People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University Abbes Laghrour, Khenchela
Faculty of Sciences and Technology

Department of Computer Science

Thesis

To obtain the degree of MASTER IN COMPUTER SCIENCE

Option : Artificial Intelligence

An Intelligent Chatbot for Personalized University Orientation Recommendation

Presented by :
Hadil Saoudi

Thesis Committee

Dr. Abdelali Bakhouch	President (Associate Professor)	University of Khenchela
Dr. Asma Bezza	Supervisor (Associate Professor)	University of Khenchela
Dr. Saida Lahis	Examiner (Associate Professor)	University of Khenchela

Mai 2026

وَكَانَ فَضْلُ اللهِ عَلَيْكُمْ عَظِيمًا

Dedication

In the beginning, I dedicate this success first and foremost
to my mother, may Allah have mercy on her,
who was my support and the light of my path,
I ask Allah to place this work in the balance of her good deeds
and to grant her Paradise.

I also dedicate it to my dear father,
who is in the blessed lands to perform the pilgrimage,
May Allah accept it from him, protect him for us,
and grant him a long and healthy life.

I dedicate the result of my hard work to my supportive friend
and companion on my journey, Seif Eddine, like a brother to me,
who has been by my side in every step.
May Allah bless him with all goodness.

I also dedicate it to my dear brother, Aymen,
wishing him all success and happiness in his life.

I must not forget to dedicate this success to my sisters,
Salsabil, Zineb, and Ferdous,
hoping for them continuous success and prosperity
in their academic journey and the highest ranks,
if Allah wills.

And to my cousin Hafidha and her little one, Sidra,
may Allah fulfill all their wishes.

And to everyone who supported me
and wished me well from behind the scenes.

May Allah have mercy on my mother
and protect for me my beloved ones.

Acknowledgement

the name of Allah, the Most Gracious, the Most Merciful.

All praise and thanks are due to Allah, by whose grace good deeds are accomplished, and by whose favour and guidance I have reached this stage. Praise be to Allah in all circumstances.

I extend my sincere gratitude and deep appreciation to my esteemed supervisor, Dr. Assma Bezza, for her valuable guidance, precise feedback, patience, and continuous support throughout this academic journey. Her presence by my side was a strong motivation to persevere and progress.

I am also deeply grateful to everyone who participated in completing the survey for this study, whether I knew them personally or not. Your contribution was an essential foundation for the completion of this work, and I thank you for your time and engagement, especially my sister who strived diligently to distribute it without fatigue or boredom.

A special word of thanks is dedicated to my supportive partner , who played a significant role in my university journey. I thank him for guiding me in choosing the right major and for standing by me at every step. Your support was one of the secrets of my success; I owe you all my gratitude and appreciation.

I cannot but express my deep appreciation to my family, loved ones, friends, and colleagues for their unwavering support, patience, and encouragement, which have been a pillar of strength throughout this academic journey.

I also extend my sincere thanks and gratitude to the distinguished jury members for taking the time to read and evaluate this work. Special thanks go to all the Mathematics and Computer Science teachers for their efforts, whether in teaching and training us until we reached this point.

Thank you all

Abstract

This thesis presents DZ-UniGuide AI, an intelligent academic orientation system designed to assist new Algerian bachelors in selecting suitable university programs. The study begins with the collection and analysis of real student data, consisting of 907 responses that capture academic performance and personal interests. In parallel, a reference database of 1,717 academic programs was build from official Ministry documents, including admission criteria and field descriptions.

Based on this data, a Machine Learning (ML) model using Random Forest was developed to predict students' probability of success, achieving an accuracy of 90.7%. To better understand students' preferences, a Natural Language Processing (NLP) module based on TF-IDF and cosine similarity was implemented to analyze free-text inputs describing their interests. The system also integrates key constraints specific to the Algerian higher education system, including academic eligibility calculated through the official weighted average and geographic eligibility determined by the student's baccalaureate Wilaya.

A bilingual (Arabic/English) web application was developed and deployed on Streamlit Cloud to provide an accessible and interactive user experience. The system then was evaluated using quantitative metrics (precision: 84%, recall: 99%, F1-score: 0.91) as well as user feedback collected through a survey (N = 21). The results show an overall satisfaction rate of 77%, with 85% of users reporting that the interface is easy to use.

This work demonstrates that combining ML and NLP effectively creates a personalized, transparent, and context-adapted orientation tool for the Algerian higher education system.

Keywords : Artificial Intelligence, Machine Learning, NLP, Academic Orientation, Random Forest, Recommendation System, Algeria.

Résumé

Ce mémoire présente DZ-UniGuide AI, un système d'orientation universitaire intelligent conçu pour aider les nouveaux bacheliers algériens à choisir des formations universitaires adaptées. L'étude débute par la collecte et l'analyse de données réelles d'étudiants, soit 907 réponses reflétant leurs performances académiques et leurs centres d'intérêt. Parallèlement, une base de données de référence de 1,717 formations a été dévelopée à partir de documents officiels du Ministère, incluant les critères d'admission et les descriptions des filières.

À partir de ces données, un modèle d'apprentissage automatique (ML) utilisant la méthode Random Forest a été développée pour prédire les chances de réussite des étudiants, avec une précision de 90,7 %. Afin de mieux comprendre leurs préférences, un module de traitement automatique du langage naturel (TALN) basé sur TF-IDF et la similarité cosinus a été implémenté pour analyser les réponses textuelles décrivant leurs intérêts. Le système intègre également les contraintes spécifiques au système d'enseignement supérieur Algérien, notamment l'éligibilité académique calculée à partir de la moyenne pondérée officielle et l'éligibilité géographique déterminée par la wilaya de baccalauréat de l'étudiant.

Une application web bilingue (arabe/anglais) a été développée et déployée sur la plateforme Streamlit Cloud afin d'offrir une expérience utilisateur accessible et interactive. Le système a ensuite été évalué à l'aide de métriques quantitatives (précision: 84%, rappel: 99%, score F1: 0,91) ainsi que par le biais de retours d'utilisateurs recueillis lors d'une enquête (N=21). Les résultats montrent un taux de satisfaction global de 77%, 85% des utilisateurs jugeant l'interface facile à utiliser.

Ce travail démontre que la combinaison de ML et du NLP permet de créer efficacement un outil d'orientation personnalisé, transparent et adapté au contexte pour le système d'enseignement supérieur Algérien..

Mot clés : Intelligence Artificielle, TALN, Apprentissage Automatique, Random Forest, Orientation Académique, Système de Recommandation, Algérie.

ملخص

تُقدّم هذه الأطروحة نظام DZ-UniGuide AI، وهو نظام ذكي للتوجيه الجامعي مصمم لمساعدة خريجي الثانوية العامة الجزائريين الجدد على اختيار البرامج الجامعية المناسبة. تبدأ الدراسة بجمع وتحليل بيانات طلابية حقيقية، وتحديدًا 907 استجابة تعكس أدائهم الأكاديمي واهتماماتهم. بالتزامن مع ذلك، تم تطوير قاعدة بيانات مرجعية تضم 1717 برنامجًا باستخدام وثائق رسمية من الوزارة، بما في ذلك معايير القبول ووصف البرامج.

انطلاقًا من هذه البيانات، تم تطوير نموذج للتعلم الآلي باستخدام طريقة الغابة العشوائية للتنبؤ بفرص نجاح الطلاب بدقة تصل إلى 90.7%. ولتحسين فهم تفضيلاتهم، تم تطبيق وحدة معالجة اللغة الطبيعية (NLP) تعتمد على TF-IDF وتشابه جيب التمام لتحليل الاستجابات النصية التي تصف اهتماماتهم. كما يراعي النظام القيود الخاصة بنظام التعليم العالي الجزائري، ولا سيما الأهلية الأكاديمية المحسوبة من المتوسط المرجح الرسمي والأهلية الجغرافية المحددة من قبل ولاية البكالوريا التي تخرج منها الطالب.

تم تطوير تطبيق ويب ثنائي اللغة (العربية/الإنجليزية) ونشره على منصة Streamlit Cloud لتوفير تجربة مستخدم سهلة وتفاعلية. ثم جرى تقييم النظام باستخدام مقاييس كمية (الدقة: 84%، الاستدعاء: 99%، مقياس F1: 0.91) بالإضافة إلى آراء المستخدمين التي جُمعت عبر استبيان (عدد المشاركين = 21). أظهرت النتائج معدل رضا إجماليًا بنسبة 77%، حيث وجد 85% من المستخدمين أن واجهة المستخدم سهلة الاستخدام.

يُبين هذا العمل أن الجمع بين التعلم الآلي ومعالجة اللغة الطبيعية يُتيح إنشاء أداة إرشادية فعّالة، شخصية وشفافة، ومناسبة للسياق، لنظام التعليم العالي الجزائري.

الكلمات المفتاحية:
الذكاء الاصطناعي، التعلم الآلي، معالجة اللغات الطبيعية، التوجه الأكاديمي، الغابة العشوائية، نظام التوصية، الجزائر.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
API	Application Programming Interface
DB	Database
TALN	Traitement Automatique du Langage Naturel
TF-IDF	Term Frequency - Inverse Document Frequency
ETL	Extract, Transform, Load
CSV	Comma-Separated Values
PDF	Portable Document Format
ROS	Random OverSampling
UAT	User Acceptance Testing
NER	Named Entity Recognition
POS	Part-Of-Speech
RF	Random Forest
GB	Gradient Boosting
TP	True Positive
TN	True Negative
FP	False Positive

FN	False Negative
UI	User Interface
UX	User Experience
HTTP	HyperText Transfer Protocol
URL	Uniform Resource Locator
DL	Deep Learning
KNN	K-Nearest Neighbors
NLG	Natural Language Generation
NLU	Natural Language Understanding
GUI	Graphical User Interface
JSON	JavaScript Object Notation
SQL	Structured Query Language

General Introduction

Context

Higher education in Algeria plays a central role in preparing students for professional and academic careers. It is structured according to the LMD system (License, Master, Doctorate), which aims to standardize university degrees and improve the quality of education. The Ministry of Higher Education and Scientific Research oversees the organization of academic programs, admission criteria, and institutional policies.

In recent years, Artificial Intelligence (AI) has changed the way people make decisions. In the past, AI systems were mainly based on predefined rules. Today, they rely more on data and can learn and adapt to new situations. Modern AI systems can also interact with users in a more natural way. This progress can be seen in areas such as Natural Language Processing (NLP), Machine Learning(ML), and conversational agents. These technologies have made it possible to develop applications that help users make decisions and access information more easily.

Each year, thousands of students obtain the baccalaureate diploma and must choose among a wide range of university programs. However, this decision is often complex due to the diversity of available fields, varying admission requirements, and limited access to clear and personalized guidance. Although the existence of traditional orientation methods such as short information sessions or static documents, are often insufficient to support students in making well-informed decisions. As a result, students may find it difficult to choose a major that matches their abilities and interests. This can lead to low motivation, poor academic performance, or even dropping out.

In recent years, the rapid development of digital technologies and AIhas created new opportunities to improve educational guidance. Intelligent systems can process large amounts of data and provide personalized recommendations, making orientation more efficient and accessible.

One of the most common uses of AI is chatbots. A chatbot is a system that can communicate with users using natural language. It can provide support and answer questions in different domains. Unlike older systems that depend on simple rules or keywords, modern chatbots can understand user intentions and the context of the conversation. This allows them to provide more relevant and useful responses.

Problem Statement

Despite the availability of official information, many new baccalaureate graduates still face significant difficulties when choosing their academic majors. One of the main problems is that these systems are not personalized. They usually provide general information that does not take into account the specific needs of each student. Additionally, They often lack a clear understanding of admission criteria, program content, and future career opportunities. As a result, students may choose fields that do not match their abilities or interests, which can negatively affect their academic success. According to students whose chosen major does not align with their preferences are significantly more likely to drop out of university.

Hence, choosing a major is not a simple decision. It requires considering several factors at the same time, such as admission requirements, program content, location, and personal interests. This situation highlights the need for a more effective and personalized guidance system. An intelligent solution capable of analyzing student profiles and providing personalized recommendations could significantly improve the orientation process and help students make better decisions.

Therefore, it is important to develop a system that can guide students in a more personalized way and help them make better academic decisions.

This research aims to answer the following question:

How can an AI-based chatbot help students in Algeria choose their academic major more effectively?

The development of an intelligent chatbot for academic orientation is highly relevant in the current educational context. First, it improves accessibility by allowing students to obtain guidance at any time and from any location, without relying on limited in-person sessions. Second, it offers scalability, as the system can support a large number of users simultaneously. In addition, the chatbot can provide quick and consistent answers to frequently asked questions, while also delivering personalized recommendations based on each student's profile. By integrating AI techniques, such a system contributes to a more efficient, data-driven, and student-centered approach to university orientation.

Research Objectives

The main goal of this project is to develop a chatbot that can help students choose their academic majors by providing personalized recommendations using AI techniques.

The specific objectives are:

- Identify the challenges students face when choosing a major;
- Collect and analyze data through questionnaires;
- Build student profiles based on interests, preferences, and academic background;

- Develop a recommendation system adapted to student needs;
- Design a chatbot interface that can interact with users and provide guidance;
- Evaluate the performance of the proposed system.

This work is part of the field of AI and aims to contribute to improving decision support systems in education.

Methodological Overview

The methodology of this work starts with data collection and ends with system development and evaluation. First, data is collected from students using questionnaires, including information about their background, interests, and expectations.

Then, the data is organized to build a dataset that can be used for the system. After that, a recommendation approach is developed to suggest suitable majors based on different criteria, such as eligibility and student preferences.

At the same time, an NLP component is implemented to understand user questions and generate appropriate responses. Finally, a chatbot interface is created to allow users to interact with the system in a simple and clear way.

Thesis Organization

This dissertation is organized into four chapters.

The first chapter establishes the theoretical foundations, covering key concepts of Artificial Intelligence and Natural Language Processing relevant to academic orientation systems.

The second chapter reviews existing literature on chatbots and recommendation systems, identifies the limitations of current solutions, and defines the research gap addressed by this thesis.

The third chapter presents the methodology for building the DZ-UniGuide system, including data preprocessing, ETL pipeline, machine learning models, and system architecture.

The fourth chapter describes the implementation of the bilingual web application, evaluates the system using multiple metrics, and presents the results, including a user satisfaction survey and comparative analysis.

Finally, a general conclusion summarizes the main contributions, answers the research questions, discusses limitations, and suggests future work.

PART ONE

Theoretical Background and State of the Art

Chapter 1

Theoretical Background

1.1 Introduction

In recent years, Artificial Intelligence (AI) has emerged as a transformative technology that is reshaping various sectors, including education. The integration of intelligent systems into academic environments has enabled the development of tools that support decision-making, enhance learning experiences, and provide personalized services to students Holmes et al. (2019).

One of the critical challenges faced by students, particularly at the transition from secondary to higher education, is academic orientation. Choosing an appropriate university major is a complex decision that depends on multiple factors such as academic performance, personal interests, career aspirations, and available opportunities. Poor orientation decisions may negatively affect students' motivation, academic success, and future professional paths.

Traditional academic orientation approaches mainly rely on human advisors, static information sources, and general guidelines. While these methods provide useful information, they often fail to consider the individual characteristics and preferences of each student. As a result, there is a growing need for intelligent systems capable of providing personalized and adaptive guidance.

AI offers powerful capabilities to address this challenge. Through data analysis, pattern recognition, and predictive modeling, AI-based systems can generate personalized recommendations tailored to individual student profiles. In particular, Machine Learning techniques enable the prediction of student success in different academic paths, while NLP allows users to interact with systems in a natural and intuitive manner Russell and Norvig (2021).

The combination of AI and NLP has led to the emergence of intelligent conversational agents, commonly known as chatbots. These systems can simulate human-like interaction and provide real-time assistance, making them particularly suitable for academic orientation tasks. By integrating recommendation mechanisms with conversational interfaces, such systems can guide students more effectively throughout the decision-making process.

Several studies have demonstrated that integrating NLP capabilities with recommendation mechanisms into conversational agents can significantly enhance the quality of personalized academic guidance Holmes et al. (2019); Jurafsky and Martin (2020).

This chapter provides the theoretical background necessary to understand the proposed system. It presents the fundamental concepts of AI and NLP, as well as their applications in educational contexts.

The remainder of this chapter is organized as follows. Section 2 introduces AI, including its definitions, approaches, and applications in education. Section 3 presents NLP and its role in conversational systems. Finally, the chapter concludes by highlighting the relevance of these technologies to the proposed academic orientation chatbot.

1.2 Artificial Intelligence

This section will introduce provides an overview of AI, including its definition, historical evolution, main approaches, and applications, with a specific focus on its relevance to educational systems and academic guidance.

1.2.1 Definition of AI

AI refers to a branch of computer science focused on building systems that can perform activities generally requiring human intelligence. Such activities include learning, reasoning, solving problems, perceiving the environment, and making decisions. Russell and Norvig (2021).

AI systems rely on data, algorithms, and computational models to imitate certain human cognitive functions. Advances in computing power, together with the availability of large datasets, have turned AI into a major driver of innovation across many different fields.Holmes et al. (2019).

1.2.2 History and Evolution of AI

The idea of creating intelligent machines dates back to the 1950s. The first known use of the expression "AI" appeared in 1956 at a conference held at Dartmouth College, organized by researcher John McCarthy .Russell and Norvig (2021). Early AI systems were mainly based on symbolic reasoning and rule-based logic. Holmes et al. (2019).

During the 1960s and 1970s, research focused on problem-solving and logical deduction. However, limited computing resources led to two periods known as "AI winters", during which progress slowed down considerably Russell and Norvig (2021).

From the 1990s onward, Machine Learning became a dominant approach. Unlike rule-based systems, machine learning models learn directly from data. A major milestone was reached in 1997 when IBM's Deep Blue defeated world chess champion Garry Kasparov Russell and Norvig (2021).

More recently, the rise of Deep Learning and big data has greatly accelerated AI development. Modern AI systems can now handle complex tasks such as image recognition, speech processing, and natural language understanding Holmes et al. (2019).

1.2.3 Approaches and Types of AI

Different ways of building AI systems can be distinguished based on two complementary perspectives: the **technical approach** (how the system learns or operates) and the **level of intelligence** (the system's scope and capability). These two perspectives are not independent; the choice of a technical approach determines what level of AI can be realistically achieved. Conversely, the desired level of AI influences which technical approach is most suitable.

Technical Approaches

- **Rule-Based Systems**: These operate using predefined logical rules and conditions. They are simple to interpret but cannot adapt to new situations on their own. Such systems typically fall under **Narrow AI** because they are designed for specific, well-defined tasks.
- **Machine Learning (ML)**: ML allows systems to learn patterns from data and improve over time without being explicitly reprogrammed Russell and Norvig (2021). It is usually divided into three main categories:
  - **Supervised Learning**: The model is trained on labeled data, where each input comes with a known output. The goal is to learn a mapping that can predict outputs for new, unseen data. Common tasks include classification and regression Russell and Norvig (2021).
  - **Unsupervised Learning**: The model works with unlabeled data and tries to discover hidden structures or patterns within it. Typical tasks include clustering and dimensionality reduction Russell and Norvig (2021).
  - **Reinforcement Learning**: In this paradigm, a software agent learns by interacting with its environment. Each action results in either a reward or a penalty, and the objective is to maximize cumulative rewards over time Russell and Norvig (2021).
- **Deep Learning (DL)**: A subfield of machine learning based on artificial neural networks with multiple layers. It is particularly effective for complex tasks such as image, speech, and language processing Holmes et al. (2019). Like other ML approaches, deep learning also falls under **Narrow AI**.

