ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

BÀI TẬP ĐÁNH GIÁ CHẤT LƯỢNG DỮ LIỆU

BÁO CÁO ĐÁNH GIÁ CHẤT LƯỢNG DỮ LIỆU

MÔN HỌC: KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG (CS313)

Nhóm 2

GVHD

ThS. Nguyễn Anh Thư

TP. HO CHI MINH, 3/2025

DANH SÁCH THÀNH VIÊN

STT	Họ và tên	MSSV
1	**Võ Đình Trung**	**22521571**
2	Trương Phúc Trường	22521587
3	Mai Dương	22520302
4	Trần Qui Linh	22520779
5	Triệu Tấn Huy	22520581
6	Nguyễn Trần Anh Phong	22521092
7	Nguyễn Tấn Lợi	22520801

LỜI CẢM ƠN

Chúng tôi xin chân thành cảm ơn cô giáo, ThS. Nguyễn Thị Anh Thư, người đã định hướng, giúp đỡ, trực tiếp hướng dẫn và tận tình chỉ bảo chúng tôi trong suốt quá trình nghiên cứu, xây dựng và hoàn đồ án này.

Chúng tôi cũng xin được cảm ơn tới gia đình, những người thân và bạn bè thường xuyên quan tâm, động viên, chia sẻ kinh nghiệm, cung cấp các tài liệu hữu ích trong thời gian học tập, nghiên cứu cũng như trong suốt quá trình thực hiện đồ án.

TP. HCM, ngày 18 tháng 3 năm 2025

ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

……...

……...

……

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……

TP. HCM, ngày 3 tháng 3 năm 2025

GVHD

I. Giới thiệu đề tài

Tên đề tài

Hệ thống dự đoán mức độ hoàn thành của người dùng đối với khóa học.

Thời gian thực hiện

8 tuần

Tổng kinh phí

Nhóm thực hiện

Giảng viên hướng dẫn: ThS. Nguyễn Thị Anh Thư
Nhóm 2, gồm các thành viên:

STT	Họ và tên	Mã số sinh viên
1	**Võ Đình Trung**	**22521571**
2	Trương Phúc Trường	22521587
3	Mai Dương	22520302
4	Trần Qui Linh	22520779
5	Triệu Tấn Huy	22520581
6	Nguyễn Trần Anh Phong	22521092
7	Nguyễn Tấn Lợi	22520801

II. Mô tả đề tài

Giới thiệu

Trong thời đại số hóa ngày nay, việc học trực tuyến đã trở thành một phần không thể thiếu của ngành giáo dục. Tuy nhiên, mặc dù có sự tiện lợi và linh hoạt, việc duy trì sự tham gia của sinh viên trong một khóa học trực tuyến vẫn là một thách thức đối với các nhà quản lý và giáo viên. Vấn đề đặt ra là làm thế nào để dự đoán và ứng phó với việc bỏ học của sinh viên một cách hiệu quả, trước khi tình trạng này diễn ra.

MOOCs là nền tảng các khóa học trực tuyến mở rộng, cho phép hàng triệu người tham gia từ khắp nơi trên thế giới. Chúng cung cấp các tài liệu giảng dạy, bài giảng video, bài tập và cơ hội giao tiếp trực tuyến.

Đề tài này tập trung vào việc áp dụng các phương pháp học máy và phân tích dữ liệu để dự đoán mức độ hoàn thành khóa học của sinh viên trong một khóa học trực tuyến cụ thể từ dữ liệu thu thập trên MOOCs. Bằng cách sử dụng dữ liệu từ các hoạt động của sinh viên như xem video, thực hiện bài tập, tương tác với bài giảng, và nhận xét, chúng ta có thể xây dựng một mô hình dự đoán chính xác.

Ứng dụng

Mô hình này có ứng dụng cụ thể trong lĩnh vực giáo dục trực tuyến. Nó giúp giáo viên và quản lý trường học nhận diện học viên có nguy cơ bỏ học và phân bổ nguồn lực hỗ trợ một cách hiệu quả, từ đó cải thiện tỷ lệ tốt nghiệp và chất lượng giáo dục trực tuyến.

Các nghiên cứu liên quan

- *Predicting MOOCs Dropout Using Only Two Easily Obtainable Features from the First Week's Activities (2019)*: Nghiên cứu này nhằm mục đích dự đoán sớm tình trạng bỏ học của người học, ngay từ tuần đầu tiên, bằng cách so sánh một số phương pháp học máy, bao gồm Random Forest, Adaptive Boost, XGBoost and Gradient Boosted Classifiers. Kết quả cho thấy độ chính xác hứa hẹn (82%-94%) chỉ sử dụng 2 tính năng, vượt qua các phương pháp tiên tiến nhất, ngay cả khi triển khai nhiều tính năng.
- *Predicting Student Dropout based on Machine Learning and Deep Learning: A Systematic Review (2023)*: Đánh giá tính hiệu quả của các thuật toán Machine Learning và Deep Learning trong việc dự đoán tình trạng học sinh hoàn thành khóa học, cho thấy Random Forest là thuật toán được sử dụng nhiều nhất với tỷ lệ chính xác đáng chú ý là 99%.

- *Understanding Dropouts in MOOCs (2019)*: Bài viết điều tra các vấn đề bỏ học trong MOOCs bằng cách sử dụng dữ liệu từ XuetangX, đề xuất Mạng tương tác tính năng nhận biết bối cảnh (CFIN) để mô hình hóa và dự đoán hành vi bỏ học, hoạt động này vượt trội hơn các phương pháp hiện có và đã được triển khai để nâng cao khả năng giữ chân người dùng.
- *Deep Learning for Dropout Prediction in MOOCs (2019)*: Nghiên cứu này xây dựng bài toán dự đoán bỏ học bằng cách dự đoán xem học sinh có thể hoàn thành bao nhiêu nội dung trong toàn bộ giáo trình khóa học. Mô hình dự đoán tỷ lệ bỏ học dựa trên mạng thần kinh tái phát (RNN) và lớp nhúng URL được đề xuất để giải quyết vấn đề này. Kết quả cho thấy độ chính xác dự đoán của mô hình cao hơn đáng kể so với mô hình học máy truyền thống.

