

Project: AI0 Classifier

Upgraded Project Presentation

GrID034 Team

Thành viên Team

Hiện tại nhóm có 5 thành viên chính thức

Đàm Nguyên Khánh	Leader
Vũ Thái Sơn	Tech leader
Bùi Đức Xuân	Member
Trịnh Nguyễn Huy Hoàng	Member
Võ Hoàng	Member
Vương Nguyệt Bình	Member

Quản lý team: **Discort**

Các công cụ sử dụng cho AIO Conquer: **Overleaf** | **GG Colab** | **MS Office**



Table of contents

01

**Objectives of the
project**

02

**Structure of the
project**

03

**Upgrade of the
project**

04

Result

05

Live Demo Feature

06

Conclusion



01

Objectives of the project



Vấn đề hiện tại

Data scientists dành 60-70% thời gian cho :

1. Data preprocessing & vectorization
2. Model testing & comparison
3. Code duplication & maintenance



Vấn đề hiện tại

Thiếu công cụ tích hợp:

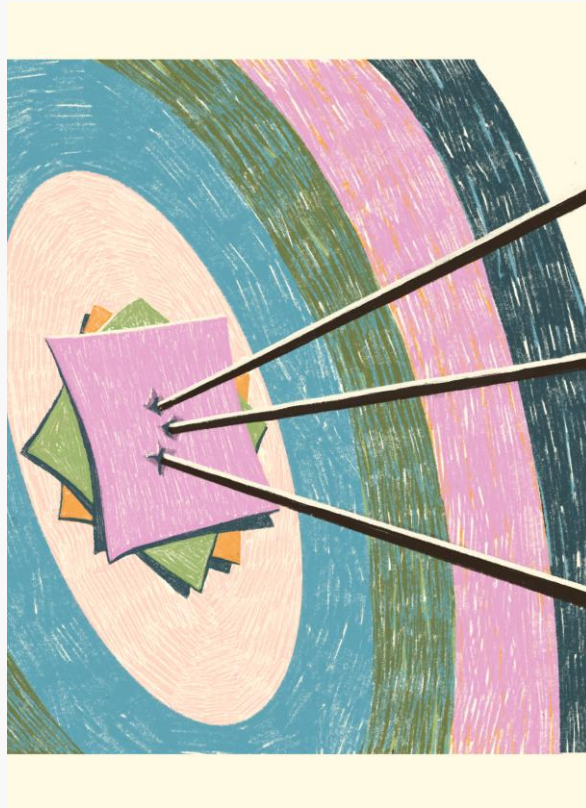
1. Manual workflow từng bước
2. Không có GUI thân thiện
3. Khó so sánh performance



Mục tiêu AIO Classifier

All-in-One Solution:

1. Tự động hóa toàn bộ pipeline
2. GUI trực quan với Streamlit
3. So sánh nhiều model combinations
4. GPU acceleration & caching



Objectives



Our aim

Tự động hóa và dễ quản lý, ứng dụng



The goal

Hệ thống hoạt động trơn tru và mượt mà, phục vụ tốt nhất khi cần.

02

Structure of the project



Cấu trúc Dự án Email Classifier

```
1 models/
2 |   __init__.py      # Package initialization
3 |   base/            # Base classes và interfaces
4 |   |   base_model.py # Abstract base class
5 |   |   interfaces.py # Protocol definitions
6 |   |   metrics.py    # Common evaluation metrics
7 |   clustering/      # Clustering models
8 |   |   kmeans_model.py # K-Means implementation
9 |   classification/  # Classification models
10 |  |   knn_model.py   # K-Nearest Neighbors
11 |  |   decision_tree_model.py
12 |  |   naive_bayes_model.py
13 |  |   logistic_regression_model.py
14 |  |   linear_svc_model.py
15 |  |   svm_model.py
16 |  ensemble/        # Ensemble learning
17 |  |   ensemble_manager.py
18 |  |   stacking_classifier.py
19 |  utils/           # Utility modules
20 |  |   model_factory.py # Factory pattern
21 |  |   model_registry.py # Model registration
22 |  |   validation_manager.py
23 |  new_model_trainer.py # Advanced trainer
```

Project structure

Kiến trúc Modular với Factory Pattern - Tổ chức Models theo chức năng và dễ mở rộng.