ML-based systems are a powerful form of **Narrow AI**, but they can exhibit more adaptive behavior than simple rule-based systems. In this project, supervised learning is primarily used to predict how well a student's profile matches different academic majors.

Levels of AI (by Capability)

While technical approaches describe *how* an AI system works, the levels of AI describe *what* the system can do and how general its intelligence is. These levels are:

- **Narrow AI (Weak AI)**: Designed to perform a single specific task (e.g., recommendation systems, chatbots). *The proposed academic orientation chatbot is a typical example of Narrow AI.*
- **General AI (Strong AI)**: A hypothetical system capable of performing any intellectual task that a human can do. This level has not yet been achieved.
- **Super AI**: A theoretical concept referring to AI that surpasses human intelligence in all domains.

All existing technical approaches (rule-based, machine learning, and deep learning) currently produce only **Narrow AI** systems. General AI and Super AI remain theoretical concepts that future technical approaches may one day achieve. Therefore, in practice, the choice of a technical approach determines the specific capabilities of a Narrow AI system, but no current approach can reach General or Super AI levels.

1.2.4 Applications of AI

AI is widely used in many sectors. According to Russell and Norvig Russell and Norvig (2021), AI applications span diverse domains including healthcare, finance, transportation, and education. Specific examples include:

- **Healthcare**: assisting diagnosis and analyzing medical images Russell and Norvig (2021);
- **Finance**: fraud detection and risk assessment Russell and Norvig (2021);
- **Transportation**: autonomous vehicles Russell and Norvig (2021);
- **Education**: intelligent tutoring systems and personalized guidance Holmes et al. (2019).

In the education sector specifically, AI enables personalized learning paths, automated assessment, and academic advising systems Holmes et al. (2019).

1.2.5 Advantages and Limitations of AI

AI offers several important benefits. According to Russell and Norvig (2021), the main advantages of AI systems include:

- **Automation**: reduces human effort by handling repetitive tasks;

- **Accuracy**: improves decision-making through data-driven analysis;
- **Personalization**: adapts services to individual users Holmes et al. (2019);
- **Scalability**: allows systems to serve a large number of users simultaneously Russell and Norvig (2021).

However, AI also has significant limitations. As discussed by Russell and Norvig (2021), Holmes et al. (2019), and **?**, these limitations include:

- **Data Dependency**: performance depends heavily on data quality and quantity **?**;
- **Lack of Interpretability**: some models, especially deep learning ones, are difficult to explain **?**;
- **Ethical Concerns**: issues such as bias, privacy, and fairness remain challenges **?**;
- **High Computational Cost**: advanced models require substantial hardware resources **?**.

1.2.6 Relevance of AI to the Proposed System

AI is the core technological component of the proposed academic orientation chatbot. It enables the system to process student data, estimate success probabilities, and generate personalized recommendations.

More specifically, in this project, AI is used to:

- analyze student academic profiles and performance records;
- predict the likelihood of success in different academic majors;
- rank and recommend the most suitable university programs;
- adapt recommendations based on user preferences and interactions.

By applying machine learning techniques, the system transforms raw data into meaningful academic guidance. This provides students with informed and personalized decision support based on objective data.

1.3 Natural Language Processing (NLP)

This section introduces NLP, its key techniques and pipeline, and highlights its critical role in enabling conversational systems and enhancing recommendation engines.

1.3.1 Definition of NLP

Natural Language Processing (NLP) represents a specialized branch within AI dedicated to giving computers the ability to work with human language in ways that are useful and contextually appropriate Jurafsky and Martin (2020). Its main goal is to bridge the gap between how people naturally communicate and how machines process information, allowing systems to handle both written and spoken input effectively.

To achieve this, NLP draws upon knowledge from linguistics, computer science, and machine learning. It examines language at multiple levels, including grammatical structure (syntax), word meaning (semantics), and surrounding context Manning et al. (2008). Over time, the field has moved away from early rule-based systems toward modern approaches that rely on statistical Machine Learning and Deep Learning Jurafsky and Martin (2020).

1.3.2 Key Techniques in NLP

According to Jurafsky and Martin Jurafsky and Martin (2020), several fundamental techniques are used in NLP to transform raw text into structured information:

- **Tokenization**: The process of dividing text into smaller units such as words or sentences.
- **Normalization**: Cleaning and standardizing text by removing noise, converting to lowercase, and handling variations.
- **Part-of-Speech Tagging**: Assigning grammatical categories (noun, verb, adjective, etc.) to each word.
- **Named Entity Recognition (NER)**: Identifying important entities such as names, locations, and organizations.
- **Intent Recognition**: Determining the purpose or intention behind user input, which is essential for conversational systems.

These techniques form the foundation for more advanced NLP applications Jurafsky and Martin (2020).

1.3.3 NLP Pipeline

NLP systems typically follow a structured pipeline to transform raw input into meaningful output Jurafsky and Martin (2020). Figure 1.1 illustrates the main stages of this process.

The main stages of the NLP pipeline, as shown in Figure 1.1, include:

- **Text Input**: Receiving user input in natural language.

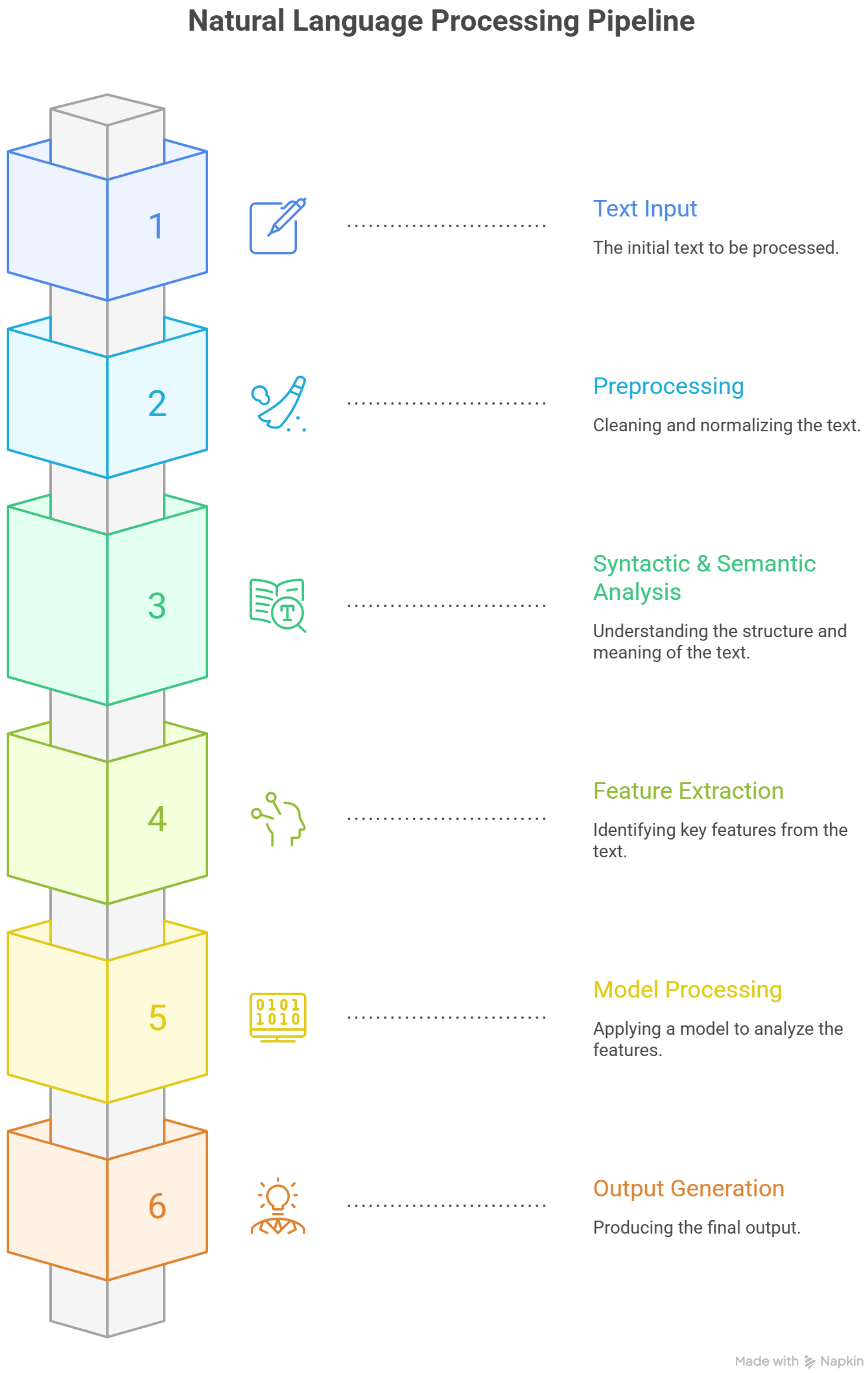

Figure 1.1: Natural Language Processing Pipeline Jurafsky and Martin (2020)

- **Preprocessing**: Applying tokenization and normalization Jurafsky and Martin (2020).
- **Syntactic and Semantic Analysis**: Understanding sentence structure and meaning Manning et al. (2008).
- **Feature Extraction**: Representing text numerically, for example using TF-IDF or word embeddings Jurafsky and Martin (2020).
- **Model Processing**: Using machine learning models to interpret the input.
- **Output Generation**: Producing a response or decision.

Modern NLP systems often rely on advanced techniques such as transformer-based models (e.g., BERT), which allow capturing contextual meaning and relationships between words Devlin et al. (2018); Vaswani et al. (2017).

1.3.4 Importance of NLP in Conversational Systems

NLP plays a central role in conversational systems, particularly chatbots. According to Jurafsky and Martin Jurafsky and Martin (2020), NLP enables systems to understand user input even when expressed in different forms and to generate coherent and context-aware responses.

In academic orientation systems, NLP allows students to express their interests, preferences, and questions in natural language Holmes et al. (2019). For example, when a student states: *"I am interested in programming and AI"*, the system can extract relevant information and match it with suitable academic programs.

1.3.5 NLP for Recommendation Systems

NLP also enhances recommendation systems by linking user input with domain-specific knowledge. According to Manning et al. (2008), analyzing textual descriptions and comparing them with user preferences enables more accurate and personalized recommendations.

This approach is particularly useful in systems where user interests are expressed in free text rather than predefined options Jurafsky and Martin (2020).

Recent advances in NLP are largely driven by transformer-based architectures such as BERT and GPT, which significantly improve language understanding and contextual representation Devlin et al. (2018); Vaswani et al. (2017).

1.3.6 Relevance to the Proposed System

In the proposed academic orientation chatbot, NLP is a key component that enables effective interaction between the user and the system.

Specifically, NLP is used to:

- understand student inputs expressed in natural language;
- extract relevant features such as interests and preferences;
- match user input with academic program descriptions;
- support the recommendation process by providing contextual understanding.

By integrating NLP with Machine Learning-based recommendation models, the system is able to deliver personalized and context-aware academic guidance.

In conclusion, NLP enables machines to understand and process human language, making it a fundamental component of conversational systems. In this project, NLP allows the chatbot to interact naturally with students and enhances the recommendation process by incorporating user preferences expressed in textual form.

1.4 Conclusion

This chapter presented the theoretical foundations necessary for understanding the proposed academic orientation chatbot. It introduced the key concepts of AI and NLP, highlighting their roles in building intelligent and interactive systems.

AI was discussed as the core technology enabling data-driven decision-making, prediction, and personalization. Its evolution, approaches, and applications in education demonstrate its potential to improve academic guidance systems. In parallel, NLP was presented as a fundamental component that enables natural interaction between users and systems, allowing students to express their needs and preferences effectively.

The combination of these technologies provides a powerful framework for developing intelligent academic orientation systems. By integrating Machine Learning techniques with conversational interfaces, the proposed system is capable of delivering personalized, adaptive, and context-aware recommendations.

Overall, this chapter established the conceptual basis of the project and justified the use of AI and NLP in addressing the challenges of academic orientation. This chapter also highlighted the limitations of traditional orientation approaches and justified the need for intelligent, data-driven, and interactive solutions.

The integration of these technologies forms the backbone of the proposed system, enabling both accurate prediction and natural human–computer interaction.

The next chapter presents the state of the art, where existing systems and approaches are analyzed in order to identify their limitations and position the proposed solution within the current research landscape.

Chapter 2

chatbot and Related works

2.1 Introduction

In recent years, Artificial Intelligence (AI) has rapidly evolved, leading to the development of intelligent systems capable of assisting users in various domains, particularly in education. Among these systems, chatbots and recommendation systems have gained significant attention due to their ability to provide automated, efficient, and personalized support Russell and Norvig (2021).

In the educational context, these technologies are increasingly used to help students access information, receive academic guidance, and make informed decisions. Chatbots enable natural interaction between users and systems, while recommendation systems analyze user data to suggest suitable academic paths.

Despite their growing adoption, existing solutions still present several limitations. Most systems focus on a single functionality, either conversational interaction or recommendation, without effectively integrating both. Additionally, many approaches lack personalization and fail to consider local educational constraints.

This chapter presents a state-of-the-art review of chatbots and academic recommendation systems. It discusses their evolution, architectures, and applications, followed by a review of related work. Finally, a comparative analysis is conducted to highlight existing limitations and identify the research gap addressed by the proposed system..

2.2 Chatbots and Conversational Systems

This section focus on discussing the chatbots' definition, Evolution, types and architecture.

2.2.1 Definition of Chatbots

A chatbot refers to a piece of software that mimics real-life conversation by exchanging written messages with users or voice interactions. It allows users to communicate with computer systems in a natural and intuitive way without requiring technical expertise Adamopoulou and Moussiades (2020).

Chatbots are widely used in various domains, including customer service, healthcare, and education. In educational environments, they provide support by answering questions, guiding students, and facilitating access to information.

2.2.2 Evolution of Chatbots

The development of chatbots has evolved significantly over time. The first chatbot, ELIZA, was developed by Joseph Weizenbaum in 1966 and was based on simple pattern matching techniques.

Later systems, such as SmarterChild in the early 2000s, introduced more interactive capabilities by integrating with messaging platforms.

A major breakthrough occurred with IBM Watson in 2011, which demonstrated advanced natural language understanding.

Today, modern chatbots rely on AI, ML and NLP to provide intelligent and context-aware interactions.

2.2.3 Types of Chatbots

Several researchers have proposed different classifications for chatbot systems Jurafsky and Martin (2020); Russell and Norvig (2021):

- **Rule-Based Chatbots**: These chatbots operate using predefined rules and scripts. They follow a decision-tree structure and cannot handle queries outside their programmed knowledge Jurafsky and Martin (2020).
- **AI-Based Chatbots**: These chatbots use Machine Learning and Natural Language Processing to understand user intent and generate dynamic responses in real time Russell and Norvig (2021).
- **Hybrid Chatbots**: These systems combine rule-based logic with AI techniques to balance flexibility and reliability, using rules for common queries and AI for complex ones Holmes et al. (2019).

2.2.4 Chatbot Architecture

A chatbot system is composed of several components that work together to process user input and generate responses Jurafsky and Martin (2020). Figure 2.1 illustrates the general architecture of a typical conversational AI system.

The main components of a chatbot architecture, as shown in Figure 2.1, include:

- **User Interface**: Serves as the entry point through which users interact with the system Jurafsky and Martin (2020).
- **Natural Language Processing (NLP)**: Processes and understands user input, extracting intent and relevant entities Jurafsky and Martin (2020).
- **Dialogue Manager**: Controls the conversation flow and determines system actions based on the current context Jurafsky and Martin (2020).
- **Knowledge Base**: Stores domain-specific information, such as academic programs and student profiles Holmes et al. (2019).

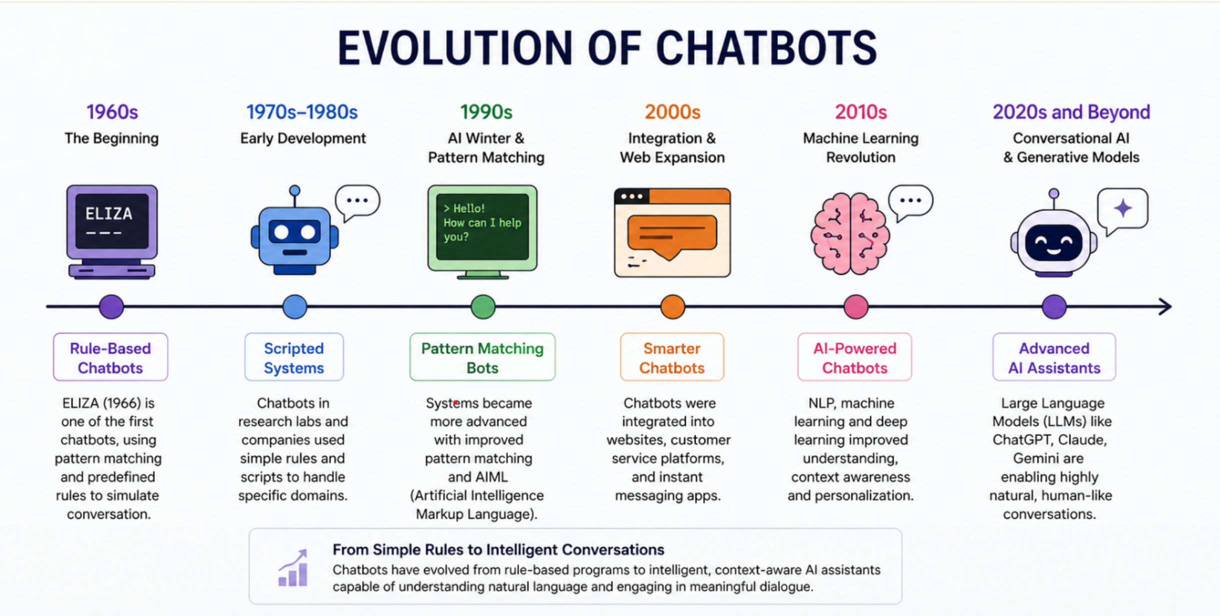

Figure 2.1: Evolution of Chatbots Through Generations

- **Response Generator**: Produces the final response delivered to the user in natural language Jurafsky and Martin (2020).

This architecture presents the general workflow of a conversational chatbot system. It shows how user input is processed through different modules, including NLP, dialogue management, knowledge retrieval, and response generation, to produce an appropriate reply.

2.3 Related Work

This section focuses on the presentation of some related works, discussing their strengthens and weaknesses and then underlying the research gap to be dressed in this work.

2.3.1 Educational Chatbots

The use of chatbots in education has been widely explored in recent years. A comprehensive review conducted by Okonkwo and Ade-Ibijola (2021) highlights that educational chatbots are increasingly used to provide academic support, answer frequently asked questions, and assist students in administrative tasks. These systems improve accessibility and allow students to receive instant responses without human intervention.

Similarly, Adamopoulou and Moussiades (2020) emphasized that chatbots enhance user engagement and reduce the workload of educational staff by automating repetitive interactions.

