

UNIVERSITÉ HASSAN II – FACULTÉ DES SCIENCES BEN M'SIK

RAPPORT DE PROJET DE FIN DE MODULE

LICENCE D'EXCELLENCE

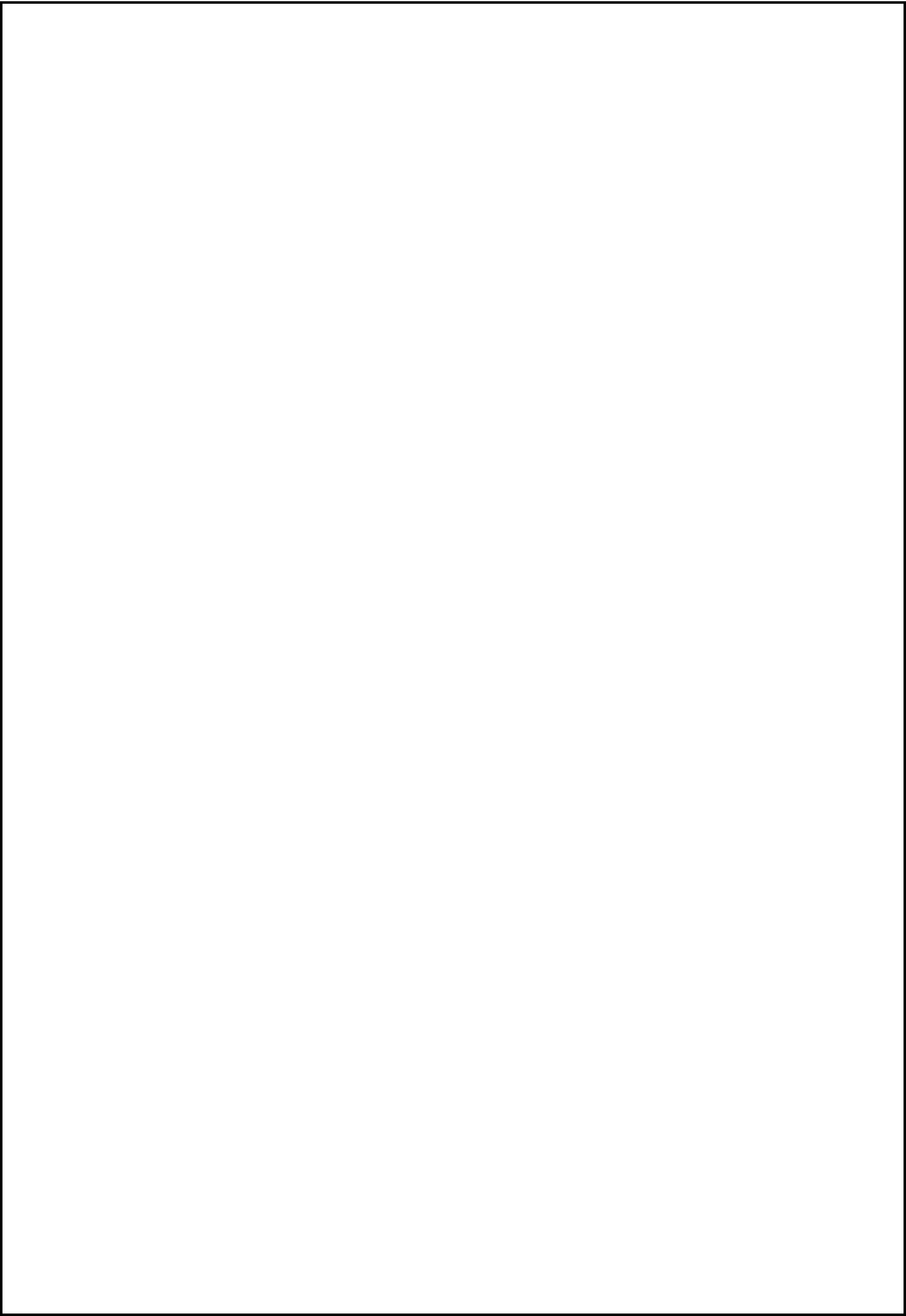
Prédiction de la Réussite Étudiante par Apprentissage Automatique

Réalisé par : Mohamed Nafii

Marouane Aittamanait

Devant le jury:

Pr. Tarik Ahajjam	Professeur à FSBM	Encadrant
Pr. Belangour Abdessamad	Professeur à FSBM	Examineur
Pr. Ait Daoud Mohamed	Professeur à FSBM	Examineur
Pr. Khalid KANDALI	Professeur à FSBM	Examineur



Remerciement

Au terme de ce travail, nous souhaitons exprimer notre profonde gratitude à notre cher professeur et encadrant, Tarik Ahajam, pour son suivi attentif et son immense soutien, qu'il n'a cessé de nous prodiguer tout au long de la réalisation de ce projet.

Nous adressons également nos sincères remerciements aux membres du jury pour avoir bien voulu examiner et évaluer ce travail.

Nous ne saurions laisser passer cette occasion sans remercier tous les enseignants et le personnel de la Faculté des Sciences Ben M'Sick, en particulier ceux de la section Mathématiques et Informatique, pour leur aide, leurs précieux conseils, et l'intérêt qu'ils portent à notre formation.

Enfin, nous adressons nos remerciements à tous ceux qui ont contribué de près ou de loin au bon déroulement de ce projet.

Résumé

Ce projet de machine learning a pour objectif d'analyser et de prédire les performances académiques des étudiants en s'appuyant sur des techniques d'intelligence artificielle. L'application développée se compose de deux parties : une première page sous forme de tableau de bord interactif permettant d'explorer les données étudiantes à travers des visualisations dynamiques, et une seconde page proposant un formulaire permettant de prédire si un étudiant est susceptible de réussir ou non. Pour mieux comprendre les comportements et caractéristiques des étudiants, une méthode de clustering a été appliquée afin de regrouper les individus ayant des profils similaires. Pour la prédiction binaire du succès académique (réussite ou échec), un modèle de classification Random Forest a été utilisé. Ce projet vise à offrir un outil d'aide à la décision aux responsables pédagogiques, afin d'identifier les étudiants à risque et de mettre en place des actions ciblées pour améliorer le taux de réussite universitaire.

Abstract

This machine learning project aims to analyze and predict students' academic performance using artificial intelligence techniques. The developed application consists of two parts: a first page in the form of an interactive dashboard allowing students to explore data through dynamic visualizations, and a second page offering a form allowing students to predict whether or not a student is likely to succeed. To better understand students' behaviors and characteristics, a clustering method was applied to group individuals with similar profiles. For the binary prediction of academic success (pass or fail), a Random Forest classification model was used. This project aims to provide a decision-making tool for educational leaders, in order to identify at-risk students and implement targeted actions to improve the university success rate.

Introduction générale

L'enseignement supérieur traverse aujourd'hui une période de transformation profonde, marquée par l'essor du numérique et l'émergence de nouvelles approches pédagogiques. Dans ce contexte, la réussite étudiante demeure un défi majeur pour les établissements d'enseignement supérieur, qui doivent composer avec des publics de plus en plus diversifiés et des parcours académiques hétérogènes.

L'échec universitaire représente un coût considérable, tant sur le plan humain qu'économique. Pour les étudiants, il peut engendrer des conséquences psychologiques durables et compromettre leur insertion professionnelle. Pour les institutions, il constitue une perte de ressources et peut affecter leur réputation. Face à ces enjeux, l'identification précoce des étudiants en difficulté devient une priorité stratégique.

Les technologies d'intelligence artificielle et d'analyse de données offrent aujourd'hui des opportunités inédites pour améliorer le suivi pédagogique et optimiser les stratégies d'accompagnement. L'exploitation des traces numériques laissées par les étudiants dans leur parcours académique permet d'identifier des patterns comportementaux et de prédire les risques d'échec avec une précision remarquable.

Ce projet de fin d'études s'inscrit dans cette démarche d'innovation pédagogique en proposant la conception et le développement d'un tableau de bord intelligent capable d'analyser les performances étudiantes et de prédire les risques d'échec académique. Notre approche combine l'utilisation d'algorithmes d'apprentissage automatique avec une interface utilisateur intuitive, permettant aux équipes pédagogiques de prendre des décisions éclairées pour améliorer l'accompagnement des étudiants.

Listes des abréviations

Abréviation	Signification
IA	Intelligence Artificielle
ML	Machine Learning (Apprentissage Automatique)
RF	Random Forest
CSV	Comma-Separated Values (Valeurs séparées par des virgules)
API	Application Programming Interface
UI	User Interface (Interface Utilisateur)
KPI	Key Performance Indicator (Indicateur Clé de Performance)
S1	Semestre 1
S2	Semestre 2
EDA	Exploratory Data Analysis (Analyse exploratoire des données)

Table de matière

Remerciement.....	1
Résumé.....	2
Abstract	3
Introduction générale.....	4
Listes des abréviations	5
Table de matière.....	6
Liste de figure.....	9
Chapitre I : État de l’art	11
I. Introduction.....	11
II. Travaux existants sur la prédiction de la réussite étudiante	11
III. Travaux existants sur la prédiction de la réussite étudiante	12
IV. Comparaison des approches et justification du choix du modèle ...	13
1. Modèles comparés.....	13
2. Évaluation des performances	13
3. Justification du choix de Random Forest	14
V. Conclusion	15
Chapitre II : Contexte du project	16
I. Introduction.....	16
II. Problématique :.....	16
III. Objectif du project :	17
IV. Contraintes du Project :	18

V. Conclusion:	19
Chapitre III : Fondaments théoriques.....	20
I. Introduction:.....	20
II. Analyse des étudiants par machine learning	20
IV. La Classification Binaire avec Random Forest.....	25
V. L'importance de la qualité des données.....	26
VI. Conclusion	27
Chapitre IV : Analyse et Conception	28
I. Introduction.....	28
II. Analyse des besoins	28
2. Objectifs non fonctionnels du système	28
III. Acteurs et utilisateurs cibles.....	29
IV. Diagramme de cas d'utilisation (UML).....	30
V. Conclusion	30
Chapitre V : Outils Utilisée	32
I. Intoduction	32
II. Python	32
III. NumPy.....	32
IV. Pandas.....	33
V. Matplotlib et Seaborn	33
VI. Scikit-learn (sklearn) :	33
VII. Interface utilisateur	34
Chapitre VI : Réalisation	35

I. Introduction:.....	35
II. Création du modèle de Machine Learning	35
III. Interface utilisateur	42
Conclusion	50
Bibliothèque.....	51

Liste de figure

Figure 1: apprentissage supervise	21
Figure 2: apprentissage non supervisé.....	21
Figure 3 : architecture du project	22
Figure 4 : Etapes de traitement.....	23
Figure 5: k-means fonctionnement.....	24
Figure 6 : Etapes k-means.....	25
Figure 7: Classification binaire	25
Figure 8 : random forest	26
Figure 9 : etapes de pétreatment.....	27
Figure 10 : acteurs et utilisateurs cibles.....	29
Figure 11 : diagramme de cas d'utilisation	30
Figure 12 : distribution des données.....	36
Figure 13 : suppression des valeurs aberrantes	36
Figure 14: répartition des classes target.....	37
Figure 15 : encodage des données	39
Figure 16 : separation des données.....	40
Figure 17 :elbow method	40
Figure 18 : kmeans exemple	41
Figure 19 : evaluation performance model	41
Figure 20 :dashboard utilisateur.....	43
Figure 21 :kpi globaux.....	43
Figure 22 :kpi par segment	44
Figure 23 : kpi comportement.....	44
Figure 24 : kpi de progression.....	45

Figure 25 : prediction des tables	46
Figure 26 : analyse des tables prédits	46
Figure 27: analyse cluster	47
Figure 28 : formulaire prédiction	48
Figure 29 : Resultat prediction.....	48

Chapitre I : État de l'art

I. Introduction

L'analyse prédictive de la réussite académique a suscité un intérêt croissant ces dernières années, notamment grâce à l'émergence des technologies d'intelligence artificielle et de l'apprentissage automatique. Ces outils permettent d'exploiter les données éducatives de manière plus fine et plus proactive, afin de mieux comprendre les facteurs qui influencent le parcours des étudiants et de mettre en place des stratégies d'accompagnement plus efficaces.