Core Components



BaseModel Interface

- Standardized fit/predict methods
- GPU management
- Progress tracking



Model Factory

- Dynamic model creation
- Parameter optimization
- Error recovery



Advanced Features

- FAISS KNN acceleration
- Ensemble learning
- Intelligent caching



03

Upgrade of the project



Các nâng cấp



Giao diện

Giao diện Wizard UI Streamlit chuyên nghiệp, có điều hướng, phân tích, chỉnh sửa



Kiến trúc

Chia module rõ ràng



Vectorization

SVD optimization + caching
Sentence-BERT + GPU



KNN Model

FAISS integration (GPU acceleration)
90.9% accuracy (Best performer)



Decision Tree

Cost Complexity Pruning
77.2% accuracy



Naive Bayes

88.7% accuracy (Fastest)



Các nâng cấp



Ensemble Learning

Voting Strategy
88.2% accuracy (Most
stable)



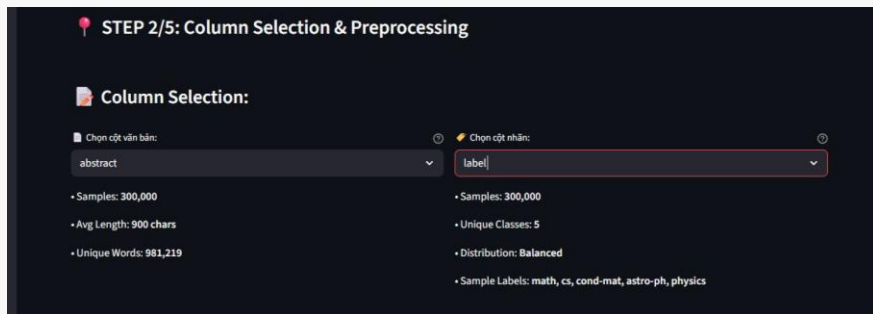
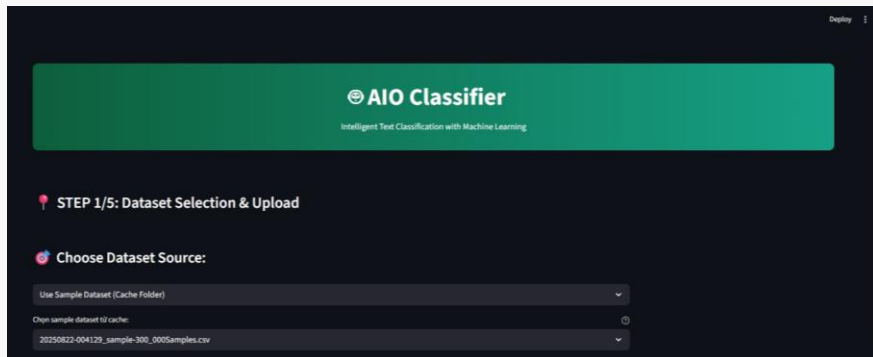
Giao diện Streamlit

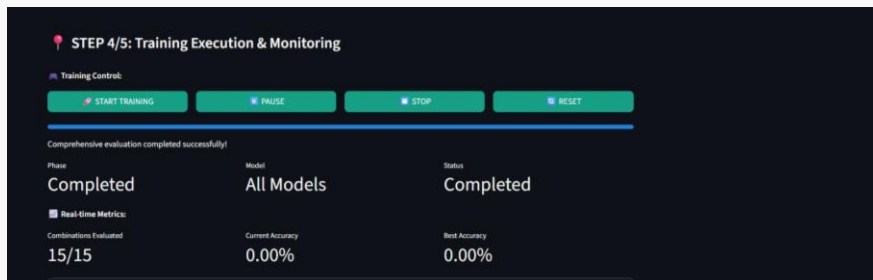
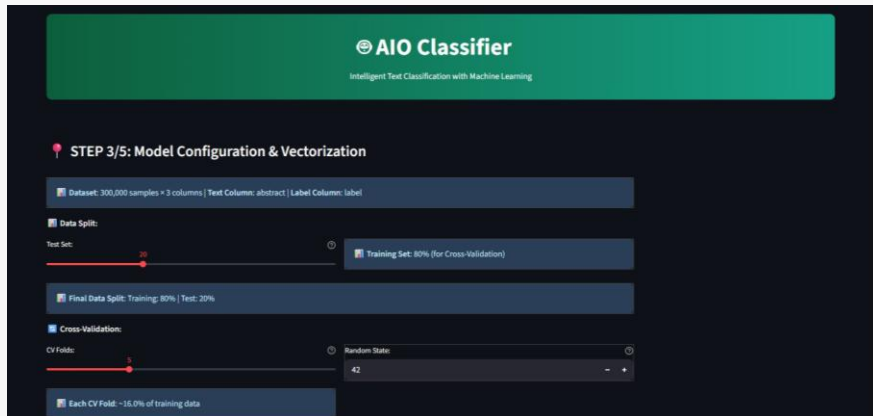
Bước 1: Chọn Dataset