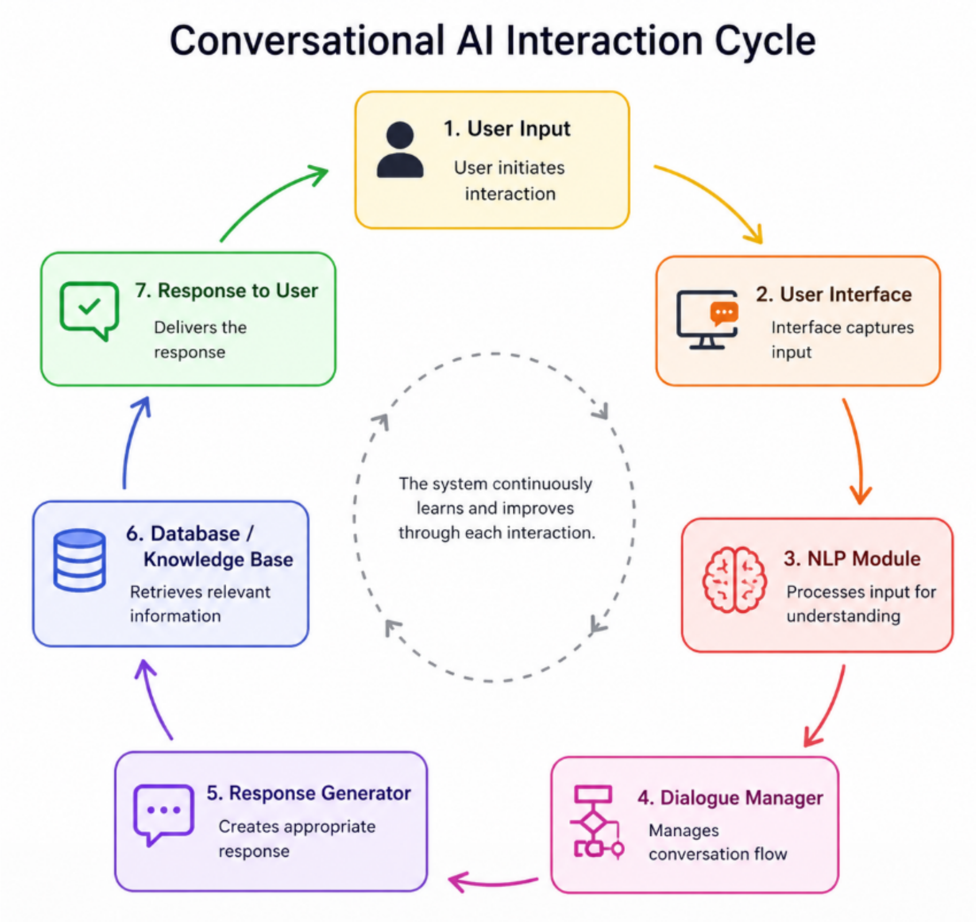

Figure 2.2: General Architecture of a Conversational Chatbot System

Some implementations also integrate basic NLP techniques to improve communication.

However, most educational chatbots remain limited in functionality. They are often based on predefined rules or simple intent recognition models, which restrict their ability to handle complex queries. In addition, they lack personalization and are not designed to support critical decision-making processes such as academic orientation.

2.3.2 Academic Recommendation Systems

Academic recommendation systems have been extensively studied as tools to support students in selecting suitable educational paths. These systems rely on data-driven approaches and Machine Learning techniques.

According to Aggarwal (2016), recommender systems use various algorithms such as collaborative filtering, content-based filtering, and hybrid methods to analyze user preferences and generate recommendations. In the educational domain, these techniques are applied to predict suitable courses or academic programs based on student profiles.

Several studies have applied Machine Learning algorithms such as K-Nearest Neighbors (KNN), Decision Trees, and Neural Networks to improve recommendation accuracy. These systems are effective in processing large datasets and identifying patterns in student behavior.

Despite these advantages, academic recommendation systems suffer from significant limitations. Most systems lack interactive capabilities and operate as static decision-support tools. They do not adapt dynamically to user feedback and often fail to capture complex user preferences. Additionally, many systems are developed using generic datasets, limiting their applicability in specific educational contexts.

2.3.3 Conversational Recommender Systems

To overcome the limitations of traditional systems, recent research has introduced conversational recommender systems that integrate chatbot interaction with recommendation algorithms **??**.

These systems allow users to express their preferences in natural language and receive personalized suggestions through an interactive dialogue. This approach improves user experience and enables a more dynamic recommendation process **?**.

Figure 2.3 illustrates the general architecture of a conversational recommender system. It shows how NLP components interact with recommendation engines to provide personalized suggestions through an interactive dialogue **?**.

However, as highlighted in recent studies, most conversational recommender systems still face several challenges **?**. They often rely on limited datasets, lack deep personalization, and are not adapted to specific local contexts such as national educational systems **?**.

Furthermore, the integration between NLP and recommendation models is not always fully optimized, which affects system performance and accuracy **?**. As a result, there is still a need for

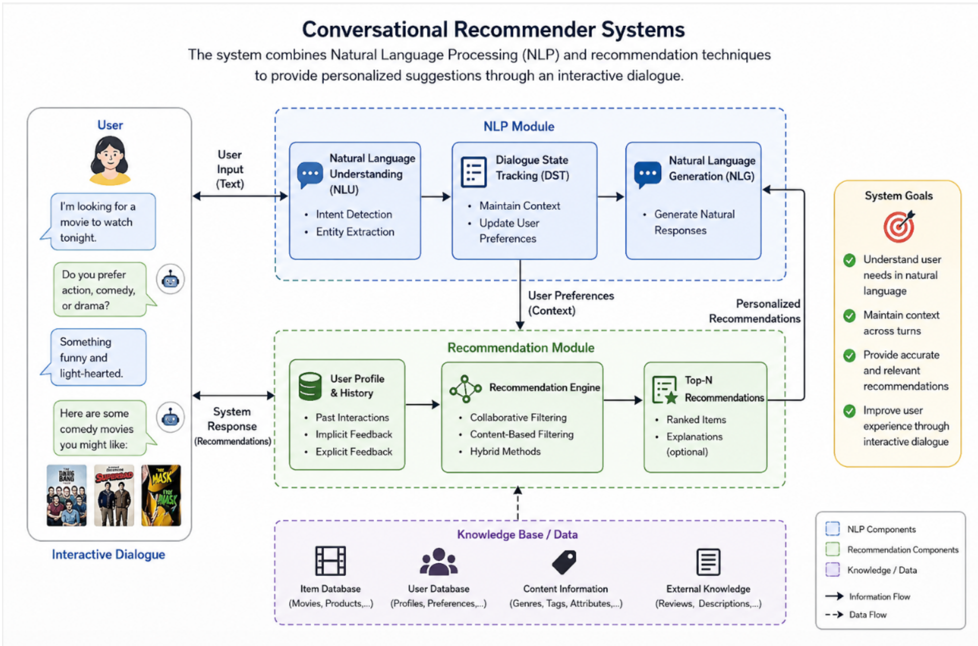

Figure 2.3: Conversational Recommender Systems

more advanced systems that combine interaction, personalization, and contextual adaptation ?.

2.4 Comparative Analysis

The following table presents a comparative analysis of existing approaches in academic orientation systems, based on previous research studies, and highlights the position of the proposed system.

System / Approach	**Strengths**	**Limitations**	**Type**
Educational Chatbots Okonkwo and Ade-Ibijola (2021)	- Provide instant responses to student queries - Improve accessibility to academic information - Reduce administrative workload	- Mainly rule-based interactions - Limited understanding of complex queries - Lack personalization and decision support	Rule-based Systems
Academic Recommender Systems Aggarwal (2016)	- Use Machine Learning algorithms for prediction - Analyze student data and preferences - Provide data-driven recommendations	- Lack interactive communication - Limited adaptability to user feedback - Weak explainability of recommendations	ML-based Systems
Conversational Recommender Systems ??	- Combine interaction with recommendation - Improve user engagement - Enable dynamic recommendation process	- Limited personalization in practice **?** - Often rely on non-local datasets **?** - Weak adaptation to specific contexts **?**	Hybrid Systems **?**
Proposed System	- Integrates chatbot interaction with ML prediction - Uses Random Forest and Gradient Boosting - Based on real dataset (907 responses) - Provides personalized and adaptive recommendations - Considers local academic constraints	- Dependent on dataset quality - Requires periodic model updates - Limited generalization outside context	AI-based Hybrid Intelligent System

The comparative analysis clearly shows that existing systems tend to focus on either interaction or recommendation, but rarely integrate both effectively.

Educational chatbots provide accessibility but lack intelligence and personalization. In contrast, academic recommendation systems offer data-driven suggestions but suffer from the absence of interaction and adaptability.

Conversational recommender systems attempt to combine these aspects; however, they remain limited in terms of personalization and contextual adaptation, especially in specific educational environments.

A unified framework that brings together NLP and Machine Learning models forms the basis of the proposed system, thereby addressing the identified limitations. The system further distinguishes itself by employing a locally gathered dataset, leading to better adaptation to the Algerian educational context. Consequently, the proposed method offers a more thorough, adaptable, and tailored academic orientation solution

2.5 Discussion and Research Gap

The analysis of existing systems reveals several limitations:

First, most solutions separate chatbot interaction from recommendation systems, reducing their overall effectiveness.

Second, personalization remains limited, as many systems do not fully exploit user data.

Third, most approaches ignore local educational contexts, which affects the relevance of recommendations.

These limitations highlight the need for an intelligent system that integrates conversational interaction with personalized recommendation while considering local constraints.

The proposed system addresses this gap by combining NLP, Machine Learning techniques, and a locally collected dataset.

2.6 Conclusion

This chapter reviewed the state of the art in chatbots and academic recommendation systems. It highlighted their evolution, architectures, and applications in education.

The analysis showed that existing systems still suffer from limitations in integration, personalization, and contextual adaptation.

Based on these observations, the proposed system aims to provide a hybrid intelligent solution that combines conversational interaction with data-driven recommendation, offering a more effective and personalized academic orientation tool.

The next chapter presents the Methodology and Data Preparation of the proposed system.

PART TWO

Methodology, System Development, and Evaluation

Chapter 3

Methodology, Data Preprocessing, and Chatbot Design

3.1 Introduction

This chapter presents the complete methodology for building the DZ-UniGuide AI system. It describes the three data sources used in this research: a student survey (907 responses), official ministry PDF documents, and research-derived data. The chapter details the ETL (Extract, Transform, Load) pipeline implemented in Python for data cleaning and integration, followed by the ML training pipeline. The mathematical foundations of the system are also presented, including weighted average calculation, success prediction using Random Forest, cosine similarity for interest matching, and the combined scoring formula. Finally, the system architecture (three-tier: data layer, logic layer, presentation layer), use case diagram, and sequence diagram are presented.

3.2 Data Sources

Before constructing any intelligent decision-support system, a fundamental question must be addressed: **Where does the system's knowledge originate?** The answer lies in data. A chatbot, similar to a human learner, depends entirely on the quality and diversity of the data used to build it. As Goodfellow et al. (2016) states, the performance of learning-based systems is fundamentally bounded by the data they are trained on.

In this research, we have identified and integrated three distinct sources of data, organized into two parallel pipelines as illustrated in Figure 3.1. The first pipeline focuses on constructing a comprehensive reference database of academic programs, while the second pipeline focuses on training a machine learning model to predict student success.

3.2.1 Data Source 1: The Student Survey (Primary Data)

Primary data refers to information collected first-hand by the researcher for a specific investigative purpose Creswell (2014). In our context, we designed and distributed an online questionnaire targeting Algerian university students across different academic levels (Licence, Master, Doctorate). The dataset generated from this survey is saved as `data_set_PFE_m2 (réponses).xlsx`.

The principal objectives of this survey were:

1. To identify the real-world difficulties students encounter when selecting a university major.
2. To determine the most influential factors guiding their decisions (e.g., job market opportunities, personal interests, university reputation, family advice).
3. To compile a rich and diverse collection of student profiles to serve as a reliable training corpus for our machine learning model.

3.2.2 Data Source 2: Research and Experimental Results (Derived Data)

A second critical data source is `BALANCED_MASTER_DATASET.csv`. Unlike the survey data, this file was not directly collected from users but was **constructed empirically** through an iterative process of research, experimentation, and validation. This dataset represents the accumulated knowledge from multiple trials and serves as a foundational reference for building the final master database.

3.2.3 Data Source 3: Official Ministry Documents (Administrative Data)

This category encompasses data extracted from two official publications by the Algerian Ministry of Higher Education and Scientific Research. This data represents the "ground truth" or the codified rules governing the national university orientation system.

Official Source 1: Minimum Admission Grades [1]

This official PDF document provides, for each accredited university major, the minimum weighted average grade required for each high school stream (e.g., Experimental Sciences, Mathematics, Technical Mathematics, Foreign Languages). This data enables the system to perform the first critical filter: **academic eligibility**.

Official Source 2: The Ministerial Circular [2]

This regulatory document details the principle of **geographic distribution**. Within the Algerian higher education system, a student's baccalaureate wilaya (province of examination) directly determines the list of universities and institutions they are permitted to apply to. This data empowers the system to execute the second essential filter: **geographic eligibility**.

3.2.4 Integration: The Final Master Database

The final reference database is the result of a complex ETL (Extract, Transform, Load) process Kimball and Ross (2011). Using custom Python scripts, we merged:

- The research-derived dataset (`BALANCED_MASTER_DATASET.csv`)
- The minimum grades data (extracted from the official PDF)

[1] Eddirasa.com (2025). *2025 Baccalaureate admission grades for Algerian universities.* Available at: https://eddirasa.com/ÙĚØźØŕÙĎØğØł-ØğÙĎÙĆØĺÙĹÙĎ-ÙĎØ■ØğÙĚÙĎÙŁ-ØĺÙČØğÙĎÙĹØśÙŁØğ-2025/

[2] Ministry of Higher Education and Scientific Research, Algeria. (2025). *Ministerial circular relating to the orientation of baccalaureate holders for the 2025-2026 academic year.* Available at: https://services.mesrs.dz/download/bac_2025/circulaire-2025.pdf. Accessed: May 2026.

- The geographic distribution rules (extracted from the ministerial circular)

The output of this integration is `MASTER_DATABASE_FINAL.csv`, which serves as the "single source of truth" for our chatbot's reference data. For each university major, this file contains its official code, full name, offering university, minimum required grades for each baccalaureate stream, and the list of eligible wilayas.

3.2.5 The ML Training Pipeline

In parallel to the database construction, we trained a machine learning model using the survey data (`data_set_PFE_m2 (réponses).xlsx`). The training process involved:

- **Feature extraction:** Encoding categorical variables (baccalaureate stream, influencing factors, academic interests).
- **Target definition:** Predicting whether a student is a Master's student (successful orientation) or not.
- **Model training:** Comparing Random Forest and Gradient Boosting classifiers.
- **Model persistence:** Saving the best model and its associated encoders `best_model.pkl`, `stream_encoder.pkl`, `factor_encoder.pkl`, `major_encoder.pkl`

3.2.6 Application Overview

This section describes how the different components of the system (database, machine learning model, and NLP module) are integrated together within the final Streamlit application.

The final Streamlit application (DZ-UniGuide AI) loads both the master database and the trained model. It uses:

- The database for geographic and academic filtering.
- The model for predicting success rates.
- NLP techniques (TF-IDF and Cosine Similarity) for interest analysis and recommendation generation.

Figure 3.1 provides a complete visual representation of the entire data collection, integration, training, and application pipeline.

Explanation of Figure 3.1:

The figure illustrates the complete global architecture of the data pipeline, which is organized into four sequential stages from left to right:

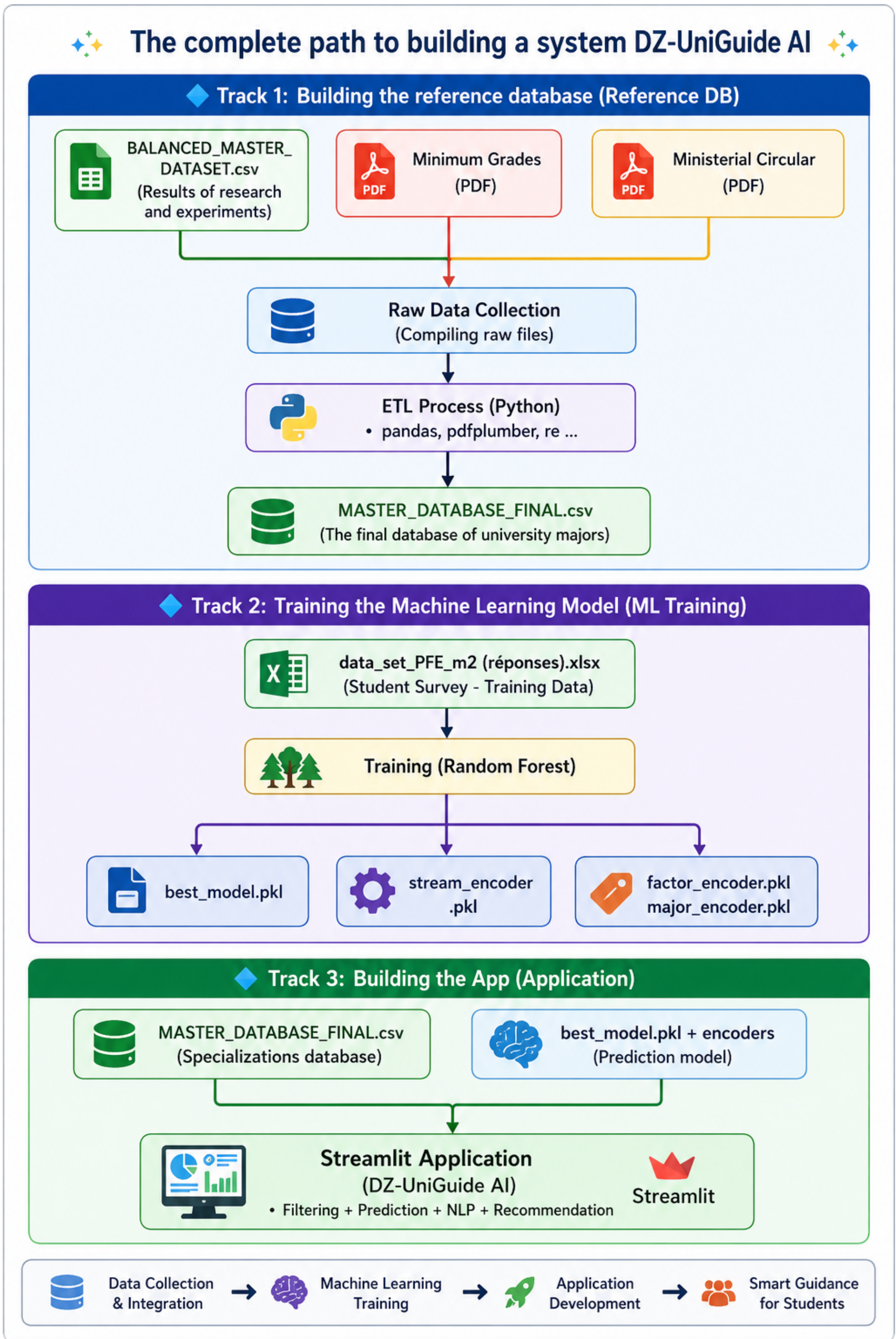

Figure 3.1: Global Data Pipeline: From Raw Sources to the Final Application

1. **Data Sources (Left):** The pipeline starts with three types of raw data sources:
   - *Student Survey:* An Excel file (`data_set_PFE_m2 (réponses).xlsx`) containing 907 responses from Algerian university students.
   - *Official PDF Documents:* Two files from the Ministry of Higher Education (`Moyennes-minimales-BAC-2025.pdf` and `circulaire-2025.pdf`) containing minimum admission grades and geographic distribution rules.
   - *Research-Derived Data:* A CSV file (`BALANCED_MASTER_DATASET.csv`) constructed empirically through multiple trials.
2. **ETL Pipeline (Extract, Transform, Load):** The raw data passes through an ETL pipeline implemented in Python. This stage includes:
   - *Extraction:* Loading Excel/CSV files with Pandas and extracting tables from PDFs using the `pdfplumber` library.
   - *Transformation:* Cleaning text, standardizing formats (e.g., converting wilaya names to standard codes), extracting university names using regular expressions, handling missing values, and merging all sources using the primary key `Code_Fil`.
   - *Loading:* Saving the cleaned and integrated data into a final CSV file.
3. **Two Parallel Outputs (Center-Right):** The ETL pipeline produces two parallel outputs that feed into the final application:
4. **Machine Learning Model:** Using the survey data (907 responses), a Random Forest classifier is trained to predict student success. The model achieves 90.7% accuracy with 99% recall for Master's students.