Avant de concevoir notre propre solution, il est essentiel de s'intéresser aux travaux existants dans ce domaine. Ce chapitre vise à présenter un aperçu des recherches précédentes portant sur la prédiction de la réussite étudiante, les méthodes d'analyse de données généralement employées, ainsi que les modèles de machine learning les plus utilisés. Il mettra également en évidence les avantages et les limites de ces approches, pour justifier les choix méthodologiques adoptés dans notre projet.

II. Travaux existants sur la prédiction de la réussite étudiante

Depuis quelques années, de nombreuses études se sont intéressées à l'utilisation du machine learning pour prédire la réussite ou l'échec des étudiants. L'objectif principal est d'aider les établissements à repérer les étudiants à risque, afin de leur proposer un meilleur accompagnement.

Par exemple, plusieurs projets ont utilisé des modèles comme Random Forest, Support Vector Machine (SVM) ou K-Nearest Neighbors (KNN) pour analyser des données telles que les notes des étudiants, leur âge, leur statut social ou leur situation financière. Ces modèles permettent d'apprendre à partir des données passées et de prédire si un nouvel étudiant risque d'échouer.

Sur la plateforme Kaggle, un dataset très utilisé contient des données détaillées sur les étudiants : leurs performances, leur profil, leur statut de boursier, etc. Plusieurs utilisateurs ont partagé des notebooks montrant que les modèles comme Random Forest obtiennent souvent de bons résultats en précision, surtout après un bon nettoyage des données et un équilibrage des classes.

Des chercheurs ont aussi montré que des variables comme les notes des semestres précédents, l'obtention d'une bourse, ou encore le statut d'endettement sont souvent liées à la réussite ou à l'échec. Ces facteurs ont donc été largement utilisés dans la majorité des projets.

En résumé, les travaux existants montrent que :

- Les données étudiantes permettent de faire des prédictions utiles ;
- Les modèles d'arbres comme Random Forest sont souvent efficaces ;
- Le prétraitement des données est une étape essentielle ;
- Il est utile de combiner les données académiques, personnelles et financières.

Notre projet s'inscrit dans cette lignée, en proposant une application simple et interactive, qui exploite ces méthodes pour aider les responsables pédagogiques à mieux comprendre et accompagner leurs étudiants.

III. Travaux existants sur la prédiction de la réussite étudiante

Dans la majorité des travaux liés à la prédiction de la réussite académique, plusieurs algorithmes de machine learning sont souvent utilisés. Le choix de l'algorithme dépend généralement du type de données disponibles et de l'objectif recherché (prédire un score, une réussite, un abandon, etc.).

Voici les techniques les plus courantes :

1. Random Forest

C'est l'un des modèles les plus populaires. Il s'agit d'un ensemble d'arbres de décision qui prennent chacun une décision, et le modèle final choisit la réponse la plus fréquente. Il est très utilisé car :

- Il est robuste,
- Il gère bien les données complexes,
- Et il offre de bonnes performances même sans réglages très poussés.

2. Support Vector Machine (SVM)

Cet algorithme essaie de tracer une frontière entre les étudiants qui réussissent et ceux qui échouent. Il fonctionne bien surtout quand les données sont bien séparées, mais peut être plus difficile à comprendre et à paramétrer.

3. K-Nearest Neighbors (KNN)

Ce modèle classe un étudiant en regardant les "k" étudiants les plus proches dans le dataset. Si la majorité a réussi, il prédit que l'étudiant réussira. C'est un modèle simple à comprendre, mais lent avec de grands volumes de données.

4. Régression Logistique

Ce modèle prédit la probabilité de réussite ou d'échec. Il est très utilisé dans les premiers projets car il est simple, rapide, et facile à interpréter.

5. Réseaux de neurones

Moins utilisés dans les projets éducatifs simples, ils peuvent être utiles pour des données très complexes. Mais ils demandent plus de puissance de calcul et sont souvent moins interprétables.

IV. Comparaison des approches et justification du choix du modèle

La prédiction de la réussite étudiante a fait l'objet de nombreuses expérimentations à l'aide de différents algorithmes d'apprentissage automatique. Chaque modèle possède ses propres avantages et limites, selon la nature des données disponibles, le niveau de précision attendu, et l'interprétabilité recherchée. Afin de sélectionner le modèle le plus adapté à notre problématique, nous avons effectué une comparaison rigoureuse entre plusieurs méthodes classiques de classification.

1. Modèles comparés

Nous avons testé les algorithmes suivants :

- **Régression Logistique** : simple et rapide, mais parfois limitée dans sa capacité à modéliser des relations non linéaires.
- **K-Nearest Neighbors (KNN)** : efficace pour les petits ensembles de données, mais sensible aux données bruitées.
- **Support Vector Machine (SVM)** : robuste, mais plus lent à l'entraînement, et difficile à interpréter dans un contexte éducatif.
- **Decision Tree** : facile à interpréter, mais sujet à l'overfitting.
- **Random Forest** : performant, robuste, interprétable, et bien adapté aux données hétérogènes.

2. Évaluation des performances

Chaque modèle a été entraîné sur le même jeu de données, puis évalué à l'aide d'une **validation croisée à 5 plis**. Les principales métriques utilisées sont:

- **Accuracy (précision globale)**
- **Recall (rappel des cas d'échec détectés)**
- **F1-score (équilibre entre précision et rappel)**

Le tableau suivant résume les performances obtenues :

Modèle	Accuracy (%)	Recall (%)	F1-score (%)
Régression Logistique	83	78	80
KNN	81	75	77
SVM	85	76	79
Decision Tree	86	82	84
Random Forest	90	87	88

3. Justification du choix de Random Forest

Parmi les modèles testés, **Random Forest** a obtenu les meilleurs résultats, avec une précision moyenne de 90 % et un excellent équilibre entre les prédictions de réussite et d'échec. Il offre également plusieurs avantages décisifs :

- Il **réduit les risques de surapprentissage**, en combinant plusieurs arbres.
- Il **gère bien les données hétérogènes**, ce qui est crucial dans notre dataset contenant à la fois des variables numériques et catégorielles.
- Il **fournit des scores d'importance des variables**, facilitant l'interprétation des résultats et l'identification des facteurs les plus influents.

Ce choix est également soutenu par la littérature. Par exemple, **Albreiki et al. (2021)** ont montré que Random Forest surpasse les modèles classiques dans la prédiction académique. De plus, dans les travaux de **Marquez-Vera et al. (2016)**, ce modèle est recommandé pour sa performance et sa lisibilité, critères essentiels dans un contexte éducatif.

V. Conclusion

Ce chapitre a permis d'explorer les travaux existants sur la prédiction de la réussite étudiante ainsi que les principales techniques de machine learning utilisées dans ce domaine. À travers l'analyse de plusieurs études, nous avons constaté que des modèles comme la régression logistique, les SVM, les KNN et surtout le Random Forest sont largement adoptés pour traiter ce type de problématique.

La revue de la littérature a également mis en évidence l'importance de bien préparer les données, de choisir les bonnes variables explicatives, et de privilégier des modèles à la fois performants et interprétables dans un contexte éducatif.

Enfin, la comparaison expérimentale menée dans notre projet a confirmé que le modèle Random Forest offrait les meilleurs résultats en termes de précision, de stabilité et de compréhension des facteurs influents. C'est donc ce modèle que nous avons retenu pour la suite du projet.

Ce socle théorique et comparatif nous permet désormais d'aborder sereinement les étapes suivantes, à commencer par l'analyse fonctionnelle et la conception de notre solution.

Chapitre II : Contexte du project

I. Introduction

Dans le domaine de l'enseignement supérieur, le suivi et l'accompagnement des étudiants jouent un rôle crucial dans l'amélioration du taux de réussite universitaire. Face à l'augmentation du nombre d'étudiants et à la diversité de leurs profils, les établissements sont confrontés à de nombreux défis pour identifier les facteurs qui influencent la réussite ou l'échec académique. Dans ce contexte, les outils traditionnels d'évaluation montrent leurs limites et ne permettent pas toujours de détecter précocement les étudiants en difficulté.

Avec l'émergence de l'intelligence artificielle et du machine learning, de nouvelles opportunités apparaissent pour analyser de grandes quantités de données éducatives et en tirer des informations utiles à la prise de décision. Ces technologies permettent non seulement de visualiser et d'analyser les performances passées des étudiants, mais aussi de prédire leur probabilité de réussite future.

C'est dans cette optique que s'inscrit ce projet, qui vise à développer une application interactive combinant analyse des données et prédiction automatique, afin de proposer un outil d'aide à la décision destiné aux responsables pédagogiques. Ce projet se base sur des techniques de machine learning telles que le clustering pour la segmentation des étudiants selon leurs caractéristiques, et le modèle Random Forest pour la classification binaire du succès académique.

II. Problématique :

Aujourd'hui, les établissements universitaires font face à un défi majeur : **améliorer le taux de réussite des étudiants** tout en tenant compte de la diversité croissante des profils et des situations personnelles. Les causes de l'échec universitaire sont multiples et souvent liées à une combinaison de facteurs académiques, financiers, sociaux et psychologiques. Identifier ces facteurs de manière précoce et objective est une tâche complexe mais essentielle pour proposer des solutions adaptées.

Malgré la disponibilité de nombreuses données sur les étudiants, celles-ci sont souvent sous-exploitées. Or, ces données contiennent des informations précieuses permettant de mieux comprendre les profils à risque. Dans le cadre de ce projet, nous avons utilisé un jeu de données provenant d'une plateforme en ligne, contenant des caractéristiques variées telles que :

- Le **profil personnel** de l'étudiant (âge, situation matrimoniale, statut de résidence, etc.) ;
- Sa **situation financière** (niveau d'endettement, obtention d'une bourse, revenus familiaux) ;
- Des **informations académiques** (notes obtenues, fréquence de participation, engagement sur la plateforme) ;
- Ainsi que des **indicateurs comportementaux** (accès aux ressources pédagogiques en ligne, interactions avec le système).

Face à cette richesse de données, une problématique centrale se pose : **Comment analyser efficacement ces informations hétérogènes afin de détecter les étudiants à risque et prédire leur probabilité de réussite ou d'échec académique ?**

Pour répondre à cette question, nous avons mis en œuvre des techniques de machine learning, telles que le **clustering** pour regrouper les étudiants selon leurs caractéristiques similaires, et un modèle de **classification Random Forest** pour prédire le succès ou l'échec à partir de ces variables. L'objectif est de fournir un outil décisionnel fiable et compréhensible aux responsables pédagogiques pour améliorer l'accompagnement étudiant.

III. Objectif du projet :

L'objectif principal de ce projet est de **concevoir une solution intelligente d'analyse et de prédiction** des performances académiques des étudiants, afin de soutenir les établissements universitaires dans leur démarche d'amélioration du taux de réussite.

Plus précisément, le projet vise à :

- **Analyser** les données disponibles sur les étudiants (personnelles, financières, académiques et comportementales) afin d'identifier les facteurs les plus significatifs influençant la réussite universitaire.
- **Segmenter** les étudiants en groupes homogènes à l'aide d'algorithmes de **clustering**, pour mieux comprendre les différents profils et adapter les stratégies d'accompagnement.
- **Prédire** la probabilité de réussite ou d'échec d'un étudiant en utilisant un modèle de **classification binaire** basé sur l'algorithme **Random Forest**, connu pour sa robustesse et son efficacité.

