une étape cruciale consiste à vérifier l'équilibrage des classes. L'interface affiche la distribution des valeurs de la colonne cible afin que l'utilisateur puisse évaluer si les classes sont équilibrées ou s'il est nécessaire d'appliquer des techniques supplémentaires de rééchantillonnage.

L'utilisateur peut également définir la taille de l'échantillon de test, en ajustant un curseur ou en saisissant directement la proportion souhaitée. Ce paramètre est important pour s'assurer que les données de test sont bien représentatives de l'ensemble du jeu de données.

Une fois toutes ces étapes validées, l'utilisateur sélectionne le modèle qu'il souhaite entraîner. L'interface propose un menu déroulant avec une liste de modèles de machine learning populaires, comme les forêts aléatoires, les régressions logistiques, ou d'autres algorithmes supervisés. Pour les problèmes non supervisés, l'option par défaut est l'algorithme K-Means, adapté aux tâches de clustering.

Après avoir choisi le modèle, l'utilisateur clique sur le bouton pour entraîner le modèle. L'interface affiche alors les résultats de l'entraînement, incluant des métriques de performance comme l'exactitude, la précision, le rappel, et la matrice de confusion, pour

évaluer la qualité du modèle dans le cadre d'un problème de classification. Si nécessaire, un rapport de classification détaillé est également fourni pour donner un aperçu plus complet de la performance du modèle.

Dans le cas où l'utilisateur traite un problème de clustering avec l'algorithme K-Means, il pourra spécifier le nombre de clusters souhaité et visualiser les résultats sous forme d'un graphique montrant la répartition des points dans les différents clusters. Cette représentation visuelle permet à l'utilisateur de mieux comprendre comment les données sont regroupées.

En somme, l'interface d'entraînement des modèles est conçue pour être aussi simple et interactive que possible, tout en offrant des outils puissants pour l'évaluation des performances et la personnalisation des paramètres. Elle permet à l'utilisateur de gérer facilement toutes les étapes, depuis la préparation des données jusqu'à l'évaluation des modèles, tout en fournissant des informations claires et utiles pour prendre des décisions éclairées.



Figure 24 : Prétraitement automatique et Aperçu des données



Figure 25: Sélectionner La variable cible cas « supervised »