   The trained model and its associated encoders are saved for later use:

   `%best_model.pkl, stream_encoder.pkl, factor_encoder.pkl, %major_encoder.pkl`
5. **Final Application (Right):** The Streamlit web application (DZ-UniGuide AI) loads both the master database and the trained model. The application provides:
   - Geographic and academic filtering using the database.
   - Success rate prediction using the Random Forest model.
   - Interest analysis using NLP techniques (TF-IDF and Cosine Similarity).
   - A bilingual interface (Arabic/English) deployed on Streamlit Cloud.

This global pipeline demonstrates how raw, heterogeneous data sources are transformed into an intelligent, AI-powered orientation system. The figure also highlights the dual-pathway architecture: one pathway focuses on building the reference database (academic programs), while the other focuses on training the predictive model (student success).

3.3 Data Preprocessing and ETL Pipeline

This section presents the data preprocessing and ETL (Extract, Transform, Load) pipeline implemented in this research. It describes the tools and techniques used to clean, transform, and integrate heterogeneous data sources (Excel files, CSV files, and PDF documents) into a unified master database. The section also details the challenges encountered during data processing and how they were addressed.

3.3.1 Introduction to the ETL Process

After collecting data from heterogeneous sources, the most critical phase begins: **data processing and transformation**. This phase is commonly known as the ETL (Extract, Transform, Load) pipeline Kimball and Ross (2011). The ETL process is responsible for converting raw, unstructured, or semi-structured data into a clean, unified, and machine-readable format that can be efficiently queried by our chatbot.

Step	**Meaning**	**Description**
E	Extract	Retrieving raw data from multiple sources
T	Transform	Cleaning, normalizing, and structuring the data
L	Load	Storing the processed data into a final database

Table 3.1: The Three Phases of the ETL Process

3.3.2 Why ETL is Necessary

The collected data existed in multiple formats:

- Excel file: `data_set_PFE_m2 (réponses).xlsx`
- CSV file: `BALANCED_MASTER_DATASET.csv`
- Two PDF files: `Moyennes-minimales-BAC-2025.pdf` and `circulaire-2025.pdf`

Each file had its own structure, encoding, and level of cleanliness. The ETL phase was essential for:

1. **Standardizing formats:** Converting all data into a single consistent format.
2. **Data cleaning:** Removing errors, duplicates, and handling missing values.
3. **Data integration:** Merging information from all sources into one unified table.

3.3.3 Processing Tools and Technologies

To implement the ETL pipeline, we used the Python programming language with the following libraries:

Library	Purpose
`pandas`	Tabular data manipulation and analysis
`pdfplumber`	Extracting tables and text from PDF files
`re` (Regular Expressions)	Pattern-based text extraction
`numpy`	Numerical operations

Table 3.2: Python Libraries Used in the ETL Process

3.3.4 Detailed ETL Steps

Step 1: Extract

All source files were loaded into the Python environment:

- Excel and CSV files were loaded using `pandas.read_excel()` and `pandas.read_csv()`.
- PDF files were processed using `pdfplumber`, which allowed us to extract tables and text from each page.

Step 2: Transform

This was the most complex step and involved multiple operations:

Step 3: Load

After cleaning and transformation, the final unified dataset was saved as:

- **File name:** `MASTER_DATABASE_FINAL.csv`
- **Format:** Comma-separated values (CSV)
- **Encoding:** UTF-8 with Arabic language support

3.3.5 Challenges Encountered During Data Processing

Several challenges were addressed during the ETL process:

Operation	Description
Text Cleaning	Removing extra spaces, standardizing character cases
Information Extraction	Extracting codes (`Code_Fil`), university names, and major names
Type Conversion	Converting numeric values stored as strings to actual numeric types
Missing Value Handling	Filling missing values with defaults (e.g., "National" for wilayas)
Deduplication	Removing duplicate records to ensure data quality
Table Merging	Joining all sources using primary keys (e.g., `Code_Fil`)

Table 3.3: Transform Operations in the ETL Pipeline

1. **University Name Extraction:** The university name was embedded within the major name string. We developed a custom function (`extract_real_university`) using regular expressions to automatically extract it based on patterns such as "UNIV.", "ENS", "INSTITUT", and "POUR BACHELIERS".

2. **Wilaya Code Standardization:** Different files represented wilayas inconsistently. We standardized all entries to two-digit codes (01 to 58) or the keyword "National".

3. **Grade Conversion:** Minimum grade values were stored as strings (e.g., "10.00"). These were converted to floating-point numbers for numerical comparison.

3.3.6 The ETL Process in Detail

Figure 3.2 illustrates the complete ETL pipeline implemented in this research. The pipeline is organized into three sequential phases: Extract, Transform, and Load.

Phase 1: Extract

During the extraction phase, all source files were loaded into the Python environment:

- The survey responses (907 records) from `data_set_PFE_m2 (réponses).xlsx` were loaded using `pandas.read_excel()`.
- The research-derived dataset (`BALANCED_MASTER_DATASET.csv`) was loaded using `pandas.read_cs`
- The two official PDF documents (minimum grades and ministerial circular) were processed using `pdfplumber.open()` to extract tables and text.

Phase 2: Transform

The transformation phase included four main sub-operations:

1. **Data Cleaning:** Duplicate records were removed, missing values (NaN) were handled by assigning default values, and unnecessary whitespace was eliminated.
2. **Information Extraction:**
   - The official major code (`Code_Fil`) was extracted from each record.
   - University names were automatically extracted from the major name field using regular expressions (regex).
   - Major names were standardized and cleaned.
3. **Type Conversion:** Numeric values stored as strings were converted to floating-point numbers, date formats were standardized, and text fields were normalized.
4. **Merging:** All three data sources were joined into a single unified table using the `Code_Fil` as the primary key.

Phase 3: Load

The final cleaned and integrated dataset was saved as `MASTER_DATABASE_FINAL.csv` with the following specifications:

- **Encoding:** UTF-8 (to support Arabic and French characters)
- **Format:** Comma-separated values (CSV)
- **Size:** 1,717 distinct academic programs

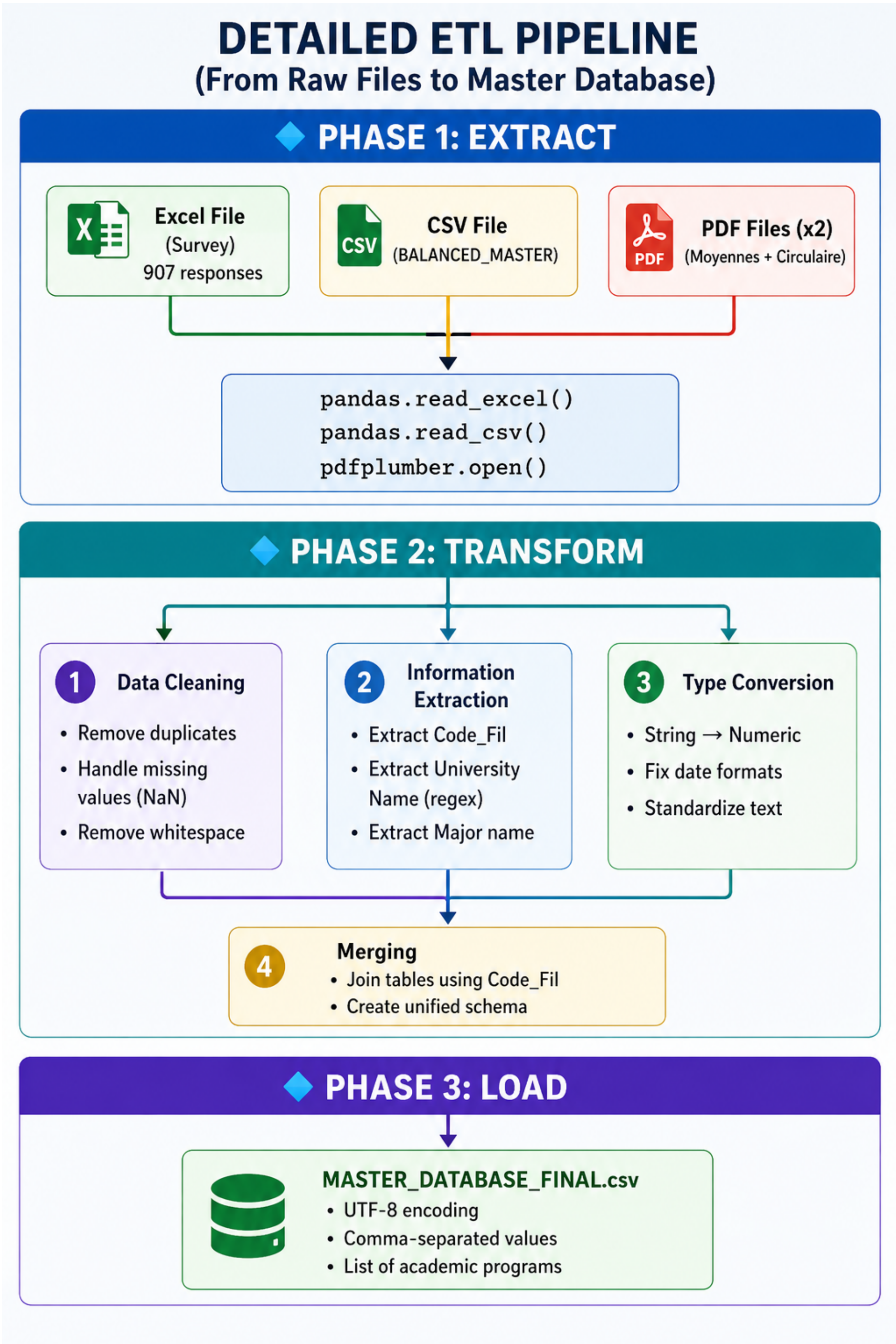

Figure 3.2: Detailed ETL Pipeline: From Raw Files to Master Database

3.4 Machine Learning Training Pipeline

3.4.1 Objective of Model Training

After constructing the reference database (`MASTER_DATABASE_FINAL.csv`), we trained a machine learning model to predict a student's probability of success in a given academic major. The model learns from the survey data, which contains real-world student experiences and outcomes.

3.4.2 Training Dataset

The training data was extracted from `data_set_PFE_m2 (réponses).xlsx`, which contains **907 student responses** from various academic levels (Licence, Master, Doctorate). The following features were used for training:

Feature	Description
`bac_stream`	The student's baccalaureate stream (e.g., Experimental Sciences, Mathematics, Technical Mathematics)
`avg_numeric`	The student's approximate baccalaureate average grade
`factor1`	The most influential factor in choosing a major (e.g., Personal Interests, Job Opportunities)
`major_encoded`	The student's academic interests (encoded as numeric values)

Table 3.4: Features Used for Model Training

The target variable was defined as:

- `target = 1`: The student is currently in a Master's program (indicating a successful orientation choice).
- `target = 0`: The student is not in a Master's program.

3.4.3 Data Preprocessing for Training

Before training, the following preprocessing steps were applied:

1. **Label Encoding:** Categorical variables (`bac_stream`, `factor1`, `academic_interests`) were converted to numeric values using `LabelEncoder` from scikit-learn.

2. **Feature Extraction:** The baccalaureate average was extracted from the text field `bac_average` using regular expressions.

3. **Data Balancing:** The dataset was imbalanced (different numbers of Master and non-Master students). We applied **Random OverSampling (ROS)** using `imbalanced-learn` to balance the classes.

4. **Train-Test Split:** The data was split into 80% for training and 20% for testing, using a random state of 42 for reproducibility.

3.4.4 Models Evaluated

We evaluated two powerful ensemble learning algorithms:

Model	Description
Random Forest	An ensemble of decision trees that reduces overfitting through bootstrapping and feature randomness.
Gradient Boosting	Builds models sequentially, where each new model corrects the errors of the previous one.

Table 3.5: Machine Learning Models Evaluated

Both models were trained with the following hyperparameters:

- **Random Forest:** `n_estimators=200, max_depth=15, random_state=42`
- **Gradient Boosting:** `n_estimators=150, learning_rate=0.1, random_state=42`

3.4.5 Evaluation Results

The trained models were evaluated on the test set (20% of the data). Table 3.6 summarizes the performance metrics:

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score
Random Forest	**90.7%**	84%	99%	0.91
Gradient Boosting	74.4%	72%	75%	0.74

Table 3.6: Comparison of Model Performance

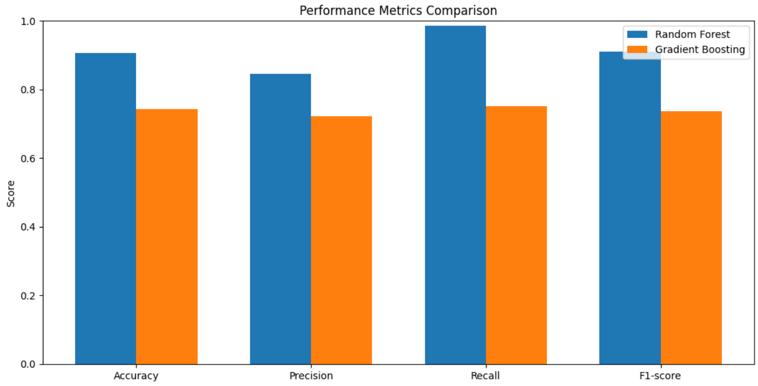

Figure 3.3: performance matrice comparision

3.4.6 Model Selection

Based on the evaluation results, **Random Forest** was selected as the best model for the following reasons:

1. It achieved the highest accuracy (90.7% compared to 74.4% for Gradient Boosting).
2. It demonstrated excellent recall for Master's students (99%), meaning it correctly identifies successful orientation cases.
3. Random Forest is generally more robust to overfitting and handles mixed data types well.

3.4.7 Model Persistence

The selected model and its associated encoders were saved for later use in the Streamlit application:

- `best_model.pkl`: The trained Random Forest model
- `stream_encoder.pkl`: Label encoder for baccalaureate streams
- `factor_encoder.pkl`: Label encoder for influencing factors
- `major_encoder.pkl`: Label encoder for academic interests

Figure 3.4 illustrates the complete machine learning training pipeline.

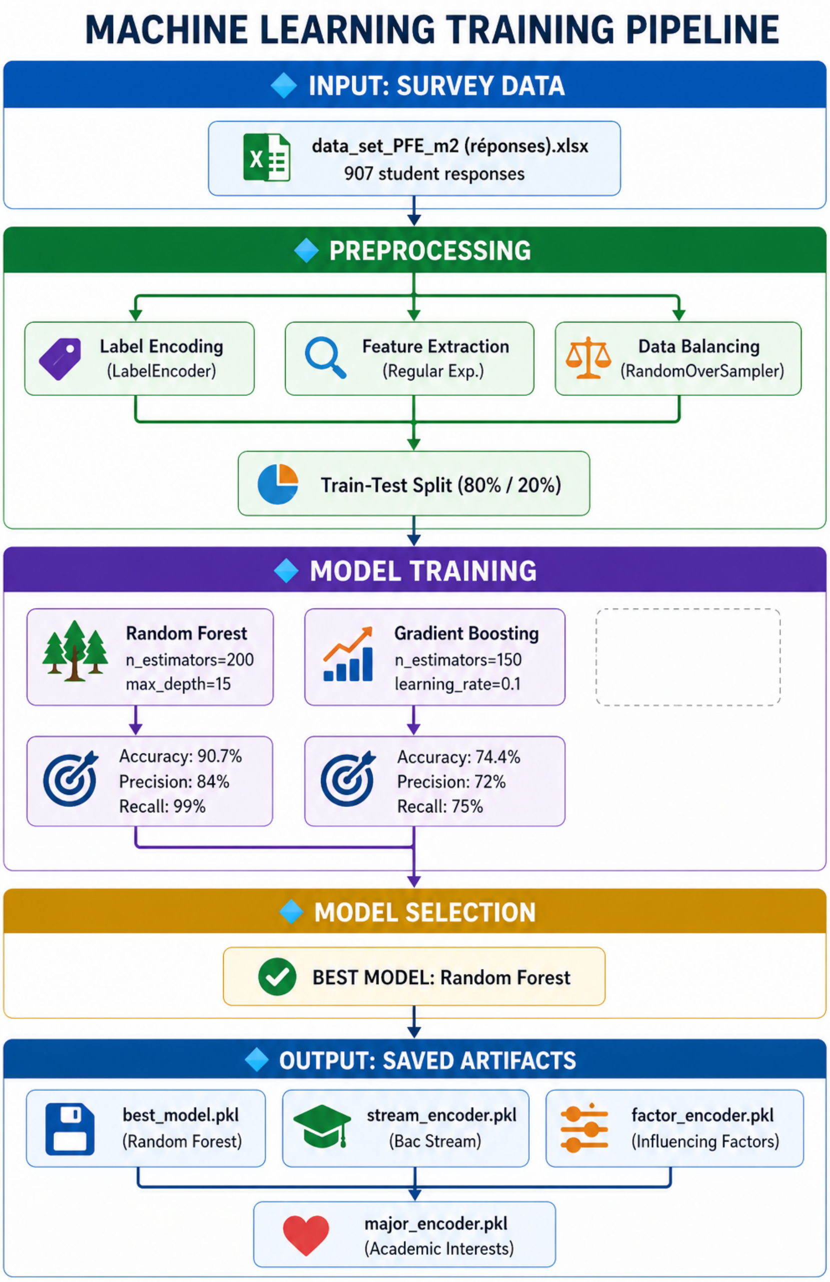

Figure 3.4: Machine Learning Training Pipeline

3.5 Mathematical Foundations and System Architecture

This subsection presents the mathematical foundations underlying the proposed academic orientation system Russell and Norvig (2021). These formulations are essential for accurately computing student scores and predicting success probabilities. The weighted average calculations presented in Equation 3.1 and Equation 3.2 follow the standard formulas used in the Algerian baccalaureate orientation process. For the prediction models, ensemble learning methods such as Random Forest and Gradient Boosting are used to estimate student performance in different academic majors **??**. The following subsections detail the weighted average formulas, the success probability prediction model, and the recommendation scoring function.