- **Développer une interface web** intuitive comprenant :
 - Un **Dashboard interactif** pour la visualisation et l'analyse des données étudiantes
 - Un **formulaire de prédiction** permettant de saisir les informations d'un étudiant et d'obtenir une estimation de son risque d'échec ou de succès.
- **Fournir un outil d'aide à la décision** aux responsables pédagogiques, afin qu'ils puissent repérer les étudiants en difficulté dès les premières phases de leur parcours et mettre en place des actions correctives ciblées.

En résumé, ce projet allie analyse de données, modélisation prédictive et visualisation interactive pour proposer une solution concrète et utile au service de la réussite étudiante.

IV. Contraintes du Project :

Comme tout projet de data science, la mise en œuvre de cette solution a été confrontée à plusieurs **contraintes techniques et méthodologiques** qui ont influencé certaines décisions de conception et de traitement des données.

Parmi les principales contraintes rencontrées :

- **Qualité des données** : Le jeu de données utilisé contenait un grand nombre de colonnes, dont certaines étaient **mal nommées, peu explicites ou non documentées**. Cela a nécessité un travail important de nettoyage, d'exploration et de sélection des variables pertinentes. Certaines colonnes ont été **exclues de l'analyse**, car leur signification était incertaine ou leur contribution aux modèles difficile à interpréter.
- **Déséquilibre des classes** : Une autre contrainte majeure a été la **présence de données déséquilibrées** au niveau de la variable cible (réussite vs échec). En effet, le nombre d'étudiants ayant réussi était largement supérieur à celui des étudiants en situation d'échec, ce qui a pu influencer la performance du modèle de classification. Des techniques d'évaluation spécifiques (comme la matrice de confusion, la précision, le rappel et le F1-score) ont été privilégiées pour compenser ce déséquilibre.
- **Temps de traitement et complexité des modèles** : L'application d'algorithmes comme le clustering et Random Forest, combinée à l'intégration dans une interface web, a également imposé des **choix techniques** pour garantir un bon compromis entre **performance, précision et temps de réponse**.

Malgré ces contraintes, des solutions adaptées ont été mises en place pour assurer la fiabilité et

la cohérence du projet, tout en respectant ses objectifs initiaux.

V. Conclusion:

Ce projet s'inscrit dans une dynamique d'innovation pédagogique, en tirant parti des outils de l'intelligence artificielle pour répondre à un enjeu majeur de l'enseignement supérieur : l'amélioration du taux de réussite des étudiants. À travers l'analyse de données variées issues d'une plateforme en ligne et l'utilisation d'algorithmes de machine learning, il propose une approche proactive et personnalisée du suivi étudiant.

L'introduction a permis de poser le cadre général du projet, la problématique a mis en évidence les défis liés à l'identification des étudiants à risque, et les objectifs ont clairement défini les apports attendus de l'application développée.

Enfin, les contraintes rencontrées ont souligné la complexité du traitement des données éducatives et l'importance d'une préparation rigoureuse des données pour garantir la fiabilité des résultats.

Cette première partie pose ainsi les fondations théoriques et pratiques du projet. La suite du rapport détaillera les méthodes utilisées, le développement de l'application, ainsi que les résultats obtenus et les perspectives d'amélioration.

Chapitre III : Fondements théoriques

I. Introduction:

Avant d'aborder la mise en œuvre pratique du projet, il est essentiel de présenter les fondements théoriques qui ont guidé les choix méthodologiques et techniques. Le machine learning, au cœur de ce travail, repose sur des concepts clés et des algorithmes spécifiques qui permettent d'extraire de l'information à partir de données brutes.

Ce chapitre a pour objectif d'exposer les principes de base du machine learning ainsi que les méthodes utilisées dans le cadre de ce projet, notamment le clustering pour la segmentation des étudiants et la classification binaire pour la prédiction de la réussite. L'algorithme Random Forest, choisi pour la tâche de classification, sera présenté en détail. Ce chapitre abordera également l'importance de la qualité des données, le prétraitement nécessaire à toute analyse fiable, ainsi que les outils technologiques mobilisés pour la réalisation du système.

Ces connaissances théoriques constituent le socle indispensable à la compréhension des choix opérés dans la phase de développement présentée dans les chapitres suivants.

II. Analyse des étudiants par machine learning

1. Impact du profil socio-académique sur la réussite

La situation financière et le profil académique d'un étudiant constituent des facteurs déterminants dans son parcours universitaire. Des études en sciences de l'éducation et en psychologie sociale ont démontré que les difficultés économiques peuvent augmenter le stress cognitif et réduire l'engagement académique (Jury et al., 2015). Un étudiant endetté ou sans bourse peut être contraint de travailler en parallèle de ses études, ce qui limite le temps et l'énergie consacrés à l'apprentissage. À l'inverse, bénéficier d'un soutien financier (bourse, exonération de frais) est souvent corrélé à une meilleure stabilité émotionnelle et à une progression plus régulière.

Sur le plan académique, les antécédents scolaires comme les notes obtenues dans les semestres précédents sont également des indicateurs prédictifs puissants. Les modèles de machine learning que nous avons utilisés confirment que les variables telles que la réussite aux unités d'enseignement et la régularité des inscriptions sont fortement liées à la probabilité de succès.

Ainsi, l'analyse croisée du profil financier et académique permet de détecter plus tôt les

étudiants à risque, et d'envisager des politiques d'accompagnement ciblées pour améliorer le taux de réussite globale à l'université.

2. Les principaux types d'apprentissage appliquée

Apprentissage supervisé :

- Dans ce cas, le modèle est entraîné à partir d'un jeu de données étiqueté, c'est-à-dire que chaque exemple d'entraînement est associé à **une valeur cible connue** (par exemple, réussite = 1 ou échec = 0). L'objectif est d'apprendre une fonction capable de prédire cette valeur sur de nouvelles données. C'est le type d'apprentissage utilisé dans ce projet pour la **classification binaire**.

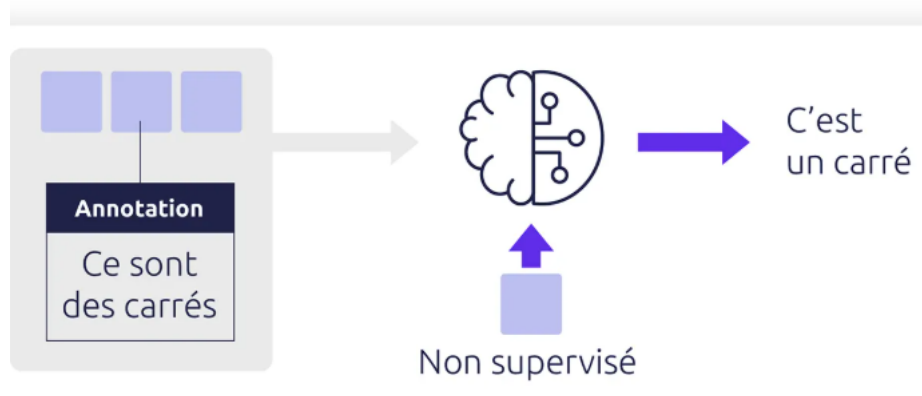


Figure 1: apprentissage supervise

Apprentissage non supervisé :

- Contrairement à l'apprentissage supervisé, ici **aucune étiquette** n'est fournie. L'objectif est de détecter des structures cachées ou des regroupements dans les données. Le clustering (ou regroupement) en est un exemple typique, utilisé dans notre projet pour **segmenter les étudiants selon leurs profils**.

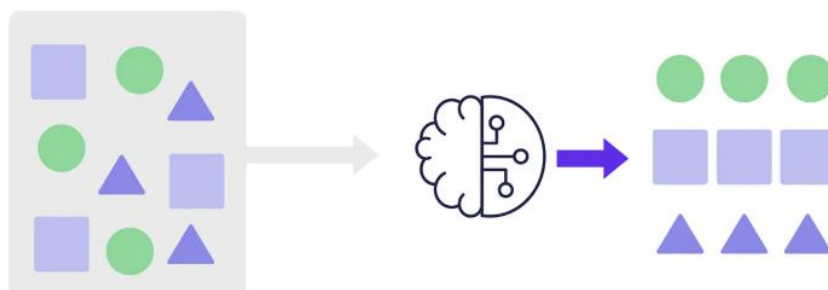


Figure 2: apprentissage non supervisé

Apprentissage semi-supervisé :

- Ce type d'apprentissage utilise à la fois des données **étiquetées et non étiquetées**, et est utile lorsque l'étiquetage manuel est coûteux ou limité. Ce n'est pas le cas de notre projet, mais il représente un domaine intéressant dans le contexte de l'éducation.

3. Architecture de notre projet

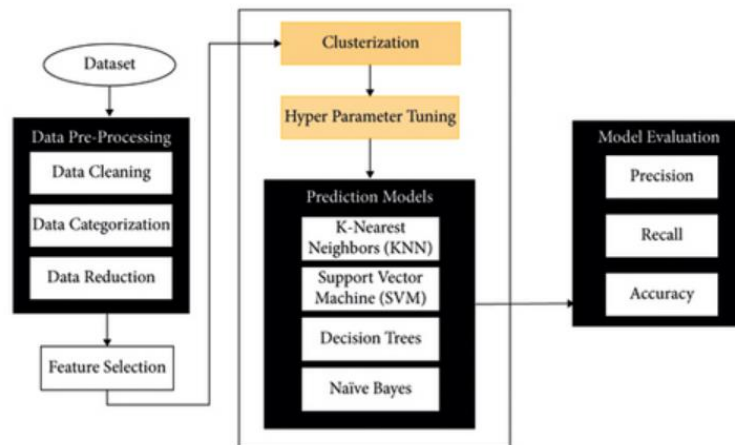


Figure 3 : architecture du projet

L'architecture de notre projet suit une approche structurée en plusieurs étapes clés pour l'analyse prédictive des performances étudiantes. Le processus commence par la préparation des données, incluant le nettoyage, la catégorisation et la réduction des données pour garantir leur qualité et leur pertinence. Ensuite, une sélection rigoureuse des caractéristiques est effectuée pour identifier les variables les plus influentes. La phase de clustering permet de segmenter les étudiants en groupes homogènes, facilitant une analyse ciblée.

Pour la modélisation, plusieurs algorithmes de prédiction sont testés, tels que K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees et Naïve Bayes, suivis d'un réglage fin des hyperparamètres pour optimiser leurs performances. Enfin, les modèles sont évalués à l'aide de métriques comme la précision, le rappel et l'exactitude, assurant ainsi leur robustesse et leur fiabilité. Cette architecture permet une analyse complète et des prédictions précises pour soutenir la prise de décision pédagogique.

4. Les étapes de traitement effectuée

Un projet de machine learning suit généralement un cycle composé des étapes suivantes :

Étape	Description
Collecte des données	Réunir les informations nécessaires pour l'entraînement du modèle.
Prétraitement des données	Nettoyage, traitement des valeurs manquantes, normalisation, encodage.
Séparation du jeu de données	Division en jeu d'entraînement, de validation et de test.
Choix de l'algorithme	Sélection du modèle le plus adapté (ex. : Random Forest).
Entraînement du modèle	Ajustement des paramètres à partir des données d'entraînement.
Évaluation	Analyse des performances du modèle sur des données non vues.
Déploiement	Intégration du modèle dans une application

Figure 4 : Etapes de traitement

III. Le Clustering (Apprentissage non supervisé)

1. Definition et objectif:

Le clustering, ou regroupement non supervisé, est une technique d'apprentissage automatique visant à regrouper des objets ou des individus similaires au sein de groupes (appelés clusters) sans connaissance préalable des étiquettes ou des classes. L'objectif est de découvrir des structures naturelles dans les données et de mieux comprendre la diversité des profils présents.

