



Université de la Manouba  
École Nationale des Sciences de l'Informatique



## RAPPORT DU PROJET DE CONCEPTION ET DE DÉVELOPPEMENT

### Sujet : Système de Recommandation pour les Universités et les Programmes

*Auteurs :*

Mr. Ghassene TANABENE      Mr. Idriss BEN HMIDA

*Organisme :* Edutest

*Encadré par :* Mr. Makrem Bekalti

*Supervisé par :* Dr. Ines Alaya

*Adresse :* Rue Omar Kaddeh, Cite Montplaisir, R5FW+45  
TUNIS TUNISIA.



Année Universitaire :2019/2020

## ملخص

---

يلخص هذا التقرير العمل المنجز كجزء من مشروع التصميم والتطوير لطلاب السنة الثانية من المدرسة الوطنية لعلوم الإعلامية. يتم تنفيذ هذا المشروع مع شركة يدتست الناشئة وهو يتألف من إنشاء تطبيق ويب هدفه الرئيسي هو التوصية بمنح أجنبية لمستخدمي منصة يدتست بناءً على تفضيلاتهم الشخصية.

---

## Résumé

---

Ce rapport synthétise le travail réalisé dans le cadre du « Projet de Conception et de Développement » pour les étudiants de la deuxième année de l'Ecole Nationale des Sciences de l'Informatique. Ce projet est réalisé avec la startup Edutest. Il consiste à créer une application web dont le but principal est de recommander des bourses étrangères pour les utilisateurs de la plate-forme « Edutest » en se basant sur leurs préférences personnelles.

**Mots clés:** Django, Apprentissage Automatique, grattage du Web, NLP

---

## Abstract

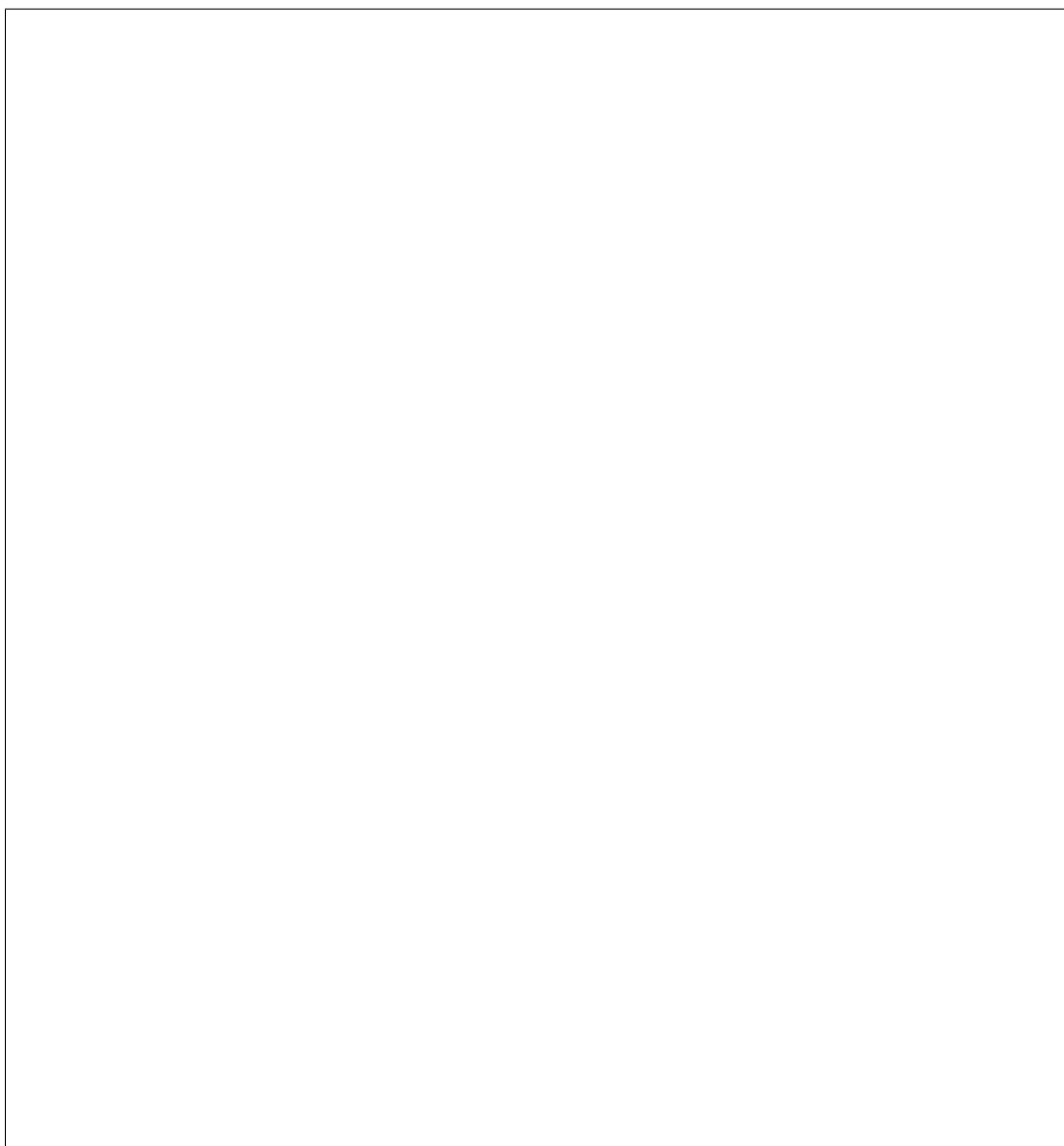
---

This report summarizes the work carried out as part of the « Design and Development Project » for the students of the second year of the National School of Computer Science. This project is carried out with the startup Edutest. It consists in creating a web application whose main goal is to recommend foreign scholarships for users of the platform « Edutest » based on their personal preferences.

**Key words:** Django, Machine Learning, Web Scraping, NLP

---

## Appréciations et signature de l'encadrant

A large, empty rectangular box with a thin black border, intended for the appraiser's comments and signature.

# Remerciements

Nous profitons par le biais de ce rapport, pour exprimer nos vifs remerciements et notre profonde gratitude à toute personne ayant contribué de près ou de loin à la réalisation de cet humble travail.

Nous tenons à remercier les membres du Jury qui nous ont fait l'honneur en acceptant de juger ce travail.

Particulièrement, nous exprimons nos profonds remerciements à Mme. Ines Alaya de nous avoir encadrés durant toute la période d'élaboration de notre projet. Sa disponibilité, sa confiance, et ses conseils nous ont beaucoup aidés pour atteindre les objectifs du projet dans les délais convenus.

Nous remercions également la startup Edutest pour cette expérience enrichissante et en particulier Mr. Makrem Bekalti pour son suivi, son soutien et ses orientations qui ont mené à bien notre travail.

Enfin, nous exprimons nos sentiments de gratitude envers nos familles et nos proches pour leur soutien moral et leur sollicitude.

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Présentation générale et étude de l'existant</b>	<b>3</b>
1.1 Présentation générale du projet . . . . .	3
1.1.1 Cadre du projet . . . . .	3
1.1.2 Présentation de l'organisme d'accueil . . . . .	3
1.2 Étude de la solution . . . . .	4
1.2.1 Présentation des solutions existantes . . . . .	4
1.2.2 Bilan des solutions . . . . .	6
1.3 Solution proposée . . . . .	6
1.4 Conclusion . . . . .	7
<b>2 État de l'art</b>	<b>8</b>
2.1 Apprentissage automatique (Machine Learning) . . . . .	8
2.1.1 Définition . . . . .	8
2.1.2 Apprentissage non-supervisé . . . . .	8
2.2 Processus d'apprentissage automatique . . . . .	9
2.2.1 Collecte de données . . . . .	10
2.2.2 Nettoyage des données . . . . .	10
2.2.3 Exploration de données . . . . .	11
2.2.4 Construction du modèle . . . . .	11
2.2.5 Évaluation de modèle . . . . .	14
2.2.6 Réglage des paramètres . . . . .	14
<b>3 Analyse et spécifications des besoins</b>	<b>15</b>
3.1 Spécification des besoins . . . . .	15
3.1.1 Identification des acteurs . . . . .	15
3.1.2 Besoins fonctionnels . . . . .	16
3.1.3 Besoins non fonctionnels . . . . .	17
3.2 Modélisation . . . . .	17
3.2.1 Langage de Modélisation Unifié (UML) . . . . .	17

---

3.2.2	Diagramme de cas d'utilisation . . . . .	17
3.2.3	Diagrammes de séquence . . . . .	18
<b>4</b>	<b>Conception de la solution</b>	<b>23</b>
4.1	Conception générale . . . . .	23
4.1.1	Conception de l'architecture physique . . . . .	23
4.1.2	Conception de l'architecture logique . . . . .	25
4.2	Conception détaillée . . . . .	26
4.2.1	Diagramme d'activité . . . . .	26
4.2.2	Diagrammes de séquence . . . . .	28
<b>5</b>	<b>Réalisation</b>	<b>32</b>
5.1	Environnement de développement . . . . .	32
5.1.1	Environnement matériel . . . . .	32
5.1.2	Environnement logiciel . . . . .	32
5.2	Choix Techniques . . . . .	33
5.2.1	Langages de Programmation . . . . .	33
5.2.2	Cadres de développement (Frameworks) . . . . .	33
5.2.3	Bibliothèques . . . . .	34
5.2.4	Construction du modèle de prédiction . . . . .	34
5.3	Travail achevé . . . . .	38
5.3.1	Application Web . . . . .	38
5.3.2	Interface administrateur . . . . .	41
	<b>Conclusion et perspectives</b>	<b>44</b>
	<b>Bibliographie</b>	<b>46</b>
	<b>Netographie</b>	<b>47</b>

# Table des figures

1.1	Page d'accueil de University Recommendation System . . . . .	4
1.2	Page d'accueil du site Scholarship Portal . . . . .	5
1.3	Page d'accueil du site Niche . . . . .	5
1.4	Tableau résumant les solutions existantes . . . . .	6
2.1	Processus d'apprentissage automatique . . . . .	9
2.2	Pipeline du traitement automatique du langage naturel . . . . .	10
2.3	Principe de regroupement K-moyennes . . . . .	12
2.4	Principe de regroupement « DBSCAN » . . . . .	13
2.5	Principe de regroupement spectral . . . . .	13
3.1	Diagramme de cas d'utilisation . . . . .	18
3.2	Diagramme de séquence l'application web côté administrateur . . . . .	19
3.3	Diagramme de séquence l'application web coté utilisateur . . . . .	20
3.4	Diagramme de séquence "Inscription de l'utilisateur" . . . . .	21
3.5	Diagramme de séquence "Authentification de l'utilisateur" . . . . .	22
4.1	Diagramme de déploiement du système . . . . .	24
4.2	Les niveaux de l'architecture 3-tiers . . . . .	24
4.3	Diagramme d'architecture MVT . . . . .	25
4.4	Diagramme d'activité du modèle de prédiction . . . . .	27
4.5	Diagramme d'activité de l'algorithme de regroupement k-moyennes . . . . .	28
4.6	Diagramme de séquence l'application web côté utilisateur . . . . .	29
4.7	Diagramme de séquence "Inscription de l'utilisateur" . . . . .	30
4.8	Diagramme de séquence "Authentification de l'utilisateur" . . . . .	30
5.1	Grattage Web . . . . .	35
5.2	Nettoyage du texte . . . . .	35
5.3	TF-IDF Vectoriseur . . . . .	36
5.4	Visualisation des données . . . . .	36
5.5	Le coefficient de Silhouette . . . . .	37
5.6	Evaluation du modèle . . . . .	38

---

5.7	Page d'accueil . . . . .	38
5.8	Les interfaces d'authentification et d'inscription de l'utilisateur . . . . .	39
5.9	Formulaire de spécification des préférences . . . . .	39
5.10	Résultat de la recommandation . . . . .	40
5.11	Bourse en détails . . . . .	40
5.12	Historique des recherches . . . . .	41
5.13	Interface Authentification de l'administrateur . . . . .	41
5.14	Interface « Django Administration » . . . . .	42
5.15	Interface de la table « Users » des utilisateurs . . . . .	42
5.16	Interface de la table « Scholarships » des bourses . . . . .	43
5.17	Formulaire pour ajouter une nouvelle bourse . . . . .	43



# Liste des tableaux

5.1	Machines . . . . .	32
-----	--------------------	----

# Liste des sigles et acronymes

<b>UML</b>	<i>Unified Modelling Language</i>
<b>Sklearn</b>	<i>Scikit-learn</i>
<b>NLP</b>	<i>Natural Language Processing</i>
<b>NLTK</b>	<i>Natural Language Toolkit</i>
<b>KNN</b>	<i>K Nearest Neighbor</i>
<b>TF-IDF</b>	<i>Term Frequency - Inverse Document Frequency</i>
<b>DBSCAN</b>	<i>Density-Based Spatial Clustering of Applications with Noise</i>
<b>CSV</b>	<i>Comma Separated Value</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>MVC</b>	<i>Model View Controller</i>
<b>MVT</b>	<i>Model View Template</i>
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>CRUD</b>	<i>Create-Read-Update-Delete</i>

# Introduction

De nos jours, nombreux sont ceux qui ont exprimé leur satisfaction suite au recours à l'Intelligence Artificielle dans la résolution des différents problèmes. En particulier, l'apprentissage automatique ou « Machine Learning » est capable de donner dans les brefs délais une solution précise et efficace pour les difficultés touchant plusieurs aspects de notre vie. Parmi ces difficultés, nous nous intéressons aux problèmes concernant l'éducation vue son importance majeure, et plus précisément à la recherche d'une bourse d'études à l'étranger.

Plusieurs jeunes étudiants ont tendance à vouloir relever de nouveaux défis en partant aux Etats-Unis, à l'Allemagne, en France ... pour apprivoiser d'autres méthodes de travail et poursuivre leurs cursus dans des grandes universités internationales. Toutefois, passer du rêve à la réalité est plus compliqué qu'il n'y paraît où le mauvais choix de l'université peut mettre une fin précoce de carrière à ses étudiants.

C'est dans ce contexte que s'inscrit notre projet de développement et de conception intitulé « Système de Recommandation des Universités et des Programmes ». En effet, il s'agit de réaliser une application web permettant de recommander des bourses d'études aux étudiants de la plateforme Edutest offrant une solution intelligente au problème présenté. Notre système de recommandation effectuera la recherche cyclique des opportunités d'aide financière sur internet, la recherche des universités correspondantes et leurs programmes d'études, et enfin la recommandation qui va dépendre nécessairement des préférences saisies de la part de l'utilisateur. Cette application permet à l'étudiant de savoir tous les détails non seulement sur les bourses d'études, mais aussi sur les universités de ses rêves tels que leurs rangs mondiaux, leurs localisations, etc.

