People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

FERHAT ABBAS UNIVERSITY SETIF1
FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Master's Thesis

Submitted in Partial Fulfillment of the Requirements for

Master's Degree in Computer Science

(Specialization: Data Engineering and Web Technologies IDTW*)*

Theme :

EduRecom AI

An Educational Recommendation System Based on Students' Programming Errors

Prepared by:

1- Lina BOUABDALLAH

2- Ines Amira MANALLAH

Supervised by:

Dr _Lamri DOUIDI

Members of the Jury:

Quality	Name & Surname	University
President		FERHAT ABBAS UNIVERSITY SETIF1
Examiner		FERHAT ABBAS UNIVERSITY SETIF1
Supervisor	Dr _Lamri DOUIDI	FERHAT ABBAS UNIVERSITY SETIF1

Academic Year: 2025 – 2026

Title: Pedagogical Recommendation System Based on Student Errors

Abstract:

Learning programming is often challenging for students due to the difficulty of understanding technical error messages, creating a gap between error detection and meaningful feedback. This thesis presents **EduRecom AI**, a hybrid intelligent system designed to support beginner Python students by providing pedagogical explanations and personalized recommendations. Unlike conventional debugging tools, the proposed system employs a four-stage diagnostic pipeline that combines deterministic techniques (compilation, execution, and static analysis) with Machine Learning methods. To ensure reliable error classification, a Multinomial Naive Bayes model was trained on a refined subset of 150,000 authentic student errors extracted from the IBM Project CodeNet dataset. In addition, a guarded generative AI layer based on GPT-4o-mini converts technical diagnostics into human-readable explanations and code corrections adapted to students' needs. Implemented using FastAPI and React, the platform provides personalized feedback and progress monitoring, offering a scalable and effective solution for enhancing programming education.

Keywords: Students' Errors, Pedagogical Recommendation System, Hybrid AI, Machine Learning, Naive Bayes, Project CodeNet, Programming Education.

Résumé (Français)

Titre : Système de recommandation pédagogique basé sur les erreurs des étudiants

Résumé :

L'apprentissage de la programmation représente souvent un défi pour les étudiants en raison de la difficulté à interpréter les messages d'erreur techniques, ce qui crée un écart entre la détection des erreurs et la compréhension du retour fourni. Ce mémoire présente **EduRecom AI**, un système intelligent hybride conçu pour accompagner les étudiants débutants en Python en leur fournissant des explications pédagogiques et des recommandations personnalisées. Contrairement aux outils de débogage classiques, le système proposé repose sur une chaîne de diagnostic en quatre étapes combinant des approches déterministes (compilation, exécution et analyse statique) avec des techniques d'apprentissage automatique. Afin d'assurer une classification fiable des erreurs, un modèle Multinomial Naive Bayes a été entraîné sur un sous-ensemble raffiné de 150 000 erreurs authentiques d'étudiants issues du jeu de données IBM Project CodeNet,. En complément, une couche d'intelligence artificielle générative sécurisée basée sur GPT-4o-mini transforme les diagnostics techniques en explications compréhensibles et en corrections de code adaptées aux besoins des étudiants. Développée avec FastAPI et React, la plateforme offre un suivi des progrès et des retours personnalisés, constituant ainsi une solution robuste et évolutive pour l'enseignement moderne de la programmation.

Mots-clés : Système de recommandation pédagogique, erreurs des étudiants, intelligence artificielle hybride, apprentissage automatique, Naive Bayes, Project CodeNet, enseignement de la programmation.

ملخص (باللغة العربية)

العنوان: نظام توصية بيداغوجي قائم على أخطاء الطلبة

الملخص:
يمثل تعلم البرمجة تحديًا بالنسبة للعديد من الطلبة بسبب صعوبة فهم مضمون رسائل الأخطاء التقنية، مما يخلق فجوة بين اكتشاف الخطأ وفهم التوصيات المقدمة. يقدم هذا العمل **EduRecom AI**، وهو نظام ذكي هجين صُمم لمساعدة المبتدئين في تعلم لغة Python من خلال توفير شروحات تربوية وتوصيات مخصصة. وعلى عكس برامج تصحيح الأخطاء التقليدية، يعتمد النظام المقترح على سلسلة تشخيص مكونة من أربع مراحل تجمع بين مجموعة أساليب (الترجمة، التنفيذ والتحليل الساكن) وتقنيات التعلم الآلي. ولضمان تصنيف موثوق للأخطاء، تم تدريب نموذج **Multinomial Naive Bayes** على مجموعة فرعية مُنقحة تضم 150 ألف خطأ حقيقي ارتكبه الطلبة، مستخرجة من قاعدة بيانات **IBM Project CodeNet**، يعتمد النظام على طبقة ذكاء اصطناعي توليدي مؤمنة باستخدام **GPT-4o-mini** لتحويل التشخيصات التقنية إلى شروحات مبسطة وتصحيحات برمجية تتناسب مع احتياجات الطلبة. وقد تم تطوير المنصة باستخدام FastAPI وReact، حيث توفر تغذية راجعة مخصصة ومتابعة لتطور المتعلم، مما يجعلها حلاً فعالاً وقابلاً للتوسع لدعم تعليم البرمجة الحديث.