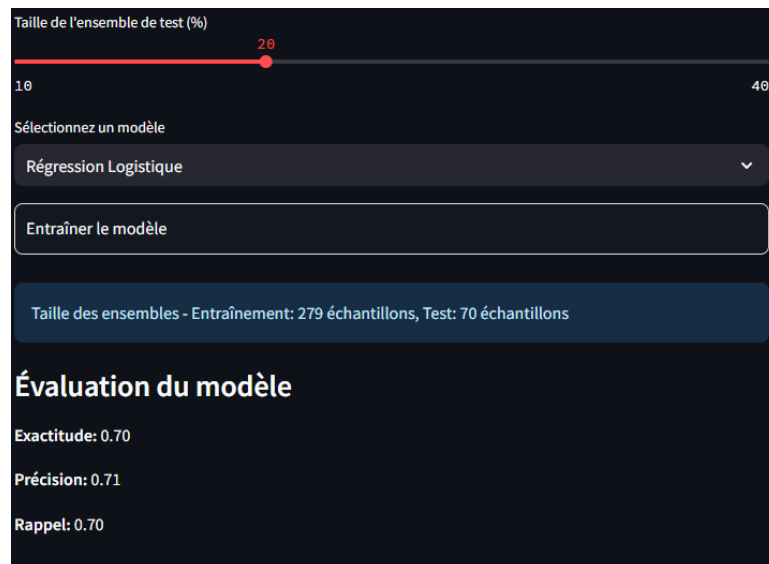


Figure 26 : Sélection De Taille Pour les donnees dut test et Choix De Modèle

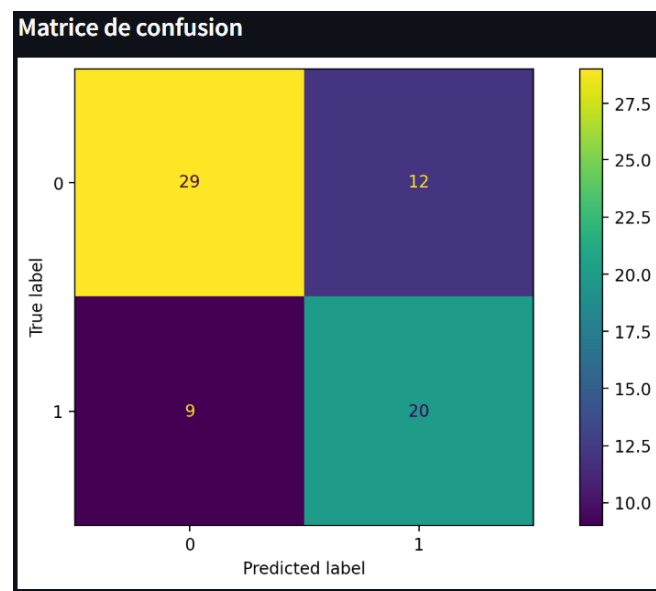


Figure 27 : Matrice De Confusion

	precision	recall	f1-score	support
0	0.7632	0.7073	0.7342	41
1	0.625	0.6897	0.6557	29
accuracy	0.7	0.7	0.7	0.7
macro avg	0.6941	0.6985	0.695	70
weighted avg	0.7059	0.7	0.7017	70

Figure 28 : Rapport De Classification

6. Interface de prédictions :

La page de prédiction permet à l'utilisateur de réaliser des prédictions avec un modèle précédemment entraîné. Tout d'abord, l'utilisateur doit s'assurer qu'un modèle a été entraîné et que les données sont prêtes. Si ce n'est pas le cas, des avertissements apparaissent pour l'informer des étapes manquantes. Ensuite, l'utilisateur peut entrer les valeurs des différentes colonnes de l'ensemble de données, à l'exception de la colonne cible, et soumettre ces informations pour effectuer une prédiction. Selon le type de problème, le modèle retourne une classe prédite dans le cas d'une classification, ou une valeur continue dans le cas d'une régression. Si des erreurs surviennent durant cette procédure, des messages d'erreur explicites sont affichés pour guider l'utilisateur. De plus, l'interface permet à l'utilisateur d'exporter le modèle entraîné sous un format spécifique, facilitant ainsi sa réutilisation. Un bouton de téléchargement est également mis à disposition, offrant à l'utilisateur la possibilité de sauvegarder et de partager son modèle pour d'autres analyses ou applications.

Prédiction avec le modèle entraîné

Entrée des données pour la prédiction

Valeur pour pclass

1

Valeur pour sex

0

Valeur pour age

22

Faire une prédiction

Classe prédite : 1

Exporter le modèle

Nom du fichier du modèle (sans extension)

model

Exporter le modèle

Figure 29 : Interface De Prédictions

7. Interface D'importation et utilisation d'un modèle :

L'interface permettant d'importer un modèle existant est conçue pour offrir à l'utilisateur la possibilité de charger un modèle déjà entraîné afin de l'utiliser pour des prédictions. Après avoir sélectionné et chargé un fichier de modèle au format .pkl, le système extrait les informations essentielles telles que le modèle lui-même, les colonnes des données, la colonne cible, le type de problème (classification ou régression), ainsi que les données associées. Une fois le modèle chargé avec succès, l'utilisateur peut saisir les valeurs des différentes colonnes du jeu de données (sauf la colonne cible) pour effectuer une prédiction. Si le problème est de type classification, la classe prédite sera affichée, tandis que pour une régression, la valeur continue prédite est présentée. L'interface prend également en charge les erreurs, offrant des messages explicites en cas de problème lors du chargement du modèle ou de l'exécution des prédictions. Ce processus permet à l'utilisateur de réutiliser un modèle existant, facilitant ainsi l'application de modèles précédemment entraînés à de nouveaux jeux de données.