Le présent rapport comprend cinq chapitres décrivant notre projet de conception et de développement. Le premier chapitre offre une présentation du contexte général du projet et une étude de l'existant. Le deuxième fournit une étude préliminaire qui explique des notions fondamentales et nécessaires pour l'élaboration de notre solution. Quant au troisième, intitulé « Analyse et spécifications des besoins », nous allons le consacrer pour la description des exigences fonctionnelles et non fonctionnelles de notre système. Dans le quatrième chapitre, une étude détaillée du projet est effectuée à travers la modélisation

conceptuelle de la solution proposée. Enfin, le dernier chapitre est dédié à la mise en oeuvre et l'évaluation de notre système de recommandation. Ainsi, nous clôturons ce rapport avec une conclusion générale pour synthétiser le travail effectué et les perspectives futures.

# Chapitre 1

## Présentation générale et étude de l'existant

Dans ce chapitre introductif, nous allons mettre le projet dans son contexte général en premier lieu tout en présentant brièvement le cadre du projet ainsi que l'organisme d'accueil. En deuxième lieu, nous effectuons une étude de l'existant en soulignant les insuffisances et nous arrivons finalement à exposer la solution que nous proposons pour le problème posé.

### 1.1 Présentation générale du projet

Dans cette section, nous commençons par décrire le cadre du projet et l'organisme d'accueil en précisant son secteur d'activité.

#### 1.1.1 Cadre du projet

Le présent projet consiste à développer un système de recommandation des universités offrant des bourses d'études ainsi que leurs programmes. Il est inscrit dans le cadre du projet de conception et développement pour les étudiants de la deuxième année de l'Ecole Nationale des Sciences de l'Informatique et également réalisé avec la startup Edutest.

#### 1.1.2 Présentation de l'organisme d'accueil

Edutest [URL1] est une startup fondée en 2018 ayant pour objectif d'identifier, habiliter, et connecter les élèves et les étudiants tunisiens aux opportunités, aux universités et surtout aux bourses pour continuer leurs études à l'étranger. A travers « EDUTEST » : la première plateforme d'études en Anglais en ligne en Tunisie, cette startup aide les étudiants à poursuivre leurs cursus essentiellement aux Etats Unis en leur assurant une

bonne préparation aux examens d'admission, tel que le TOEFL, le SAT, le GRE ou encore le GMAT, et également des cours d'Anglais.

## 1.2 Étude de la solution

Pour répondre à notre problématique, certains projets ont proposé des solutions différentes en vue de faire une recommandation des universités. Dans cette section, nous présenterons quelques exemples de ces projets.

### 1.2.1 Présentation des solutions existantes

Dans cette sous-section, nous essaierons de mettre en relief les caractéristiques et les limites de trois exemples de solutions qui traitent notre problème.

- **University Recommendation System**

Le site [URL2] présente une recommandation des universités pour les étudiants en se basant sur le filtrage collaboratif. En réalité, cette technique utilise les notes attribuées par les utilisateurs pour réaliser la recommandation.

Nous présentons dans la figure 1.1 ci-dessous la page d'accueil de ce site.

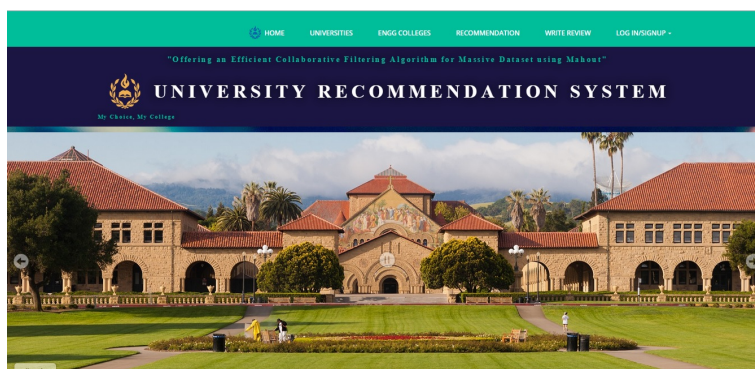


FIGURE 1.1 – Page d'accueil de University Recommendation System

**Limites :** Ce projet ne présente pas de recommandation pour les bourses d'études et n'expose pas les informations relatives aux universités.

- **Scholarship Portal**

Le site [URL3] est considéré parmi les sites les plus populaires qui facilitent la recherche des bourses d'études partout dans le monde. Ce site affichera une description détaillée de la bourse souhaitée de la part de l'utilisateur.

La figure 1.2 représente la page d'accueil du site Scholarship Portal.

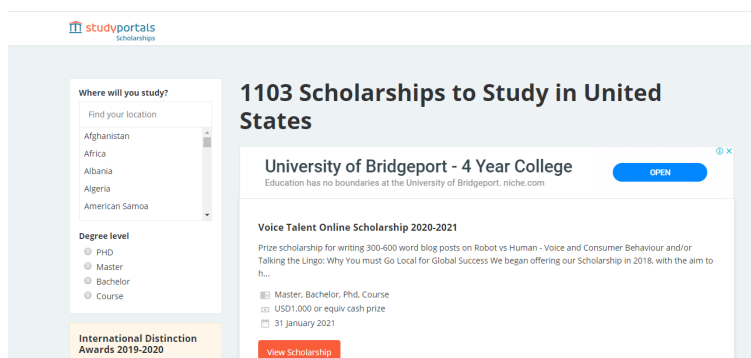


FIGURE 1.2 – Page d'accueil du site Scholarship Portal

**Limites :** Ce projet ne présente pas les informations relatives aux universités telles que les programmes d'études. Il recommande également des bourses qui ont déjà expiré.

- **Niche**

Le site [URL4] est utilisé pour la recherche des bourses d'études aux Etats-Unis. Grâce à son interface facile à manipuler, l'utilisateur trouvera les propositions des bourses d'études accompagnées de leurs descriptions détaillées. La page d'accueil du site Niche est représentée par la figure 1.3.

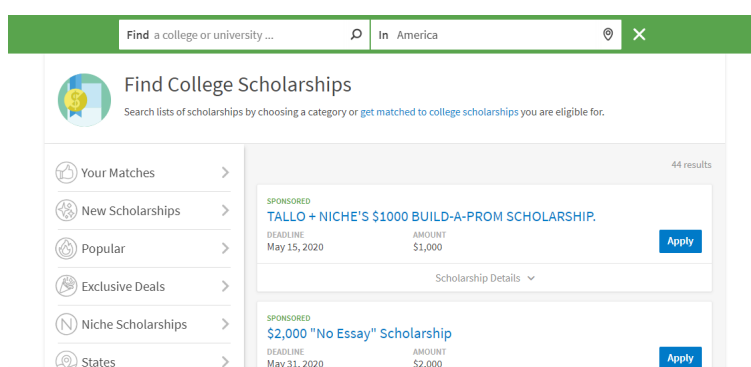


FIGURE 1.3 – Page d'accueil du site Niche

**Limites :** La limite principale de ce site se manifeste dans le nombre limité de données où nous trouvons uniquement une dizaine de bourses d'études. De même, ce site ne présente pas les informations relatives aux universités et ne demande pas suffisamment de préférences pour effectuer la bonne recommandation.

### 1.2.2 Bilan des solutions

Afin d'effectuer l'évaluation des solutions existantes décrites précédemment, nous avons sélectionné les critères suivants : la recommandation des bourses, la recommandation des universités, la recommandation du niveau d'étude, l'affichage trié par rang de l'université, la description détaillée des résultats et la recommandation du programme d'études.

Le tableau 1.4 ci-dessous récapitule le bilan de cette étude des cas existants :

Site	Critères					
	Recommandation des bourses	Recommandation des universités	Recommandation du niveau	Affichage trié par rang de l'université	Description détaillée des résultats	Recommandation du programme
<b>University Recommendation System</b>	✗	✓	✗	✗	✗	✗
<b>Scholarship Portal</b> ( <a href="http://www.scholarshipportal.com">www.scholarshipportal.com</a> )	✓	✗	✓	✗	✓	✗
<b>Niche</b> ( <a href="http://www.niche.com">www.niche.com</a> )	✓	✓	✗	✗	✓	✗

FIGURE 1.4 – Tableau résumant les solutions existantes

En guise de conclusion, le tableau récapitulatif précédent nous a montré les faiblesses des solutions existant face aux critères qui répondent aux besoins de l'étudiant. Ceci nous a menés à proposer une nouvelle solution plus pertinente qui sera décrite dans la partie suivante.

## 1.3 Solution proposée

La solution que nous allons concevoir consiste à créer une application web dont le but principal est de faire la recommandation des bourses d'études à l'étranger. Cette application répondra à tous les critères et besoins cités précédemment en s'appuyant sur les préférences des étudiants. Pour ce faire, l'utilisateur doit s'authentifier initialement puis indiquer les informations nécessaires à la prédiction des bourses d'études ainsi que les universités qu'il souhaite. A une précision bien déterminée choisie lors de la construction du modèle d'apprentissage automatique, notre système affichera une liste bien ordonnée de bourses d'études selon les préférences de l'étudiant, les classements mondiaux des universités, les dates limites d'inscription etc...

Par conséquent, l'utilisateur trouvera tous les détails concernant les bourses et les universités et aura la possibilité de consulter les résultats de ses recherches précédentes. En prenant en considération les changements quotidiens des opportunités et même les rangs des universités, notre système manipule souvent des données qui sont mises à jour automatiquement grâce à des algorithmes de grattage de web. De cette façon, notre objectif sera bien atteint.



## **1.4 Conclusion**

Dans ce chapitre, nous avons fait une présentation générale du projet et de l'organisme d'accueil. Puis, nous avons réalisé l'étude de l'existant qui intègre des critiques afin d'arriver à décrire notre solution proposée.

De cette manière, nous passons à l'état de l'art dans le chapitre suivant.

# Chapitre 2

## État de l’art

Dans ce deuxième chapitre, nous allons établir l’étude théorique des techniques fondamentaux adoptés pour mener à bien notre projet. En fait, ces informations vont affecter par la suite la prise de décision concernant le choix de la solution la plus adéquate à notre problème. Nous expliquerons ainsi les notions de bases en commençant par la définition de l’apprentissage automatique dont nous allons mettre en relief sa technique non-supervisée et en arrivant finalement à clarifier toute le processus de la préparation des données à utiliser et obtenir notre modèle de classification et atteindre notre objectif.

### 2.1 Apprentissage automatique (Machine Learning)

Dans cette section, nous allons expliquer la notion d’apprentissage automatique et également son approche non-supervisé.

#### 2.1.1 Définition

L’apprentissage automatique ou « Machine Learning » en anglais, fait partie de l’une des approches de l’intelligence artificielle. C’est une technologie qui offre aux machines la capacité d’apprendre à partir des données sans être explicitement programmés. On distingue plusieurs types d’apprentissage automatique qui sont l’apprentissage supervisé, l’apprentissage non-supervisé, l’apprentissage semi-supervisé et l’apprentissage par renforcement.

#### 2.1.2 Apprentissage non-supervisé

##### Définition

L’apprentissage non supervisé [URL5] est une technique d’apprentissage automatique, qui ne demande pas de supervision du modèle. En effet, ce dernier apprend tout seul et découvre les relations cachées dans l’ensemble des données par regroupement sans que les

données ne soient étiquetées ni classifiées. On distingue deux types d'apprentissage non-supervisé : le regroupement et l'association.

### Regroupement de données (Data Clustering)

Ce concept permet de déterminer des groupes homogènes dans les données en fonction de leurs similitudes en se basant principalement sur des critères de proximité. Parmi les méthodes de partitionnement on trouve le partitionnement en K-Moyennes, le regroupement hiérarchique , des algorithmes basés sur la densité telle que DBSCAN etc...

### Association de données

L'association est une technique d'apprentissage non-supervisé qui consiste à découvrir des relations intéressantes entre les attributs d'un grand ensemble de données et établir des associations entre des objets de données.

## 2.2 Processus d'apprentissage automatique

L'apprentissage automatique est un domaine d'étude de l'intelligence artificielle basé sur des approches mathématiques et statistiques. Il comporte plusieurs étapes illustrées dans la figure 2.1.

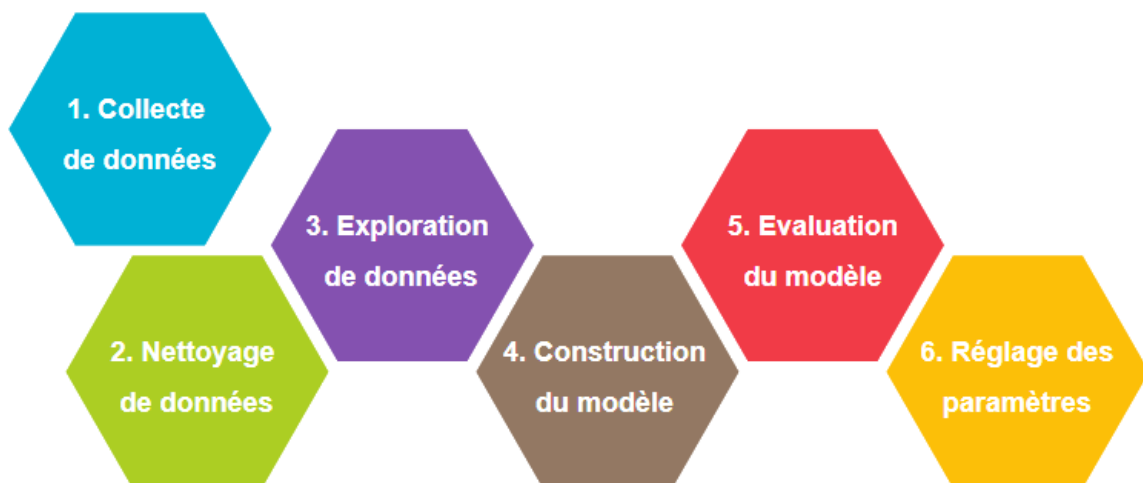


FIGURE 2.1 – Processus d'apprentissage automatique

### 2.2.1 Collecte de données

Cette étape consiste à rassembler les données nécessaires en vue de les analyser, et ce, à travers la recherche dans les jeux de données, ou « datasets » en anglais, et notamment grâce à la technique de grattage web ou « web scraping ».

**Grattage Web (Web Scraping) :** C'est une technique utilisée pour extraire le contenu des sites Web via un script ou un programme.