- Hệ thống cho người dùng tùy chọn files.
- Preview dữ liệu (title, abstract, label, etc.)
- Kiểm tra cấu trúc dataset

Bước 2: Cấu hình & Tiền xử lý

- Chọn cột text (abstract) và label (label)
- Tự động làm sạch văn bản
- Kiểm tra dữ liệu rỗng
- Chuẩn bị cho training





Giao diện Streamlit

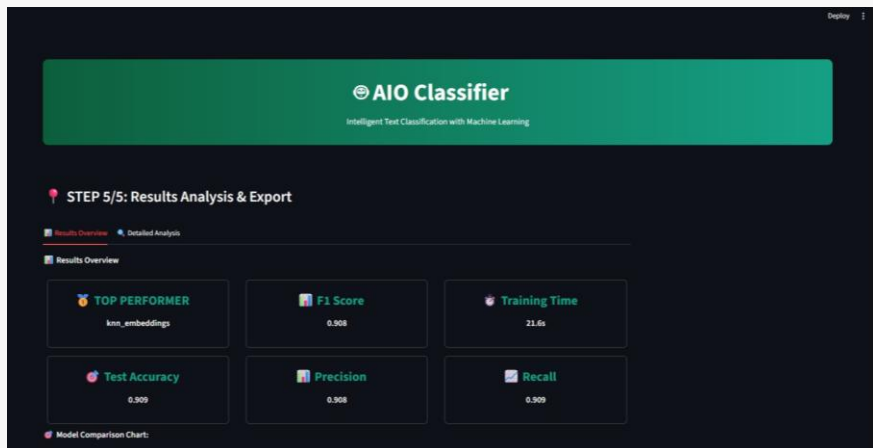
Bước 3: Cấu hình Model & Vectorization

- Thiết lập tỷ lệ train/test (80%/20%)
- Cross-Validation: 5 folds
- Hiển thị thông tin 300K samples, 3 cột.
- Xác định cột Text và label

Bước 4: Thực thi & Giám sát Training

- Trình Điều khiển
- Giám sát: Tiến độ training
- Cache: Quản lý thông tin cache training





STEP 5/5: Results Analysis & Export

Results Overview **Detailed Analysis**

Model	Vectorization	F1 Score	Accuracy	Precision	Recall	Time (s)
Knn	Bow	84.9%	85.2%	85.1%	85.2%	498.6
Knn	Tfidf	85.9%	86.1%	86.0%	86.1%	416.9
Knn	Embeddings	90.8%	90.9%	90.8%	90.9%	21.6
Decision Tree	Bow	75.7%	75.7%	75.8%	75.7%	465.0
Decision Tree	Tfidf	74.6%	74.5%	74.6%	74.5%	469.0
Decision Tree	Embeddings	77.2%	77.2%	77.1%	77.2%	730.6