III. Tổng quan đề tài

Ngữ cảnh

Trong bối cảnh giáo dục hiện đại, đặc biệt là với sự phát triển mạnh mẽ của các nền tảng học trực tuyến, ngày càng có nhiều người tham gia vào các khóa học trực tuyến nhằm nâng cao kiến thức và kỹ năng. Đối tượng chính là học viên tham gia khóa học trực tuyến, bao gồm học sinh, sinh viên muốn học hỏi và phát triển kỹ năng. Tuy nhiên, một trong những thách thức lớn nhất mà các nền tảng học trực tuyến phải đối mặt là tỷ lệ bỏ học cao và mức độ hoàn thành khóa học thấp.

Sự thiếu động lực hoặc những khó khăn trong việc theo kịp nội dung giảng dạy là một số nguyên nhân phổ biến khiến học viên không thể hoàn thành khóa học. Việc xây dựng một hệ thống dự đoán mức độ hoàn thành khóa học của người học không chỉ giúp các giảng viên và nhà quản lý khóa học hiểu rõ hơn về hành vi học tập của học viên, mà còn cho phép họ can thiệp kịp thời thông qua các biện pháp hỗ trợ, từ đó nâng cao hiệu quả và chất lượng giảng dạy.

Đề tài này nhằm xây dựng một mô hình dự đoán dựa trên dữ liệu hành vi của người học, phân tích dữ liệu giúp xây dựng mô hình dự đoán mức độ hoàn thành khóa học và đề xuất biện pháp can thiệp kịp thời.

Input

Đầu vào bao gồm: Dữ liệu về thông tin khóa học, thông tin của người dùng, hoạt động và kết quả học tập của người dùng. Đây là 1 dataframe với mỗi dòng có user_id, course_id

duy nhất chứa các thuộc tính sau: (Link file nhỏ chứa các object có ý nghĩa cao nhất để demo cấu trúc dataframe) (đồng thời chia dataset thành time-series với 4 tuần).

- user_id: Mã định danh duy nhất của người dùng/học viên
- course_id: Mã định danh duy nhất của khóa học
- gender: Giới tính của người dùng/học viên
- year_of_birth: Năm sinh của người dùng, học viên
- user_school_encoded: Mã hóa trường của người dùng, học viên
- num_teacher: Số lượng giáo viên giảng dạy khóa học
- num_school: Số lượng trường học liên kết với khóa học
- field_encoded: Mã hóa lĩnh vực/chủ đề của khóa học
- prerequisites_encoded: Mã hóa các điều kiện tiên quyết của khóa học
- num_videos: Tổng số video trong khóa học
- num_exercises: Tổng số bài tập trong khóa học
- num_problems: Tổng số problem trong khóa học
- num_students: Tổng số học viên đăng ký khóa học
- total_default_video_time: Tổng thời lượng mặc định của tất cả video trong 1 khóa học
- total_comments: Tổng số bình luận trong khóa học
- total_replies: Tổng số phản hồi trong khóa học
- avg_comments_per_student: Số bình luận trung bình mỗi học viên của mỗi khóa học
- avg_replies_per_student: Số phản hồi trung bình mỗi học viên của mỗi khóa học

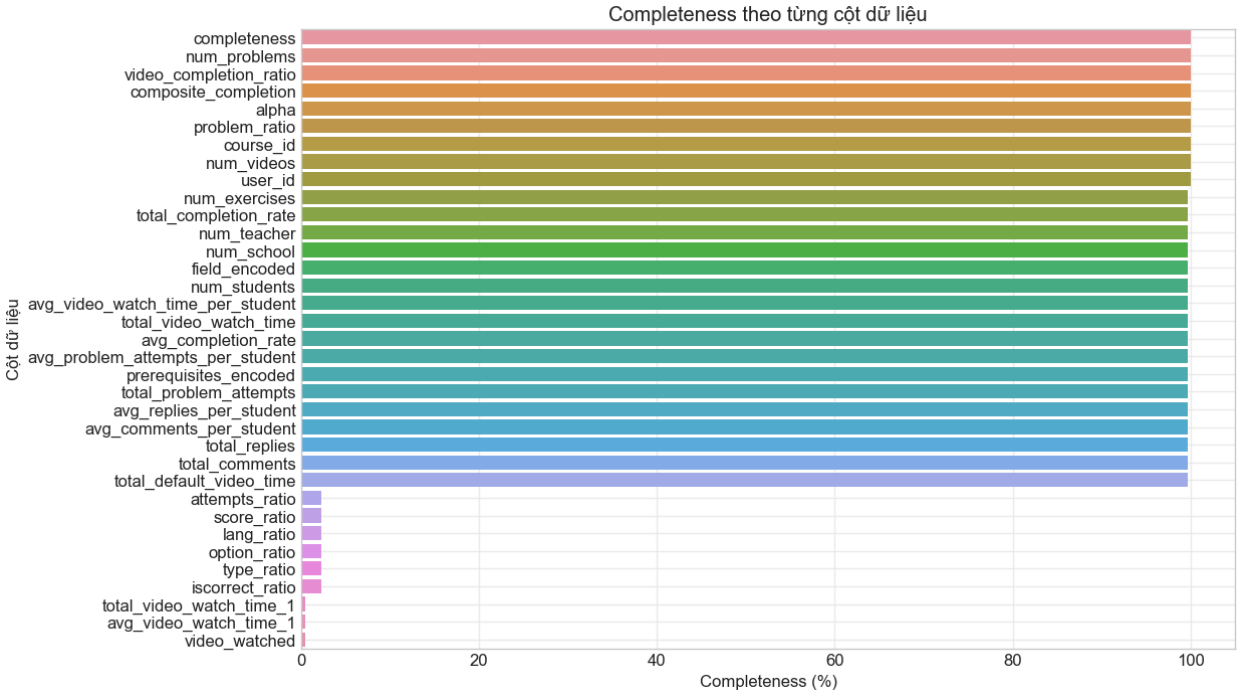
total_problem_attempts: Tổng số lần thử giải bài tập của mỗi khóa học