### 2.2.2 Nettoyage des données

#### Traitement automatique du langage naturel (NLP)

Le traitement naturel du langage, ou « Natural Language Processing (NLP) » en anglais, est une technique qui permet à la machine de lire, déchiffrer et comprendre le langage humain d'une manière intelligente et artificielle. C'est la transformation de texte en représentation numérique compréhensible par des modèles de l'apprentissage automatique. La figure 2.2 présente le pipeline du traitement automatique du langage naturel.

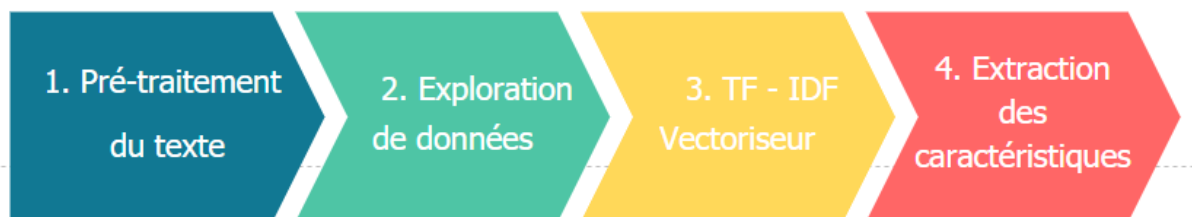


FIGURE 2.2 – Pipeline du traitement automatique du langage naturel

#### Pré-traitement du texte

Cette étape permet de nettoyer les données textuelles et les standardiser afin de rendre son usage plus facile et compréhensible par les algorithmes de l'apprentissage automatique. Le pré-traitement [URL6] comporte plusieurs phases telles que :

- Élimination des espaces blancs
- Suppression des ponctuations et symboles.
- Transformation du texte en minuscule.
- Élimination des mots vides ou « stop words » en anglais.
- **Tokenisation** : c'est la transformation du texte en une série de jetons « tokens » individuels dont chacun représente un mot.
- **Racination** : c'est la réduction du mot dans sa forme « racine ».

- **Reconnaissance d'entités nommées (Named-entity recognition)** : c'est la catégorisation des mots en des entités telles que des organisations, des lieux etc...
- Remplissage des valeurs manquantes des données à l'aide de l'algorithme « KNN ».
- **K plus proches voisins (KNN)** : La méthode des k plus proches voisins, ou « k-nearest neighbors (KNN) » en anglais, fait correspondre un point avec ses k voisins les plus proches dans un espace multidimensionnel. Cet algorithme est utilisé dans cette étape pour approximer la valeur d'un point par les valeurs des points qui en sont les plus proches.
- **Normalisation des données** : C'est la transformation des valeurs numériques en nombres réels compris entre zéro et un, afin de faciliter la manipulation.
- **Vectorisation des données** : C'est la transformation du texte en une représentation vectorielle compréhensible par la machine à l'aide des techniques de sac de mots (Bag of Words) ou dans notre cas le vectoriseur Tf-IdF [URL7].

**TF-IDF vectoriseur** : C'est une méthode de pondération [URL8] utilisée en vue d'évaluer l'importance d'un terme contenu dans un document textuel. Le poids s'obtient en multipliant les deux mesures suivantes :

$$TF\ IDF(t) = TF(t) \times IDF(t)$$

où :

$$\begin{cases} TF(t) = \frac{\text{Nombre d'apparition du terme } t \text{ dans le document } d}{\text{Nombre total de termes dans le document } d} \\ IDF(t) = \log\left(\frac{\text{Nombre total de documents dans le corpus}}{\text{nombre de documents où le terme } t \text{ apparaît}}\right) \end{cases}$$

### 2.2.3 Exploration de données

Après le nettoyage des données rassemblées de différentes sources, le processus d'apprentissage automatique nécessite à ce stade la visualisation des données afin de dégager les relations entre les caractéristiques ou « features » en anglais. Grâce à cette étape, la sélection des caractéristiques se fait en gardant celles qui sont les plus importantes pour construire le modèle de prédiction.

### 2.2.4 Construction du modèle

Après avoir préparé les données sous leur forme numérique et sélectionnée les caractéristiques les plus importantes, l'étape suivante comprend le choix de l'algorithme le plus meilleur pour effectuer la recommandation. En effet, différents algorithmes d'apprentissage automatique seront utilisés et testés pour construire un modèle approprié qui nous permet d'atteindre notre objectif.

### • Algorithme de regroupement K-moyennes

L'algorithme de regroupement K-moyennes ou « K-means Clustering » en anglais [Saint-Cirgue, 2019], est l'algorithme d'apprentissage non supervisé le plus simple et le plus connu. Il s'agit d'une méthode de quantification vectorielle ayant pour objectif de partitionner  $n$  observations en  $k$  groupes « clusters ».

#### Définition mathématique [URL9] :

Étant donné un ensemble de points  $(x_1, x_2, \dots, x_n)$ , on cherche à partitionner les  $n$  points en  $k$  ensembles  $S = S_1, S_2, \dots, S_k$  ( $k \leq n$ ) en minimisant la distance entre les points à l'intérieur de chaque partition :

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

où  $\mu_i$  est le barycentre des points dans  $S_i$ .

La figure 2.3 illustre le principe de fonctionnement de cet algorithme.

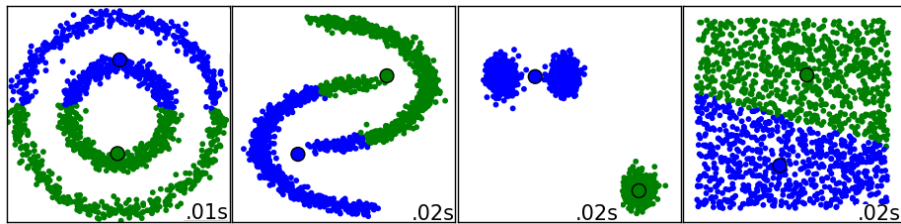


FIGURE 2.3 – Principe de regroupement K-moyennes

### • Algorithme DBSCAN :

L'algorithme de regroupement spatial basé sur la densité des applications avec bruit ou encore « DBSCAN » en anglais est une méthode populaire d'apprentissage non supervisé. Cet algorithme est fondé sur la densité dans la mesure qui s'appuie sur la densité estimée des groupes pour effectuer le partitionnement.

#### Définition mathématique [URL10] :

L'algorithme « DBSCAN » s'appuie sur les notions fondamentales suivantes :

- Le voisinage  $V_\epsilon(p)$  d'un point  $p$  qui est défini par :  $V_\epsilon(p) = \{q \in D \mid d(q, p) < \epsilon\}$  avec  $\epsilon$  une distance et  $d(p, q)$  est la distance entre les points  $p$  et  $q$ .
- Epsilon-voisinage dense : Un  $\epsilon$ -voisinage  $V_\epsilon(p)$  est dit dense si son cardinal vérifie la relation :  $|V_\epsilon(p)| \geq \text{minPts}$ ; avec « MinPts » le nombre minimum de points se trouvant dans la boule de rayon  $\epsilon$  formant un groupe.

La figure 2.4 illustre le principe de fonctionnement de cet algorithme.

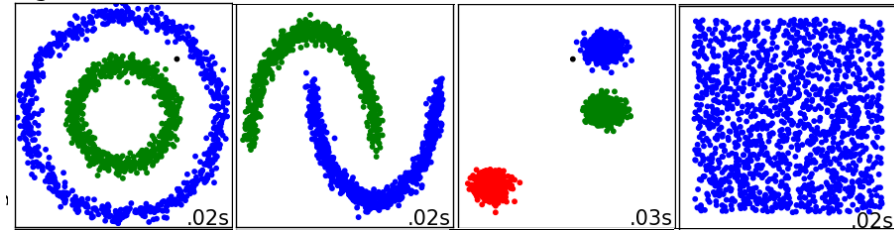


FIGURE 2.4 – Principe de regroupement « DBSCAN »

### • Algorithme de regroupement spectral :

Le partitionnement spectral ou « spectral clustering » en anglais, est une méthode de regroupement basée sur des éléments de la théorie des graphes. Effectivement, les points représentant l'ensemble de données sont interprétés comme des sommets de ce graphe pondéré où cet algorithme utilise le plus souvent les vecteurs propres de la matrice de similarités.

#### Définition mathématique [URL11] :

Le partitionnement se fait en  $K$  groupes travaillant sur la minimisation d'un critère de type « coupe » (coupe simple à  $K=2$ , ou coupe multiple à  $K \geq 2$ ).

Afin de construire la matrice de similarité  $S$  entre les objets, l'algorithme utilise les deux formules suivantes :

- **Noyau Gaussien** :  $S_{ij} = \exp\left(\frac{-d^2(x_i, x_j)}{2\sigma^2}\right)$  avec  $d$  la distance entre les points  $S_i$  et  $S_j$ , et  $\sigma$  un paramètre d'échelle donné par l'utilisateur.
- **Distance cosinus** :  $S = B^T B$  avec  $B$  désignant la matrice de données (constituée de  $N$  objets décrits chacun par  $M$  attributs) normalisée en ligne.

La figure 2.5 illustre le principe de fonctionnement de cet algorithme.

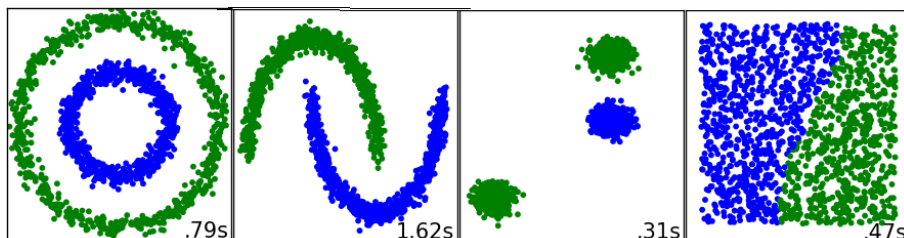


FIGURE 2.5 – Principe de regroupement spectral

### 2.2.5 Évaluation de modèle

La construction du modèle, et plus précisément le choix de l'algorithme le plus adéquat, ne se fait pas d'une manière arbitraire. A ce stade, il s'agit de tester les performances du modèle avec des données invisibles et ensuite à l'aide de différentes techniques pour savoir le degré de fiabilité des résultats et vérifier que le modèle ne fait pas de sur-apprentissage ou un sous-apprentissage. Pour ce faire, la manière la plus connue est de chercher la valeur de la précision du modèle d'apprentissage automatique ou « Accuracy » en anglais. Concernant l'approche non supervisée, il est recommandé d'utiliser des méthodes de calcul de précision basées principalement sur la mesure des distances entre les points comme le coefficient de silhouette.

### 2.2.6 Réglage des paramètres

Arrivant à la fin du processus d'apprentissage automatique, le modèle ne donne pas en général un bon résultat du premier coup. Cependant, la phase de réglage des paramètres du modèle est souvent la solution qui va améliorer la prédiction. Ainsi, cette étape sera répétée plusieurs fois jusqu'à atteindre la précision souhaitée et par suite valider notre modèle.

## Conclusion

A l'issue de ce chapitre, nous avons défini l'apprentissage automatique et notamment son approche non-supervisée. Par ailleurs, nous avons décrit les étapes formant le processus de développement du modèle de prédiction en débutant par la préparation des données et en arrivant finalement à la comparaison des algorithmes et la validation. Cette étude théorique a été élaborée dans le but d'adoucir la transition à la phase d'analyse de notre projet qui sera le sujet du chapitre suivant.



# Chapitre 3

## Analyse et spécifications des besoins

La réalisation d'un projet exige avant tout une bonne compréhension de ses fonctionnalités tout en respectant les normes fixées dans le cahier des charges. Pour ce faire, nous consacrons ce chapitre en vue d'identifier les acteurs de notre système, analyser ses besoins fonctionnels et non fonctionnels et enfin modéliser ses caractéristiques à l'aide des diagrammes du langage UML tels que le diagramme de cas d'utilisation et celui de séquence.

### 3.1 Spécification des besoins

Dans cette section, nous précisons les acteurs de notre système. D'une autre part, nous détaillons les exigences fonctionnelles et celles non fonctionnelles relatives à notre solution.

#### 3.1.1 Identification des acteurs

Un acteur peut être défini comme une entité externe qui interagit avec le système. Dans notre cas, nous avons identifié les deux acteurs suivants :

- **L'utilisateur de l'application :**

C'est l'étudiant de la plate-forme « Edutest » qui cherchera une aide financière à l'étranger. Il sera le bénéficiaire principal des services fournis par notre système.

- **L'administrateur de l'application :**

C'est la personne qui va contrôler la plate-forme « Edutest » ayant le droit d'ajouter, modifier et supprimer les données stockées dans la base de données. Celles-ci sont relatives aux bourses d'études et des universités et également aux utilisateurs de la plate-forme.

### 3.1.2 Besoins fonctionnels

Les exigences fonctionnelles décrivent les fonctionnalités de notre solution, formée par le système de prédiction et l'application web, et pareillement les services pour chacun de ses acteurs résumés comme suit :

- **Pour l'utilisateur de l'application**

- Créer compte : L'application permet à son utilisateur de créer son compte personnel pour qu'il puisse se bénéficier de la recommandation nécessitant son authentification.
- Chercher des bourses d'études : L'application donne à l'étudiant la possibilité d'introduire ses préférences pour prédire les aides financières qu'il souhaite chercher.
- Consulter la recommandation : Une fois le système présente la prévision pour les aides financières, l'utilisateur a toujours l'accès pour consulter son résultat en parcourant les informations en sortie et éventuellement celui des recommandations précédentes.

- **Pour l'administrateur de l'application**

- Gérer les jeux de données des bourses et universités : L'admin possède les droits d'écriture, modification et suppression des données qui sont en entrée de notre système de prédiction, et ce, en insérant des nouvelles informations dans la base de données ou en éditant celles additionnées automatiquement à la base de données de l'application à l'aide des algorithmes de grattage web.
- Gérer les utilisateurs de l'application : L'administrateur peut ajouter, modifier et supprimer des utilisateurs de la base de données de notre application web.

- **Autres besoins relatifs à notre système de prédiction**

- Mettre à jour les données : Le système de prédiction se charge d'apporter les nouvelles opportunités d'aides financières et les ajouter à la base de données qui est gérée par l'administrateur. Il se charge également de supprimer automatiquement les opportunités dont les dates limites sont dépassées.
- Réaliser la recommandation : Le système de prévision doit effectuer immédiatement la recommandation des bourses d'études avec le minimum possible d'erreur. Pour ce faire, il doit analyser les choix saisis par l'utilisateur en se basant sur les algorithmes d'apprentissage automatique.