Giao diện Streamlit

Bước 5: Phân tích Kết quả & Xuất Báo cáo

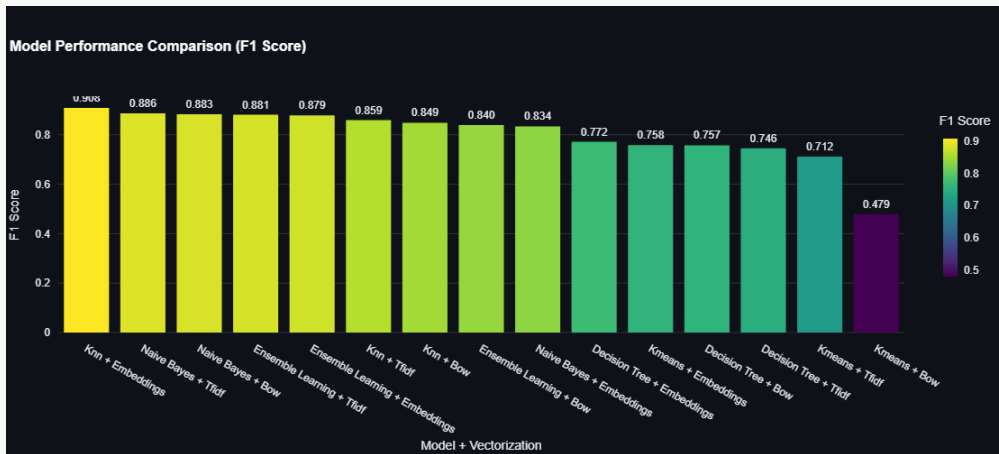
- Tổng quan Hiệu suất: Hiển thị bảng so sánh chi tiết các mô hình
- Đánh giá Metrics: Cung cấp các chỉ số F1 Score, Accuracy, Precision, Recall và thời gian huấn luyện cho từng sự kết hợp
- Lựa chọn Phân tích: Cho phép người dùng chọn một mô hình cụ thể từ bảng để xem phân tích chi tiết hơn
- Xuất Báo cáo: Chuẩn bị dữ liệu để xuất báo cáo kết quả



04

Result





Kết quả Training 300K Samples

- Embeddings cho hiệu suất tốt nhất với KNN
- Naive Bayes ổn định với mọi phương pháp vectorization
- K-Means kém hiệu quả (47.9% - 75.8%)
- Ensemble Learning đảm bảo tính ổn định cao