- avg_problem_attempts_per_student: Số lần thử giải bài tập trung bình mỗi học viên
- total_completion_rate: Tổng tỷ lệ hoàn thành của khóa học của mỗi khóa học
- avg_completion_rate: Tỷ lệ hoàn thành trung bình của khóa học của mỗi khóa học
- total_video_watch_time: Tổng thời gian xem video của tất cả học viên của mỗi khóa học
- avg_video_watch_time_per_student: Thời gian xem video trung bình mỗi học viên
- problem_days_since_enroll: Thời gian trung bình học viên bắt đầu làm bài tập
- video_days_since_enroll: Thời gian trung bình học viên bắt đầu xem video
- comment_days_since_enroll: Thời gian trung bình học viên bắt đầu bình luận
- reply_days_since_enroll: Thời gian trung bình học viên bắt đầu trả lời bình luận
- problem_type: Loại problem
- problem_language_encoded: Mã hóa loại ngôn ngữ của problem
- problem_num_options: Số lượng đáp án của problem
- problem_typetext_encoded: Mã hóa loại problem
- problem_chapter: Số chương của problem
- problem_index: Chỉ mục của problem

- ex_do_week1: Bài kiểm tra làm trong tuần 1
- problem_done_week1: Problem hoàn thành trong tuần 1
- total_correct_answer_week1: Tổng số câu trả lời đúng trong tuần 1
- total_attempt_week1: Tổng số lượt nộp trong tuần 1
- problem_do_MaxScore_week1: Điểm số cao nhất làm được trong tuần 1
- total_score_week1: Tổng số điểm của tuần 1
- ex_do_week2: Bài kiểm tra làm trong tuần 2
- problem_done_week2: Problem hoàn thành trong tuần 2
- total_correct_answer_week2: Tổng số câu trả lời đúng trong tuần 2
- total_attempt_week2: Tổng số lượt nộp bài trong tuần 2
- problem_do_MaxScore_week2: Điểm số cao nhất làm được trong tuần 2
- total_score_week2: Tổng số điểm của tuần 2
- total_video_watching_week1: Tổng số video xem trong tuần 1
- video_watching_speed_week1: Tốc độ xem video trong tuần 1
- video_watching_duration_week1: Thời gian xem video trong tuần 1
- total_video_watching_week2: Tổng số video xem trong tuần 2
- video_watching_speed_week2: Thời gian xem video trong tuần 2
- video_watching_duration_week2: Tổng số video xem trong tuần 2
- total_comments_week1: Tổng số bình luận trong tuần 1
- total_replies_week1: Tổng số phản hồi trong tuần 1
- total_comments_week2: Tổng số bình luận trong tuần 2
- total_replies_week2: Tổng số phản hồi trong tuần 2
- week3, week4 tương tự với week1, week2

	user_id	course_id	video_completion_ratio	num_teacher	num_school	field_encoded	pr
0	U_10031789	C_1017355	0.0	5.0	1.0	4.0	
1	U_10035815	C_1017355	0.0	5.0	1.0	4.0	
2	U_1003583	C_1017355	0.0	5.0	1.0	4.0	
3	U_10064340	C_1017355	0.0	5.0	1.0	4.0	
4	U_10102621	C_1017355	0.0	5.0	1.0	4.0	
...	...	...	...	...	...	...	...
6310435	U_9478971	C_956130	0.0	3.0	1.0	4.0	
6310436	U_9478974	C_956130	0.0	3.0	1.0	4.0	
6310437	U_9530948	C_956130	0.0	3.0	1.0	4.0	
6310438	U_9849139	C_956130	0.0	3.0	1.0	4.0	
6310439	NaN	C_956450	0.0	3.0	1.0	4.0	

6310440 rows × 34 columns



	user_id	course_id	video_completion_ratio	num_teacher	num_school	field_encoded	pre
0	U_10448165	C_1017355	0.000000	5.0	1.0	4.0	
1	U_10448484	C_1017355	0.000000	5.0	1.0	4.0	
2	U_10682780	C_1017355	0.000000	5.0	1.0	4.0	
3	U_10682785	C_1017355	0.000000	5.0	1.0	4.0	
4	U_10682787	C_1017355	0.000000	5.0	1.0	4.0	
...	...	...	...	...	...	...	
158414	U_31818726	C_956130	0.014085	3.0	1.0	4.0	
158415	U_34173615	C_956130	0.000000	3.0	1.0	4.0	
158416	U_36406914	C_956130	0.000000	0.0	0.0	NaN	
158417	U_36655655	C_956130	0.000000	0.0	0.0	NaN	
158418	U_8394424	C_956130	0.056338	3.0	1.0	4.0	

158419 rows × 34 columns

Output

Nhóm lựa chọn 2 output để thử nghiệm là **completion** và **user_final_exam_score_ratio**:

completion: Chỉ số hoàn thành tổng hợp (kết hợp video và bài tập).

Cách tính **completion** (Tổng quan):

- Tính tỷ lệ hoàn thành riêng lẻ: Tính video_completion (tỷ lệ xem video) và problem (tỷ lệ làm bài tập). Các giá trị này có thể được chuẩn hóa hoặc biến đổi (như trong final.ipynb) để cải thiện phân phối.
- Xác định tham số (Alpha - α): Với rateprob là problem_completion và ratevid là video_completion.

a.rateprob + (1-a).ratevid
với a = 0.5 nếu course tồn tại cả 2 hình thức
với a = 0 nếu course chỉ tồn tại video
với a=1 nếu course chỉ tồn tại problem

.