### 3.1.3 Besoins non fonctionnels

Les exigences non fonctionnelles forment les besoins qui caractérisent le système et améliore la qualité de son fonctionnement. Parmi les contraintes additionnelles dont nous devons tenir compte lors de la réalisation du projet, nous pouvons citer :

- Facilité d'utilisation : L'application doit être facile à utiliser ayant une interface pratique et présentable.
- Mise à jour des jeux des données : Le système doit faire l'actualisation automatique des bourses d'études à recommander d'une façon journalière.
- Performance : Le système doit répondre aux requêtes des utilisateurs dans un délai court et raisonnable. Il doit gérer les erreurs que l'utilisateur peut commettre.
- Confidentialité : L'application doit assurer la confidentialité des données personnelles des utilisateurs.
- Maintenabilité : Le code source de notre application doit être bien documenté et facile à comprendre afin de garantir sa maintenance et son évolution selon les besoins de son client.

## 3.2 Modélisation

Tout au long de cette section, nous allons présenter la modélisation basée principalement sur le diagramme de cas d'utilisation et les diagrammes de séquence. Ces diagrammes définis en langage UML décrivent la dépendance et les relations entre notre système et ses acteurs afin d'expliquer davantage son aspect fonctionnel.

### 3.2.1 Langage de Modélisation Unifié (UML)

Le Langage de Modélisation Unifié (UML) est un langage visuel commun et riche sémantiquement et syntaxiquement. Il est composé d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter.

### 3.2.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation donne une vision globale des fonctionnalités du système. Il représente les relations entre les acteurs et les différents cas d'utilisation.

Dans la figure 3.1, nous représentons le diagramme de cas d'utilisation globale de notre système.

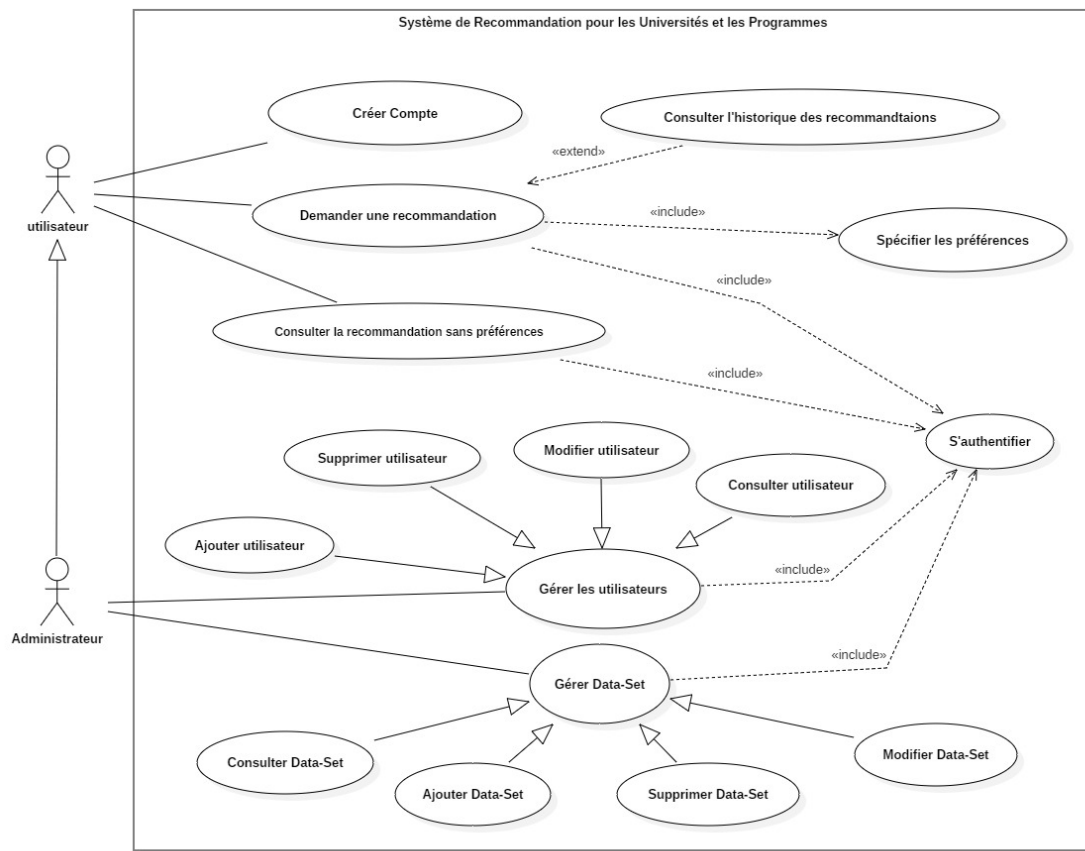


FIGURE 3.1 – Diagramme de cas d'utilisation

D'une part, l'utilisateur doit créer un compte pour qu'il puisse accéder à la plate-forme. Ensuite, il aura deux choix : Soit il spécifie ses préférences pour se bénéficier d'une recommandation qui répond à ses besoins, soit il accède à la recommandation par défaut qui sera commune pour tous les utilisateurs en se basant sur le classement, les valeurs des bourses, les dates limites etc.. Enfin, l'utilisateur peut consulter plus de détails sur les bourses d'études données en résultat ainsi que l'historique de ses anciennes recherches. D'autre part, l'administrateur a la possibilité de gérer la base de données des utilisateurs et celle des aides financières, et ce, à travers les opérations « C.R.U.D ».

### 3.2.3 Diagrammes de séquence

Le diagramme de séquence permet de représenter les interactions entre les acteurs et le système dans un ordre chronologique. Il se focalise sur l'aspect temporel [DRIRA, 2019a].

### Diagramme de séquence de l'application web côté administrateur

La figure 3.2 illustre les interaction de l'administrateur de notre application web avec le système.

Après son authentification, l'administrateur peut configurer les tables de la base de données et les modifier selon les circonstances à venir.

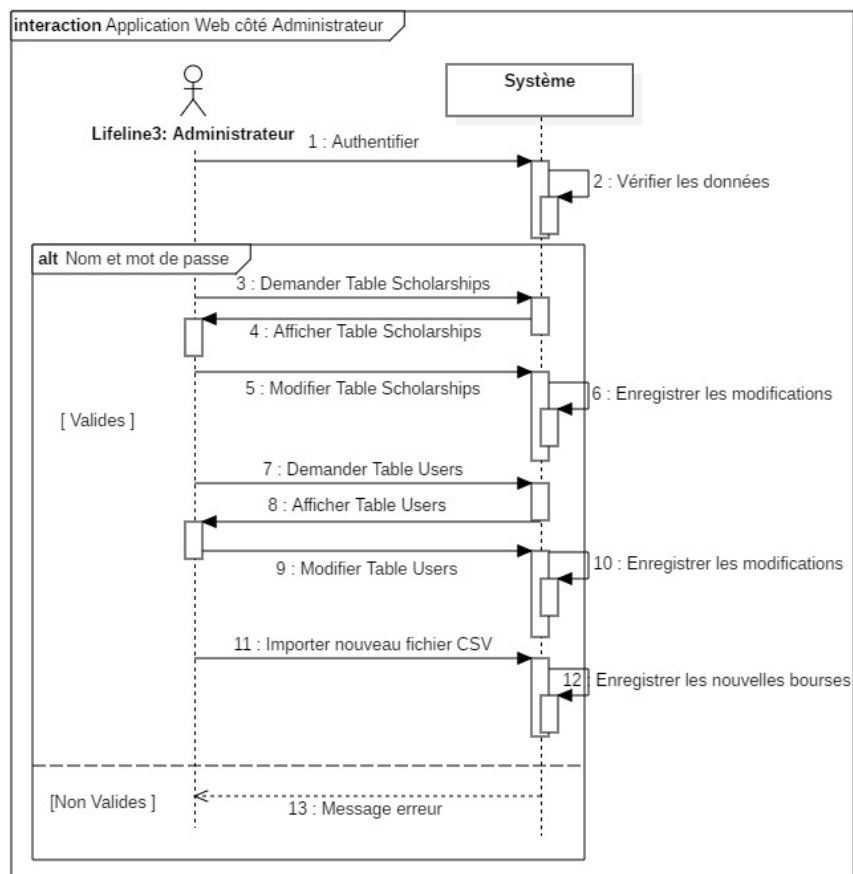


FIGURE 3.2 – Diagramme de séquence l'application web côté administrateur

### Diagramme de séquence de l'application web côté utilisateur

Dans cette partie, nous avons employé le diagramme de séquence pour schématiser l'interaction de l'utilisateur avec l'application. Cette figure 3.3 montre que celui-ci doit être authentifié tout d'abord pour qu'il puisse accéder aux services de la plate-forme.

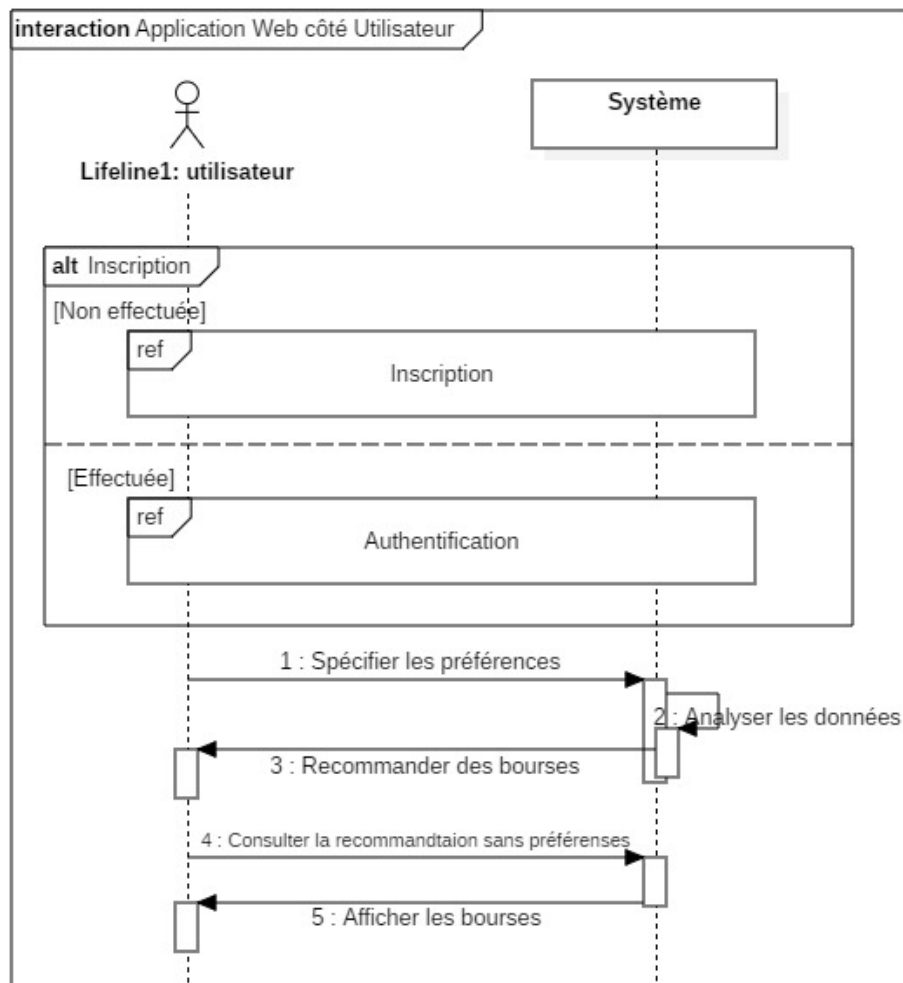


FIGURE 3.3 – Diagramme de séquence l'application web coté utilisateur

- **Inscription de l'utilisateur**

Dans la figure 3.4 nous allons considérer le scénario d'inscription de l'utilisateur. Dans le cas où l'étudiant n'admet pas de compte, il est invité à s'inscrire en remplissant le formulaire d'inscription. Une vérification des champs remplis aura lieu par notre système pour autoriser la connexion ou retourner un message d'erreur en cas d'existence des informations semblables à celles saisies.

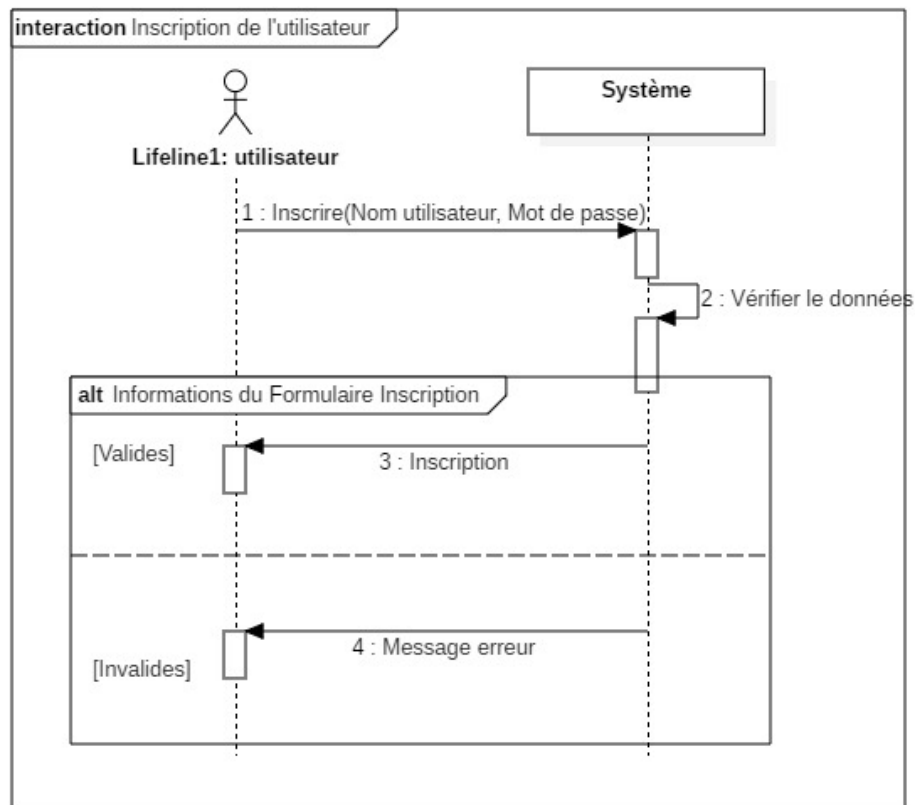


FIGURE 3.4 – Diagramme de séquence "Inscription de l'utilisateur"

- **Authentification de l'utilisateur**

La figure 3.5 qui suit modélise le scénario d'authentification de l'utilisateur. Ce dernier doit taper son nom et son mot de passe. Le système vérifiera les informations tapées et autorisera la connexion si elles sont correctes ou affichera un message d'erreur dans le cas contraire.