الكلمات المفتاحية :نظام توصية بيداغوجي، أخطاء الطلبة، الذكاء الاصطناعي الهجين، التعلم الآلي، نايف بايز، Project CodeNet، تعليم البرمجة.

Table of Contents

List of Figures / List of Tables: 8
Introduction 9
1.1Background and Context 9
1.2 Problem Statement 10
1.3 Proposed Solution: EduRecom AI 10
1.4 Project Objectives 11
1.5 Proposed System Pipeline (Project Plan) 11
1.6 Structure of the Report 12
Chapter 2: State of the Art — From Pedagogical Error to Intelligent Recommendation 13
Introduction 13
2.1 Error as a Learning Lever: Foundations and Classifications 13
2.1.1 The Status of Error in the Learning Process 13
2.1.2 Pedagogical Classification of Errors (Astolfi) 14
2.1.3 Technical Classification for Programming Errors 15
2.1.4 Bridging Pedagogical Theory and Technical Classification 16
2.2 Pedagogical Recommendation Systems: From Expert Rules to AI 16
2.2.1 Generation 1: Rule-Based Systems (1980–2000) 16
2.2.2 Generation 2: Collaborative Filtering and Data-Driven Approaches (2000–2015) 17
2.2.3 Generation 3: AI, Deep Learning, and Embeddings (2015–Present) 19
2.2.4 The Gap: Why Current Systems Fall Short for Programming Education 22
2.3 Recent Work (2024–2025) on AI-Based Error Analysis 23
2.3.1 LLMs for Programming Error Messages and Explanations 23
2.3.2 Automated Program Repair Systems 24
2.3.3 Deep Learning for Code Understanding 25
2.3.4 Educational AI Systems for Programming 26
2.3.5 Synthesis: Progress and Persistent Gaps 27
2.4 Comparative Synthesis: Mapping the Gap 29
2.4.1 Key Observations 30
Conclusion 31
Chapter 3: System Design and Architecture 32
3.1 Strategic Delimitation: The Python Pedagogical Sandbox 32
3.1.1 Rationale for Choosing Python 32
3.2 The Tri-Level Cognitive Analysis Framework 33
3.2.1 Theoretical Foundation: Cascading Diagnostics 33
3.2.2 Level 1: Static Detection Layer (The “What”) 33

3.2.3 Level 2: Taxonomic Classification (The “Why”) 34
3.2.4 Level 3: Generative Correction & Profiling (The “Who” + “How”) 34
3.2.5 Use Case Diagram 35
3.3 Functional Requirements and Actor Modeling 36
3.3.1 Unique Actor: The Autonomous Learner 36
3.3.2 Functional Requirements Specification 36
3.4 Architectural Design: The Hybrid AI Stack 37
3.4.1 Macro-Architectural View: Four Autonomous Modules 37
3.4.2 Sequence Diagram: The Correction Workflow 39
3.4.3 Class Diagram: Domain Model 40
3.4.4 Activity Diagram: The Automated Pipeline 41
3.5 Non-Functional Requirements and Constraints 42
Conclusion 42
Chapter 4: Implementation 44
Introduction 44
4.1 Error Analysis and Classification Module 44
4.1.1 Hybrid Analysis Pipeline 44
4.1.2 Machine Learning Classification 45
4.2 Recommendation and Feedback Module 47
4.2.1 Educational Feedback Logic 47
4.2.2 AI-Enhanced Feedback (OpenAI Integration) 48
4.3 Student Interface Module 48
4.3.1 Code Analysis Page 49
4.3.2 Progress and Analytics Dashboard 49
4.3.3 AI Insights and Recommendation Page 51
4.3.4 Premium Features and Monetization 53
4.3.5 User Registration (Sign-Up Page) 54
4.4 System Implementation 55
4.5 Technology Stack Summary 57
Conclusion 57
Chapter 5: Evaluation: strengths and limitations 58
Introduction 58
5.1 Experimental Setup 58
5.1.1 Evaluation Dataset 58
5.1.2 Evaluation Metrics 59
5.1.3 Compared Approaches 59

5.1.4 Rule-Based Approach 60
5.1.5 Classical Machine Learning Approach 60
5.1.6 Comparative Results 61
5.2 Qualitative Evaluation 61
5.2.1 Classification Performance 61
5.2.2 Impact of Dataset Augmentation 62
5.2.3 Recommendation Engine Evaluation 62
5.2.4 LLM Feedback Component Evaluation 62
5.2.5 Overall System Observation 63
5.3 Strengths and Limitations 63
5.3.1 Strengths 63
5.3.2 Limitations 64
Conclusion 65
General Conclusion and Future Work 66
6.1 General Conclusion 66
6.2 Future Work 67
6.2.1 Technical Enhancements 67
6.2.2 Educational Improvements 67
6.2.3 Institutional Integration 67
6.3 Final Remarks 67
References 68