- Kết hợp có tham số: Tính điểm tổng hợp bằng công thức: completion = (alpha * problem_completion) + ((1 - alpha) * video_completion)

Cách tạo chi tiết thuộc tính **completion**:

- Tính tỷ lệ hoàn thành video (video_completion_ratio):
  - Đo lường mức độ người dùng đã xem video trong khóa học.
  - Công thức cơ bản (hoặc ý tưởng): min(1,(Tổng số video đã xem của user) / (Tổng số video mặc định của khóa học))
  - Tính toán điều này bằng cách dùng dữ liệu từ user_video_interaction.csv
  - Giá trị này thường được giới hạn ở mức tối đa là 1 (hoặc 100%).
- Tính tỷ lệ hoàn thành problem (problem_ratio):
  - Đo lường mức độ người dùng đã tương tác hoặc hoàn thành các bài tập/vấn đề.
  - Tính dựa trên: min(1, Tổng số bài tập đã làm của user / Tổng số bài tập trong khóa học).
  - Giá trị này cũng được giới hạn ở mức tối đa là 1.
- Xác định tham số (Alpha - α):
  - Đây là yếu tố quan trọng để cân bằng giữa video và bài tập. Trọng số này phụ thuộc vào cấu trúc của khóa học.
- Ý tưởng: Một khóa học có thể có 3 dạng: Khóa học chỉ có video, khóa học chỉ có exercise, và khóa học có cả 2 hình thức. Vì thế để phù hợp với mỗi loại course ta sẽ dùng Alpha để tính completion, với:
  - Nếu chỉ có bài tập (num_videos == 0), alpha = 1. Vì không có video nên video_completion sẽ bị NaN hoặc bằng 0 và ta sẽ không thể dùng video_completion để đánh giá khả năng hoàn thành khóa học của user đối với một khóa học không có video được. Vì thế ta sẽ chỉ dùng problem_completion nhờ việc tiêu biến video_completion thông qua tham số Alpha.
  - Nếu chỉ có video (num_problems == 0), alpha = 0. Vì không có problem nên problem_completion sẽ bị NaN hoặc bằng 0 và ta sẽ không thể dùng problem_completion để đánh giá khả năng hoàn thành khóa học của user đối với một khóa học không có problem được. Vì thế ta sẽ chỉ dùng video_completion nhờ việc tiêu biến video_completion thông qua tham số Alpha.
  - Nếu có cả hai, alpha = 0.5, lấy trung bình video_completion và problem_completion.

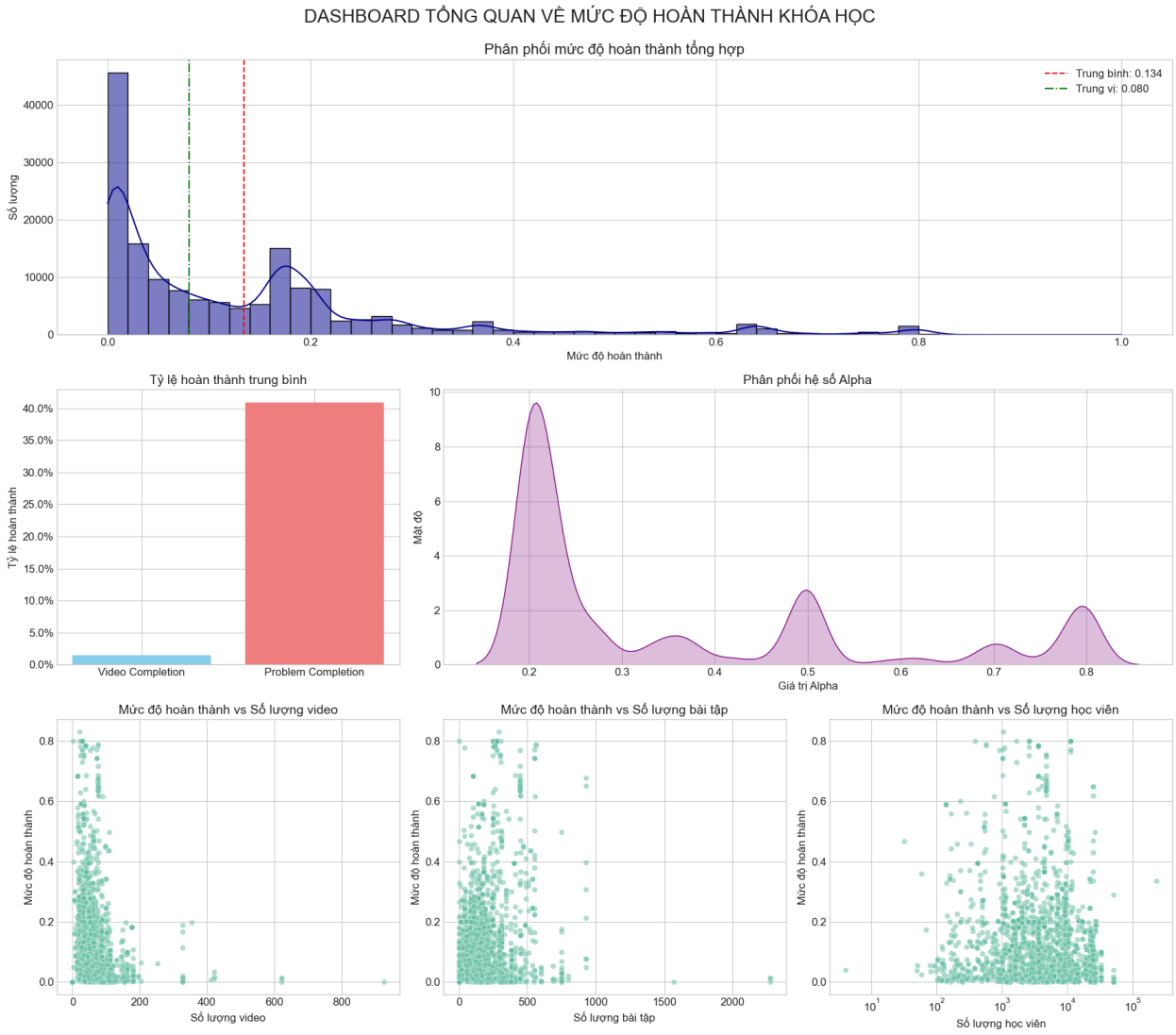
user_final_exam_score_ratio: Tỉ lệ điểm đạt được của user trên final_ex_maxScore.