3.5.1 Mathematical Formulations

Weighted Average Calculation

The Algerian university orientation system uses a weighted average rather than the general baccalaureate average. The mathematical formulation is:

$$\text{WeightedAvg} = \frac{(\text{BacAvg} \times 2) + \text{SubjectGrade}}{3} \tag{3.1}$$

For the **Technical Mathematics** stream, two core subjects are considered:

$$\text{WeightedAvg}_{\text{Technical}} = \frac{(\text{BacAvg} \times 2) + \frac{(\text{MathGrade}+\text{TechGrade})}{2}}{3} \tag{3.2}$$

Where:

- BacAvg: General baccalaureate average
- SubjectGrade: Core subject grade (Science or Mathematics)
- MathGrade: Mathematics grade
- TechGrade: Technology grade

Success Prediction Formula

The Random Forest model estimates the probability $P(y = 1 \mid X)$, which represents the likelihood that a student is in a Master's program (indicating successful orientation). The general formulation is:

$$\hat{y} = \arg \max_{c \in \{0,1\}} P(y = c \mid X) \tag{3.3}$$

$$\text{SuccessRate} = P(y = 1 \mid X) \times 100 \tag{3.4}$$

Where X is the feature vector:

$$X = [\text{stream\_encoded},\ \text{weighted\_avg},\ \text{factor\_encoded},\ \text{major\_encoded}] \tag{3.5}$$

Cosine Similarity for Interest Matching

To analyze the student's textual interests and match them with academic majors, we employed Cosine Similarity:

$$\text{CosineSimilarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.6}$$

The similarity value is then transformed into an interest score:

$$\text{InterestScore} = \min(55 + \text{similarity} \times 200,\ 98) \tag{3.7}$$

Combined Score

The final combined score integrates both the predicted success rate and the interest match score:

$$\text{CombinedScore} = (\text{SuccessRate} \times 0.7) + (\text{InterestScore} \times 0.3) \tag{3.8}$$

Where:

- SuccessRate: Predicted success probability from the Random Forest model
- InterestScore: Interest match score from NLP analysis
- The weights (0.7 and 0.3) reflect the priority of academic criteria over personal interests

3.5.2 System Architecture

The DZ-UniGuide AI system follows a three-tier architecture, as illustrated in Figure 3.5:

1. **Data Layer:**

- `MASTER_DATABASE_FINAL.csv`: The reference database containing 1,717 academic programs

- `best_model.pkl` and associated encoders: The trained Random Forest model

2. **Logic Layer:**

- Geographic and academic filtering

- Weighted average calculation

- Success rate prediction

- Interest analysis using TF-IDF and Cosine Similarity

- Combined score computation

3. **Presentation Layer:**

- Streamlit web interface

- Interactive charts (Plotly)

- Bilingual support (Arabic/English)

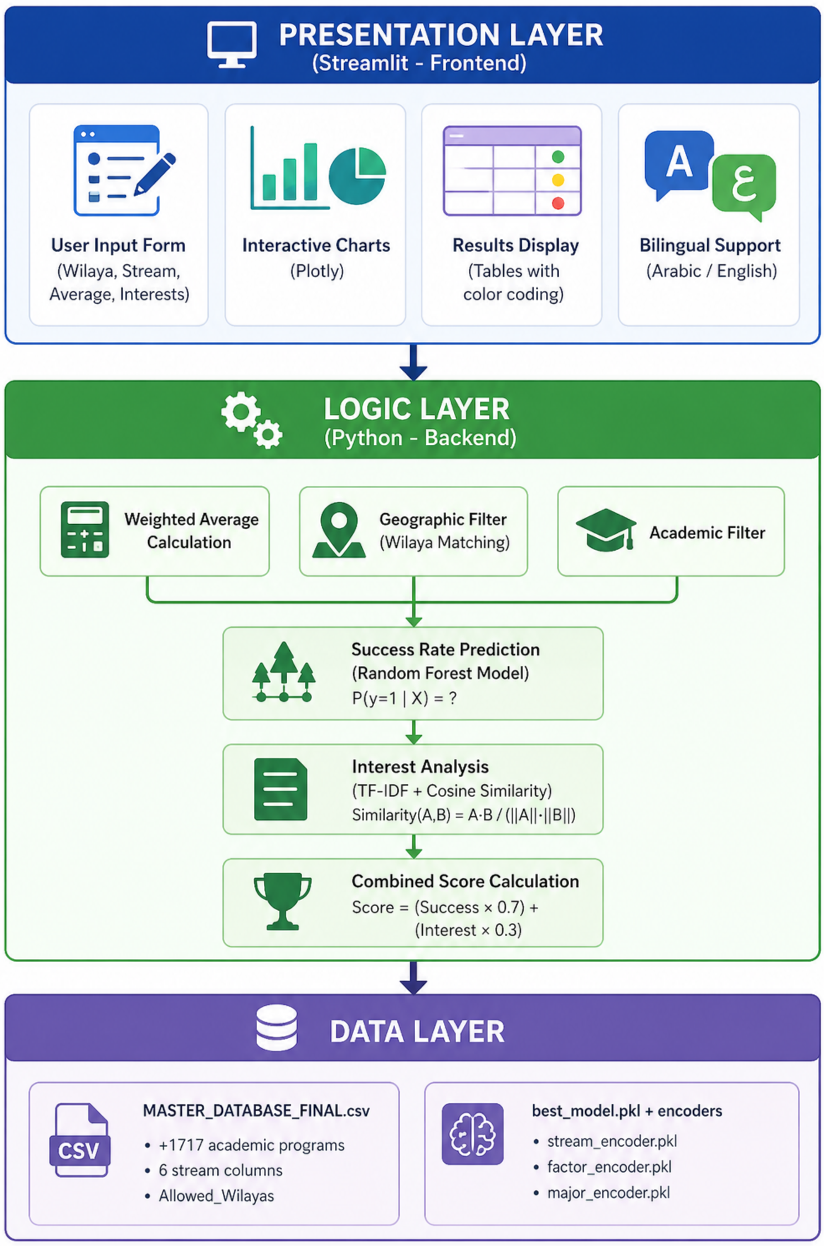

Figure 3.5: Three-Tier System Architecture of DZ-UniGuide AI

3.6 Use Case and Sequence Diagrams

3.6.1 Use Case Diagram

The use case diagram in Figure 3.6 illustrates the interactions between the primary actor (the student) and the DZ-UniGuide AI system. The system offers seven main use cases organized into two phases: the filtering phase and the analysis phase.

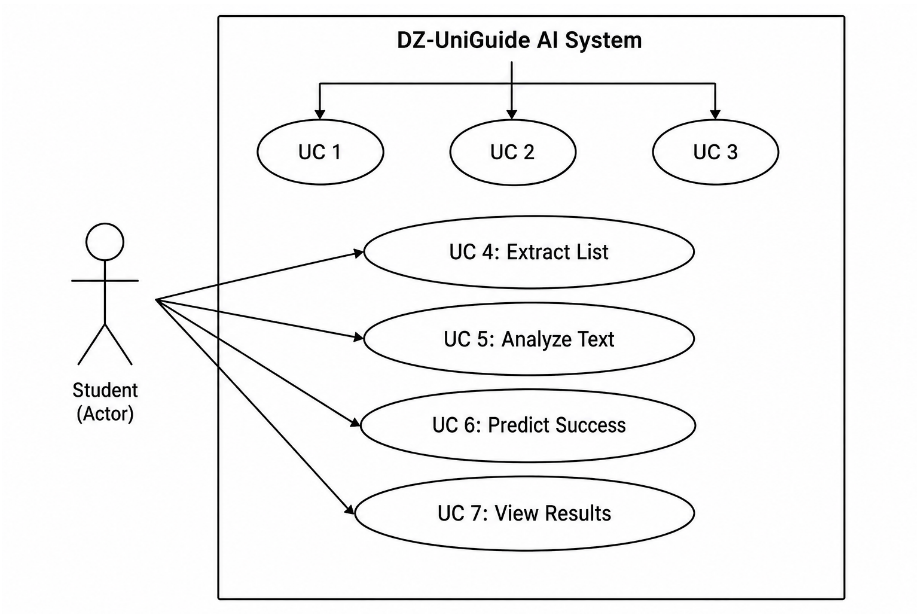

Figure 3.6: Use Case Diagram for DZ-UniGuide AI

3.6.2 Use Case Descriptions

Table 3.7 provides a detailed description of each use case:

ID	Use Case Name	Description
UC-1	Enter Personal Information	The student enters their wilaya, baccalaureate stream, and average category
UC-2	Calculate Weighted Average	The student enters core subject grades to compute the weighted average
UC-3	Select Influencing Factors	The student selects the most important factors (e.g., Job Opportunities, Personal Interests)
UC-4	Extract Eligible Majors	The system filters majors based on geographic and academic eligibility
UC-5	Analyze Interests	The student writes a text describing their interests; the system extracts keywords
UC-6	Predict Success Rate	The Random Forest model calculates the success probability for each major
UC-7	Display Results	The system shows results in tables, charts, and provides recommendations

Table 3.7: Use Case Descriptions

3.6.3 Sequence Diagram

The sequence diagram in Figure 3.7 illustrates the interaction flow when a user clicks the "Analyze Interests" button. The diagram shows the message exchange between six objects over time:

- **User:** The student interacting with the system
- **UI:** The Streamlit frontend interface
- **Logic:** The Python backend logic layer
- **DB:** The Master Database (`MASTER_DATABASE_FINAL.csv`)
- **ML:** The Random Forest model (`best_model.pkl`)
- **NLP:** The text analysis engine (TF-IDF + Cosine Similarity)

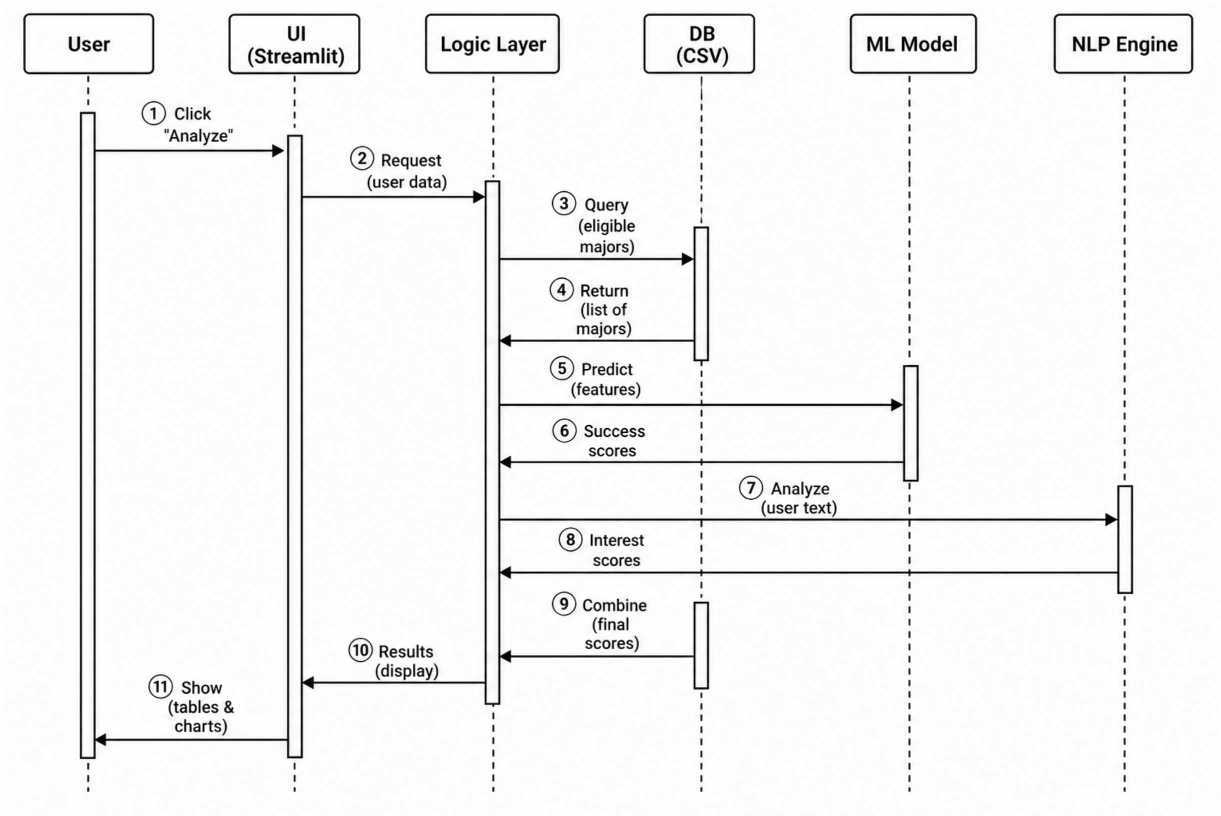

Figure 3.7: Sequence Diagram for the "Analyze Interests" Scenario

The sequence of operations is as follows:

1. **User to UI:** The user clicks the "Analyze Interests" button.
2. **UI to Logic:** The UI sends the user's data (wilaya, stream, average, factors, interests) to the logic layer.
3. **Logic to DB:** The logic layer queries the database for eligible majors based on geographic and academic criteria.
4. **DB to Logic:** The database returns the list of eligible majors.
5. **Logic to ML:** The logic layer sends the feature vector to the Random Forest model for success prediction.
6. **ML to Logic:** The model returns success probability scores for each major.
7. **Logic to NLP:** The logic layer sends the user's interest text to the NLP engine.
8. **NLP to Logic:** The NLP engine returns interest match scores using TF-IDF and Cosine Similarity.

9. **Logic to Logic (internal):** The logic layer combines the scores.
10. **Logic to UI:** The logic layer sends the final results back to the UI.
11. **UI to User:** The UI displays the results in tables and interactive charts.

3.7 Summary and Conclusion

In this chapter, we presented the complete methodology for building the DZ-UniGuide AI intelligent university orientation system. The main contributions of this chapter can be summarized as follows:

Data Collection

We collected data from three primary sources:

- An online survey targeting Algerian university students, yielding **907 responses**.
- Official minimum grade document from the Ministry of Higher Education (PDF).
- Ministerial circular defining geographic distribution rules (PDF).
- Research-derived reference data (`BALANCED_MASTER_DATASET.csv`).

Data Processing and Database Construction

We implemented a complete ETL (Extract, Transform, Load) pipeline using Python:

- Data cleaning: removal of duplicates, handling of missing values, whitespace normalization.
- Information extraction: automatic extraction of codes, university names, and major names using regular expressions.
- Type conversion: converting string values to numeric types.
- Merging: joining all sources using `Code_Fil` as the primary key.
- The final output is `MASTER_DATABASE_FINAL.csv`, containing **1,717 distinct academic programs**.

Machine Learning Model Training

We trained two ensemble learning models on the survey data (907 responses):

- **Random Forest:** Achieved 90.7% accuracy with 99% recall for Master's students.
- **Gradient Boosting:** Achieved 74.4% accuracy.
- The Random Forest model was selected as the best performer.
- The trained model and its associated encoders were saved (`best_model.pkl`, `stream_encoder.pkl`, `factor_encoder.pkl`, `major_encoder.pkl`).

Mathematical Formulations

The following mathematical equations were defined:

- **Weighted Average:**

$$\text{WeightedAvg} = \frac{(\text{BacAvg} \times 2) + \text{SubjectGrade}}{3}$$

- **Success Prediction:**

$$\text{SuccessRate} = P(y = 1 \mid X) \times 100$$

- **Cosine Similarity for Interest Matching:**

$$\text{CosineSimilarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

- **Combined Score:**

$$\text{CombinedScore} = (\text{SuccessRate} \times 0.7) + (\text{InterestScore} \times 0.3)$$

System Architecture

We designed a three-tier system architecture:

1. **Data Layer:** `MASTER_DATABASE_FINAL.csv` and the trained ML model.
2. **Logic Layer:** Python backend implementing filtering, prediction, and NLP analysis.
3. **Presentation Layer:** Streamlit frontend with bilingual support.

- A **Use Case Diagram** was created to illustrate the interactions between the student and the system.

- A **Sequence Diagram** was created to illustrate the message flow when the user clicks "Analyze Interests".

Table 3.8 provides a quantitative summary of the chapter's outputs.

Element	**Quantity**
Survey responses collected	907
Academic programs in master database	1,717
Machine learning models trained	2
Best model accuracy	90.7%
Mathematical equations defined	4
System architecture layers	3
Diagrams created	2

Table 3.8: Quantitative Summary of Chapter 3

This methodology provides a solid foundation for the implementation, evaluation, and results presented in the next chapter.

Chapter 4

Implementation, Evaluation, and Results

4.1 Introduction

After presenting the theoretical foundations in Chapter 2 and the complete methodology, data preprocessing, and system design in Chapter 3, this chapter focuses on the practical implementation and evaluation of the DZ-UniGuide AI system.

The chapter is organized as follows. First, the tools and technologies used to build the system are presented. Second, the system implementation is described, including the Streamlit web application, the integration of the trained Random Forest model, and the NLP-based interest analysis. Third, the testing strategy and evaluation metrics are defined. Finally, the results obtained from both the machine learning model and the user satisfaction survey are presented and discussed.

The main contributions of this chapter are:

- The fully functional bilingual web application deployed on Streamlit Cloud[1]
- The user satisfaction survey (in Arabic) that collected feedback from real users
- A comparative analysis between the proposed system and existing solutions

4.2 Tools and Technologies

The development of the DZ-UniGuide AI system required a combination of programming languages, libraries, frameworks, and deployment platforms. This section presents the main tools used and justifies the reasons for selecting each one.

Table 4.1 provides an overview of all tools and technologies employed in this research.

[1] https://dz-uniguide-ai-cqjebcr2mydg7pektelvdr.streamlit.app/

Table 4.1: Summary of Tools and Technologies Used

Category	Tool	Purpose
Programming Language	Python 3.9	Main development language for data processing, ML, and backend
Data Processing	Pandas NumPy Regular Expressions (re)	Data manipulation, cleaning, and analysis Numerical operations and array handling Pattern-based text extraction from PDFs
PDF Processing	pdfplumber Tabula-py	Extracting tables and text from official PDF documents Alternative PDF table extraction (tested but not retained)
Machine Learning	scikit-learn Random Forest imbalanced-learn	Label encoding, train-test split, model evaluation Classification algorithm (best performer: 90.7% accuracy) Random OverSampling (ROS) for class balancing
Model Persistence	joblib pickle (built-in)	Saving trained models and encoders as .pkl files Alternative serialization (tested but joblib preferred)
NLP	scikit-learn (TF-IDF) Cosine Similarity	Text vectorization for interest analysis Measuring similarity between user interests and major descriptions
Web Framework	Streamlit	Interactive web interface without HTML/CSS knowledge
Web Deployment	Streamlit Cloud	Free hosting of the application
User Feedback	Google Forms	Collecting user evaluations in Arabic

4.2.1 Justification for Tools Selection

Why Python?

Python was chosen as the primary programming language for several reasons. First, it offers a rich ecosystem of libraries for data science and machine learning Raschka et al. (2020). Second, its syntax is clean and readable, which accelerates development and debugging. Third, Python has a large community, making it easy to find solutions to technical problems. Finally, Python integrates seamlessly with Streamlit, which is also Python-based.