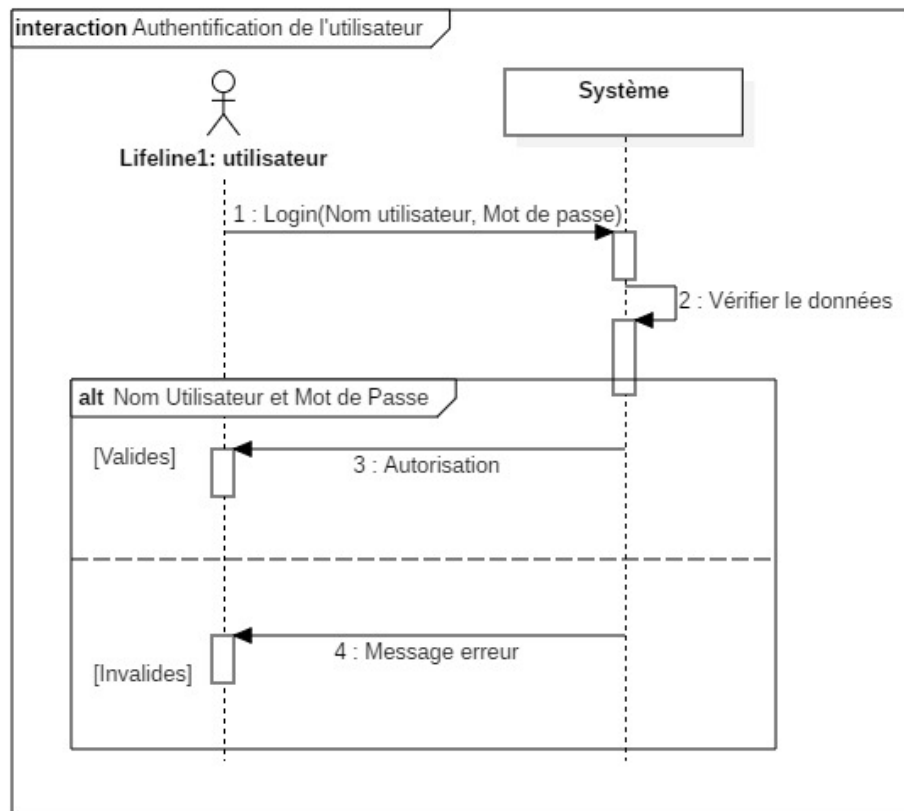


FIGURE 3.5 – Diagramme de séquence "Authentification de l'utilisateur"

## Conclusion

A la fin de ce chapitre, nous arrivons à clarifier les fonctionnalités de notre système d'une vue globale. En s'appuyant sur les diagrammes de cas d'utilisation et de séquence, nous avons effectué l'analyse et la spécification des différents besoins. Ceci va nous faciliter évidemment la réalisation et la conception de notre projet qui sera l'objectif du chapitre suivant.



# Chapitre 4

## Conception de la solution

Après avoir souligné les grandes lignes décrivant les besoins de notre système, nous mettons l'accent dans ce chapitre sur la conception qui est une phase cruciale dans le processus de développement de notre projet. Nous abordons en début la conception générale de notre solution puis nous expliquons sa conception détaillée.

### 4.1 Conception générale

Nous rappelons que notre solution se décompose en deux parties : un système de prévision et une application web. Par la suite, nous expliquons dans cette première section l'architecture physique et l'architecture logique de l'application web.

#### 4.1.1 Conception de l'architecture physique

L'architecture physique que nous avons adoptée pour notre application est l'architecture trois tiers qui représente un cas particulier du modèle multi-tiers le plus général.

En effet, cette architecture est appelée aussi une architecture à trois couches qui sont : la couche présentation, la couche applicative ou métier et la couche des données.

Notre choix de cette architecture qui nous semble la plus adéquate est fondé sur les raisons suivantes :

- Nous avons besoin d'alléger l'application du côté client vu qu'elle présente un traitement un peu lourd lors de la recommandation où nous avons préféré de l'isoler dans un autre serveur.
- Les données que les utilisateurs utilisent sont confidentielles.  
De ce fait, nous avons choisi de les placer dans un serveur de données pour assurer la sécurité.
- Nous voulons favoriser la rapidité des traitements et améliorer le temps de réponse du système en divisant les tâches entre les différents tiers.

Pour mieux comprendre les relations entre les trois tiers de notre système, nous adoptons le diagramme de déploiement 4.1 suivant :

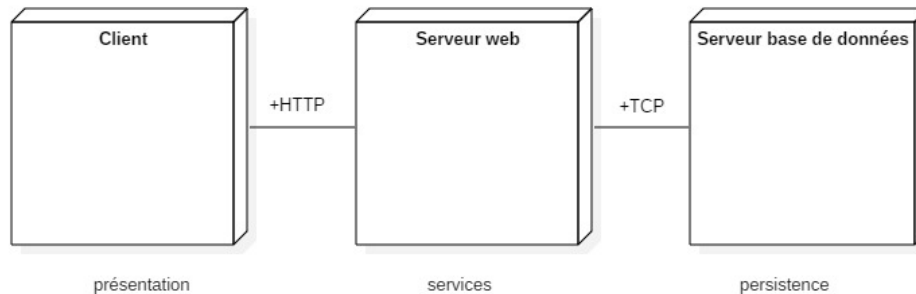


FIGURE 4.1 – Diagramme de déploiement du système

Il est bien clair que cette architecture est divisée en trois niveaux [URL12] : le client, le serveur Web et le serveur base de données.

La communication entre eux est schématisée dans la figure 4.2.

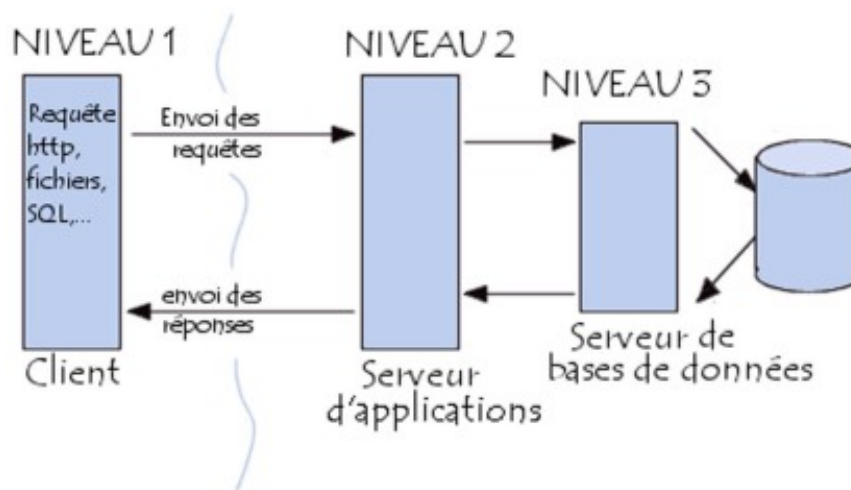


FIGURE 4.2 – Les niveaux de l'architecture 3-tiers

- **Le client** : Il représente la couche présentation. Ce niveau correspond à l'ordinateur personnel de l'utilisateur, et plus précisément à son navigateur Web qui permet l'interaction de l'application avec l'utilisateur en envoyant et recevant des requêtes HTTP.
- **Le serveur web** : Il représente la couche métier. C'est l'intermédiaire qui assure la communication entre les deux autres couches. Il est ressemblé à un moteur de toute l'application car dans cette couche, tous les traitements nécessaires pour garantir le résultat sont effectués.

- **Le serveur base de données** : Il décrit la couche de données. C'est le serveur secondaire qui stocke les données, les récupère et fournit un service au premier serveur.

#### 4.1.2 Conception de l'architecture logique

Pour l'application web de notre projet, nous avons utilisé l'infrastructure de développement Django qui utilise l'architecture MVT présentée dans la figure 4.3 suivante :

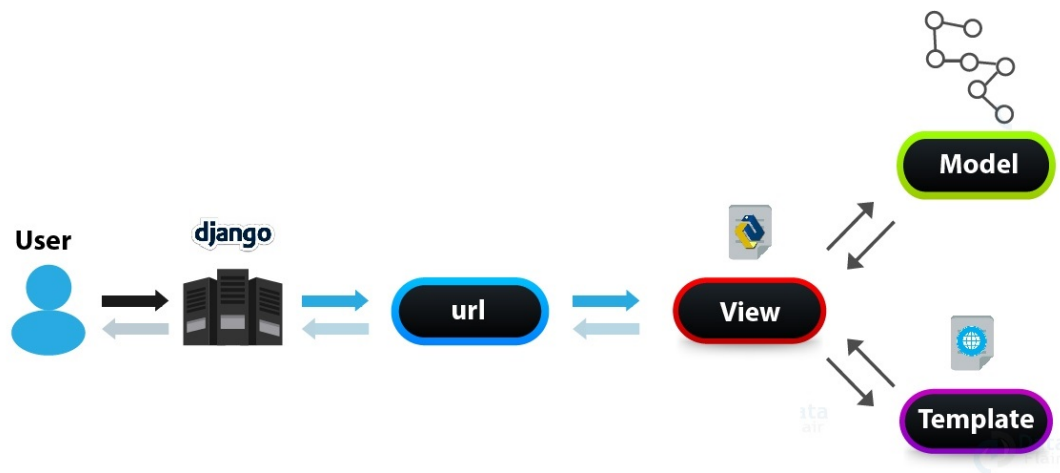


FIGURE 4.3 – Diagramme d'architecture MVT

L'architecture MVT (Model, View, Template) [URL13] est un modèle de conception de logiciels pour développer une application Web, axée sur les trois pôles suivants : le Modèle, la Vue et le Template. Son but est de séparer les responsabilités de chaque pôle afin que chacun puisse se concentrer sur ses tâches. Elle est inspirée de l'architecture classique MVC (Model, View, Controller) qui est très répandue dans les infrastructures web.

L'architecture MVT diffère légèrement de celle MVC. En effet, ce qui spécifie Django c'est le fait qu'il gère lui-même le Contrôleur.

Comme déjà mentionné, le modèle de conception MVT sépare le code en trois parties :

- **Le modèle** : Il représente les informations stockées dans la base de données. Il permet d'accéder, modifier, ajouter et supprimer les données. C'est une interface supplémentaire pour effectuer la communication avec la base de données.
- **La vue** : Elle effectue toutes les actions nécessaires pour recevoir et répondre aux requêtes HTTP. C'est l'intermédiaire qui assure l'échange entre le modèle et le template. Lorsqu'une interaction avec la base de données est nécessaire, la vue appelle le modèle. De l'autre côté, si un des traitements nécessite l'intervention du template, la vue l'appelle.

- **Le template** : C'est l'ensemble des fichiers HTML qui peuvent recevoir des objets python, afficher des variables et utiliser des structures conditionnelles et des boucles dans le code HTML grâce à un moteur fourni par Django qui sera analysé et exécuté par cette infrastructure.

## 4.2 Conception détaillée

Après avoir expliqué l'architecture globale de notre application, nous allons présenter la conception en détail en utilisant différents diagrammes UML.

Il s'agit donc de représenter les étapes à suivre pour l'apprentissage automatique à l'aide des diagrammes d'activité, concevoir la base de données et modéliser l'aspect dynamique du système en dressant le diagramme de séquence des cas d'utilisation.

### 4.2.1 Diagramme d'activité

Les diagrammes d'activité UML [DRIRA, 2019b] constituent un outil de modélisation des systèmes des flux des travaux, des modèles orientés service et des processus métiers.

#### Diagramme d'activité du modèle de prédiction

Nous commençons dans cette sous section par la conception détaillée de notre modèle d'apprentissage automatique. Le diagramme d'activité illustré dans la figure 4.4 explique le déroulement des étapes fondamentales qui aboutissent pour obtenir le bon modèle.

En réalité, la réalisation de notre modèle de prédiction commence par la phase de pré-traitement des données où se déroulent la collection des données, le nettoyage et l'extraction des caractéristiques. Par la suite, la phase de construction du modèle aura lieu comportant son entraînement, son évaluation et le réglage de ses paramètres jusqu'à l'obtention d'une bonne précision.

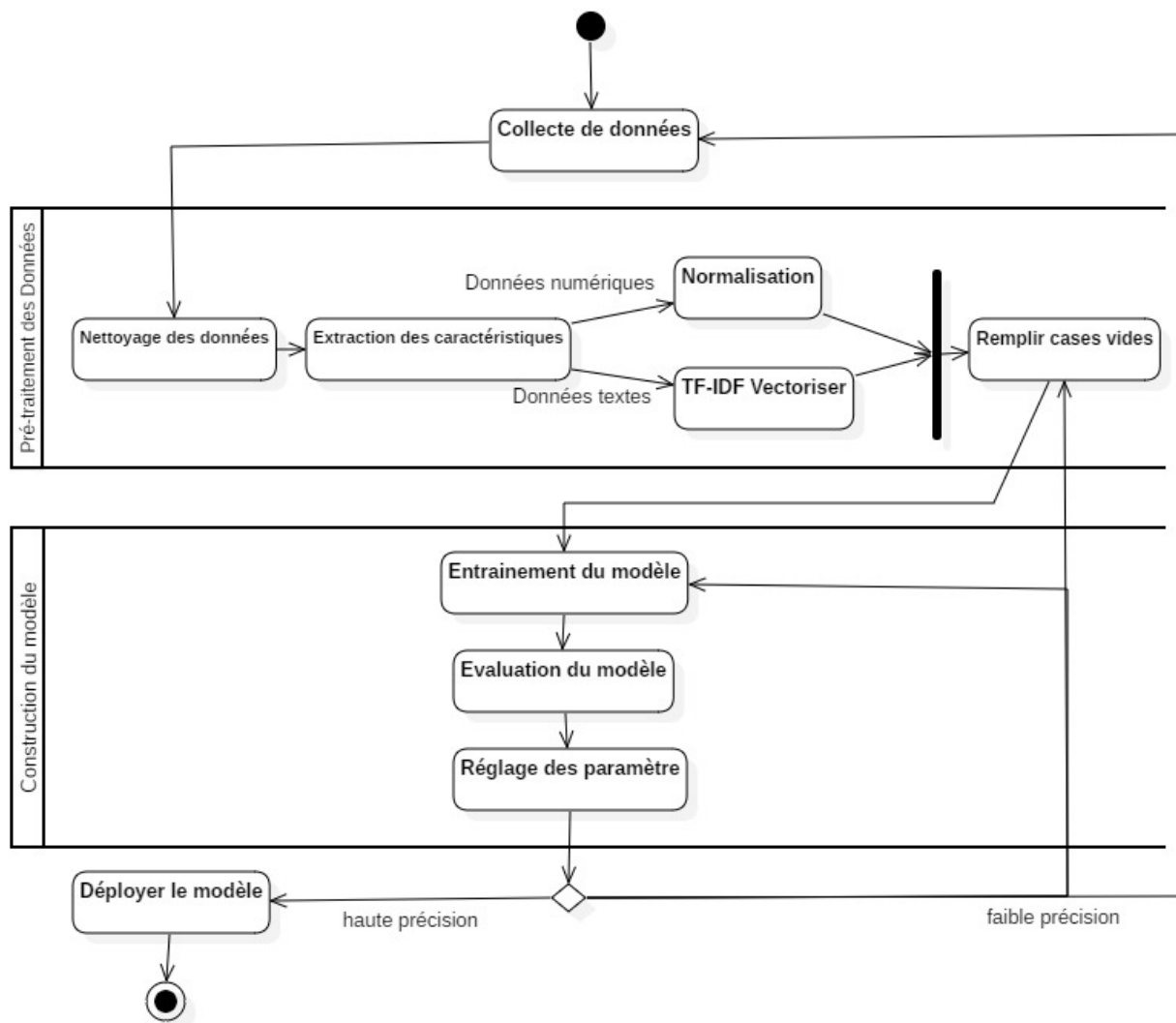


FIGURE 4.4 – Diagramme d'activité du modèle de prédiction

### Diagramme d'activité de l'algorithme de regroupement k-moyennes

Afin d'expliquer en détail le principe de fonctionnement de l'algorithme k-moyennes [URL14] qui est l'un des algorithmes utilisés pour construire notre modèle, nous présentons son diagramme d'activité dans la figure 4.5.