Cách tính **user_final_exam_score_ratio** (Tổng quan):

- Đầu tiên là từ file course_ex_ids(file được mở rộng từ file course.json), ta lấy phần từ ex_id cuối cùng trong list exercise_ids làm last_ex_id rồi xuất thành dataset final_exam_id bao gồm 2 cột là course_id và last_ex_id(id của exercise cuối cùng).
- Tiếp đó ta dùng final_exam_id mới tạo ra dùng 2 trường course_id và last_ex_id để merge với file user_problem_final(file này chỉ cần lấy tất các cột id và cột score, adjust_score là đưa vào df).
- Sau khi đã có df merge với các cột user_id, course_id, ex_id, problem_id, score(đổi tên thành max_final_exam_score), adjusted_score(đổi tên thành user_score).
- Tiến hành groupby dựa theo user_id, course_id, ex_id với lấy sum cho cột user_score(cộng tất cả điểm làm bài của từng problem lại để thành điểm làm bài của user trên cả final_ex) và cộng cả cột max_final_exam_score(cộng tất cả điểm problem lại để thành điểm của final_ex).
- File final tên là **final_exam_scores** sẽ có 5 cột là user_id, course_id, ex_id, user_score, max_final_exam_score.
- Cuối cùng ta lấy cột user_score chia cho cột max_final_exam_score và lưu kết quả vào cột **user_final_exam_score_ratio**, cột này sẽ là output của dataset với khoảng giá trị hiễn nhiên là từ 0 đến 1.

Lý do sử dụng **user_final_exam_score_ratio:**

- Theo lẽ thường thì bài tập/exercise cuối cùng sẽ quan trọng và "mang tính dự đoán" cao hơn, có nghĩa càng là những bài tập cuối cùng, ta sẽ có thể dự đoán được năng lực của user rõ hơn. Bên cạnh đó bài tập cuối cùng cũng sẽ có thể mang tính bao hàm toàn khóa học.
- Từ việc sử dụng điểm làm bài exercise cuối cùng của user làm output, ta có thể dự đoán được khả năng hoàn thành khóa học. Một cách cảm tính mà nói, có thể bài tập cuối cùng là lựa chọn chưa thuyết phục, nhưng khả năng của cách này là cao nhất trong tất cả các cách. Vì thế nhóm sẽ thử nghiệm sử dụng output là user_final_exam_score_ratio làm khả năng hoàn thành khóa học của user.



Ý tưởng

Sử dụng bộ dữ liệu MOOCCubeX để xây dựng mô hình dự đoán mức độ hoàn thành khóa học của người dùng sử dụng các đặc điểm hành vi để dự đoán những học sinh nào có nguy cơ hoàn thành khóa học cao.

Các tập dữ liệu sẽ được dùng là: bộ dữ liệu dự đoán mức độ hoàn thành khóa học. Từ những tập dữ liệu trên, nhóm sẽ chủ yếu dựa vào hành vi, sự tương tác của người dùng với các khóa học để trích xuất ra các đặc trưng hữu ích, ngoài ra còn xem xét sử dụng thêm các đặc trưng từ thông tin người dùng cũng như khóa học, chẳng hạn như:

- Các đặc trưng liên quan đến người dùng
- Các đặc trưng liên quan đến khóa học
- Các đặc trưng trích xuất từ lịch sử hoạt động người dùng

Nhiệm vụ của đề tài là nghiên cứu xây dựng các mô hình sử dụng kỹ thuật học có giám sát để phân tích các yếu tố ảnh hưởng đến việc hoàn thành khóa học, từ đó dự đoán mức độ hoàn thành khóa học của người dùng.

Tính cấp thiết

Tính cấp thiết của đề tài được thể hiện trên nhiều khía cạnh:

- *Giảm khả năng bỏ học của học viên:* Việc dự đoán khả năng bỏ học sớm giúp các nhà giáo dục và nhà quản lý MOOCs có thể thực hiện các biện pháp can thiệp kịp thời để hỗ trợ học viên có nguy cơ bỏ học, từ đó tăng cơ hội cho họ hoàn thành khóa học.
- *Tối ưu hóa tài nguyên giáo dục*: Việc hiểu và dự đoán mức độ hoàn thành khóa học giúp các tổ chức MOOCs tối ưu hóa việc sử dụng tài nguyên giáo dục, giảm thiểu lãng phí và tăng hiệu quả.
- *Cải thiện trải nghiệm học tập*: Nhận biết và can thiệp sớm vào vấn đề bỏ học có thể cải thiện trải nghiệm học tập của các học viên bằng cách cung cấp hỗ trợ phù hợp và cá nhân hóa.
- *Nâng cao chất lượng khóa học*: Việc phân tích và dự đoán mức độ hoàn thành khóa học cũng giúp các nhà giáo dục hiểu rõ hơn về những yếu tố nào ảnh hưởng đến sự thành công của một khóa học và từ đó cải thiện chất lượng của chúng.
- *Tiết kiệm chi phí và tăng hiệu quả*: Bằng cách ngăn chặn hoặc giảm bớt tỷ lệ bỏ học, tổ chức MOOCs có thể tiết kiệm được chi phí về việc quảng cáo và tái học viên mới, cũng như tăng hiệu suất sử dụng tài nguyên giáo dục.