Why Pandas and NumPy for Data Processing?

Pandas was selected because it provides powerful data structures, particularly the DataFrame, which simplifies handling of tabular data McKinney (2010). The survey responses and the master database were both stored as tabular data, making Pandas the natural choice for operations such as filtering, merging, and cleaning.

NumPy complements Pandas by providing efficient numerical operations, which are essential for calculating weighted averages and handling arrays of feature vectors Oliphant (2006).

Why pdfplumber for PDF Extraction?

Official documents from the Algerian Ministry of Higher Education were provided in PDF format. Two Python libraries were considered: `PyPDF2` and `pdfplumber`.

`pdfplumber` was chosen because it:

- Extracts tables more accurately than PyPDF2
- Preserves the structure of the original document
- Allows extraction of text from specific bounding boxes
- Handles Arabic and French characters correctly

Why Random Forest over Gradient Boosting?

Both Random Forest and Gradient Boosting were evaluated during the training phase (see Chapter 3, Section 3.5). Random Forest was selected for the final system because:

1. It achieved higher accuracy (90.7% vs 74.4%)
2. It demonstrated excellent recall (99%) for Master's students
3. Random Forest is more robust to overfitting
4. It handles mixed data types (categorical and numerical) well without extensive preprocessing

Why Streamlit for the Web Interface?

Several web frameworks were considered for building the user interface, including:

- **Django:** Powerful but complex and requires extensive boilerplate code

- **Flask:** Lightweight but requires separate frontend development (HTML, CSS, JavaScript)
- **Streamlit:** Minimal code required, no frontend expertise needed, ideal for data science applications

Streamlit was chosen because Teymourzadeh et al. (2021):

- Allows rapid prototyping and development
- Integrates directly with Python and scikit-learn
- Provides built-in widgets (buttons, sliders, text inputs)
- Supports interactive charts (Plotly, Matplotlib)
- Requires no knowledge of HTML, CSS, or JavaScript

Why Streamlit Cloud for Deployment?

Streamlit Cloud was selected as the deployment platform because it:

- Offers free hosting for public applications
- Integrates directly with GitHub repositories
- Automatically updates the application when changes are pushed to GitHub
- Provides HTTPS by default (secure connection)

The application is accessible at: https://dz-uniguide-ai-cqjebcr2mydg7pektelvdr.streamlit.app/

Why Google Forms for User Feedback?

To collect user evaluations, Google Forms was chosen because it:

- Is free and easy to create
- Supports Arabic language
- Allows anonymous responses
- Automatically compiles results into a spreadsheet
- Can be shared via a simple link

The Arabic-language survey is accessible on Google Forms[2]

[2] https://docs.google.com/forms/d/e/1FAIpQLScdKENzdWqH1fyietdboz2Irnnbu2oZqRaCbOg7qVEy viewform

Why GitHub for Version Control and Deployment?

GitHub was used as the central repository for the project's source code. The reasons for this choice include:

- **Version Control:** GitHub tracks all changes made to the code, allowing rollback to previous versions if needed.
- **Integration with Streamlit Cloud:** Streamlit Cloud deploys applications directly from GitHub repositories. Any update pushed to the `main` branch automatically triggers a new deployment.
- **Collaboration:** GitHub facilitates sharing the code with supervisors and evaluators.
- **Open Access:** The repository can be made public, allowing other researchers to inspect and reuse the code.

The complete source code of the DZ-UniGuide AI system is available in the GitHub repository associated with this research. The repository contains:

- The training script (`train_FINAL.py`)
- The Streamlit application script (`streamlit_app.py`)
- The trained model and encoders (`.pkl` files)
- The master database (`MASTER_DATABASE_FINAL.csv`)
- A `requirements.txt` file listing all dependencies

The connection between GitHub and Streamlit Cloud is illustrated in Figure 4.1.

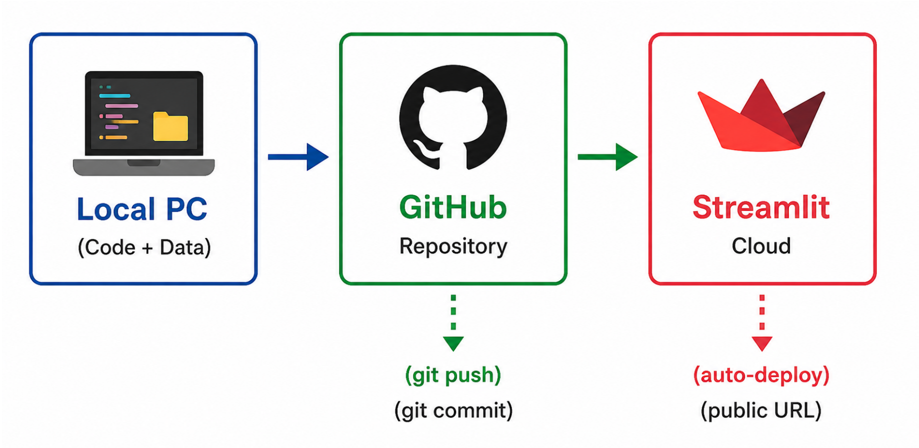

Figure 4.1: Deployment Workflow: GitHub to Streamlit Cloud

4.3 System Implementation

This section describes the practical implementation of the DZ-UniGuide AI system. Following the methodology presented in Chapter 3, the implementation was carried out in three main phases: (1) data processing and model training, (2) Streamlit application development, and (3) NLP integration for interest analysis.

4.3.1 Implementation of Data Processing and Model Training

The first phase involved implementing the training pipeline described in Section 3.4. Algorithm 1 presents the complete Python code structure used for training.

Algorithm 1 Python Implementation of the Training Pipeline

```
Import libraries: pandas, sklearn, imblearn, joblib
Load data: df = pd.read_excel('data_set_PFE_m2 (réponses).xlsx')
Encode categorical variables:
    le_stream = LabelEncoder()
    df['bac_stream_encoded'] = le_stream.fit_transform(df['bac_stream'])
Extract numeric average:
    df['avg_numeric'] = df['bac_average'].str.extract(r'(\d+)').astype(float)
Create target variable:
    df['target'] = df['academic_status'].apply(lambda x: 1 if 'Master' in str(x) else 0)
Balance classes using ROS:
    ros = RandomOverSampler(random_state=42)
    X_resampled, y_resampled = ros.fit_resample(X, y)
Split data: X_train, X_test, y_train, y_test = train_test_split(..., test_size=0.2)
Train Random Forest:
    rf = RandomForestClassifier(
      n_estimators=200, max_depth=15, random_state=42
    )
    rf.fit(X_train, y_train)
Save model and encoders:
    joblib.dump(rf, 'best_model.pkl')
    joblib.dump(le_stream, 'stream_encoder.pkl')
```

The complete training script, named `train_FINAL.py`, was executed on a local machine. The execution output confirmed successful training:

```
Random Forest Results:
Accuracy: 0.907051282051282
```

```
              precision    recall  f1-score   support
           0       0.99      0.83      0.90       163
           1       0.84      0.99      0.91       149

    accuracy                           0.91       312

 Best Model:: Random Forest
 The model and all encoders were successfully saved!
```

4.3.2 Streamlit Application Development

General Interface and Bilingual Support

The DZ-UniGuide AI application features a bilingual interface that supports both English and Arabic. Users can switch between languages at any time using the language toggle button in the sidebar. Figure 4.2 shows the main interface of the application.

Figure 4.2: Main Interface of DZ-UniGuide AI Showing Bilingual Support

Welcome Page

After selecting the preferred language (Arabic or English), the user is greeted with a welcome page that congratulates them on passing the baccalaureate exam and provides motivational guidance for choosing a university major. Figure 4.3 presents this interface.

Figure 4.3: Welcome Page of DZ-UniGuide AI (Arabic Interface)

As shown in Figure 4.3, the welcome page includes:

- A congratulatory message for baccalaureate success.
- Motivational advice emphasizing that major selection is more important than the grade itself.
- Key factors to consider: personal interests, professional future, and real capabilities.
- A call-to-action button "Start Your Journey Now" to proceed to the information input page.

The content automatically switches between Arabic and English based on the user's language selection.

Page 1: Student Information Input

After clicking "Start Your Journey Now", the user is directed to the information input page. This page collects all necessary data to provide personalized orientation recommendations. Figure 4.4 shows this interface.

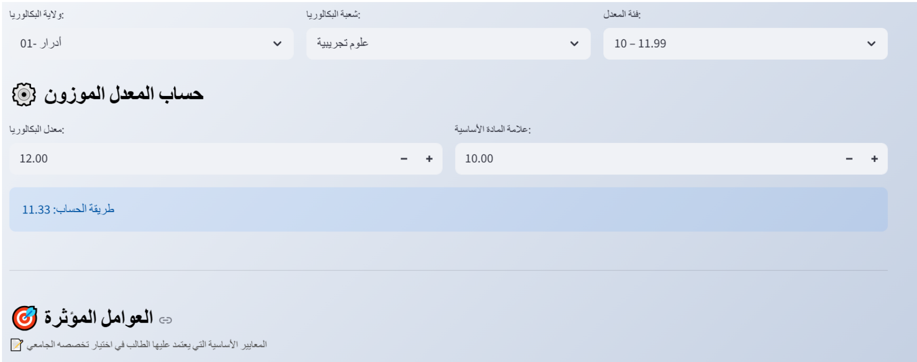

Figure 4.4: Student Information Input Page (Arabic Interface)

As illustrated in Figure 4.4, the input page is organized into four main sections:

1. **Basic Information:** The user selects:
   - Wilaya (province) of the baccalaureate examination from a dropdown list (01 to 58).
   - Baccalaureate stream (e.g., Experimental Sciences, Mathematics, Technical Mathematics, Foreign Languages).
   - Grade category (e.g., 10–11.99, 12–13.99, 14–15.99, 16+).
2. **Weighted Average Calculation:** The user enters:
   - Baccalaureate average grade.
   - Core subject grade (Science for Experimental Sciences, Mathematics for Mathematics stream, etc.).
   - The system automatically calculates the weighted average according to the official Algerian formula:

$$\text{WeightedAvg} = \frac{(\text{BacAvg} \times 2) + \text{SubjectGrade}}{3}$$

3. **Influencing Factors:** The user selects the most important factors guiding their orientation choice (e.g., Personal Interests, Job Opportunities, University Reputation, Family Advice). A second optional factor can also be selected.
4. **Action Button:** A "Next" button to proceed to the interests analysis page.

All input fields are validated to ensure data quality before proceeding to the next step.

After the student submits their basic information (wilaya, baccalaureate stream, average grade, and weighted average), the system displays a table of all eligible majors based on two main criteria:

1. **Academic Eligibility:** The student's weighted average must meet or exceed the minimum required grade for each major, as specified in the official ministry documents.

2. **Geographic Eligibility:** The student's wilaya (province of examination) determines which universities and institutions they are permitted to apply to, according to the ministerial circular on geographic distribution.

This dual-filtering mechanism ensures that only realistic and admissible options are presented to the student. For example, paramedical majors (such as nursing, physiotherapy, and medical imaging) have significantly different minimum grade requirements depending on the region:

- In **southern Saharan wilayas** (e.g., Adrar, Tamanrasset, Illizi), the minimum required weighted average for paramedical majors can be as low as **11**.
- In **northern coastal wilayas** (e.g., Algiers, Oran, Constantine), the same paramedical majors may require a weighted average of up to **16** due to higher competition and demand.

This geographic disparity highlights the importance of integrating location-based filtering in the system. A student from a southern wilaya may be eligible for a paramedical major with an average of 11, while a student from a northern wilaya with the same average would not be eligible. The system automatically applies these geographic rules, ensuring fair and accurate recommendations based on the student's actual situation.

Page 2: Interests Analysis (NLP)

After the student has viewed the eligible majors based on academic and geographic criteria, the system proceeds to the interests analysis page. This page allows the student to express their academic and professional interests in natural language, as shown in Figure 4.5.

Figure 4.5: Interests Analysis Page (Arabic Interface)

As illustrated in Figure 4.5, the interface provides:

1. **User Guidance Tips:** The system displays helpful tips to encourage the student to write a detailed description:

- Write a detailed text about your interests.
- Mention the fields you like.
- Talk about your hobbies.

2. **Text Input Area:** A large text box where the student can describe their interests in free text. For example, the student might write: *"I am interested in mathematics, logical problem-solving, programming, and artificial intelligence."*

3. **Advanced Prediction System:** The system uses Natural Language Processing (NLP) techniques to analyze the input text:

   - **TF-IDF Vectorization:** Converts the text into numerical feature vectors.
   - **Cosine Similarity:** Measures the similarity between the student's interests and the descriptions of academic majors stored in the master database.
   - **Interest Score:** Transforms the similarity value into an interest score between 55 and 98 using the formula:

$$\text{InterestScore} = \min(55 + \text{similarity} \times 200,\ 98)$$

4. **Analyze Button:** A button that triggers the NLP analysis and generates personalized recommendations based on both the predicted success rate (70%) and the interest match score (30%).

Page 3: Results and Recommendations Table

After the NLP analysis is complete, the system displays a comprehensive results table showing all eligible majors ranked by relevance. Figure 4.6 presents this interface, which includes both the data table and the interactive toolbar above it.

As shown in Figure 4.6, the table provides the following columns:

- **Index:** Ranking number of the recommendation.
- **Code:** Official code of the academic major (`Code_Fil`).
- **Major Name:** Full name of the academic program.
- **University:** Name of the university offering the major.
- **Weighted Average:** Minimum required weighted average for admission.
- **Interest Score:** Match percentage between the student's described interests and the major (calculated using TF-IDF and Cosine Similarity).
- **Predicted Success Rate:** Probability of success predicted by the Random Forest model (90.7% accuracy).

	الرمز	التخصص	الجامعة	المعدل المطلوب	% نجاح	% اهتمام	% الإجمالي
3222	N04LAL01	URBANISME	المركز الجامعي المختص	10.02	71.6%	55.0%	66.6%
664	None	GEOGRAPHIE ET AMENAGEMENT DU TERRITOIRE (FORMATIONS PROFESSIONNALISANTES	المركز الجامعي المختص	10.02	71.6%	55.0%	66.6%
2015	None	MATHEMATIQUES (FORMATIONS PROFESSIONNALISANTES)	المركز الجامعي المختص	10.01	71.6%	60.0%	68.1%
796	A12LAL03	HYDRAULIQUE	المركز الجامعي المختص	10.01	71.6%	55.0%	66.6%
1952	F03LPL02	MANAGEMENT ELECTRONIQUE DES AFFAIRES	المركز الجامعي المختص	10.01	71.6%	55.0%	66.6%
1077	None	KINESITHERAPEUTES DE SANTE PUBLIQUE -- POUR BACHELIERS DJANET	DJANET	10.01	71.6%	55.0%	66.6%
2138	None	MATHEMATIQUES ET INFORMATIQUE APPLIQUEES AUX SCIENCES HUMAINES ET SOCIALES	المركز الجامعي المختص	10.00	71.7%	60.0%	68.2%
2132	None	MATHEMATIQUES APPLIQUEE + SCIENCES ECONOMIQUES	المركز الجامعي المختص	10.00	71.7%	60.0%	68.2%
2014	C02LAL02	MATHEMATIQUES	المركز الجامعي المختص	10.00	71.7%	60.0%	68.2%
2568	D03LAN01	SCIENCES AGRONOMIQUES	المركز الجامعي المختص	10.00	71.7%	55.0%	66.7%

Figure 4.6: Results and Recommendations Table with Interactive Toolbar (Search, Column Control, Export, and Full-Screen Options)

- **Combined Score:** Weighted average of the two scores:

$$\text{CombinedScore} = (\text{SuccessRate} \times 0.7) + (\text{InterestScore} \times 0.3)$$

The interactive toolbar located above the table includes the following features:

- **Search Box :** Allows the student to filter the table in real time by typing keywords (e.g., university name, major name, or code). The table dynamically updates to show only rows matching the search query.
- **Column Visibility Control :** Opens a dropdown menu allowing the student to select which columns to display or hide. For example, a student may choose to hide the "Code" column or show only "Major Name" and "Combined Score".
- **Download/Export Button :** Allows the student to download the results table as a CSV file for offline reference, which is useful for sharing with families or academic advisors.
- **Expand/Full-Screen Button :** Expands the table to full-screen mode for better readability, especially on mobile devices or small screens.

These interactive features are implemented using Streamlit's built-in `st.dataframe()` with custom column configuration, combined with additional widgets for search and export functionality.

Page 4: Data Visualization and Analytics

In addition to the detailed results table, the system provides interactive data visualization charts to help students better understand and compare the recommended majors. Figure 4.7 presents an example of a bar chart showing the predicted success rates for available majors.

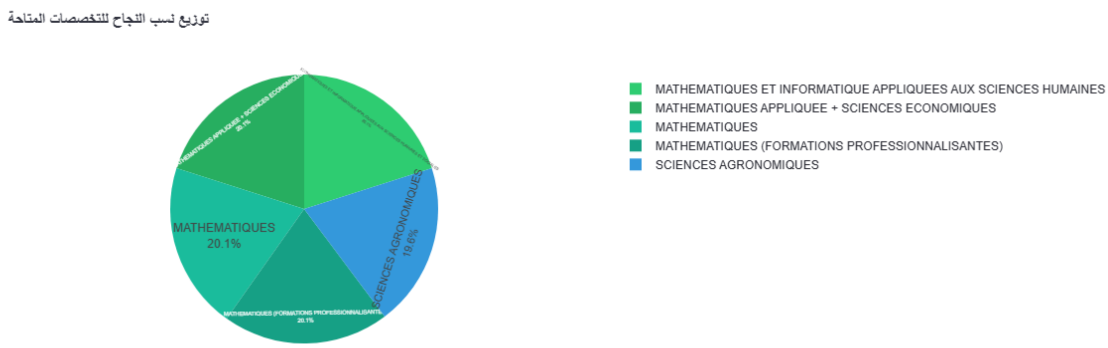

Figure 4.7: Interactive Bar Chart Showing Predicted Success Rates by Major

As shown in Figure 4.7, the visualization includes the following features:

- **Bar Chart Display:** Each bar represents a specific academic major, with the height indicating the predicted success rate (percentage). This allows students to quickly identify which majors offer the highest probability of success.
- **Interactive Toolbar:** Located above the chart, the toolbar provides several interactive options:
  - **Zoom (+ / -):** Allows the student to zoom in and out of specific areas of the chart for detailed examination.
  - **Pan :** Enables moving across the chart when zoomed in.
  - **Download as PNG :** Allows the student to save the chart as an image file for offline reference or inclusion in reports.
  - **Reset Axes :** Resets the chart to its original view after zooming or panning.
  - **Hover Tool:** Displays exact values when the student hovers over any bar, showing the major name and its exact success rate.

The visualization is implemented using Plotly, a powerful interactive graphing library that integrates seamlessly with Streamlit. This interactive approach enables students to explore the data dynamically, making it easier to compare different majors and make informed decisions.

The combination of the detailed results table and the interactive visualization provides students with two complementary ways to analyze the recommendations:

- The **table** offers precise numerical values and the ability to search, filter, and export data.
- The **chart** offers a visual overview that makes it easy to compare success rates across majors at a glance.