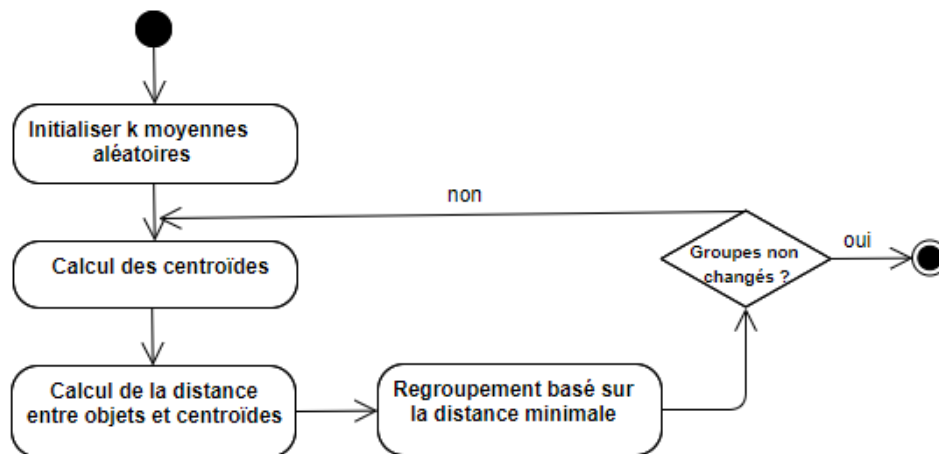


FIGURE 4.5 – Diagramme d'activité de l'algorithme de regroupement k-moyennes

### 4.2.2 Diagrammes de séquence

#### Demander une liste de bourses

La figure 4.6 représente le diagramme de séquence du système de recommandation. Dans ce diagramme nous avons présenté les interactions entre l'utilisateur, l'interface utilisateur, le « backend » API et la base de données.

Pour qu'il puisse accéder à la plate-forme, l'utilisateur doit d'abord se connecter ou créer un compte s'il n'admet pas un auparavant.

Dès qu'il sera bien authentifié, l'utilisateur peut spécifier ses préférences concernant les bourses puis les envoyer.

Ces informations seront bien traitées dans le « backend » par le modèle déjà construit pour donner la classe adéquate aux informations entrées.

Ensuite, une requête sera envoyée à la base de données pour sélectionner la liste des bourses de la classe donnée par le modèle de prédiction. Le résultat de la recherche sera affiché par ordre qui suit les meilleurs critères décrivant les bourses et leurs universités correspondantes tout en éliminant celles qui ont des dates limites achevées.

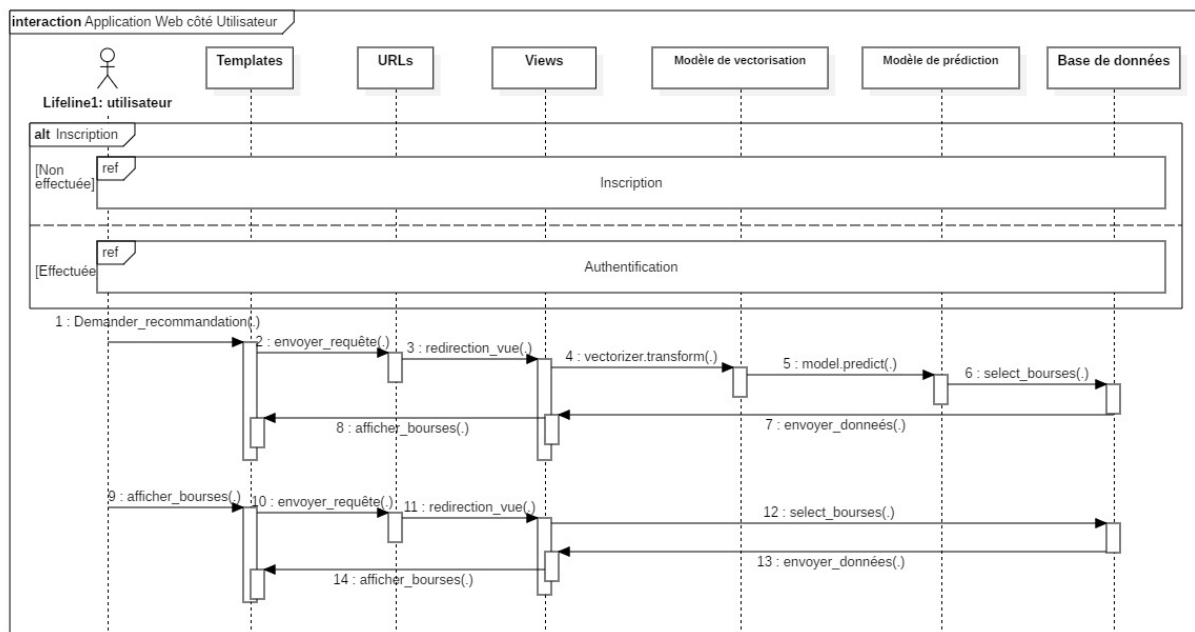


FIGURE 4.6 – Diagramme de séquence l'application web côté utilisateur

### Inscription de l'utilisateur

Cette figure 4.7 représente le diagramme de séquence de l'inscription de l'utilisateur. Ce dernier doit remplir le formulaire d'inscription et cliquer sur le bouton « Créer un compte ». A ce moment, une requête sera envoyée au « backend » contenant les données du formulaire. Celles-ci vont être vérifiées afin de limiter l'accès uniquement aux utilisateurs qui sont inscrits dans la plate-forme « Edutest »

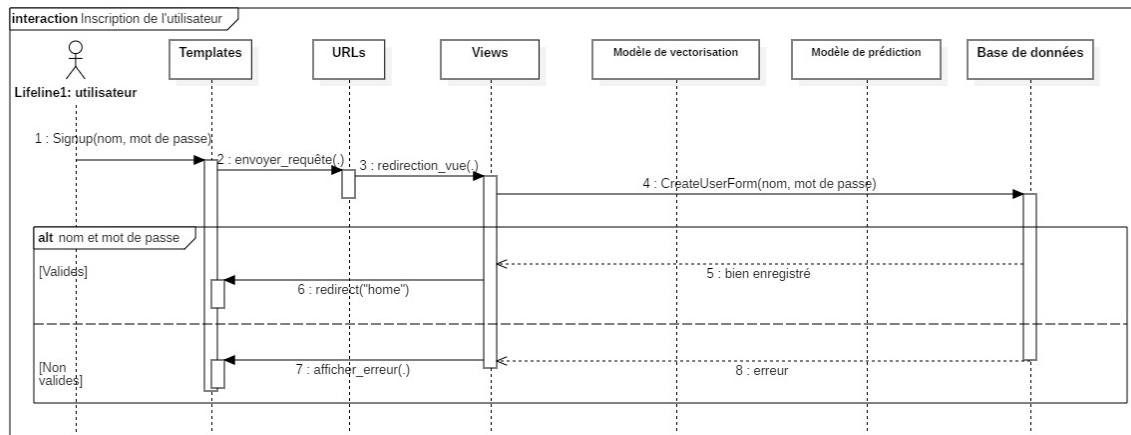


FIGURE 4.7 – Diagramme de séquence "Inscription de l'utilisateur"

### Authentification de l'utilisateur

La figure 4.8 représente le diagramme de séquence de l'authentification de l'utilisateur. Après que l'étudiant de la plate-forme « Edutest » saisisse son nom d'utilisateur et son mot de passe et clique sur le bouton « Connexion », ses informations seront envoyées au « backend » pour la vérification. Ce dernier va envoyer à son tour une requête à la base de données pour vérifier si le nom de l'utilisateur et le mot de passe sont corrects. En cas d'erreur un message sera affiché sur l'écran, sinon l'utilisateur est bien authentifié et il peut profiter du service de l'application.

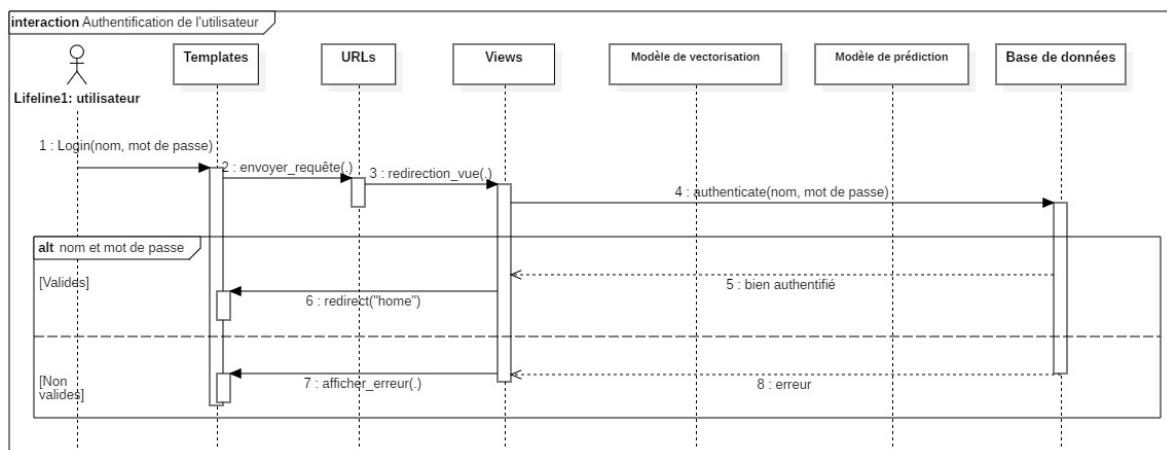


FIGURE 4.8 – Diagramme de séquence "Authentification de l'utilisateur"



## Conclusion

Dans ce quatrième chapitre, nous avons fait la conception détaillée de notre projet. Par conséquent, nous avons bien préparé le terrain pour la phase de mise en oeuvre détaillée dans le chapitre suivant. Celui-ci exposera les environnements de travail et le scénario d'exécution de l'application.

# Chapitre 5

## Réalisation

Dans ce dernier chapitre, nous allons donner les environnements de travail utilisés matériel et logiciel. Ensuite, nous allons présenter les fonctionnalités de notre système de recommandation avec des captures écrans.

### 5.1 Environnement de développement

#### 5.1.1 Environnement matériel

Tout au long de ce projet nous avons utilisé nos machines.

	PC 1	PC 2
Processeur :	Intel i5-7200U CPU 2.50 GHz	Intel i7-7500U CPU 2.70GHz
Mémoire :	8 Go de RAM	8 Go de RAM
Disque Dur :	1 To	1 To
Système d'Exploitation :	Window 10	Window 10

TABLE 5.1 – Machines

#### 5.1.2 Environnement logiciel

Dans cette partie, nous allons illustrer l'environnement logiciel utilisé pour l'implémentation de ce projet.

- **Anaconda** est un environnement de programmation code source ouvert dédié à la science des données et l'apprentissage automatique. Toutes les parties de ce projet liées à l'intelligence artificielle sont effectuées avec cet environnement.

- **Jupyter Notebook** est une application web pour développer plus de 40 langages de programmation que nous l'avons utilisé pour programmer en langage python. Elle nous fournit des cellules dans lesquelles nous pouvons écrire des codes et les exécuter au fur et à mesure accompagnés des textes de documentation.
- **Visual Studio Code** est un éditeur de code librement extensible développé par Microsoft, code source ouvert et qui supporte plusieurs langages de programmation. Nous l'avons utilisé pour le développement de l'application web.
- **Overleaf** est un éditeur collaboratif Latex en ligne que nous l'avons utilisé pour la rédaction du rapport.
- **StarUML** est un logiciel de conception UML que nous avons utilisé pour schématiser les diagrammes de conception de ce projet.

## 5.2 Choix Techniques

Dans cette section, nous présenterons les différentes technologies utilisées pour implémenter cette application et les raisons du choix de ces technologies.

### 5.2.1 Langages de Programmation

- **Python** est un langage de programmation open source interprété, orienté objet, multi-paradigme et multi-plateforme. Il est facile à apprendre et peut traiter plusieurs domaines : développement Web, science des données, intelligence artificielle, sécurité etc... Il est donc parmi les langages populaires les plus utilisés dans le monde et plus précisément pour l'intelligence artificielle.  
Nous l'avons utilisé vu sa grande performance et l'existence des bibliothèques qui rendent le travail plus efficace. Ceci nous a aidés à résoudre les problèmes dans les courts délais grâce à la présence d'une documentation bien détaillée sur ce langage.
- **HTML, CSS et JavaScript** sont les langages du « frontend » les plus populaires. HTML fournit la structure de base du site web et CSS offre les différents styles pour une bonne présentation de l'interface web. Concernant JavaScript, il est utilisé pour le contrôle de saisie côté client des différents éléments de l'application web.

### 5.2.2 Cadres de développement (Frameworks)

- **Django** est un cadre de développement programmé en Python, de haut niveau et code source ouvert utilisé pour le développement de la partie « backend » du site web. Grâce à sa simplicité et sa sécurité, Django qui suit l'architecture MVT inspirée du modèle de conception MVC demeure une infrastructure très puissante.  
Nous l'avons choisi vu sa compatibilité avec notre modèle de prédiction programmé en langage Python.