Tính mới

- *Xem xét yếu tố về môi trường học tập*: Trong môi trường học tập trực tuyến, nhóm không chỉ tập trung vào hành vi truy cập nội dung mà còn đưa vào xem xét các yếu tố môi trường như tuổi tác, bằng cấp và giới tính của sinh viên. Việc này mở ra cơ hội để hiểu rõ hơn về đa dạng của sinh viên và tác động của các yếu tố này đến trải nghiệm học tập của họ. Như vậy, việc tối ưu hóa quá trình giảng dạy và hỗ trợ sinh viên sẽ trở nên hiệu quả hơn thông qua việc cá nhân hóa và đáp ứng nhu cầu học tập cụ thể của từng sinh viên.
- *Kết hợp nhiều nguồn dữ liệu*: Sử dụng thông tin từ nhiều nguồn dữ liệu khác nhau bao gồm hành vi sử dụng khóa học, thông tin người dùng và thông tin về khóa học, từ đó tạo ra một bức tranh toàn diện về người học và môi trường học tập.
- *Trích xuất đặc trưng đa chiều*: Đề xuất sử dụng các đặc trưng đa chiều như đặc trưng liên quan đến người dùng, đặc trưng liên quan đến khóa học và đặc trưng

trích xuất từ lịch sử hành vi người dùng. Điều này giúp tạo ra một bức tranh phức tạp và đa chiều về hành vi học tập.

IV. Mục tiêu đề tài

- Xây dựng mô hình dự đoán mức độ hoàn thành khóa học của học viên.
- Đánh giá hiệu quả của các mô hình dự đoán khác nhau.
- Xác định các yếu tố ảnh hưởng đến việc hoàn thành khóa học.
- Xây dựng ứng dụng hiển thị thông tin dự đoán mức độ hoàn thành khóa học của người dùng cho nhà quản lý giáo dục.
- Đề xuất các biện pháp can thiệp để giảm thiểu tỷ lệ bỏ học.

V. Nội dung và phương pháp thực hiện

1. Mục tiêu

- Khám phá bộ dữ liệu, các thông tin, chỉ số, mối quan hệ giữa các biến trong bộ dữ liệu. Từ đó, xác định được các đặc trưng có thể khai thác và xử lý.
- Làm sạch dữ liệu, xử lý các dữ liệu có điểm khuyết thiếu như dữ liệu bị thiếu, dữ liệu ngoại lai, giá trị Null hoặc sai lệch về mặt logic,...
- Đảm bảo tính nhất quán, hợp lệ, đầy đủ và chính xác của dữ liệu.
- Khai thác được những đặc trưng hữu ích cho bài toán dự đoán khả năng bỏ học từ dữ liệu.
- Quản lý và sắp xếp dữ liệu, đảm bảo tính bảo mật và khả năng truy xuất của dữ liệu.

2. Sản phẩm

Sản phẩm của dự án là một ứng dụng web quản lý và dự đoán mức độ hoàn thành khóa học của sinh viên. Với giao diện người dùng trực quan, dễ sử dụng, ứng dụng cho phép nhà quản lý giáo dục theo dõi và quản lý thông tin chi tiết của sinh viên, được phân loại theo các khóa học mà họ đang theo học. Một trong những tính năng nổi bật của ứng dụng là khả năng dự đoán mức độ hoàn thành khóa học của sinh viên, dựa trên mô hình máy học đã được huấn luyện kỹ lưỡng.

Dữ liệu về sinh viên và các kết quả dự đoán được lưu trữ trong cơ sở dữ liệu, đảm bảo tính nhất quán và bảo mật. Nhờ vào những tính năng này, sản phẩm không chỉ cung cấp một công cụ hiệu quả cho việc quản lý sinh viên mà còn hỗ trợ nhà quản lý trong việc nâng cao chất lượng giáo dục.

3. Phương pháp thực hiện

a. Phân tích và quản lý dữ liệu ban đầu

Mục tiêu

- Khám phá bộ dữ liệu, các thông tin, chỉ số, mối quan hệ giữa các biến trong bộ dữ liệu. Từ đó, xác định được các đặc trưng có thể khai thác và xử lý.
- Làm sạch dữ liệu, xử lý các dữ liệu có điểm khuyết thiếu như dữ liệu bị thiếu, dữ liệu ngoại lai, giá trị Null hoặc sai lệch về mặt logic,...
- Đảm bảo tính nhất quán, hợp lệ, đầy đủ và chính xác của dữ liệu.
- Khai thác được những đặc trưng hữu ích cho bài toán dự đoán khả năng bỏ học từ dữ liệu.
- Quản lý và sắp xếp dữ liệu, đảm bảo tính bảo mật và khả năng truy xuất của dữ liệu.

Phương pháp thực hiện

Hợp nhất dữ liệu:

- Đọc file user.json và trích xuất tất cả các cặp (user_id, course_id) từ trường course_order của mỗi người dùng. Đây sẽ là 2 id chính cho dataframe hợp nhất.
- Loại bỏ các dòng liên quan đến người dùng tham gia cực nhiều khóa học: Tất cả các dòng dữ liệu (user_id, course_id) thuộc về những người dùng đã đăng ký nhiều hơn 20 khóa học đã bị loại bỏ.
- Loại bỏ các dòng liên quan đến khóa học rất ít người đăng ký: Tất cả các dòng dữ liệu thuộc về những khóa học có ít hơn 2 người dùng đăng ký đã bị loại bỏ.
- Sau khi áp dụng các tiêu chí lọc cải tiến trên, số dòng dữ liệu giảm từ 11,807,090 xuống còn 6,310,440 dòng (giữ lại 55.21%). DataFrame đã lọc này đã lọc ra các user_id, course_id đủ quan trọng để đánh giá, tiếp tục tiến hành các bước hợp nhất và xử lí dữ liệu tiếp theo.

Tạo output:

- Tính Tỷ lệ Hoàn thành Video (video_completion_ratio):
    - Đo lường mức độ người dùng đã xem video trong khóa học.