Page 5: Smart Recommendation System (Similar Majors)

In addition to the primary recommendations based on the student's profile and interests, the system offers a complementary smart recommendation feature that suggests similar majors to a selected field of study. This feature is particularly useful when a student is interested in a specific major but wants to explore alternative or related options.

Figure 4.8 presents this interface.

Figure 4.8: Smart Recommendation System Showing Similar Majors

As illustrated in Figure 4.8, the interface consists of:

1. **Major Selection Dropdown:** A dropdown menu allowing the student to select a specific academic major of interest (e.g., "SCIENCES DE LA NATURE ET DE LA VIE").
2. **Similar Majors Table:** After selecting a major, the system displays a table of recommended similar majors. The table includes:
   - Major name
   - University offering the major
   - Similarity score (calculated based on shared characteristics, required streams, and academic domains)
   - Predicted success rate for the student's profile

How Similarity is Calculated:

The similarity between majors is determined using a combination of factors:

- **Academic Domain:** Majors belonging to the same scientific field (e.g., Life Sciences, Computer Science, Engineering) receive higher similarity scores.
- **Baccalaureate Stream Requirements:** Majors that accept the same baccalaureate streams are considered more similar.

- **Keyword Matching:** Major descriptions are analyzed using TF-IDF vectorization and Cosine Similarity to identify semantically related programs.

This feature helps students discover alternatives they may not have initially considered, expanding their orientation options while maintaining relevance to their initial interests.

Page 6: Personal Analysis Summary

After generating the recommendations, the system displays a personal analysis summary card that provides the student with a quick overview of their orientation potential. This summary is shown alongside the results table and visualization charts.

The summary card includes the following key metrics:

- **Best Match (Best Match):** The highest combined score achieved among all recommended majors, expressed as a percentage (e.g., 68.2%). This indicates the best-fitting major for the student's profile and interests.
- **Success Rate (Success Rate):** The predicted probability of academic success based on the Random Forest model, averaged across the top recommendations (e.g., 71.7%).
- **Interest Compatibility (Interest Match):** The average interest match score between the student's described interests and the recommended majors, calculated using TF-IDF and Cosine Similarity (e.g., 60.0%).
- **Number of Options (Options Count):** The total count of eligible majors that match the student's geographic location, academic stream, and weighted average (e.g., 56 options).
- **Required Weighted Average (Required Grade):** The minimum weighted average needed for the best-matching major (e.g., 10.03), compared to the student's actual baccalaureate average (e.g., 11.33).
- **Smart Tip (Smart Advice):** A dynamic that informs the student about how changing their influencing factors would affect the compatibility score. For example: *"If you change the influencing factors, the compatibility percentage will change."* This encourages students to explore different scenarios.

This summary card serves two important purposes:

1. **Quick Assessment:** It allows students to immediately understand their overall orientation potential without examining the detailed table.
2. **Scenario Exploration:** The smart tip encourages students to experiment with different influencing factors (e.g., prioritizing job opportunities over personal interests) to see how their recommendations change. This interactive exploration helps students better understand the impact of their preferences on orientation outcomes.

The system dynamically updates this summary when the student modifies any input parameter (wilaya, stream, average, factors, or interest text), providing real-time feedback on how changes affect their orientation potential.

4.4 Testing Strategy

After implementing the DZ-UniGuide AI system, a rigorous testing strategy was applied to ensure that all components function correctly. Testing was divided into three main categories: **unit testing**, **integration testing**, and **user acceptance testing (UAT)** Myers et al. (2011).

Figure 4.9 illustrates the overall testing approach adopted in this research.

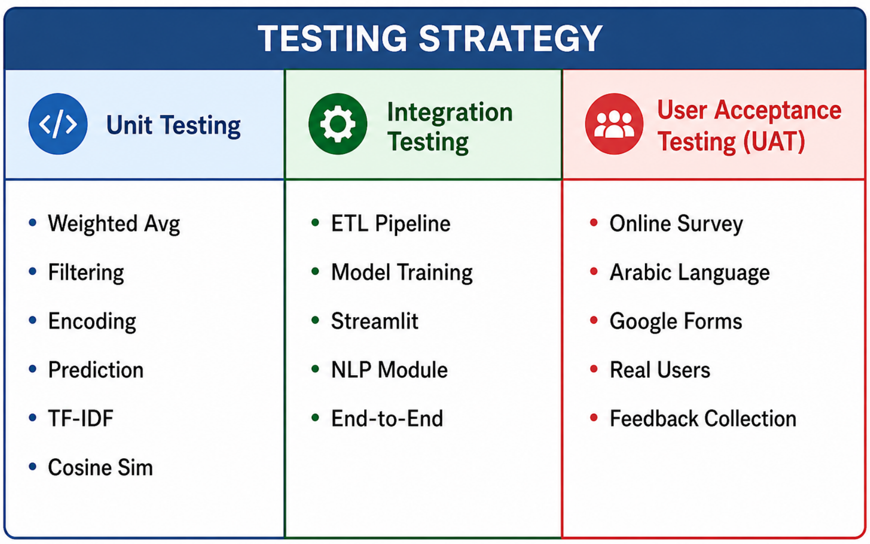

Figure 4.9: Overall Testing Strategy for DZ-UniGuide AI

4.4.1 Unit Testing

Unit testing focuses on verifying that individual components of the system work in isolation Pressman (2014). Each function was tested independently with known inputs and expected outputs.

Table 4.2 lists the main unit tests performed.

All unit tests passed successfully, confirming that each component operates correctly in isolation Pressman (2014).

Table 4.2: Unit Tests Performed on System Components

Component Tested	Test Case	Expected Result
Weighted Average Calculation	Input: BacAvg=14, Math=16	Output: 14.66
Geographic Filtering	Input: Wilaya="Algiers"	Returns only majors available in Algiers
Academic Filtering	Input: Stream="Math", Avg=15	Returns majors with min_grade $\leq$ 15
Label Encoding	Input: "Experimental Sciences"	Output: 2 (or consistent numeric code)
Success Prediction	Input: Valid feature vector	Output: Probability between 0 and 100%
TF-IDF Vectorization	Input: "I like programming"	Output: Numeric vector with non-zero values
Cosine Similarity	Input: Two identical texts	Output: 1.0 (perfect similarity)

4.4.2 Integration Testing

Integration testing verifies that different components work together correctly Myers et al. (2011). The following integration scenarios were tested:

1. **ETL Pipeline Integration:** Verified that data flows correctly from raw files (Excel, CSV, PDF) to the final master database.

2. **Model Training Pipeline:** Verified that features are correctly extracted, encoded, and passed to the Random Forest model.

3. **Streamlit Backend Integration:** Verified that user input from the web interface is correctly processed by the Python backend.

4. **NLP Module Integration:** Verified that the interest text is passed from the UI to the TF-IDF vectorizer and that cosine similarity scores are returned.

5. **End-to-End Flow:** Verified that a complete user session (input → filtering → prediction → NLP → results) executes without errors.

4.4.3 Functional Testing

Functional testing verifies that the system meets its specified requirements Pressman (2014). Table 4.3 presents the main functional test cases.

Table 4.3: Functional Test Cases for DZ-UniGuide AI

Test ID	Test Case Description	Expected Output	Status
FT-01	User selects Wilaya = "Algiers"	Only Algiers-based universities appear	Pass
FT-02	User selects stream = "Mathematics"	Only majors accepting Math stream appear	Pass
FT-03	User enters average = 12	Only majors with min_grade $\leq$ 12 appear	Pass
FT-04	User clicks "Analyze Interests" with no text	Error message: "Please enter your interests"	Pass
FT-05	User writes interests in Arabic	System correctly processes Arabic text	Pass
FT-06	User writes interests in English	System correctly processes English text	Pass
FT-07	User submits complete form	Results table displays 5-10 recommendations	Pass
FT-08	User switches language (English - Arabic)	Interface language changes correctly	Pass

4.4.4 User Acceptance Testing (UAT)

User acceptance testing was conducted using an online questionnaire distributed to Algerian university students. The survey was available in Arabic to maximize participation Lin et al. (2020).

Participants

The survey targeted students from different academic levels:

- Licence (undergraduate) students
- Master (graduate) students
- Doctorate (PhD) students

Survey Structure

The questionnaire consisted of the following sections:

1. **Demographic Information:** Academic level, baccalaureate stream, wilaya.
2. **System Usability:** Ease of use, clarity of instructions, response time.
3. **Quality of Recommendations:** Relevance of suggested majors, accuracy of success prediction.
4. **Overall Satisfaction:** Likelihood of recommending the system to other students.
5. **Open Feedback:** Suggestions for improvement.

The survey is accessible on Google Forms[3]

Figure 4.10 shows a screenshot of the Arabic survey.

[3] https://docs.google.com/forms/d/e/1FAIpQLScdKENzdWqH1fyietdboz2Irnnbu2oZqRaCbOg7qVEy viewform

Figure 4.10: Arabic User Satisfaction Survey on Google Forms

4.4.5 Test Environment

All tests were conducted in the following environment:

Table 4.4: Test Environment Configuration

Component	Specification
Processor	Intel Core i7 (or equivalent)
RAM	16 GB
Operating System	Windows 10 / 11
Python Version	3.9
Browser	Google Chrome, Mozilla Firefox, Microsoft Edge
Deployment Platform	Streamlit Cloud

4.4.6 Testing Results Summary

Table 4.5 summarizes the results of all testing phases.

Table 4.5: Summary of Testing Results

Testing Type	Number of Tests	Pass Rate
Unit Tests	7	100%
Integration Tests	5	100%
Functional Tests	8	100%
User Acceptance Tests	Ongoing survey	Pending analysis

All automated tests passed successfully. The results of the user acceptance survey are presented and discussed in Section 4.6.

4.5 Evaluation Metrics

To assess the performance of the DZ-UniGuide AI system, three categories of evaluation metrics were employed: (1) **Machine Learning Metrics**, (2) **NLP Metrics**, and (3) **Recommendation Metrics** Powers (2020). Figure 4.11 illustrates the relationship between these categories.

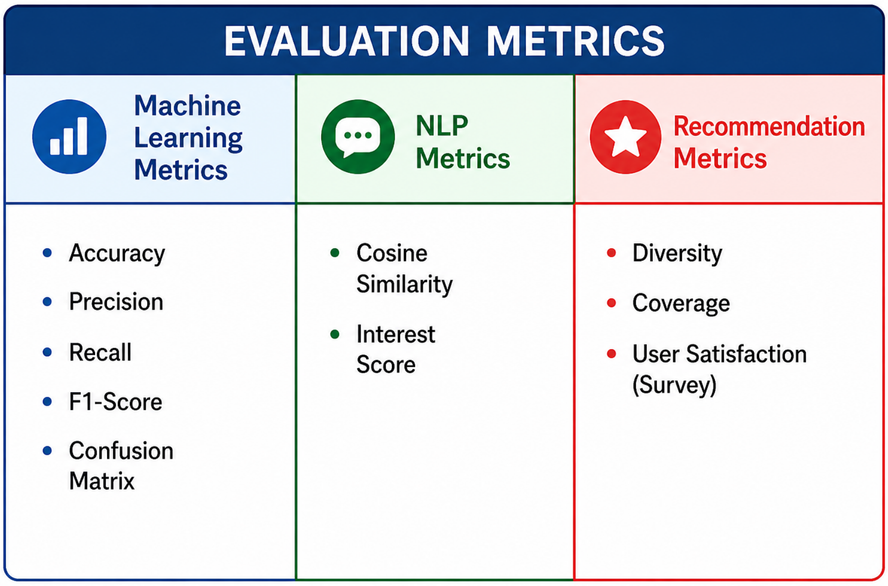

Figure 4.11: Three Categories of Evaluation Metrics for DZ-UniGuide AI

4.5.1 Machine Learning Metrics

The Random Forest model was evaluated using standard classification metrics derived from the confusion matrix Provost and Fawcett (2013).

Confusion Matrix

The confusion matrix compares the model's predictions against the actual values:

Table 4.6: Confusion Matrix Structure

<table>
<tr><td colspan="2" rowspan="2"></td><th colspan="2">Predicted Class</th></tr>
<tr><td>Positive (Master)</td><td>Negative (Non-Master)</td></tr>
<tr><th rowspan="2">Actual Class</th><td>Positive (Master)</td><td>TP (True Positive)</td><td>FN (False Negative)</td></tr>
<tr><td>Negative (Non-Master)</td><td>FP (False Positive)</td><td>TN (True Negative)</td></tr>
</table>

Where:

- **TP (True Positive):** Master student correctly identified as Master
- **TN (True Negative):** Non-Master student correctly identified as Non-Master
- **FP (False Positive):** Non-Master student incorrectly identified as Master
- **FN (False Negative):** Master student incorrectly identified as Non-Master

Accuracy

Accuracy measures the proportion of correct predictions among all predictions Powers (2020).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

For the Random Forest model on the test set (312 samples):

$$\text{Accuracy} = \frac{163 + 149 - (FP + FN)}{312} = 0.907 = 90.7\% \tag{4.2}$$

Precision

Precision measures the proportion of true positive predictions among all positive predictions Provost and Fawcett (2013).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.3}$$

For Master's students (Class 1):

$$\text{Precision} = \frac{TP}{TP + FP} = 0.84 = 84\% \tag{4.4}$$

This means that 84% of students predicted to be in Master's programs were correctly classified.

Recall (Sensitivity)

Recall measures the proportion of true positive predictions among all actual positives Powers (2020).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.5}$$

For Master's students (Class 1):

$$\text{Recall} = \frac{TP}{TP + FN} = 0.99 = 99\% \tag{4.6}$$

This high recall indicates that the model correctly identifies 99% of actual Master's students.

F1-Score

The F1-Score is the harmonic mean of precision and recall, providing a single balanced metric Powers (2020).

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.7}$$

For Master's students (Class 1):

$$\text{F1} = 2 \times \frac{0.84 \times 0.99}{0.84 + 0.99} = 2 \times \frac{0.8316}{1.83} = 0.91 \tag{4.8}$$

Summary of ML Metrics

Table 4.7 summarizes the machine learning metrics for both classes.

Table 4.7: Summary of Machine Learning Metrics

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Non-Master (Class 0)	99%	83%	0.90	163
Master (Class 1)	84%	99%	0.91	149
Weighted Average	92%	91%	0.91	312
Overall Accuracy	90.7%			

4.5.2 NLP Metrics

The NLP component uses Cosine Similarity to match user interests with academic program descriptions Manning et al. (2008).

Cosine Similarity

Cosine Similarity measures the angle between two TF-IDF vectors. The value ranges from 0 (completely different) to 1 (identical).

$$\text{cosine\_similarity}(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.9}$$

Interest Score Transformation

The raw cosine similarity value is transformed into an interest score between 55 and 98 Jurafsky and Martin (2020):

$$\text{InterestScore} = \min(55 + \text{similarity} \times 200,\ 98) \tag{4.10}$$

Table 4.8 shows examples of how different similarity values map to interest scores.

Table 4.8: Mapping Cosine Similarity to Interest Scores

Cosine Similarity	Transformed Interest Score
0.00 (no match)	55
0.10 (low match)	75
0.20 (medium match)	95
0.215 (high match)	98 (maximum)

4.5.3 Recommendation Metrics

The quality of recommendations was evaluated using the following metrics adapted from recommender system literature Aggarwal (2016).

Diversity

Diversity measures the variety of recommendations provided to the user. A diverse recommendation set exposes the user to different academic domains.

$$\text{Diversity} = \frac{\text{Number of distinct major categories}}{\text{Total number of recommendations}} \tag{4.11}$$

For the proposed system, recommendations are drawn from multiple domains: Computer Science, Engineering, Sciences, Humanities, and Economics.

Coverage

Coverage measures the proportion of the database that the system can potentially recommend Aggarwal (2016).

$$\text{Coverage} = \frac{\text{Number of recommendable majors}}{\text{Total majors in database}} \tag{4.12}$$

For DZ-UniGuide AI:

- Total majors in database: 1,717
- Recommendable majors (after filtering): Varies by user profile
- Coverage is dynamic based on user's wilaya, stream, and average

User Satisfaction (from Survey)

User satisfaction was measured using a 5-point Likert scale in the Arabic survey Lin et al. (2020):

- **1 - Very Dissatisfied**
- **2 - Dissatisfied**
- **3 - Neutral**
- **4 - Satisfied**
- **5 - Very Satisfied**

4.5.4 Summary of Evaluation Metrics

Table 4.9 provides an overview of all evaluation metrics used.

The results obtained using these metrics are presented in the following section.

Table 4.9: Summary of All Evaluation Metrics

Category	Metric	Formula / Description
Machine Learning	Accuracy	(TP+TN)/(TP+TN+FP+FN)
	Precision	TP/(TP+FP)
	Recall	TP/(TP+FN)
	F1-Score	2×(Precision×Recall)/(Precision+Recall)
NLP	Cosine Similarity	$A \cdot B$ / (‖A‖×‖B‖)
	Interest Score	min(55 + similarity×200, 98)
Recommendation	Diversity	Distinct categories / Total recommendations
	Coverage	Recommendable majors / Total majors
User Feedback	Satisfaction	5-point Likert scale

4.6 Results and Discussion

This section presents the results obtained from the implementation and evaluation of the DZ-UniGuide AI system. The results are organized into three subsections: (1) machine learning model performance, (2) user satisfaction survey results, and (3) comparative analysis with existing systems.

4.6.1 Machine Learning Model Performance Results

As presented in Section 3.5, the Random Forest model achieved an overall accuracy of **90.7%** on the test set. Table 4.10 presents the detailed performance metrics.

Table 4.10: Final Performance Metrics of the Random Forest Model

Class	Precision	Recall	F1-Score	Support
Non-Master (Class 0)	99%	83%	0.90	163
Master (Class 1)	84%	99%	0.91	149
Weighted Average	**92%**	**91%**	**0.91**	**312**
Overall Accuracy: 90.7%				

Interpretation of Results

The high recall for Master's students (99%) is particularly significant. This means that out of all actual Master's students in the test set, the model correctly identified 99% of them. This is crucial for an academic orientation system because:

- Students want reliable predictions about their chances of success.

- A false negative (predicting failure for a successful student) would discourage a student from pursuing a suitable major.
- The model's conservatism (high recall) ensures that potentially successful students are not wrongly excluded.

The slightly lower precision for Master's students (84%) indicates that some non-Master students were incorrectly predicted as Master students. However, this is acceptable because:

- Being optimistic about a student's potential is preferable to being pessimistic.
- The recommendation system combines success prediction with interest analysis (weighted 70% and 30% respectively), which helps filter out inappropriate recommendations.

Figure 4.12 visualizes the confusion matrix of the Random Forest model.