Dans le cadre de ce projet, le clustering a été utilisé pour identifier des groupes d'étudiants ayant des caractéristiques communes, tels que la situation financière, le statut marital, l'endettement ou l'obtention d'une bourse. Ces regroupements permettent d'adapter les stratégies d'accompagnement pédagogique en fonction des besoins spécifiques de chaque groupe.

2. Algorithme utilisé: K-Means Clustering

L'algorithme de K-Means a été retenu pour ce projet en raison de sa simplicité, de sa rapidité et de son efficacité pour traiter des jeux de données de taille moyenne. Il s'agit d'un algorithme itératif qui partitionne les données en K groupes distincts, en minimisant la distance intra-cluster (distance entre chaque point et le centre de son groupe).

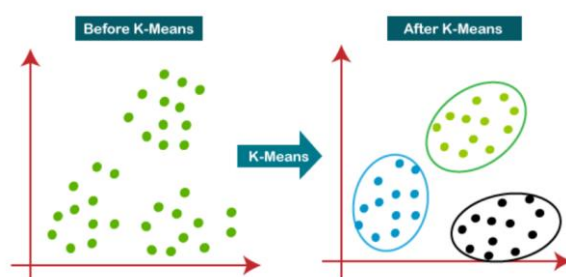


Figure 5: k-means fonctionnement

3. Étapes du clustering avec K-Means

Voici les étapes générales appliquées lors de l'utilisation de K-Means

Étape	Description
Prétraitement des données	Sélection des variables pertinentes (finances, profil social, statut académique, etc.) Normalisation des données
Choix du nombre de clusters (K)	- Utilisation de la méthode du coude (elbow method) et visualisation de la courbe d'inertie pour repérer le point de rupture
Initialisation des centroïdes	Placement aléatoire de K centroïdes dans l'espace des données
Affectation des points aux clusters	Chaque point (étudiant) est affecté au centroïde le plus proche selon la distance euclidienne
Mise à jour des centroïdes	Recalcul des centroïdes à partir des points assignés à chaque cluster

Itération jusqu'à convergence	Répétition des étapes d'affectation et de mise à jour jusqu'à stabilisation ou nombre maximal d'itérations
Interprétation des résultats	- Analyse des caractéristiques de chaque cluster- Exploitation des clusters pour améliorer le dashboard ou guider des actions pédagogiques

Figure 6 : Etapes k-means

IV. La Classification Binaire avec Random Forest

I. Définition:

La classification binaire est une technique d'apprentissage supervisé visant à prédire une variable cible ayant deux classes possibles, généralement représentées par 0 et 1. Dans notre projet, il s'agit de prédire si un étudiant va réussir (1) ou échouer (0) à son parcours académique, à partir d'un ensemble de caractéristiques (profil financier, statut social, situation familiale, etc.).

Ce type de classification nécessite un jeu de données étiqueté, où chaque observation est associée à un résultat connu. Le modèle apprend ainsi à reconnaître les relations entre les variables d'entrée et la classe cible, pour pouvoir faire des prédictions sur de nouveaux cas.

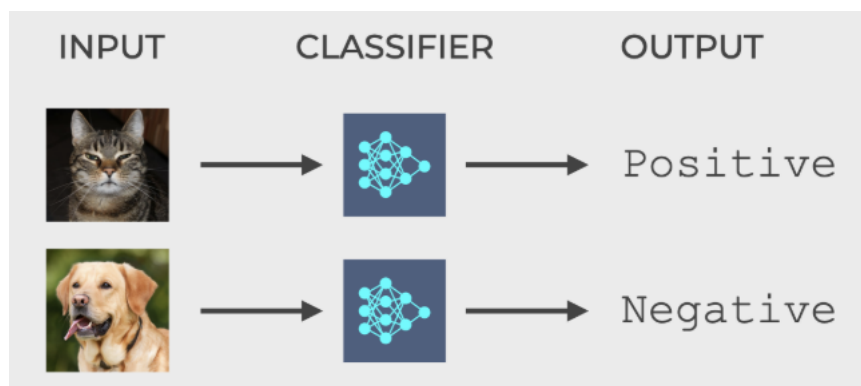


Figure 7: Classification binaire

II. Algorithme utilisé : Random Forest

Pour accomplir cette tâche de classification, nous avons choisi d'utiliser l'algorithme **Random Forest**, un modèle robuste et largement utilisé dans les projets de machine learning supervisé.

Random Forest est un **ensemble d'arbres de décision**, construit selon la méthode du bagging (Bootstrap Aggregation). Au lieu d'entraîner un seul arbre, le modèle en entraîne plusieurs,

chacun sur un **échantillon aléatoire** du jeu de données. Lorsqu'un nouvel échantillon est présenté, chaque arbre **donne une prédiction**, et la forêt vote pour la classe majoritaire (en classification).

Ce fonctionnement permet de **réduire la variance** des modèles individuels (arbres), et donc d'obtenir des prédictions plus stables et plus fiables.

Random Forest

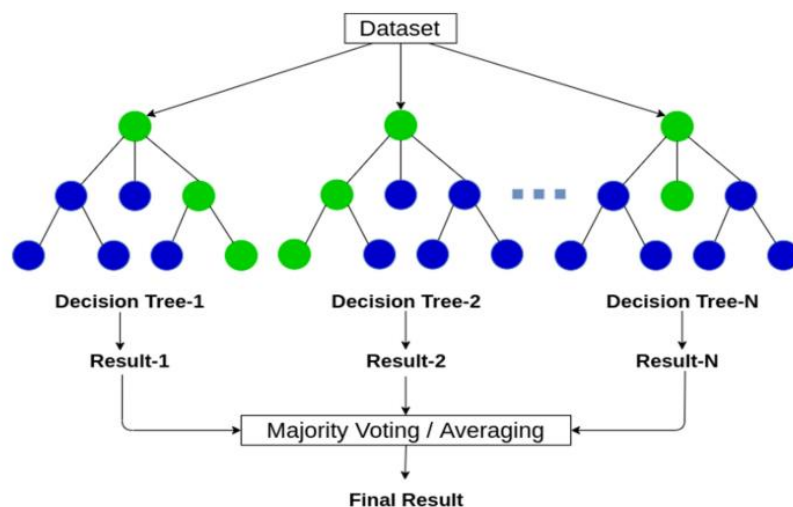


Figure 8 : random forest

V. L'importance de la qualité des données

1. Pourquoi la qualité des données est essentielle

Dans tout projet de machine learning, la qualité des données utilisées est un facteur déterminant pour la performance du modèle. Même les algorithmes les plus performants ne peuvent donner de bons résultats s'ils sont alimentés par des données erronées, incomplètes, ou mal structurées. Une donnée de mauvaise qualité peut conduire à des modèles biaisés, à des prédictions incohérentes et, dans le cas de notre projet, à des décisions pédagogiques potentiellement inefficaces ou injustes.

2. Les étapes de prétraitement effectuées

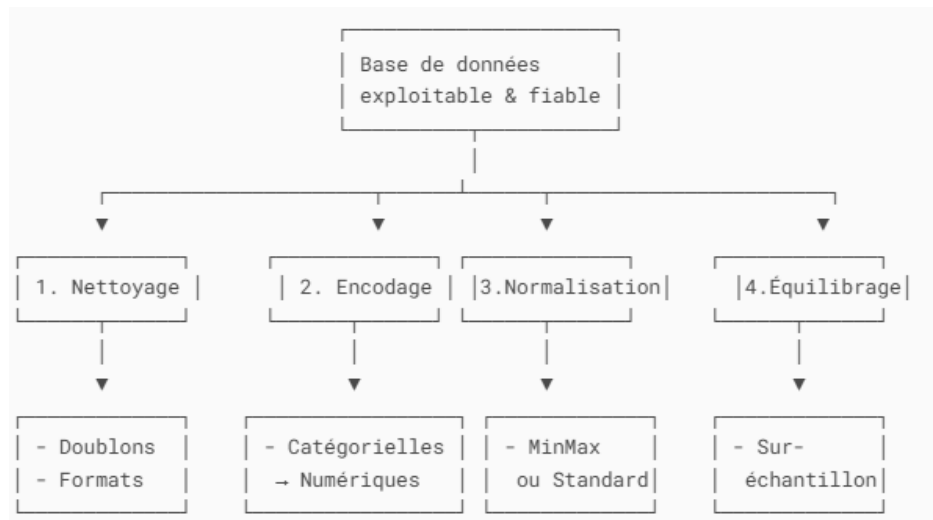


Figure 9 : étapes de prétraitement

Une base de données bien préparée a un impact direct sur la **précision des prédictions**, la **stabilité du modèle**, la **vitesse d'apprentissage**, ainsi que sur la **compréhension des résultats**. En effet, un modèle entraîné sur des données propres, cohérentes et représentatives est beaucoup plus fiable et performant. Dans notre projet, une attention particulière a été portée à cette étape cruciale du prétraitement, car elle conditionne la **qualité globale et la fiabilité du système prédictif** que nous avons mis en place.

VI. Conclusion

Dans ce chapitre, nous avons présenté les principes fondamentaux qui sous-tendent notre projet de machine learning. Nous avons d'abord introduit les notions de clustering avec l'algorithme K-Means, qui nous a permis d'identifier des groupes d'étudiants partageant des caractéristiques similaires. Ensuite, nous avons détaillé le fonctionnement de la classification binaire, en mettant l'accent sur l'algorithme Random Forest, choisi pour sa robustesse, sa capacité à gérer des données complexes

Chapitre IV : Analyse et Conception

I. Introduction

Ce chapitre a pour objectif de présenter l'analyse fonctionnelle et technique du système développé. Il détaille les différents acteurs qui interagissent avec l'application, les cas d'utilisation associés, ainsi que la structure générale du système à travers des diagrammes UML. L'objectif est de définir clairement les fonctionnalités attendues et l'architecture du projet, avant la phase de réalisation.

II. Analyse des besoins

1. Objectifs fonctionnels du système

Le système vise à fournir un outil intelligent et interactif pour :

- **Analyser les performances académiques** des étudiants à l'aide de visualisations graphiques.
- **Identifier des groupes d'étudiants** aux profils similaires à l'aide de l'algorithme de **clustering K-Means**.
- **Prédire la réussite ou l'échec** d'un étudiant en se basant sur un ensemble de caractéristiques personnelles, financières et académiques via un modèle **Random Forest**.
- **Permettre à l'administrateur** de remplir un formulaire de prédiction personnalisé et d'obtenir une **analyse détaillée du résultat**.
- **Afficher les cas d'étudiants non encore prédits**, afin de mieux cibler les suivis pédagogiques.

2. Objectifs non fonctionnels du système

Le système doit également respecter un certain nombre de contraintes techniques et ergonomiques :

- **Simplicité d'utilisation** : une interface intuitive, accessible même sans compétence technique.

- **Temps de réponse rapide** pour la génération de prédictions et l’affichage des graphiques.
- **Robustesse des prédictions**, grâce à un algorithme fiable et performant.
- **Modularité et évolutivité** : possibilité d’adapter ou d’étendre le système à d’autres cas d’usage.
- **Lisibilité des résultats** : rendre les sorties du modèle compréhensibles pour des utilisateurs non techniques.