    - Công thức cơ bản : (Tổng số video đã xem của user) / (Tổng số video mặc định của khóa học)
    - Tính toán điều này bằng cách dùng dữ liệu từ user_video_interaction.csv
    - Giá trị này thường được giới hạn ở mức tối đa là 1 (hoặc 100%).
- Tính Tỷ lệ Hoàn thành Bài tập/Vấn đề (problem_ratio):
    - Đo lường mức độ người dùng đã tương tác hoặc hoàn thành các bài tập/vấn đề.
    - Tính dựa trên: min(1, Tổng số bài tập đã làm của user / Tổng số bài tập trong khóa học)
    - Xác định Trọng số (Alpha - α): Đây là yếu tố quan trọng để cân bằng giữa video và bài tập. Trọng số này phụ thuộc vào cấu trúc của khóa học.
    - Tính tỷ lệ: ratio = num_problems / (num_videos + epsilon) (epsilon để tránh chia cho 0).

      Chuẩn hóa tỷ lệ về khoảng [0, 1): alpha = ratio / (1 + ratio).

      Xử lý trường hợp đặc biệt:

      Nếu chỉ có bài tập (num_videos == 0), alpha = 1.

      Nếu chỉ có video (num_problems == 0), alpha = 0.

      Nếu không có cả hai, alpha = 0.5 (trung lập).

- Tính Điểm Hoàn thành Tổng hợp (composite_completion):
    - Công thức chung:

composite_completion = (alpha * problem_completion_ratio) + ((1 - alpha) * video_completion_ratio)

Trong đó:

- alpha là trọng số cho việc hoàn thành bài tập.
- (1 - alpha) là trọng số cho việc hoàn thành video.

b. Triển khai mô hình

Mục tiêu

Huấn luyện mô hình máy học có hiệu suất cao nhất trên bài toán của nhóm.

Phương pháp thực hiện

- Các tham số và độ đo chung dùng trong quá trình train:
  - Tham số khi dùng grid search:
  - pipeline: Đối tượng Pipeline chứa các bước xử lý dữ liệu
  - param_grid: Lưới tham số cần tìm kiếm
  - cv: Chiến lược cross-validation (KFold với n_splits=3)
  - scoring: Cách tính điểm để đánh giá mô hình ('neg_mean_squared_error' - âm của MSE)
  - verbose: Mức độ hiển thị thông tin (1: hiển thị tiến trình cơ bản)
  - n_jobs: Số lượng CPU cores sử dụng (-1: sử dụng tất cả các cores)
  - Độ đo sử dụng:
  - RMSE: Căn bậc hai của trung bình bình phương sai số, đo lường sai số dự đoán
  - MAE: Trung bình sai số tuyệt đối, ít bị ảnh hưởng bởi các ngoại lai
  - R^2: Hệ số xác định, cho biết mô hình giải thích được bao nhiêu phần trăm biến thiên của dữ liệu
  - MAPE: Trung bình phần trăm sai số tuyệt đối, cho biết mức độ sai lệch theo tỷ lệ phần trăm

1.1. Train mô hình với thuật toán Random Forest Regressor

- Loại mô hình:
  - Học máy dựa trên tập hợp. Kết hợp nhiều Decision Trees để tạo ra một mô hình mạnh mẽ hơn.
  - Sử dụng phương pháp bootstrap để tạo ra các tập dữ liệu con.
- Nguyên lý hoạt động:
  - Tạo ra nhiều Decision Trees (cây quyết định) khác nhau.
  - Mỗi cây được huấn luyện trên một tập dữ liệu con được tạo ra bằng phương pháp bootstrap.
  - Dự đoán của mô hình là kết quả đa số (majority vote) của các cây.
- Ưu điểm:
  - Hiệu quả cao với nhiều loại dữ liệu.
  - Khả năng khái quát tốt. Chống quá khớp (overfitting).
  - Dễ dàng điều chỉnh tham số.
  - Khả năng giải thích tốt.

- Nhược điểm:
    - Tốc độ huấn luyện chậm hơn Decision Trees.
    - Khó lựa chọn số lượng cây.
    - Kích thước mô hình có thể lớn.
- Các bước thực hiện:

Bước 1: Tiền xử lý dữ liệu

- Đọc dữ liệu từ nguồn JSON và CSV.

- Gộp dữ liệu từ nhiều bảng liên quan đến học viên, khóa học, video, và bài tập.

- Trích xuất và tính toán các đặc trưng như: video_completion_ratio, problem_ratio, avg_completion_rate, v.v.

- Mã hóa các trường phân loại (field, school, prerequisites,...) bằng kỹ thuật Label Encoding hoặc One-hot Encoding.

- Chuẩn hóa dữ liệu (nếu cần) để đảm bảo đồng đều về thang đo.

Bước 2: Tạo tập huấn luyện và kiểm thử

- Chia dữ liệu thành:

+ Tập huấn luyện + validation

+ Tập kiểm thử riêng biệt (hold-out test set)

- Sử dụng K-Fold Cross-Validation để đánh giá độ ổn định mô hình trên tập huấn luyện + validation.

Bước 3: Huấn luyện mô hình

- Sử dụng mô hình RandomForestRegressor từ thư viện sklearn.

- Huấn luyện trên tập train với các đặc trưng đã chọn.

- Kiểm tra độ chính xác và hiệu suất mô hình qua các vòng cross-validation.

Bước 4: Đánh giá mô hình trên tập kiểm thử

- Dự đoán trên tập test.

- Tính toán các chỉ số đánh giá: RMSE, MAE, R^2, MAPE.

- Kiểm tra mức độ sai số và khả năng tổng quát hóa của mô hình.

Bước 5: Phân tích tầm quan trọng của đặc trưng

- Trích xuất giá trị feature importance từ mô hình Random Forest.

- Vẽ biểu đồ thể hiện mức độ đóng góp của từng đặc trưng đến kết quả dự đoán.