- **Bootstrap** est un cadre de développement de CSS pour la conception de la partie « frontend » des sites web élaboré par Twitter en 2011. Il est code source ouvert et le projet le plus populaire sur la plate-forme GitHub. Contenant des codes en HTML, CSS et JavaScript, Bootstrap offre des composants réutilisables et facilite diverses tâches comme la création des formulaires, des boutons etc...

### 5.2.3 Bibliothèques

- **Pandas** est une bibliothèque Python pour l'analyse des données, qui nous a permis de traiter des fichiers d'extensions "CSV" ou "XLSX" et d'enregistrer leurs informations dans les « Data-Frame » et vice-versa.
- **Numpy** est une bibliothèque Python conçue pour traiter des tableaux multidimensionnels. Elle fournit plusieurs fonctions mathématiques pour les matrices.
- **Matplotlib** est une bibliothèque Python pour dessiner et visualiser des données dans des graphiques 2D ou 3D. Elle nous a aidés à visualiser la distribution des données afin que nous puissions régler les paramètres de notre modèle et choisir l'algorithme le plus adéquat.
- **Sklearn** est une bibliothèque Python code source ouvert pour l'apprentissage automatique. Elle contient de nombreux algorithmes pour nous aider à construire le bon modèle de recommandation.
- **NLTK et Spacy** sont des bibliothèques Python utilisées pour le nettoyage, la normalisation et la vectorisation des données textuelles. Spacy est plus rapide que NLTK en termes de traitement, mais NLTK fournit certaines fonctionnalités que Spacy ne fournit pas. C'est pourquoi nous avons utilisé les deux bibliothèques qui sont complémentaires.
- **BeautifulSoup** est une bibliothèque Python utilisée pour l'extraction des données d'un fichier HTML ou XML. Elle nous a aidés à collecter l'ensemble de données des bourses d'études et des universités à partir des différents sites web.
- **Pickle** est une librairie Python que nous avons utilisé pour enregistrer notre modèle de prédiction ainsi que le modèle de vectorisation dans un fichier binaire.

### 5.2.4 Construction du modèle de prédiction

#### Collecte des données

Dans l'entame de notre projet de conception et de développement, nous avons cherché des données sur des plates-formes différentes comme Kaggle, Zindi... Cependant, nous n'avons pas trouvé les données nécessaires et suffisantes pour la construction de notre modèle et dans le meilleur des cas nous avons trouvé des informations qui ne sont pas mises à jour. Par conséquent, nous avons recours au « Web Scraping » . De ce fait, nous

avons préparé 6 algorithmes pour collecter les données de 6 sites Web différents et les enregistrer dans un fichier CSV, comme le montre la figure suivante 5.1. Pour effectuer la mise à jour de ces données et apporter des nouvelles opportunités, il suffit de réexécuter ces algorithmes.

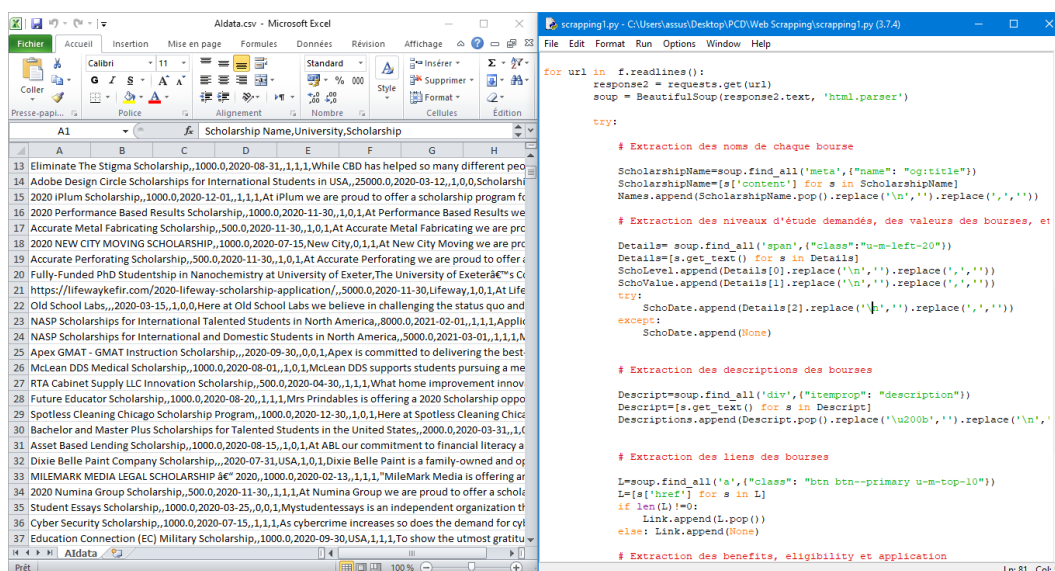


FIGURE 5.1 – Grattage Web

## Pré-traitement des données

C'est l'étape la plus difficile qui a pris beaucoup de temps vu que les informations extraites sous forme de texte ne sont pas structurées. Pour cette raison, nous avons commencé par le nettoyage des données en supprimant les caractères et les mots d'arrêts (stop words) puis nous effectuons la racinisation des mots. Ces étapes sont illustrées dans la figure 5.2.

```
#Text pre-processing
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
nltk.download('stopwords')
nltk.download('wordnet')
import string

def text_process(text):
    stemmer = WordNetLemmatizer()
    nopunc = [char for char in text if char not in string.punctuation] #supprimer ponctuation
    nopunc = ''.join([i for i in nopunc if not i.isdigit()])
    nopunc = [word.lower() for word in nopunc.split() if word not in stopwords.words('english')] #supprimer Les mots d'arrêts
    return [stemmer.lemmatize(word) for word in nopunc] #Racinisation des mots
```

FIGURE 5.2 – Nettoyage du texte

Suite à la phase de nettoyage des données textuelles, nous avons extrait les caractéristiques telles que valeur de la bourse, la date limite d'inscription, le nom d'université, la localisation... Maintenant les données sont divisées en deux catégories : des données numériques qui doivent être normalisées (valeur varie entre 0 et 1) et les données de type texte qui doivent être vectorisées à l'aide de « TF-IDF » Vectoriseur illustré dans la figure ci-dessous 5.3.

```
from sklearn.feature_extraction.text import TfidfVectorizer

def vectorise(text):
    vectorizer = TfidfVectorizer() # créer la transformation
    vectorizer.fit(text) # tokenize et construire vocab
    vector = vectorizer.transform([text]) # encoder document
    return(vector)
```

FIGURE 5.3 – TF-IDF Vectoriseur

Après avoir corrélié toutes les données dans le même tableau, nous avons remarqué qu'il y avait plusieurs cases vides qui traduisent un manque de données pour certaines caractéristiques. Nous devons donc remplir ces valeurs vides en utilisant l'algorithme k-voisins les plus proches (KNN).

### Visualisation des données

Afin de pouvoir connaître la distribution des données, il est nécessaire de tracer la courbe 3D représentée sur cette figure 5.4

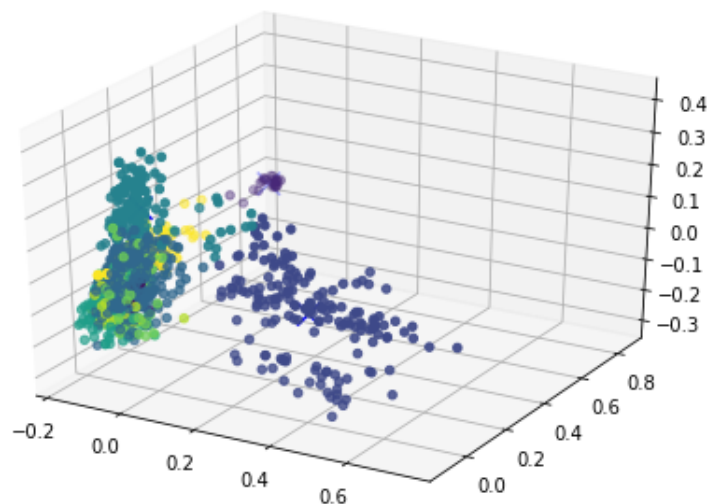


FIGURE 5.4 – Visualisation des données

### Choix de l'algorithme d'apprentissage automatique

Comme notre jeu de données n'a pas du label, nous avons utilisé l'apprentissage non-supervisé pour construire le modèle. Nous avons essayé plusieurs algorithmes pour ce type d'apprentissage : algorithme de regroupement K-moyennes, algorithme DBSCAN, algorithme de regroupement spectral, algorithme de propagation d'affinité et algorithme agglomératif.

Ainsi, nous devons choisir l'un de ces algorithmes et essayer de régler ses paramètres pour atteindre la meilleure précision. D'une autre part, puisque nous traitons un problème d'apprentissage non-supervisé, nous ne pouvons pas calculer la précision de manière classique. Cependant, nous avons trouvé la méthode de coefficient Silhouette [URL15], qui se base sur le calcul de la distance entre les points comme le montre cette figure 5.5


$$\begin{aligned} a(i) &= \frac{1}{|C(i)|-1} \sum_{C(i), i \neq j} d(i, j) \\ b(i) &= \min_{i \neq j} \left( \frac{1}{|C(j)|} \sum_{j \in C(j)} d(i, j) \right) \\ s(i) &= \frac{b(i) - a(i)}{\max(a(i), b(i))} \end{aligned}$$

FIGURE 5.5 – Le coefficient de Silhouette

Avec :

- $C(i)$  : Le cluster affecté au  $i$ -ème point de données.
- $|C(i)|$  : Le nombre de points de données dans le cluster affecté au  $i$ ème point de données.
- $a(i)$  : Une mesure de la qualité de l'attribution du  $i$ ème point de données à son cluster.
- $b(i)$  : La dissemblance moyenne au cluster le plus proche qui n'est pas son cluster.
- $s(i)$  : Le coefficient de silhouette.

Nous déterminons la valeur moyenne des coefficients de Silhouette de tous les points pour chaque algorithme comme illustrée dans la figure suivante 5.6. La meilleure valeur de coefficient de Silhouette est 1 et la valeur d'erreur est -1. D'après les valeurs retournées par les différents algorithmes testés, nous constatons que l'algorithme K-moyennes donne les meilleurs résultats.



Out[47]:

	Silhouette
<b>K-means</b>	0.621232
<b>Spectral</b>	0.346534
<b>DBSCAN</b>	0.325782
<b>Affinity</b>	0.044342
<b>Agglomerative</b>	-0.032420

FIGURE 5.6 – Evaluation du modèle

## 5.3 Travail achevé

Après avoir terminé la phase de l'apprentissage automatique, nous passons à la partie Web. Nous allons présenter dans cette section l'application web que nous avons réalisée et ses différentes fonctionnalités.

### 5.3.1 Application Web

Dans la page d'accueil exposée dans la figure 5.7, l'utilisateur trouvera une explication de sa finalité. Il doit par la suite s'authentifier en cliquant sur le bouton « Connexion ». S'il n'a pas encore de compte, il est invité à créer un en cliquant sur le bouton « Créer un Compte » comme indiqué dans la figure 5.8.

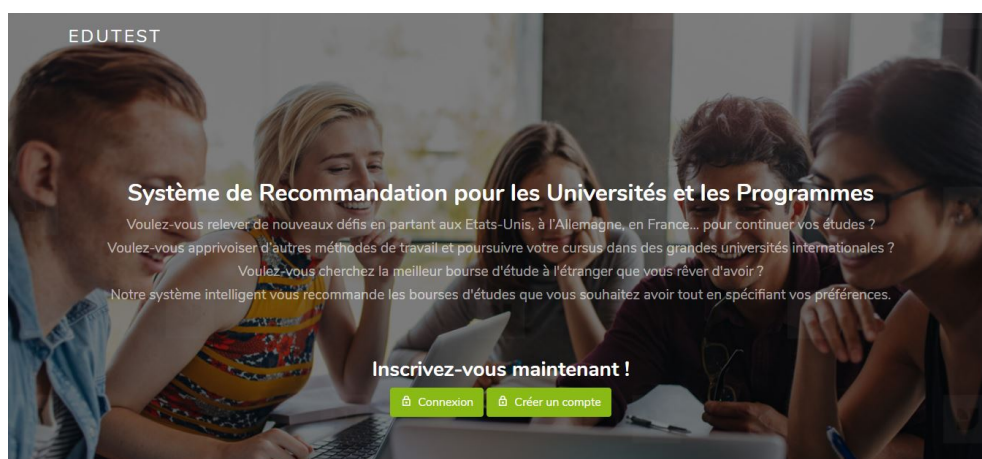


FIGURE 5.7 – Page d'accueil



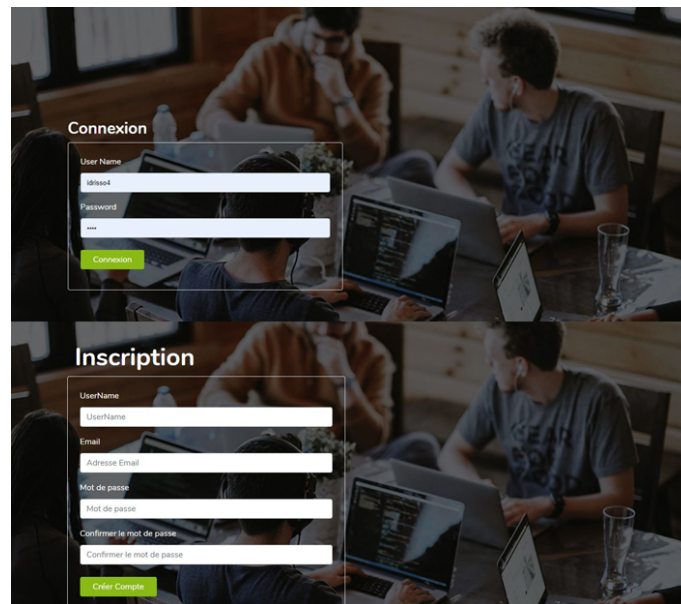


FIGURE 5.8 – Les interfaces d’authentification et d’inscription de l’utilisateur

Après l’authentification, la page principale 5.9 apparaît. Elle présente le formulaire utilisé pour spécifier les préférences que l’utilisateur doit remplir. Il y a des champs pour la valeur de la bourse, le pays, le domaine d’étude et les niveaux. S’il a d’autres spécifications, il peut les saisir dans le champ de description.