III. Acteurs et utilisateurs cibles

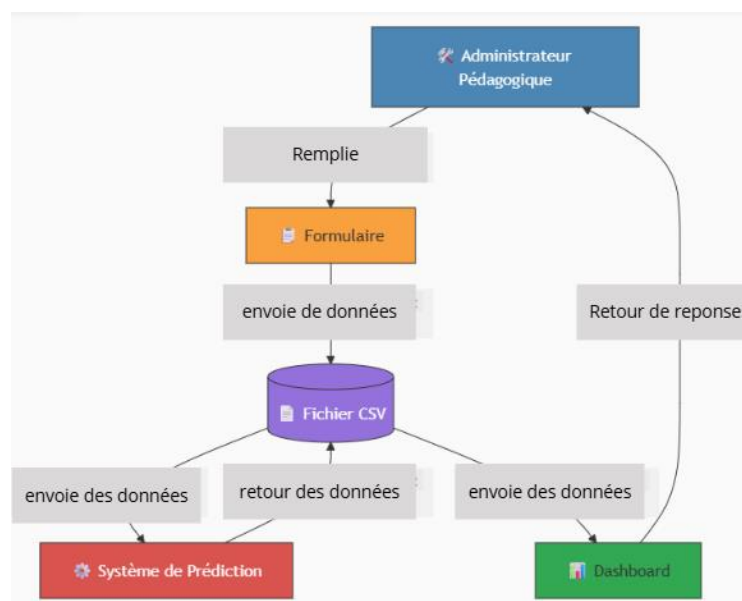


Figure 10 : acteurs et utilisateurs cibles

Le système est destiné principalement à :

- **Administrateur pédagogique** : acteur principal qui interagit avec les deux modules (dashboard et formulaire).
- **Système de prédiction** (composant interne) : exécute les calculs à partir des données saisies.
- **Étudiant (indirectement)** : bien qu’il n’utilise pas directement le système, ses données sont au cœur de l’analyse.

IV. Diagramme de cas d'utilisation (UML)

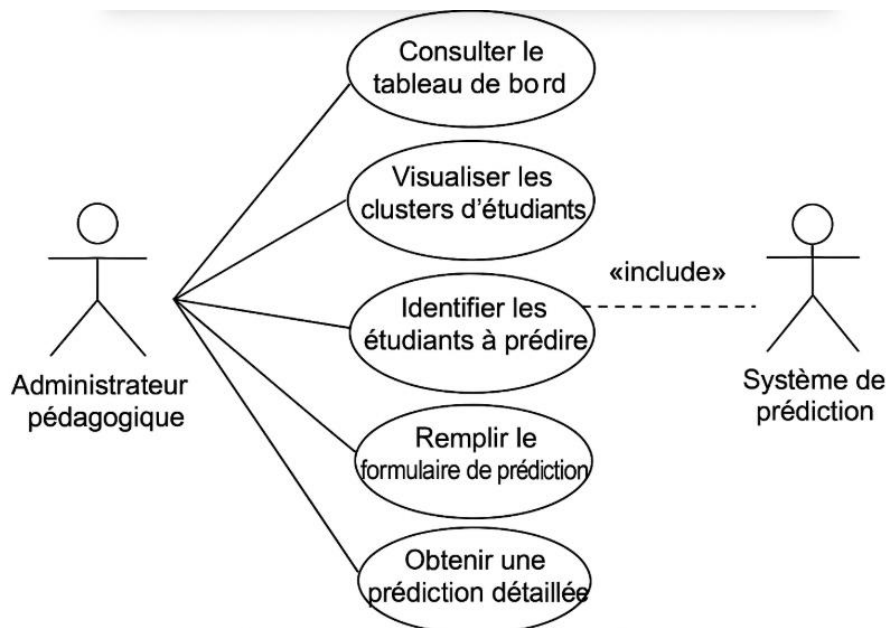


Figure 11 : diagramme de cas d'utilisation

Ce diagramme de cas d'utilisation illustre les interactions entre l'**Administrateur pédagogique** et un système d'analyse prédictive pour la réussite étudiante. L'acteur principal, placé à gauche, peut **consulter le tableau de bord** pour visualiser les **clusters d'étudiants** (groupes homogènes), **uploader un fichier CSV** afin d'effectuer une **prédiction automatisée** de réussite, ou **identifier manuellement** des étudiants pour une **prédiction individuelle** via un formulaire. Les résultats détaillés, issus des deux modes de prédiction (fichier ou formulaire), peuvent ensuite être **exportés ou enregistrés**. Les flèches indiquent les dépendances logiques entre les cas d'utilisation, comme l'obligation d'importer un fichier avant une prédiction groupée, ou de remplir un formulaire pour obtenir une analyse ciblée. Ce diagramme met en évidence un flux cohérent, allant de la collecte des données à l'exploitation des résultats, avec des options flexibles adaptées aux besoins de l'administrateur.

V. Conclusion

Ce chapitre a permis d'explorer les bases théoriques indispensables à la compréhension et à la mise en œuvre de notre projet. Nous avons d'abord défini les principaux facteurs influençant la réussite académique des étudiants, en tenant compte des aspects personnels, académiques et socio-économiques. Ensuite, nous avons présenté les méthodes d'analyse de données et les étapes essentielles de la préparation des données, notamment la sélection des variables, le traitement des valeurs manquantes et la normalisation.

Enfin, une attention particulière a été accordée aux techniques de machine learning utilisées dans notre application, avec un focus sur les algorithmes de classification, notamment le Random Forest, choisis pour leur efficacité, leur robustesse et leur capacité à gérer des données hétérogènes. Ces éléments théoriques posent ainsi un cadre solide pour passer à l'implémentation technique, en assurant que les choix effectués soient scientifiquement justifiés et adaptés aux objectifs du projet.

Chapitre V : Outils Utilisée

I. Introduction

Nous avons également insisté sur l'importance cruciale de la qualité des données, en soulignant les différentes étapes de nettoyage, de transformation et d'adaptation des variables. Ces fondements théoriques nous ont permis de bâtir une base solide pour le développement de notre application, qui vise à améliorer la compréhension et la prédiction des performances académiques des étudiants. Le chapitre suivant portera sur la mise en œuvre pratique du projet, de l'acquisition des données jusqu'à l'évaluation des résultats obtenus.

La réalisation de ce projet de fin d'études a nécessité l'utilisation d'un ensemble cohérent d'outils, de bibliothèques logicielles et de langages de programmation. Le choix de ces technologies s'est basé sur leur efficacité, leur popularité dans la communauté Data Science, leur compatibilité entre elles, ainsi que sur leur facilité d'utilisation dans un contexte académique

II. Python

Python est un langage de programmation polyvalent, apprécié pour sa simplicité de syntaxe et la richesse de ses bibliothèques. Il est particulièrement adapté aux projets de data science grâce à sa large communauté et à ses nombreux outils spécialisés dans le traitement des données, la visualisation et l'apprentissage automatique



III. NumPy

(Numerical Python) permet la manipulation de tableaux multidimensionnels et offre une multitude de fonctions mathématiques performantes.



IV. Pandas

Pandas est une bibliothèque orientée vers la manipulation de données sous forme de tableaux (DataFrames). Elle permet :

- L'importation et l'exportation de fichiers CSV.
- La gestion des données manquantes.
- Le filtrage, le tri, la fusion et l'agrégation des données



V. Matplotlib et Seaborn

Matplotlib : est la bibliothèque de base permettant de générer des graphiques personnalisés (courbes, histogrammes, nuages de points, etc.).

Seaborn : repose sur Matplotlib mais fournit des visualisations plus élaborées et esthétiques, notamment pour les corrélations, les répartitions de variables ou les heatmaps



VI. Scikit-learn (sklearn) :

Scikit-learn est une bibliothèque de référence pour le machine learning en Python. Elle propose

- Des algorithmes de régression (linéaire, ridge, lasso...).
- Des modèles d'ensemble (Random Forest, Gradient Boosting...).

- Des outils de prétraitement (encodage, normalisation).
- Des métriques d'évaluation (MAE, RMSE, R^2).
- Des méthodes de validation croisée.

Son intégration dans le projet a permis d'entraîner et d'évaluer différents modèles de prédiction des prix des véhicules.

Interface utilisateur



VII. Interface utilisateur

Streamlit : est un framework Python permettant de créer des interfaces web interactives pour la data science et le machine learning. Il a été utilisé pour concevoir une interface utilisateur intuitive dans laquelle l'utilisateur peut entrer les caractéristiques d'une voiture (année, kilométrage, marque, etc.) et obtenir instantanément une prédiction du prix.

Caractéristiques principales :

- Développement rapide sans HTML/CSS
- Widgets interactifs (menus déroulants, sliders, boutons)
- Intégration directe avec le backend Flask
- Adapté aux démonstrations de projets de machine learning



Plotly : est une bibliothèque de visualisation de données interactive utilisée en Python (et dans d'autres langages comme R et JavaScript). Elle permet de créer des graphiques dynamiques, intuitifs et esthétiques, tels que des histogrammes, des diagrammes en barres, des courbes, des cartes ou encore des dashboards interactifs.



Chapitre VI : Réalisation

I. Introduction:

Ce chapitre détaille les différentes étapes de mise en œuvre technique de notre projet de machine learning dédié à l'analyse et à la prédiction des performances académiques des étudiants. Après avoir défini les besoins fonctionnels et conçu l'architecture générale du système, la phase de réalisation a permis de concrétiser le projet à travers le développement de deux composants principaux : le traitement des données et la construction de l'interface utilisateur.

La première partie a consisté à préparer et analyser les données, en appliquant des techniques de clustering (K-Means) pour identifier les groupes d'étudiants similaires, et en construisant un modèle de classification binaire basé sur l'algorithme Random Forest pour prédire les cas de réussite ou d'échec.

La seconde partie s'est concentrée sur la création d'une interface web interactive à l'aide de Dash et Plotly, comprenant un dashboard d'analyse et un formulaire de prédiction personnalisé. Ce chapitre présente donc les choix technologiques, les étapes de développement, ainsi que l'intégration des différents modules qui composent le système final.

II. Création du modèle de Machine Learning

1. Chargement des données de Kaggle

Les données utilisées dans ce projet ont été importées depuis la plateforme **Kaggle**, sous forme d'un fichier CSV contenant initialement **35 colonnes** décrivant le profil, la situation financière, et les performances académiques des étudiants universitaires.

Cependant, après une phase de nettoyage et de sélection des variables pertinentes pour la modélisation, seules les colonnes jugées significatives ont été conservées. Ce processus a permis de réduire le bruit, d'améliorer la qualité des données et de faciliter l'entraînement du modèle.

2. Nettoyage des données

Avant d'entraîner un modèle de machine learning, il est essentiel de passer par une étape rigoureuse de nettoyage des données. Cette phase a pour but de garantir la qualité, la cohérence et la pertinence des variables utilisées.

Le processus de nettoyage des données a suivi plusieurs étapes essentielles pour garantir la qualité et la fiabilité des analyses :

- **Suppression des colonnes indésirables** : Plusieurs colonnes ont été retirées du jeu de données initial en raison de leur faible valeur ajoutée, de leur redondance ou d'un grand nombre de valeurs manquantes. Cela a permis d'alléger la structure et de se concentrer sur les variables les plus pertinentes pour l'analyse et la prédiction.
- **Traitement des valeurs manquantes** : Les lignes contenant trop de valeurs nulles ont été supprimées pour éviter de fausser les résultats. Pour certaines colonnes quantitatives partiellement incomplètes, une **imputation par la médiane** a été appliquée, afin de conserver une distribution robuste sans être influencée par des extrêmes.

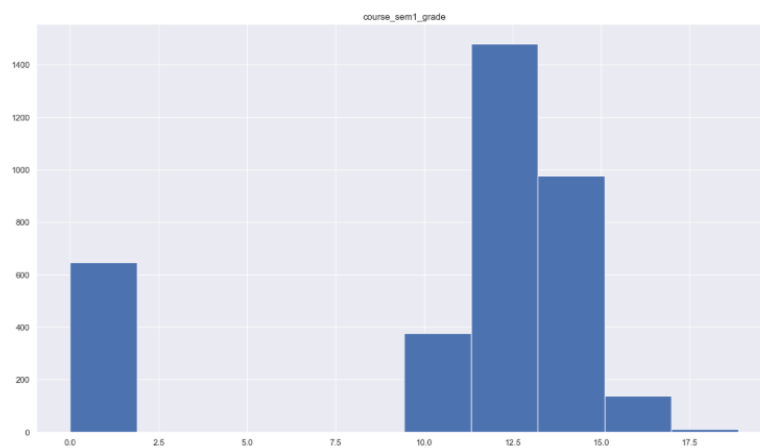


Figure 12 : distribution des données

- **Suppression des valeurs aberrantes (outliers)** : Une analyse statistique a permis d'identifier et d'éliminer les données anormalement éloignées de la moyenne, susceptibles de nuire à l'apprentissage du modèle.