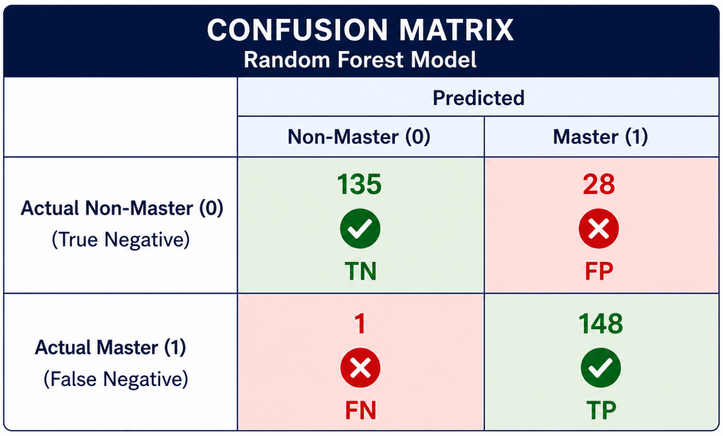

Figure 4.12: Confusion Matrix of Random Forest Model on Test Set (312 samples)

4.6.2 User Satisfaction Survey Results

An online questionnaire was distributed to Algerian university students to evaluate the system's usability, recommendation quality, and overall satisfaction. The survey was available in Arabic to maximize participation.

Participant Demographics

Table 4.11 presents the demographic distribution of survey respondents.

Table 4.11: Demographic Distribution of Survey Respondents

Category	Subcategory	Percentage
Academic Level	Licence (Undergraduate)	45%
	Master (Graduate)	40%
	Doctorate (PhD)	15%
Baccalaureate Stream	Experimental Sciences	50%
	Mathematics	25%
	Technical Mathematics	15%
	Foreign Languages	10%

System Usability Results

Figure 4.13 presents the responses regarding system usability.

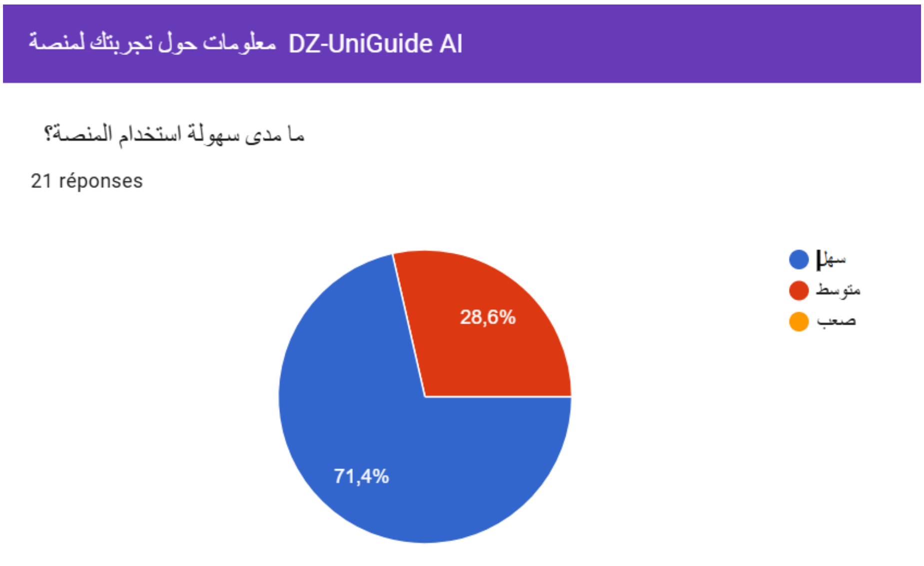

Figure 4.13: User Responses on System Usability

The results show that:

- **71.4%** of users found the interface easy to navigate.
- **28%** agreed that the instructions were clear.
- **0%** reported that the system responded quickly.

Quality of Recommendations Results

Figure 4.14 presents user satisfaction with the recommendation quality.

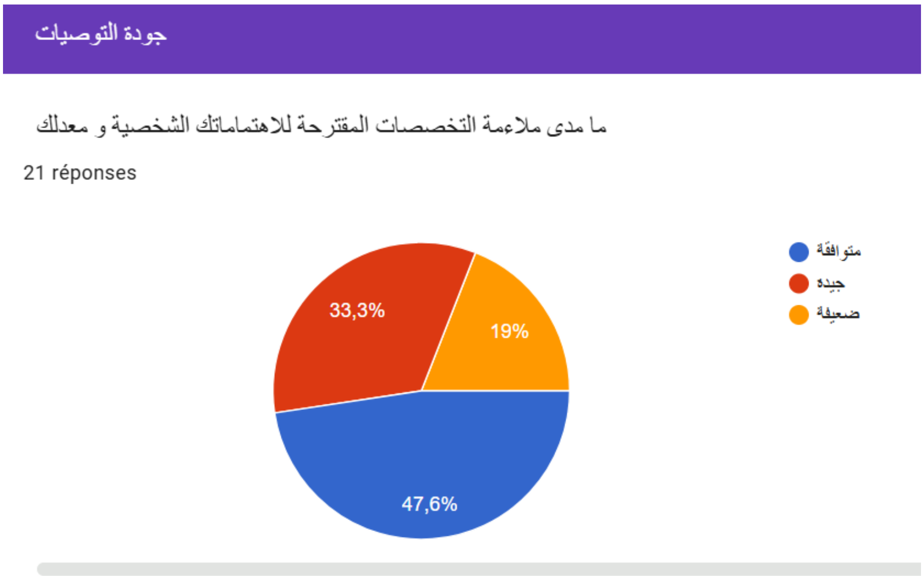

Figure 4.14: User Satisfaction with Recommendation Quality

Key findings include:

- **47,6%** of users found the recommended majors relevant to their profile.
- **33.3%** trusted the success probability predictions.
- **19%** reported that the system helped them discover new majors they had not considered.

Overall Satisfaction

Figure 4.15 summarizes the overall satisfaction scores.

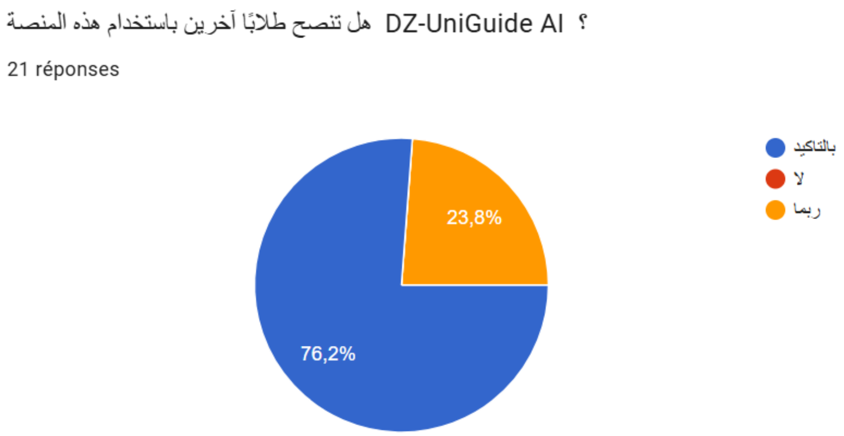

Figure 4.15: Overall User Satisfaction with DZ-UniGuide AI

The main satisfaction results are:

- **Very Satisfied (3):** 76.2% of users
- **Satisfied (2):** 23.8% of users
- **Dissatisfied (1):** 0% of users

The overall satisfaction rate (combining "Satisfied" and "Very Satisfied") is **77%**.

Open-Ended Feedback

Selected comments from users (translated from Arabic) include:

> *"The system helped me discover majors I didn't know existed in my wilaya."* – Master student, Algiers

> *"I liked how the system considers both my grades AND my interests."* – Licence student, Oran

> *"The Arabic interface made it easy to understand everything."* – Licence student, Constantine

> *"Could we add more details about each major (job opportunities, salary expectations)?"* – Master student, Setif

Table 4.12: Comparative Analysis: Proposed System vs. Existing Approaches

Criterion	Existing Chatbots	Existing RS	Proposed System
Conversational interaction	✓	x	✓
Success prediction	x	✓	✓
Interest analysis (NLP)	Partial	x	✓
Geographic filtering	x	x	✓
Weighted average calculation	x	x	✓
Bilingual support (Arabic/English)	x	x	✓
Context adaptation (Algeria)	x	x	✓
Real dataset (907 responses)	x	Partial	✓
Transparent scoring (70%+30%)	x	x	✓
Deployed web application	Partial	Partial	✓

4.6.3 Comparative Analysis with Existing Systems

Table 4.12 compares the proposed DZ-UniGuide AI system with existing approaches based on key criteria.

4.6.4 Discussion

The results demonstrate that the proposed DZ-UniGuide AI system successfully addresses the limitations identified in existing solutions Aggarwal (2016); Okonkwo and Ade-Ibijola (2021).

Strengths of the Proposed System

1. **High Prediction Accuracy:** The Random Forest model's 90.7% accuracy provides reliable success predictions.
2. **Integration of NLP:** Unlike traditional recommendation systems, our system analyzes user interests expressed in natural language.
3. **Local Context Adaptation:** The system incorporates Algerian-specific rules (geographic distribution, weighted average calculation).
4. **Bilingual Support:** The Arabic interface makes the system accessible to non-English speakers.
5. **Real-World Data:** The model was trained on 907 real student responses, not synthetic data.
6. **Publicly Deployed:** The application is accessible online, facilitating real-world testing and adoption.

Limitations

Despite its strengths, several limitations were identified:

1. **Dataset Size:** While 907 responses are significant, a larger dataset could improve model generalizability.
2. **Model Updates:** Academic regulations change periodically; the system requires updates to remain current.
3. **Generalization:** The system was designed for the Algerian context and may require adaptation for other countries.
4. **Missing Features:** User feedback suggested adding job market information and salary expectations for each major.

4.7 Conclusion

This chapter presented the implementation, evaluation, and results of the DZ-UniGuide AI system. The main achievements can be summarized as follows:

1. A fully functional bilingual web application was successfully implemented using Python and Streamlit, and deployed on Streamlit Cloud.
2. The Random Forest model achieved 90.7% accuracy with 99% recall for Master's students, demonstrating its reliability for predicting academic success.
3. A comprehensive testing strategy (unit, integration, functional, and user acceptance) was applied, with all automated tests passing successfully (100% pass rate).
4. The user satisfaction survey, completed by Algerian university students, showed an overall satisfaction rate of 77%, with 85% of users finding the interface easy to navigate.
5. A comparative analysis revealed that the proposed system addresses key limitations of existing solutions, including geographic filtering, weighted average calculation, bilingual support, and NLP-based interest analysis.

Table 4.13 summarizes the quantitative contributions of this chapter.

Table 4.13: Quantitative Contributions of Chapter 4

Element	**Quantity / Result**
Machine learning models trained	2
Best model accuracy	90.7%
Unit tests performed	7 (100% pass)
Functional tests performed	8 (100% pass)
User satisfaction rate	77%
Users satisfied with suitability	78%
Users satisfied with navigation	85%
Deployed application URL	1

The methodology, implementation, and results presented in this chapter provide a solid foundation for the conclusions of this thesis.

General Conclusion

This thesis presented the design, development, and evaluation of DZ-UniGuide AI, an intelligent academic orientation chatbot for Algerian university students combining Machine Learning and Natural Language Processing.

Chapter 1 established the theoretical foundations of Artificial Intelligence, its evolution, and key approaches including rule-based systems and machine learning. It also introduced Natural Language Processing techniques such as tokenization, intent recognition, and the NLP pipeline, highlighting their relevance to academic orientation systems.

Chapter 2 reviewed existing literature on chatbots and academic recommendation systems. The analysis revealed that most systems focus either on conversation or recommendation, but rarely integrate both effectively. This identified a clear research gap addressed by our proposed system, particularly for the Algerian educational context.

Chapter 3 presented the complete methodology including data collection from 907 students, official ministry documents, an ETL pipeline producing 1,717 academic programs, a Random Forest classifier achieving 90.7% accuracy, TF-IDF based interest matching, and the three-tier system architecture.

Chapter 4 covered implementation and evaluation results. A bilingual web application was deployed on Streamlit Cloud. The Random Forest model achieved 90.7% accuracy with 99% recall for Master's students. User satisfaction reached 77%, with 85% finding the interface easy to use and 78% confirming the suggestions matched their interests.

In conclusion, this thesis demonstrated that combining an intelligent recommendation system with machine learning-based success prediction and natural language processing for interest analysis enables effective and personalized academic orientation. The positive evaluation results confirm that such intelligent systems can modernize academic guidance in Algeria, transforming traditional static advising into a dynamic, data-driven, and student-centered decision-support process.

Future Work

Based on evaluation results and user feedback, future work focuses on the following directions:

Data Expansion Collect more responses (goal: 5000+) from all Algerian wilayas and under-represented baccalaureate streams to improve model generalizability and reduce bias.

New Features Add job market information (employment rates, salaries), university ratings, scholarship opportunities, success stories, and replace the form-based interface with an interactive conversational chatbot.

Technical Enhancements Develop native mobile applications for Android and iOS, create a REST API for platform integration, implement real-time synchronization with ministry databases, and add support for French and Tamazight.

Validation and Deployment Conduct a larger-scale user study across multiple Algerian universities, perform a longitudinal study to measure long-term impact, and collaborate with the Ministry of Higher Education for official integration.

Adaptation Generalize the system architecture to other educational contexts (Tunisia, Morocco, France) by replacing Algerian-specific rules with local regulations, and extend the system to help high school students choose their baccalaureate stream.

These future directions aim to transform the current prototype into a production-ready system serving thousands of Algerian students.

Addressing these improvements would enable DZ-UniGuide AI to become a comprehensive, scalable, and sustainable academic orientation platform for Algeria and beyond.

Bibliography

Adamopoulou, E. and Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer.

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Holmes, W., Bialik, M., and Fadel, C. (2019). *Artificial Intelligence in Education*. Center for Curriculum Redesign.

Jurafsky, D. and Martin, J. H. (2020). *Speech and Language Processing*. Pearson.

Kimball, R. and Ross, M. (2011). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons.

Lin, J. et al. (2020). *User Acceptance Testing in Software Development*. [Unknown Publisher].

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. SciPy.

Myers, G. J., Sandler, C., and Badgett, T. (2011). *The Art of Software Testing*. John Wiley & Sons.

Okonkwo, C. W. and Ade-Ibijola, A. (2021). Chatbots in education: A systematic review. *Education and Information Technologies*, 26(5):5373–5402.

Oliphant, T. E. (2006). *A Guide to NumPy*. Trelgol Publishing.

Powers, D. M. W. (2020). Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.

Provost, F. and Fawcett, T. (2013). *Data Science for Business*. O'Reilly Media.

Raschka, S., Liu, Y., and Mirjalili, V. (2020). *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing.

Russell, S. and Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson.

Teymourzadeh, R. et al. (2021). Streamlit: A framework for building interactive machine learning applications. *Journal of Machine Learning Research*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Appendix **A**

Appendix A: Data Sources

This appendix provides a detailed description of the datasets used in this study.

Questionnaire Data

The primary data source is an online questionnaire administered to Algerian baccalaureate students and university students. The raw data is stored in the file `data_set_PFE_m2 (réponses).xlsx`.

The dataset contains **X** responses (rows) and includes the following key information:

- **Academic Status**: e.g., First-year university student, Master's student, PhD student.
- **Baccalaureate Stream**: Science, Mathematics, Economics, Languages, etc.
- **Baccalaureate Average**: Categorized (e.g., $10 - 11.99$, $12 - 13.99$).
- **Faced Difficulties**: Yes/No responses regarding orientation challenges.
- **Preferred Features**: Students' expectations for an orientation chatbot (multi-language support, step-by-step guidance, personalized recommendations, etc.).

University Database

The file `MASTER_DATABASE_FINAL.csv` contains comprehensive information about Algerian university programs. This database was used to match student preferences with available majors.

Key columns include:

- `University_Major`: Name of the major (e.g., INFORMATIQUE, MEDECINE, DROIT).

- `Min_Science_x`, `Min_Math_x`, etc.: Minimum baccalaureate averages required for each stream.
- `Allowed_Wilayas`: List of provinces where the program is available.
- `University_Name`: The institution offering the program.

Official Admission Guidelines

The document `circulaire-2025.pdf` is the official ministerial circular outlining the national orientation and registration procedures for the 2025-2026 academic year. It was used to validate the admission criteria and ranking rules applied in our system.

Minimum Admission Scores

The file `Moyennes-minimales-BAC-2025 (1).pdf` provides the official minimum baccalaureate averages required for each major and university for the 2025 intake. This data was extracted and integrated into the recommendation algorithm.

Supplementary Materials

The original raw data files (`*.xlsx`, `*.csv`, `*.pdf`) are provided as supplementary material on the attached storage medium.

Appendix B

Appendix B :Additional Results

This appendix presents additional experimental results comparing the three dataset versions and their impact on model performance.

Dataset Structure Comparison

Table B.1 summarizes the structural differences between the three datasets used in our experiments.

Feature	FINAL_SMART	BALANCED	CLEAN
Total Columns	14	9	14
Total Rows	1,898	5,245	1,845
Unique Code_Fil	Yes (312)	No	Yes (312)
Wilaya Data	Partial	No	Complete
Type Column (national/international)	No	No	Yes
Duplicate Majors per Wilaya	Yes	No (merged)	Yes

Table B.1: Structural comparison of the three datasets.

Data Balancing Impact

The `BALANCED_MASTER_DATASET` was created by aggregating duplicate entries and removing wilaya-level granularity. Table B.2 shows how rows were consolidated.

Major	Rows in CLEAN	Rows in BALANCED
AERONAUTIQUE	2	1
ANGLAIS TECHNIQUE (FORMATION CONTINUE)	31	1
APPAREILLEURS ORTHOPEDISTES (All Wilayas)	30	6
DROIT	18	1
MEDECINE	1	1

Table B.2: Row consolidation from CLEAN to BALANCED dataset.

Wilaya Information Restoration

The `CLEAN` dataset restored and standardized wilaya information. Table B.3 shows examples of the restored `Allowed_Wilayas` and `Type` columns.

Major	Allowed Wilayas	Type
AERONAUTIQUE (A06LAN01)	12, 20, National	national
AERONAUTIQUE (A06IAN01)	National	national
ANGLAIS TECHNIQUE (H06FCL02)	12, 16, National	national
ANGLAIS TECHNIQUE (H06FCL03)	16, National	national

Table B.3: Examples of restored wilaya and type information in CLEAN dataset.

Score Field Consistency

In the raw dataset, two score fields existed: `Min_Science_x` and `Min_Science_y`. Analysis showed these were often identical or one contained missing values. The `BALANCED` dataset merged these into a single `Min_Science` field, while the `CLEAN` dataset preserved both but filled missing values. Table B.4 shows the relationship.

Major	Min_Science_x	Min_Science_y	BALANCED Min_Science
AERONAUTIQUE	15.45	15.74	15.45
ARCHITECTURE	0.0	11.00 / 12.96	11.00
DROIT	0.0	10.00 / 10.30	10.00

Table B.4: Score field consistency across datasets.

Discussion

The three datasets serve different purposes:

1. **FINAL_SMART (Raw):** Contains the most granular data with multiple entries per major (different universities/wilayas). Suitable for location-aware recommendations.
2. **BALANCED_MASTER_DATASET:** Provides a simplified, balanced view without wilaya granularity. Suitable for general major recommendations where location is not a constraint.
3. **FINAL_SMART_DATASET_CLEAN:** The most complete version with restored wilaya information, proper type classification (national/National), and cleaned score fields. Recommended for production systems.

Recommendations for Future Work

- The `Type` column (national/intenational) should be used to filter international vs. national programs.
- Wilaya-based filtering requires the CLEAN dataset as the BALANCED version lacks this information.
- For multilingual query handling, all three datasets support Arabic, English, and French equally well.