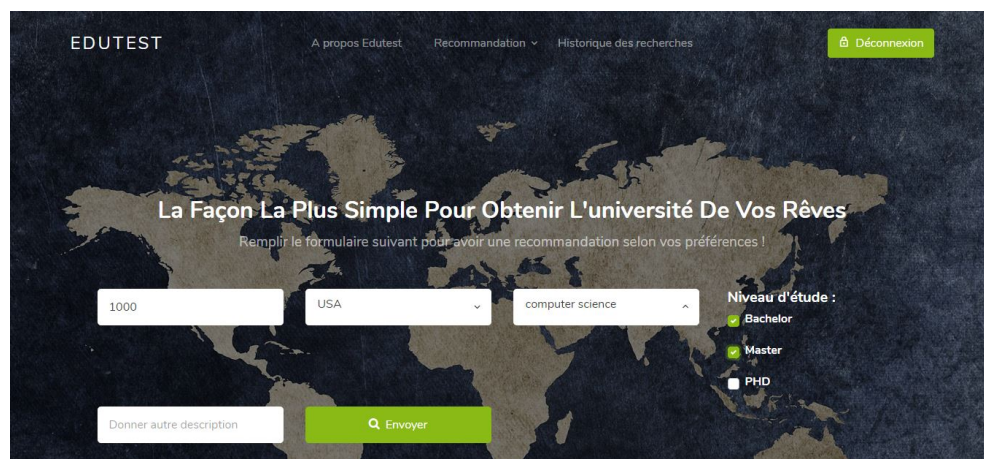


FIGURE 5.9 – Formulaire de spécification des préférences

Dès qu’il clique sur le bouton « Envoyer », une nouvelle page contenant les résultats de recherche sera affichée immédiatement sur l’écran 5.10. Seules les bourses qui n’ont pas encore expiré sont affichées et triées.






	Global Corners International Student Recruitment and Retention Award at University of Oregon in USA Université : University of Oregon Date limite d'inscription :2020-03-01	Eugene ...	45000.0 \$
	Jungle Scholar Université : Auburn University Date limite d'inscription :2019-11-15	Ironton	2000.0 \$
	\$5000 Hach & Rose LLP Annual College Scholarship Université : Hach & Rose LLP Annual College Scholarship Date limite d'inscription :2019-06-01	New York City	5000.0 \$
	The Investor's Podcast Scholarship Université : American University Date limite d'inscription :2018-10-01	Ironton	1000.0 \$
	MacCormac Honor Scholarship Université : MacCormac College Date limite d'inscription :2018-08-01	Localisation non spécifiée	1000.0 \$

FIGURE 5.10 – Résultat de la recommandation

A ce stade, l'utilisateur peut consulter les bourses une par une pour voir ses informations détaillées 5.11. Si celle-ci est satisfaisante pour lui, il est invité à s'inscrire à la bourse en cliquant simplement sur le bouton « Apply » ou en visitant son site officiel pour des renseignements supplémentaires.

### George Washington University Global Leaders Fellowship Program 2018-2019

 George Washington University  Ironton  2019-02-08

Apply Now



#### Scholarship Summary

**University:** George Washington University  
**Rank:** None  
**Program:** data science, management health informatics, information systems technology, business analytic  
**Level:** Bachelor Phd Master  
**University Location:** Ironton  
**Scholarship Value:** 30000.0 \$  
**Application Deadline:** 2019-02-08

FIGURE 5.11 – Bourse en détails

Si l'utilisateur souhaite consulter la recommandation par défaut, il doit appuyer sur le bouton « Recommandation sans préférences » qui se trouve sur la barre de navigation. Cette recommandation est commune pour tous les utilisateurs selon les critères généraux.

D'une autre part, l'utilisateur peut consulter l'historique de ses recommandations en appuyant sur « Historique des recherches » comme indiqué dans cette figure 5.12 pour revoir les anciennes recommandations.

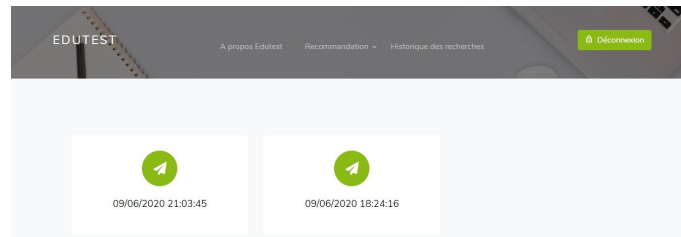


FIGURE 5.12 – Historique des recherches

### 5.3.2 Interface administrateur

Dans cette section, nous présenterons l'interface de l'administrateur et ses fonctions fournies par Django. L'admin doit s'authentifier en saisissant son nom et son mot de passe comme indiqué dans la figure 5.13.

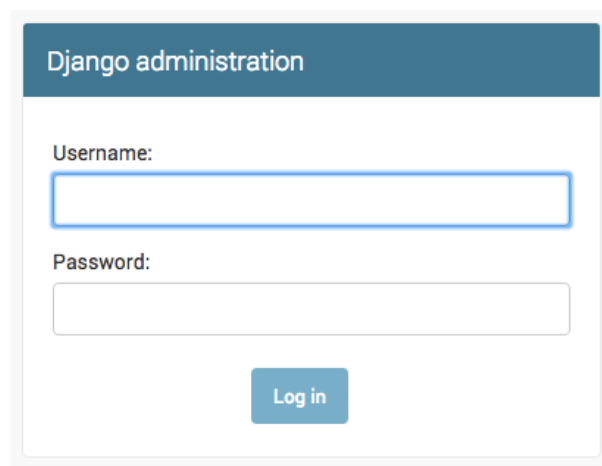


FIGURE 5.13 – Interface Authentification de l'administrateur

Après l'authentification, la page « Django Administration » se présente comme cette figure 5.14 le montre. Cette interface représente la base de données de l'application qui comporte deux tables : « Scholarships » contient les bourses d'études et « Users » contient la liste des utilisateurs et des administrateurs.



FIGURE 5.14 – Interface « Django Administration »

La figure 5.15 montre la table « Users » où l'administrateur est représenté par une croix verte alors que l'utilisateur est représenté par une croix rouge. Les administrateurs peuvent utiliser les opérations « CRUD » pour gérer tout le contenu et peuvent également gérer les privilèges des utilisateurs.

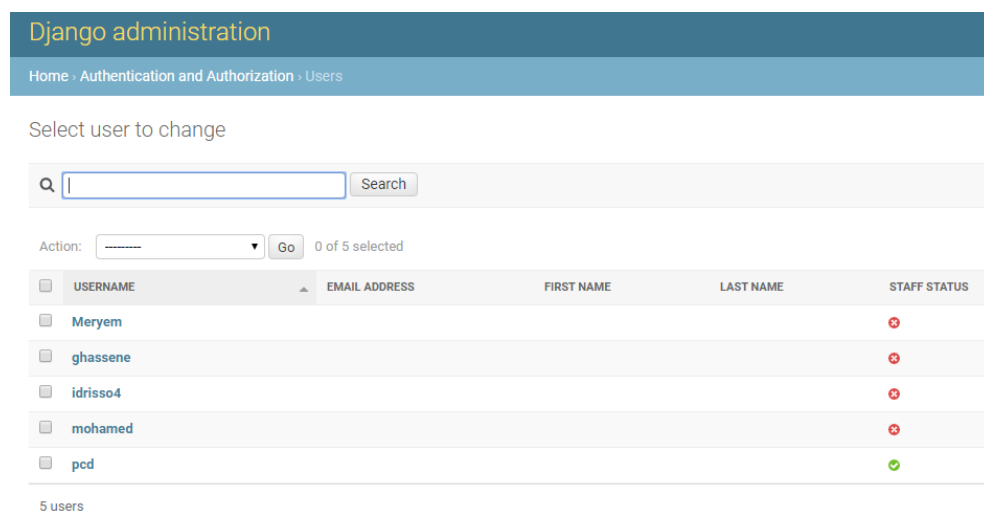


FIGURE 5.15 – Interface de la table « Users » des utilisateurs

Cette figure 5.16 représente la table « Scholarships » qui contient tous les bourses d'études. L'administrateur peut aussi les gérer à travers les opérations « CRUD ».

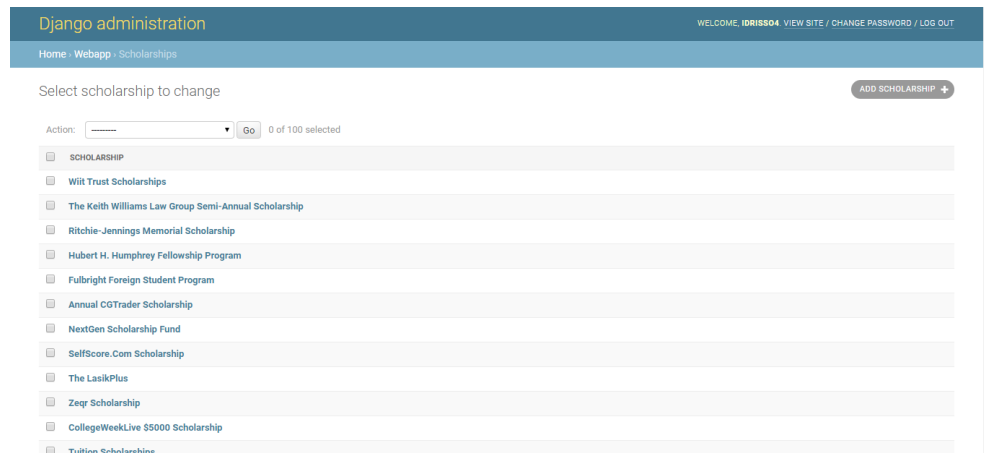


FIGURE 5.16 – Interface de la table « Scholarships » des bourses

Le formulaire dans cette figure 5.17 est utilisé pour ajouter ou modifier des bourses.

Change scholarship

HISTORY

ScholarshipName: Robert M. Helfend Criminal Defense Scholars

University: the Pepperdine University School of Law

ScholarshipValue: 500.0

Deadline: 2020-02-15

Location: USA

Level: Bachelor Phd Master

Description: Los Angeles criminal defense attorney Robert

Link: https://www.robertmhelfend.com/law-school

Delete Save and add another Save and continue editing SAVE

FIGURE 5.17 – Formulaire pour ajouter une nouvelle bourse

## Conclusion

Tout au long de ce chapitre nous avons présenté les environnements de travail, les techniques utilisées : les langages de programmation, cadres de développement, bibliothèques et les différents algorithmes pour l'apprentissage automatique ... ainsi que l'application Web et ses fonctionnalités.

# Conclusion et perspectives

L'intelligence artificielle, et en particulier l'apprentissage, automatique demeure aujourd'hui une technologie très demandée dans les différents secteurs qui a gagné la confiance du grand public pour résoudre des problèmes que l'être humain est incapable parfois de les gérer. Grâce à sa puissance en terme de temps de réponse, précision et de manipulations des données massives, cette technologie est souvent utilisée pour construire des systèmes de recommandation qui réduisent la probabilité d'erreur dans les décisions des êtres humains et favorisent la satisfaction de leurs choix.

Partant de cette constatation, nous avons été chargés dans le cadre de ce projet de conception et de développement de développer une partie importante d'un projet plus vaste avec la Startup « Edutest » qui consiste à créer une plate-forme d'études en Anglais en ligne pour aider les étudiants à poursuivre leurs cursus à l'étranger et notamment aux Etats Unis. Notre mission consiste à ajouter à cette plate-forme un système de recommandation pour les bourses d'études à l'étranger.

Nous avons présenté au cours de ce rapport les différentes étapes menant à la réalisation du projet où nous avons détaillé notre étude de la solution en cinq parties. Dans la première partie, nous avons étudié les technologies existantes pour améliorer les solutions déjà proposées par d'autres personnes. Dans la deuxième partie, nous avons exposé des recherches théoriques sur les notions de base des outils et technologies utilisés pour mettre en œuvre notre projet. Dans la troisième partie, nous avons spécifié les besoins et précisé les fonctionnalités et les services auxquels l'application devait répondre. Concernant la quatrième partie, nous avons détaillé la conception de notre application en employant les différents diagrammes UML. Finalement, nous avons exposé dans le dernier chapitre l'implémentation de la partie de l'apprentissage automatique et celle de l'application web tout en présentant l'environnement logiciel et physique. Ce qui nous a permis de mettre en oeuvre notre solution.

Au cours de la réalisation de ce projet, nous avons rencontré quelques problèmes pratiques lors de la phase de collection et nettoyage des données. D'autre part, il est souvent connu que l'apprentissage automatique non-supervisé au contraire de celui supervisé, pose une grande difficulté pour avoir des bons résultats et atteindre la performance souhaitée.

De ce fait, comme chaque projet peut être amélioré et élargi, nous avons l'intention de poursuivre le travail avec « Edutest » pour l'amélioration des données qui peuvent être enrichies après un certain temps par les expériences personnelles des étudiants de cette plate-forme. En intégrant leurs informations concernant les choix des bourses, l'affectation aux universités, les niveaux de satisfaction, etc. Nous pourrions améliorer notre modèle de recommandation et essayer de préparer une nouvelle solution en se basant cette fois sur l'apprentissage supervisé. De même, nous pouvons aller plus loin en enrichissant la plate-forme avec un chatbot intelligent pour répondre aux questions des étudiants et faciliter l'interaction avec eux.

# Bibliographie

[DRIRA, 2019a] DRIRA, D. R. (2019a). *Cours Analyse et Conception Orientées Objet*.

[DRIRA, 2019b] DRIRA, D. R. (2019b). *Cours Modèles dynamiques*.

[Saint-Cirgue, 2019] Saint-Cirgue, G. (2019). *Apprendre le Machine Learning en une semaine*.



# Netographie

[URL1] <http://edutest.tn>.

[URL2] <https://github.com/dheerajbokde/university-recommendation-system>.

[URL3] <https://www.scholarshipportal.com/scholarships>.

[URL4] <https://www.niche.com/colleges/scholarships>.

[URL5] <https://www.geeksforgeeks.org/ml-types-learning-part-2/>.

[URL6] <https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing>.

[URL7] <https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b42>.

[URL8] <https://fr.wikipedia.org/wiki/TF-IDF>.

[URL9] <https://fr.wikipedia.org/wiki/K-moyennes>.

[URL10] <https://fr.wikipedia.org/wiki/DBSCAN>.

[URL11] [https://fr.wikipedia.org/wiki/Partitionnement\\_spectral](https://fr.wikipedia.org/wiki/Partitionnement_spectral).

[URL12] <https://stph.scenari-community.org/bdd/lap2/co/webUC003archi.html>.

[URL13] <https://openclassrooms.com/fr/courses/4425076-decouvrez-le-framework-django/4631014-decouvrez-larchitecture-mvt>.

[URL14] <https://towardsdatascience.com/k-means-clustering-algorithm-applications-eval>.

[URL15] <https://www.geeksforgeeks.org/silhouette-algorithm-to-determine-the-optimal-v>.