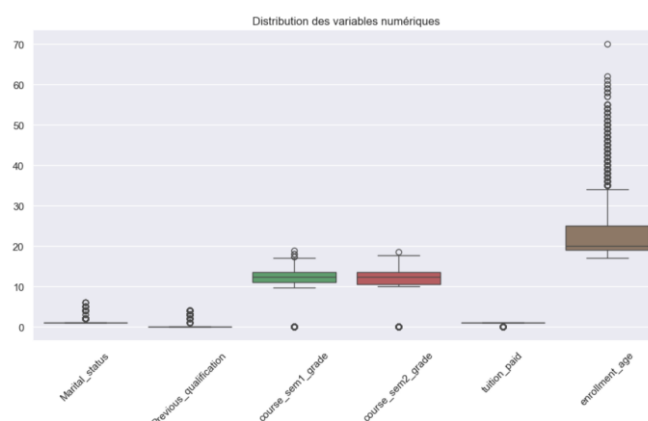


Figure 13 : suppression des valeurs aberrantes

- Dans le but d'améliorer la qualité des données et d'éviter que des valeurs extrêmes n'influencent négativement l'entraînement du modèle, une **détection des valeurs aberrantes** a été réalisée à l'aide de la méthode du **Z-score**.
- Le **Z-score** mesure l'écart entre une valeur donnée et la moyenne de sa distribution, en unités d'écart-type. Une valeur dont le Z-score est supérieur à un seuil (généralement ± 3) est considérée comme un **outlier** potentiel. Dans notre projet, ce seuil a été utilisé pour identifier et supprimer les observations anormalement éloignées de la moyenne.

Grâce à ces étapes, le jeu de données final obtenu est propre, équilibré, cohérent, et parfaitement exploitable pour les phases de clustering et de classification.

3. Prétraitements des données

Le prétraitement des données constitue une étape cruciale dans la construction d'un modèle fiable. Il comprend le nettoyage des données, le traitement des valeurs manquantes, l'encodage des variables catégorielles, la gestion des outliers ainsi que l'équilibrage des classes à l'aide de SMOTE. Ces opérations visent à améliorer la qualité des données, garantir une meilleure performance du modèle et assurer des prédictions plus précises.

Équilibrage des données avec SMOTE :

Avant l'entraînement du modèle, une analyse exploratoire de la variable cible (réussite ou échec de l'étudiant) a été réalisée afin d'évaluer la répartition des classes. Un diagramme circulaire (pie chart) a permis de mettre en évidence un déséquilibre important, avec une majorité d'étudiants appartenant à une seule classe (souvent ceux ayant réussi)

Ce déséquilibre peut entraîner un biais du modèle, qui risque de favoriser la classe majoritaire au détriment de la minoritaire.

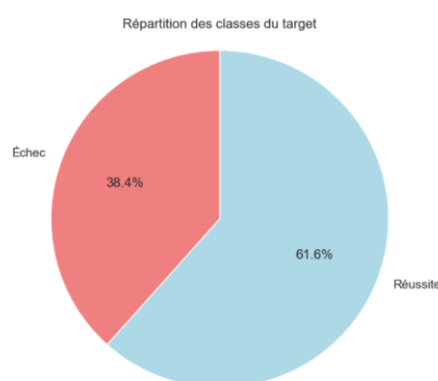


Figure 14: répartition des classes target

Pour corriger ce problème, nous avons appliqué la méthode **SMOTE (Synthetic Minority Over-sampling Technique)**. Il s'agit d'une technique de sur-échantillonnage qui permet de **générer artificiellement de nouvelles instances** pour la classe minoritaire en interpolant entre les exemples existants.

Grâce à SMOTE, les données sont devenues **plus équilibrées**, ce qui a permis au modèle d'apprentissage d'être **moins biaisé et plus performant** pour prédire les deux classes (succès ou échec académique).

Sélection des variables pertinentes :

Une fois les données préparées et encodées, il était essentiel de ne conserver que les **variables les plus utiles** pour la prédiction, afin d'**améliorer l'efficacité du modèle** et **réduire la complexité** du système.

Pour cela, nous avons utilisé une approche basée sur l'**importance des variables** fournie par le modèle Random Forest. Chaque variable se voit attribuer un score reflétant sa contribution à la qualité de la prédiction.

Les **variables ayant une importance inférieure à 0.1** ont été considérées comme **peu significatives** et donc **supprimées** du jeu de données final. Cette sélection a permis de :

- Réduire le risque de surapprentissage (overfitting),
- Accélérer le temps d'entraînement,
- Améliorer l'interprétabilité des résultats.

Ce filtrage ciblé des variables a contribué à rendre le modèle plus robuste, plus rapide et centré sur les facteurs réellement impactants dans la réussite ou l'échec académique des étudiants.

Encodage des données :

Afin de rendre les variables catégorielles exploitables par les algorithmes de machine learning, un encodage adapté a été appliqué. Pour les variables **à multiples modalités sans relation d'ordre**, telles que Marital_status et Previous_qualification, nous avons utilisé le **Label Encoding**, qui attribue un entier unique à chaque catégorie. Cette méthode est simple et efficace, notamment lorsqu'elle est combinée avec des modèles capables de gérer des variables ordinales ou non linéaires comme les forêts d'arbres.

Quant aux variables **binaires** telles que Displaced, Debtor, tuition_paid, Gender et has_scholarship, un **Binary Encoding** (encodage binaire sous forme de 0 et 1) a été appliqué.

Ce type d'encodage est bien adapté pour exprimer des états binaires (oui/non, vrai/faux) et permet de conserver la signification logique tout en restant compatible avec les modèles d'apprentissage.

Variable	Valeurs originales	Valeurs encodées
Marital_status	Single, Married, Widowed	0, 1, 2
Previous_qualification	High School, Bachelor, Other	0, 1, 2
Displaced	Yes / No	1 / 0
Debtor	Yes / No	1 / 0
tuition_paid	Yes / No	1 / 0
Gender	Male / Female	0 / 1
has_scholarship	Yes / No	1 / 0

Figure 15 : encodage des données

Standardisation des données numériques:

Dans le but d'assurer une **échelle homogène entre les variables numériques** et d'optimiser les performances du modèle, une **standardisation** a été appliquée. Cette étape consiste à centrer et réduire les données, c'est-à-dire à soustraire la moyenne puis diviser par l'écart-type. Les variables concernées par cette transformation sont : enrollment_age, course_sem1_enrolled, course_sem1_passed, course_sem1_grade, course_sem2_enrolled, course_sem2_passed et course_sem2_grade. La standardisation permet de **prévenir l'influence disproportionnée** des variables à grande échelle sur le modèle et favorise une **convergence plus rapide** lors de l'entraînement.

4. Division des données

L'entraînement du modèle a été réalisé en plusieurs étapes successives, en commençant par la **division du jeu de données** en deux sous-ensembles : **80 % pour l'entraînement** et **20 % pour les tests**, afin d'évaluer les performances du modèle sur des données jamais vues.



Figure 16 : separation des données

Dans une première phase exploratoire, nous avons appliqué un **clustering non supervisé** à l'aide de l'algorithme **KMeans**. Pour déterminer le nombre optimal de groupes, la **méthode du coude** (Elbow Method) a été utilisée.

5. Clustering utilisant k-means

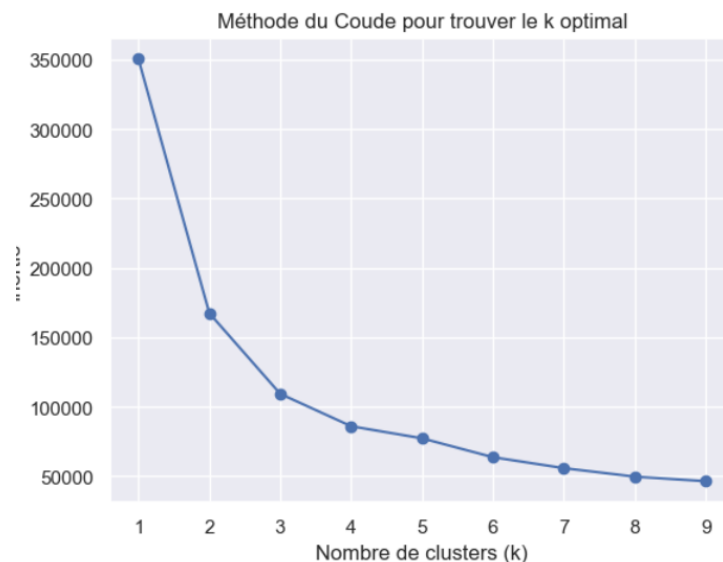


Figure 17 :elbow method

Indiquant que **3 clusters** représentaient une segmentation pertinente des profils étudiants. Le clustering a été réalisé sur un sous-ensemble de variables jugées significatives : Marital_status ,course_sem1_grade , course_sem2_grade enrollment_age.

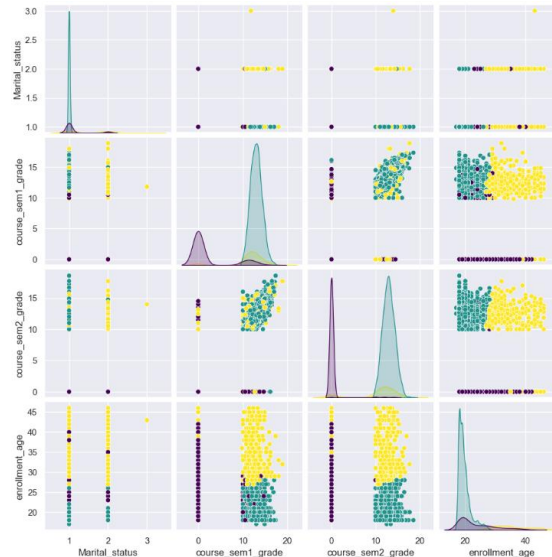


Figure 18 : kmeans exemple

6. Validation croisée et choix du modèle

Afin d'identifier le modèle le plus performant pour la classification binaire (réussite ou échec), une validation croisée (cross-validation) a été réalisée sur plusieurs algorithmes : Logistic Regression, Random Forest, Support Vector Machine, Decision Tree et K-Nearest Neighbors. Chaque modèle a été évalué selon plusieurs métriques : accuracy, precision, recall et f1-score. Parmi tous les modèles, Random Forest s'est distingué par la meilleure accuracy moyenne (0.90 ± 0.01) et des scores F1 équilibrés pour les deux classes, indiquant une excellente capacité à bien généraliser. Ce modèle a donc été retenu comme modèle final pour la prédiction des performances académiques des étudiants, en raison de sa robustesse, sa résistance au surapprentissage, et sa capacité à gérer les interactions complexes entre les variables

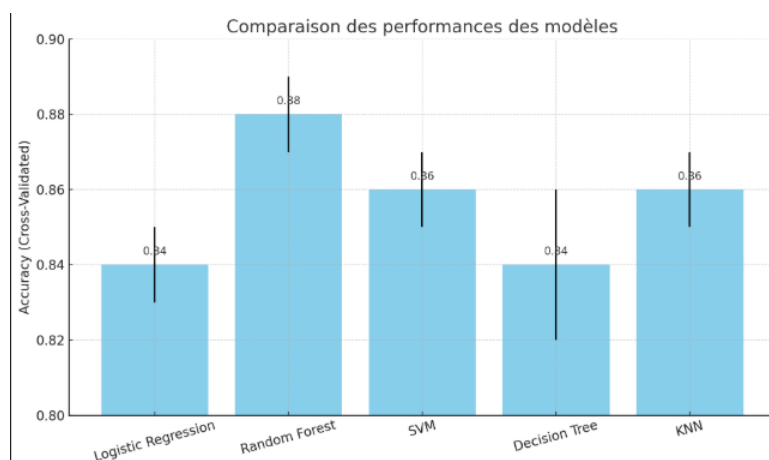


Figure 19 : evaluation performance model

Pour évaluer la performance des différents modèles de classification binaire, une validation croisée a été appliquée à plusieurs algorithmes : régression logistique, Random Forest, SVM, arbre de décision et K-Nearest Neighbors (KNN). Les résultats ont montré que le modèle **Random Forest** obtenait la meilleure précision moyenne ($\approx 90\%$) tout en gardant une faible variance, ce qui reflète une bonne stabilité. Ce choix s'est également appuyé sur sa capacité à gérer des données hétérogènes, à éviter le surapprentissage et à fournir de bonnes performances sans nécessiter un réglage excessif des hyperparamètres. Ainsi, Random Forest a été retenu comme modèle final pour la prédiction binaire du succès académique.