PERFORMANCE RANKING



KNN + Embeddings

90.9%



Naive Bayes + TF-IDF

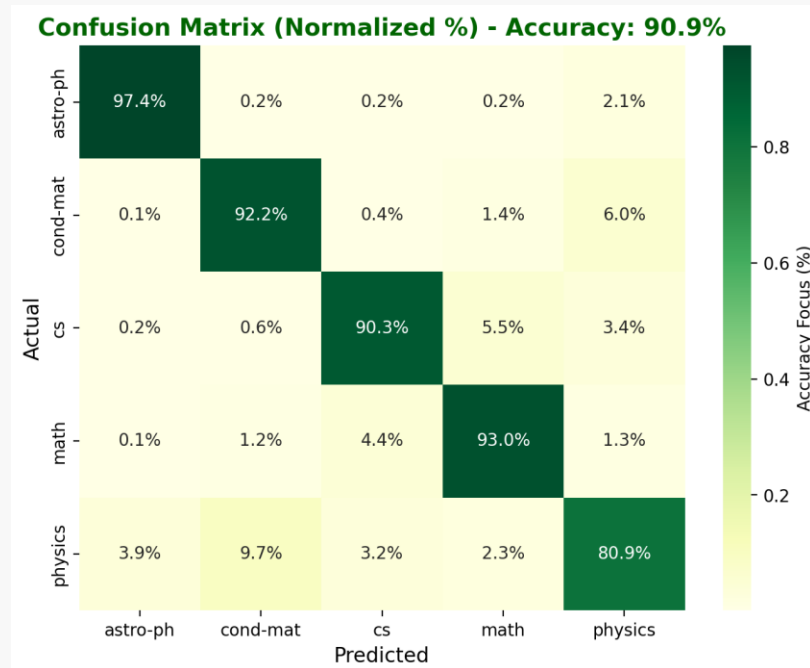
88.7%



Ensemble Learning

88.2%

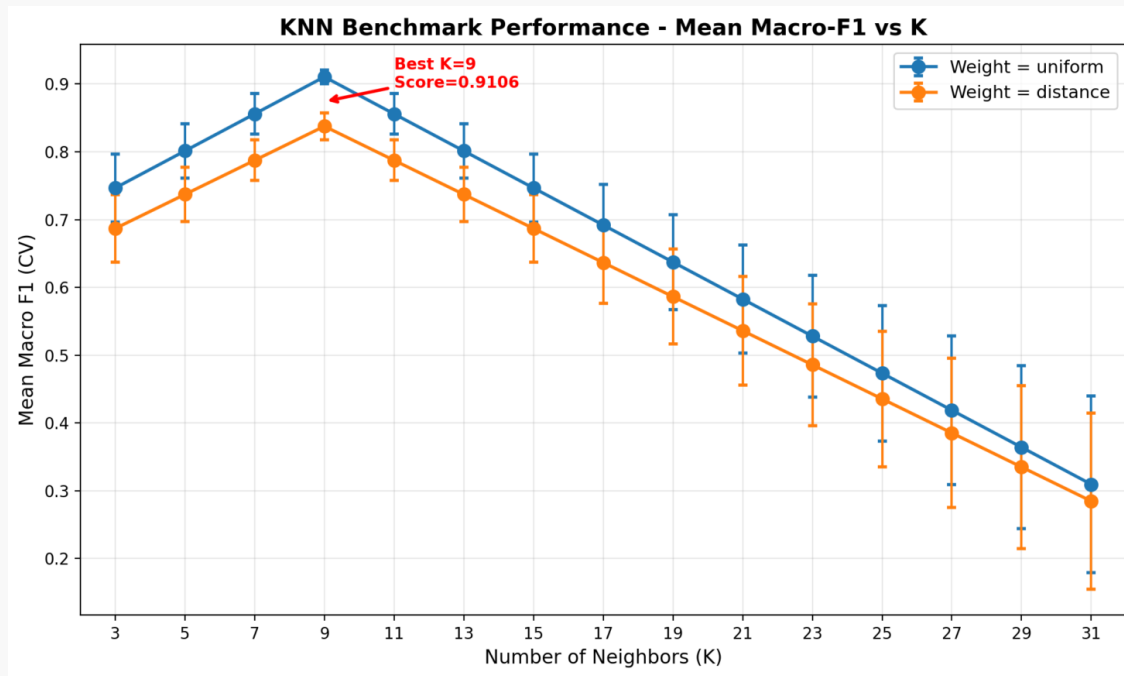




Kết quả KNN + Embeddings

- KNN + Embeddings phù hợp với các chủ đề khoa học tự nhiên
- physics và cond-mat có ranh giới mờ nhạt, cần feature engineering
- Mô hình đạt hiệu suất cao với thời gian training nhanh.





Best K

- K = 9 với Uniform weighting đạt F1-Score cao nhất: 0.9106
- Uniform ổn định hơn Distance trên hầu hết các giá trị K
- Hiệu suất tăng đến K=9, sau đó giảm dần khi K tăng

05

LIVE DEMO FEATURES



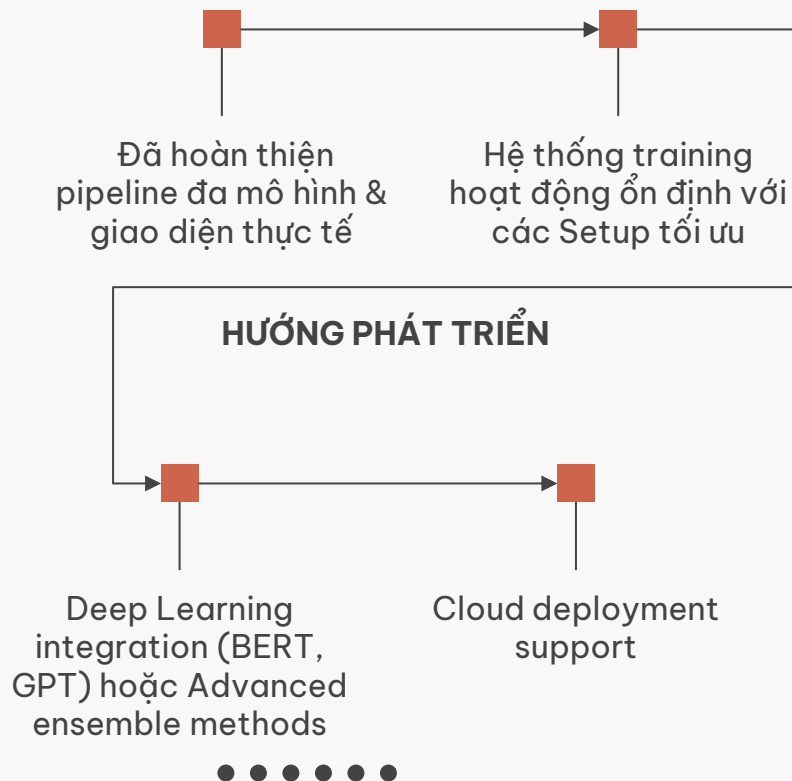
06

Conlusion





Kết quả & Hướng phát triển



Thanks!

Do you have any questions?