Figure 30 : Interface pour utiliser un modèle existant

IV. Conclusion

En conclusion, le chapitre 3 présente une description détaillée des différentes interfaces permettant à l'utilisateur d'interagir avec l'application de machine learning. Ces interfaces sont conçues pour offrir une expérience fluide et intuitive, allant de l'importation et de la préparation des données jusqu'à l'entraînement et à l'utilisation de modèles pour la prédiction. Chaque section est optimisée pour guider l'utilisateur à travers des étapes claires, tout en permettant une personnalisation des paramètres selon les besoins spécifiques du projet. L'interface d'entraînement de modèles offre des fonctionnalités avancées, telles que la sélection automatique de type de problème et la visualisation des résultats de manière détaillée. L'interface de prédiction permet une utilisation pratique des modèles entraînés pour effectuer des prédictions sur de nouvelles données, tandis que l'option de charger un modèle existant simplifie la réutilisation de modèles préalablement formés. Ainsi, ce chapitre met en avant la flexibilité et l'efficacité de l'application, qui vise à rendre l'apprentissage automatique accessible et opérationnel pour les utilisateurs, quel que soit leur niveau d'expertise.

Conclusion Générale

En conclusion, ce rapport décrit le développement et la mise en œuvre d'une application interactive basée sur Streamlit, conçue pour aider les utilisateurs à apprendre et à appliquer des techniques de machine learning dans un environnement simple et intuitif. L'objectif principal de ce projet était de créer une plateforme permettant d'effectuer diverses tâches liées au machine learning, telles que la préparation des données, l'entraînement de modèles, la prédiction, et l'évaluation des performances, tout en offrant une interface accessible même aux débutants.

Le système est structuré autour de plusieurs sections interactives, permettant à l'utilisateur de charger et de préparer les données, de sélectionner le type de problème (supervisé ou non supervisé), de choisir un modèle, puis d'entraîner et d'évaluer ce modèle. L'application gère automatiquement les différents aspects du processus de machine learning, comme la sélection des caractéristiques, la gestion des classes cibles, et la présentation des résultats sous forme de visualisations et de métriques pertinentes. Pour les problèmes supervisés, l'interface permet de visualiser des informations détaillées sur les performances du modèle, telles que l'exactitude, la précision, le rappel, et la matrice de confusion, tandis que pour les problèmes non supervisés, l'application propose une visualisation des clusters.

Un des aspects clés du projet est sa flexibilité : l'utilisateur peut charger un modèle existant, effectuer des prédictions avec de nouvelles données et exporter les modèles pour les utiliser dans d'autres projets ou contextes. Cette fonctionnalité d'exportation permet aux utilisateurs de sauvegarder leurs modèles d'entraînement dans des fichiers .pkl, garantissant ainsi la portabilité des modèles et la possibilité de les réutiliser facilement. De plus, l'application prend en charge l'importation de modèles préexistants, permettant ainsi une continuité dans les analyses et les prédictions.

L'architecture logicielle du projet a été pensée de manière modulaire, avec des sections bien distinctes pour chaque étape du processus, ce qui permet une gestion claire et évolutive du code. Chaque module est conçu pour être facilement compréhensible et extensible, garantissant ainsi que l'application peut évoluer en fonction des besoins futurs, par exemple, en ajoutant de nouveaux algorithmes de machine learning ou en améliorant les fonctionnalités existantes.

En somme, ce projet fournit une plateforme robuste et évolutive pour l'analyse de données et l'entraînement de modèles de machine learning, avec un accent particulier sur l'interactivité et l'expérience utilisateur. En facilitant l'utilisation des outils de machine learning, il offre aux étudiants, chercheurs et professionnels un moyen pratique d'expérimenter et de comprendre les algorithmes de machine learning, tout en rendant les concepts complexes plus accessibles. Ce projet peut également servir de base pour des développements futurs, notamment pour l'intégration d'autres modèles, l'amélioration de l'interface, ou l'ajout de nouvelles fonctionnalités de visualisation des résultats.

Bibliographie

1. Prof. Said Kafhali - Cours génie logiciel et UML(FSTS)
2. Documentation de draw.io : <https://www.drawio.com/doc/>
3. Streamlit Documentation : <https://docs.streamlit.io/>
4. Scikit-learn Documentation : <https://scikit-learn.org/stable/>
5. Prof. Sanae KHALI ISSA – Cours Programmation avancée (FSTT)