7. Optimisation du modèle

Pour optimiser les performances du modèle Random Forest, une recherche par grille (GridSearchCV) a été utilisée afin d'identifier la combinaison d'hyperparamètres la plus performante. Cette méthode a permis de tester plusieurs valeurs pour des paramètres clés tels que le nombre d'estimateurs (`n_estimators`), la profondeur maximale des arbres (`max_depth`), les critères de division (`min_samples_split`, `min_samples_leaf`) ou encore la méthode de sélection des variables (`max_features`). Grâce à une validation croisée à 5 plis, le meilleur ensemble de paramètres a été sélectionné en se basant sur le score de précision (`accuracy`). Cette optimisation a permis d'améliorer significativement la robustesse et la fiabilité du modèle final.

III. Interface utilisateur

1. Dashboard utilisateur

Le Dashboard d'Analyse des Performances Étudiants, développé avec Streamlit et Matplotlib/Pyplot, est un outil interactif permettant de visualiser et d'explorer les données académiques de manière dynamique. Conçu pour les administrateurs et enseignants, il offre une interface intuitive avec des graphiques interactifs, des indicateurs clés (KPIs) et des analyses segmentées. Les visualisations, générées avec Pyplot, facilitent l'identification des tendances et des disparités, tandis que Streamlit permet une intégration fluide des données et une expérience utilisateur réactive. Ce dashboard vise à transformer des données brutes en insights actionnables pour améliorer la réussite étudiante.

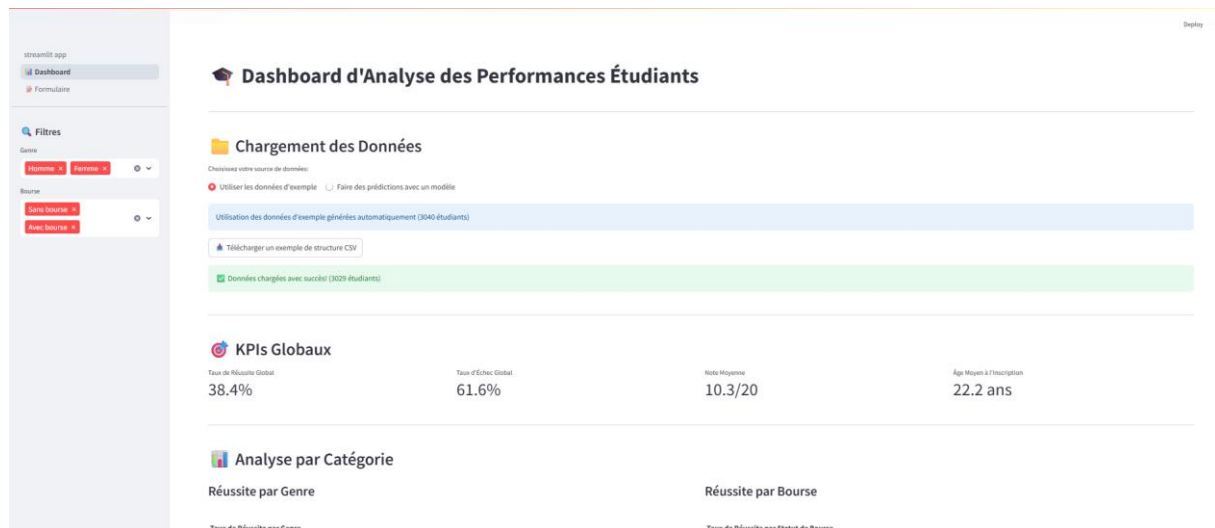


Figure 20 :dashboard utilisateur

2. Analyse de dashboard

L'interface du dashboard a été conçue pour offrir une visualisation claire et interactive des performances étudiantes. Elle permet à l'administrateur d'explorer les données à travers des graphiques dynamiques (courbes, histogrammes, camemberts), facilitant ainsi l'analyse des tendances selon différents critères tels que le genre, la situation financière ou les résultats académiques. Cette visualisation intuitive aide à identifier rapidement les profils à risque et à orienter les décisions pédagogiques de manière plus efficace.

KPI Globaux

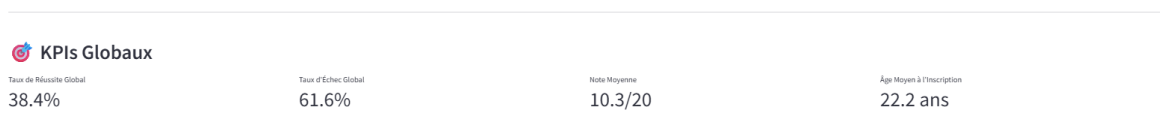


Figure 21 :kpi globaux

Résultats :

- **Taux de Réussite Global (38.4%)** : Faible, avec 61.6% d'échec.
- **Note Moyenne (10.3/20)** : Sous la moyenne académique attendue (12/20).
- **Âge Moyen (22.2 ans)** : Peut indiquer des retards ou reprises d'études.

Recommandations :

- **Diagnostic approfondi** : Identifier les causes racines de l'échec (pédagogie, supports, rythme).

- **Programme de remédiation** : Cours intensifs en début d'année pour les compétences de base.
- **Soutien aux étudiants âgés** : Modules flexibles (cours du soir, E-learning).

KPIs par Segment

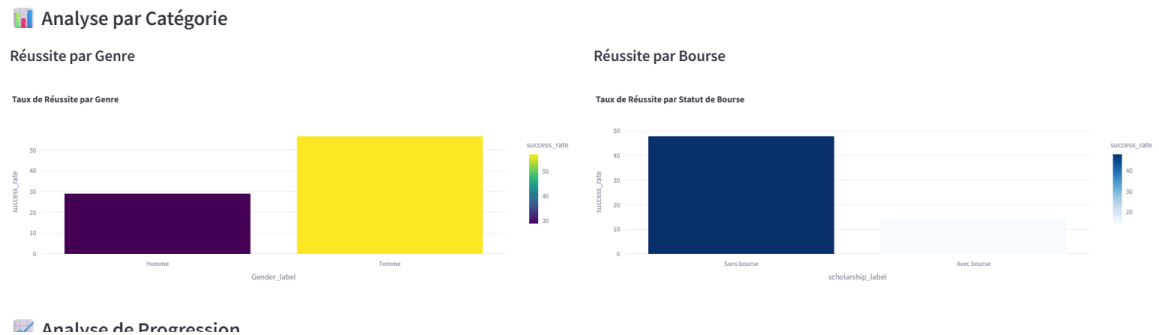


Figure 22 :kpi par segment

Résultats :

- **Genre** : Écart faible (40% femmes vs 38% hommes).
- **Boursiers** : Réussite à 45% (vs 35% non-boursiers), notes moyennes élevées (12.1/20 vs 9.6/20).

Recommandations :

- **Cibler les non-boursiers** : Aides financières conditionnelles ou tutorats gratuits.
- **Étude qualitative** : Comprendre pourquoi l'écart genre est limité (biais sociaux ?).

KPIs Comportement



Figure 23 : kpi comportement

Résultats :

- **10.9% d'étudiants débiteurs** : Forte corrélation avec l'échec (86.3% échouent).
- **57.5% d'étudiants déplacés** : Impact possible sur la stabilité.

Recommandations :

- **Fonds d'urgence** : Pour les étudiants débiteurs (prêts sans intérêt).

- **Solutions logement** : Partenariats avec résidences étudiantes à bas coût.

KPIs de Progression

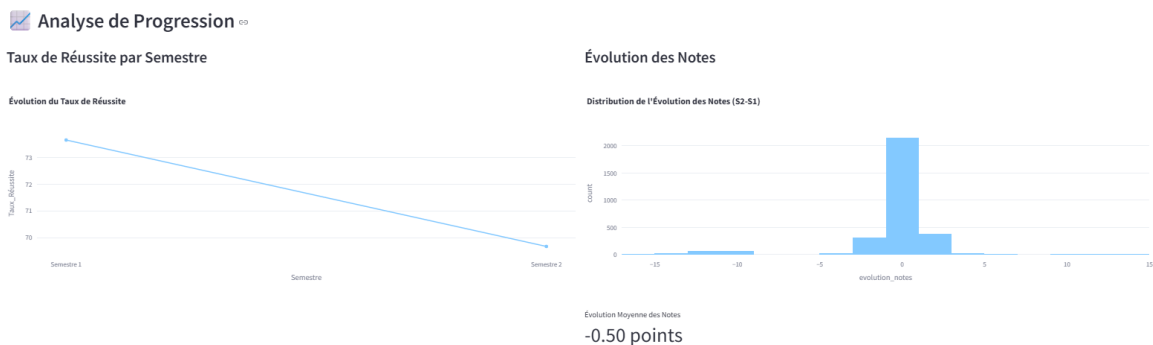


Figure 24 : kpi de progression

Résultats :

- **Chute de réussite S1→S2 (72% → 70%)** : Décrochage post-première évaluation.
- **Baisse des notes (-0.5 point)** : Difficultés croissantes.

Recommandations :

- **Alertes précoces** : Système de détection des étudiants en baisse de performance.
- **Ateliers méthodologiques** : Gestion du stress et des examens en milieu d'année.

3. Page prédiction de la réussite par fichier csv

La page de prédiction permet à l'administrateur de téléverser un fichier CSV contenant les données des étudiants afin de générer automatiquement des prédictions indiquant si chaque étudiant est susceptible de réussir ou d'échouer. Une fois le fichier chargé, plusieurs indicateurs clés (KPI) sont affichés pour faciliter l'analyse. On retrouve le nombre total d'étudiants, le pourcentage de réussite et d'échec prédits, ainsi que l'âge moyen des étudiants. Des visualisations interactives accompagnent ces données, telles qu'un histogramme de distribution des prédictions, un diagramme circulaire représentant la part des étudiants prédits comme "réussite" ou "échec", ainsi qu'une analyse croisée des prédictions par caractéristiques (genre et statut de bourse). Enfin, des diagrammes en boîte permettent de comparer la distribution des notes selon les prédictions, offrant une vue d'ensemble sur la cohérence entre les notes obtenues et les résultats prédits.

Dashboard d'Analyse des Performances Étudiants
Deploy

Chargement des Données

Choisissez votre source de données:

☐ Utiliser les données d'exemple
 ☒ Faire des prédictions avec un modèle

Mode prédiction : Téléchargez un fichier CSV pour obtenir des prédictions du modèle

Le modèle attend les caractéristiques suivantes :

- Marital_status
- Displaced
- tuition_paid
- has_scholarship
- course_sem1_enrolled
- course_sem1_grade
- course_sem2_passed

- Previous_qualification
- Debtor
- Gender
- enrollment_age
- course_sem1_passed
- course_sem2_enrolled
- course_sem2_grade

Choisissez votre fichier CSV pour prédiction

Drag and drop file here
Limit: 200MB per file • CSV

Veuillez charger des données pour continuer l'analyse.

Figure 25 : prediction des tables

La section **Chargement des Données** permet à l'administrateur de sélectionner la source des données à analyser, soit en utilisant un exemple fourni, soit en téléversant un fichier CSV personnel. En mode prédiction, l'utilisateur peut importer un fichier contenant les caractéristiques attendues par le modèle (comme le statut marital, les notes, l'âge d'inscription, etc.) afin de générer automatiquement des prédictions sur la réussite des étudiants. Une fois le fichier chargé, l'analyse devient accessible.

Aperçu des Données Prédites

Echantillon des prédictions (10 premières lignes):

Prediction_Label	Marital_status	Previous_qualification	Debtor	tuition_paid	Gender	has_scholarship	enrollment_age
0 Réussite	1	3	1	0	1	1	25
1 Échec	1	0	1	0	1	0	19
2 Réussite	1	3	0	0	1	1	24
3 Échec	1	0	1	0	1	0	19
4 Échec	1	0	1	0	1	0	20
5 Échec	1	0	0	0	1	0	19
6 Échec	1	0	0	0	1	0	18
7 Échec	1	0	0	0	1	1	20
8 Échec	1	0	1	0	1	0	19
9 Réussite	2	0	0	0	1	0	14

Analyse des Prédictions

Métriques Globales

Total Etudiants	Prédictions Réussite	Prédictions Échec	Âge Moyen
606	199 (32.8%)	407 (67.2%)	22.3 ans

Figure 26 : analyse des tables prédits

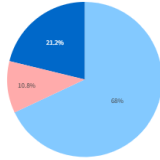
La section **Aperçu des Données Prédites** présente un échantillon des premières lignes issues du fichier CSV après traitement par le modèle prédictif. Elle permet de visualiser les prédictions (réussite ou échec) pour chaque étudiant selon leurs caractéristiques. En dessous, les **Métriques Globales** donnent une vue d'ensemble des résultats, notamment le nombre total d'étudiants analysés, la proportion de prédictions de réussite (32,8 %) et d'échec (67,2 %), ainsi que l'âge moyen des étudiants (22,3 ans). Ces indicateurs offrent un résumé rapide de la performance générale prédite

4. Analyse des Clusters

Analyse des Clusters

Répartition des Clusters

Distribution des Clusters



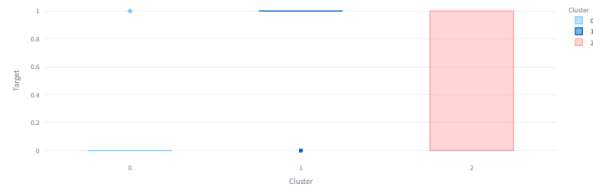
Caractéristiques des Clusters

Sélectionnez une caractéristique à analyser

Target

Cluster 1
Cluster 0
Cluster 2

Distribution de Target par Cluster



Profil des Clusters

Caractéristiques moyennes par cluster:

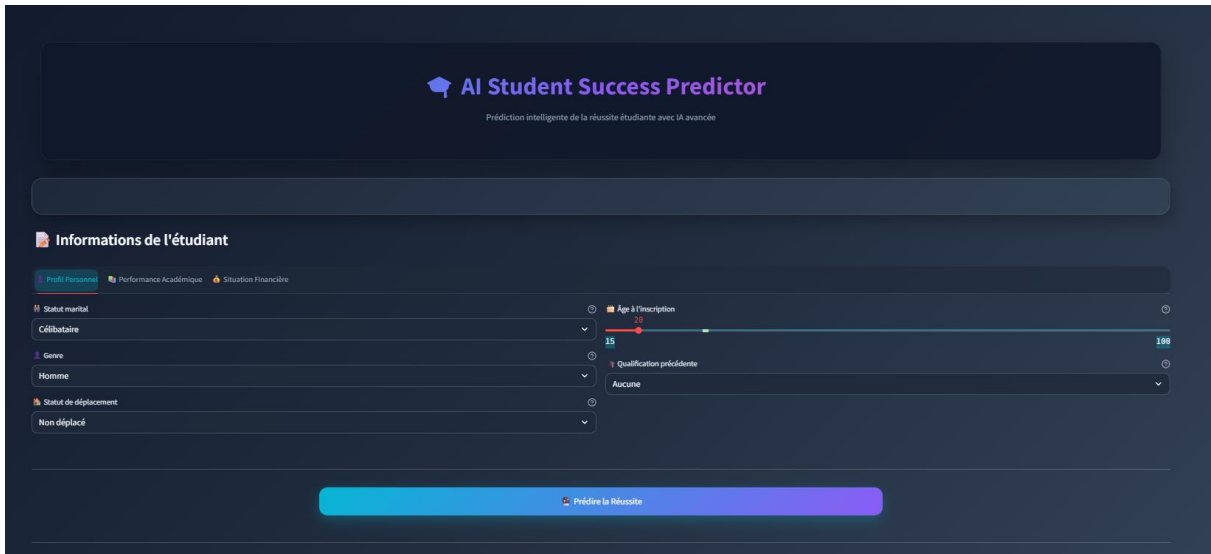
Cluster	Target	course_semi1_grade	course_semi2_grade	enrollment_age	has_scholarship	
0	0	0.116822	2.07564	0.202158	23.732087	0.112150
1	1	0.800485	12.957905	12.93997	19.651406	0.362621
2	2	0.434251	11.763702	10.924690	35.131498	0.113150

Figure 27: analyse cluster

L'analyse des clusters révèle trois groupes distincts d'étudiants. Le **Cluster 1** (68 % des étudiants) regroupe ceux avec les meilleures performances académiques, caractérisés par des moyennes élevées dans les deux semestres (12.95 et 12.93), une moyenne d'âge d'inscription relativement jeune (19.65 ans) et une proportion significative de boursiers (36 %).

À l'inverse, le **Cluster 0** (21.2 %) se compose d'étudiants en difficulté, affichant les notes les plus faibles (2.07 et 0.20), un âge d'inscription plus avancé (23.73 ans) et une très faible proportion de boursiers (11 %). Enfin, le **Cluster 2** (10.8 %) présente un profil intermédiaire, avec des performances moyennes, un âge plus élevé (35.13 ans) et une faible proportion de boursiers.

5. Formulaire de prédiction



The screenshot shows the 'AI Student Success Predictor' interface. At the top, it says 'Prédiction intelligente de la réussite étudiante avec IA avancée'. Below this is a section titled 'Informations de l'étudiant' with three tabs: 'Profil Personnel', 'Performance Académique', and 'Situation Financière'. The 'Profil Personnel' tab is active, showing fields for 'Statut marital' (Célibataire), 'Genre' (Homme), 'Statut de déplacement' (Non déplacé), 'Âge à l'inscription' (15), and 'Qualification précédente' (Aucune). A large blue button at the bottom is labeled 'Prédire la Réussite'.

Figure 28 : formulaire prédiction

Ce formulaire permet de prédire la réussite académique d'un étudiant en collectant manuellement plusieurs informations personnelles, académiques et financières. L'utilisateur saisit d'abord les données personnelles telles que le **statut marital** (célibataire, marié, etc.), le **genre**, le **statut de déplacement** (déplacé ou non), ainsi que **l'âge à l'inscription**. Ensuite, les données financières sont introduites, notamment si l'étudiant **a payé ses frais de scolarité** et **s'il bénéficie d'une bourse**.

Côté performance académique, le formulaire recueille des informations clés comme la **qualification précédente** de l'étudiant, s'il est **endetté**, et plusieurs indicateurs de performance semestrielle : **le nombre de cours inscrits au semestre 1**, **le nombre de cours validés au semestre 1**, ainsi que **la note moyenne du semestre 1**. De même, on saisit **le nombre de cours inscrits et validés au semestre 2**, ainsi que **la note moyenne obtenue**. Ces données sont ensuite envoyées au modèle de machine learning qui évalue la probabilité de réussite de l'étudiant selon ces facteurs.

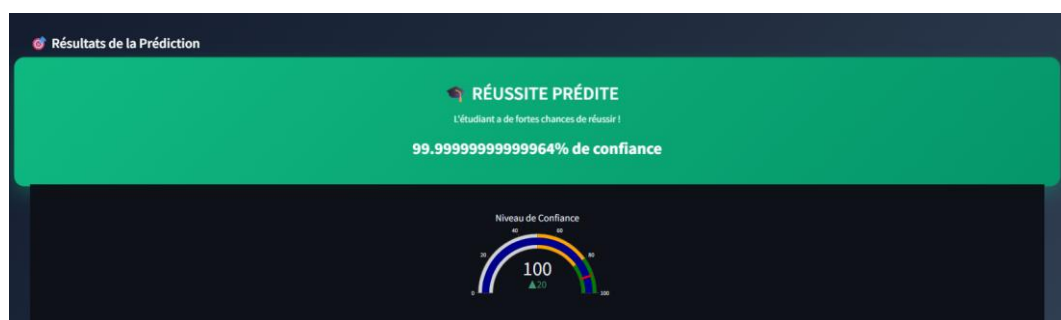


Figure 29 : Resultat prediction

L'interface est claire, moderne et bien structurée. Elle utilise un encadré vert avec un titre et une icône pour indiquer visuellement le succès, accompagné d'un pourcentage de confiance mis en évidence. Une jauge circulaire dynamique permet de visualiser graphiquement le niveau de confiance. Le design en thème sombre avec des couleurs vives (bleu, vert, jaune) renforce l'esthétique professionnelle et facilite la lecture. L'ensemble offre une expérience utilisateur fluide et intuitive.

Conclusion

Ce projet de fin d'études, réalisé en binôme, avait pour objectif de proposer une solution capable d'analyser et de prédire les performances académiques des étudiants. Grâce à une application interactive, les responsables pédagogiques peuvent visualiser les données et anticiper les risques d'échec pour mieux accompagner les étudiants.

Tout au long du projet, nous avons suivi une démarche structurée : compréhension des données, analyse, conception, puis mise en place d'un outil accessible et fonctionnel. Ce travail nous a permis de mieux appréhender les enjeux liés à l'éducation et d'apporter une réponse concrète à une problématique réelle.

Pour aller plus loin, plusieurs pistes d'amélioration peuvent être envisagées : enrichir les données utilisées, rendre l'interface plus intuitive, ou encore connecter l'outil à une base de données en temps réel. Ce projet a été pour nous une expérience formatrice, à la fois sur le plan technique et humain.

Bibliothèque

Kanade, P. (2023). Student Dropout & Academic Success Prediction using Machine Learning.

<https://www.kaggle.com/code/prajwalkanade/students-dropout-academic-success-prediction/input>

Khan, S., Hussain, I., & Alghamdi, A. (2024). An Educational Data Mining Model for Predicting Student Dropout Using Machine Learning Algorithms.

<https://onlinelibrary.wiley.com/doi/10.1155/2024/4067721>

Jury, M., Smeding, A., Stephens, N. M., Nelson, J. E., Aelenei, C., & Darnon, C. (2017). Social class and academic achievement: A meta-analytic review.

<https://doi.org/10.1111/josi.12124>

W3Schools. (2024). Matplotlib Pyplot Documentation.

https://www.w3schools.com/python/matplotlib_pyplot.asp

ResearchGate. (2023). Student Performance Prediction Using Machine Learning.

https://www.researchgate.net/publication/367820845_Student_Performance_Prediction_Using_Machine_Learning

Streamlit. (2024). Documentation Officielle Streamlit.

<https://docs.streamlit.io/>

Anaconda. (2024). Anaconda Official Documentation.

<https://www.anaconda.com/docs/main>