- Nhận diện các đặc trưng ảnh hưởng chính (ví dụ: thời gian xem video, chỉ số alpha,...).

Demo

- Giải thích tham số:

  - Mô hình sử dụng: Random Forest với số lượng cây mặc định, không tinh chỉnh sâu, tập trung kiểm chứng khả năng mô hình hóa hành vi học tập.

- Kết quả mô hình:
  Kết quả đánh giá mô hình cho thấy hiệu suất dự đoán rất cao:

Độ đo	Giá trị	Giải thích
RSME (Test)	0.0029	Sai số trung bình bình phương rất nhỏ
MAE (Test)	0.0004	Sai số tuyệt đối trung bình gần bằng 0
R^2 Score (Test)	0.9997	Mô hình giải thích tới 99.97% phương sai dữ liệu
MAPE (Test)	0.3584%	Sai số phần trăm thấp, độ chính xác cao
Cross-Validation RMSE std	0.0006	Mô hình ổn định, không overfit

```python
# Đánh giá
rmse = mean_squared_error(y_test, y_pred, squared=False)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
mape = mean_absolute_percentage_error(y_test, y_pred)

print(f"🔹 RMSE (Test): {rmse:.4f}")
print(f"🔹 MAE (Test): {mae:.4f}")
print(f"🔸 R² Score (Test): {r2:.4f}")
print(f"🔸 MAPE (Test): {mape:.4%}")
```

```
🔹 RMSE (Test): 0.0029
🔹 MAE (Test): 0.0004
🔸 R² Score (Test): 0.9997
🔸 MAPE (Test): 0.3584%
```

Feature Importance:

- Đặc trưng quan trọng nhất là total_video_watch_time_1 (~45%), cho thấy việc xem video có ảnh hưởng lớn đến khả năng hoàn thành.

- alpha chiếm tỷ trọng cao (~35%), đóng vai trò như một chỉ báo tiềm năng tổng hợp hành vi người học.

- Các đặc trưng khác như video_completion_ratio, total_completion_rate góp phần bổ sung nhưng không chi phối chính.

Kết luận
Mô hình Random Forest đã thể hiện độ chính xác và độ ổn định rất cao trong việc dự đoán khả năng hoàn thành khóa học. Các đặc trưng liên quan đến thời lượng xem video và chỉ số tổng hợp hành vi đóng vai trò quan trọng nhất. Đây là nền tảng vững chắc để phát triển hệ thống gợi ý hoặc cảnh báo sớm cho học viên có nguy cơ không hoàn thành.

1.2. Train mô hình với thuật toán Gradient Boosting

- Loại mô hình:
  - Học máy dựa trên tập hợp: Gradient Boosting là một kỹ thuật học máy mạnh mẽ kết hợp nhiều mô hình yếu (weak learners) để tạo thành một mô hình mạnh hơn.

    - Sử dụng phương pháp boosting: Mỗi mô hình mới được huấn luyện để sửa lỗi của các mô hình trước đó, dần dần cải thiện độ chính xác.
- Nguyên lý hoạt động:
    - Tạo ra nhiều mô hình yếu: Thường là các cây quyết định nông (shallow decision trees).
    - Huấn luyện mô hình mới để sửa lỗi: Mỗi cây mới cố gắng giảm thiểu sai số còn lại (residual errors) của các cây trước đó.
    - Cộng dồn dự đoán: Tổng dự đoán của tất cả các cây để đưa ra kết quả cuối cùng.
- Ưu điểm:
    - Hiệu quả cao: Gradient Boosting thường đạt hiệu suất cao trên nhiều loại dữ liệu và bài toán khác nhau.
    - Khả năng khái quát tốt: Dù có thể phức tạp hơn, Gradient Boosting có khả năng khái quát tốt hơn và ít bị overfitting hơn so với một số thuật toán khác.
    - Linh hoạt: Có thể sử dụng cho cả bài toán hồi quy và phân loại.
    - Điều chỉnh tham số: Có nhiều tham số để điều chỉnh, giúp mô hình phù hợp với từng bộ dữ liệu cụ thể.
- Nhược điểm:
    - Tốc độ huấn luyện chậm: Do tính chất tuần tự, Gradient Boosting thường chậm hơn so với một số thuật toán khác như Random Forest.
    - Khó khăn trong việc điều chỉnh tham số: Có nhiều tham số cần điều chỉnh để đạt hiệu suất tối ưu, điều này có thể phức tạp và tốn nhiều thời gian.
    - Dễ bị overfitting nếu không điều chỉnh đúng: Nếu không cẩn thận trong việc điều chỉnh tham số, mô hình có thể dễ bị overfitting.
- Các bước thực hiện:

Demo

- Giải thích tham số:

- N_estimators: số lượng cây trong mô hình boosting
- Learning_rate: Dùng để giảm tác động của mỗi cây mới thêm vào mô hình
- Max_depth: Độ sâu tối đa của mỗi cây quyết định
- Subsample: Tỉ lệ mẫu huấn luyện sử dụng cho mỗi cây
- Min_sample_split: Số lượng mẫu cần thiết để chia một node
- Mục tiêu dự đoán là một chỉ số tổng hợp (ví dụ: composite_completion_score) phản ánh khả năng hoàn thành khóa học của học viên.

- Kết quả mô hình:

Đánh giá Reliability (Mức độ tin cậy của dữ liệu và mô hình)

Kết quả kiểm tra lỗi:

- Test RMSE: 0.0031 (rất thấp)
- Test MAE: 0.0014 (rất thấp)

=> Sai số cực nhỏ cho thấy mô hình dự đoán chính xác, dữ liệu có độ tin cậy rất cao.

Kiểm tra độ ổn định qua Cross-Validation:

- Validation RMSE: 0.0033 ≈ Test RMSE: 0.0031
- Validation MAE: 0.0015 ≈ Test MAE: 0.0014

=> Chênh lệch giữa tập kiểm tra và validation là rất nhỏ → mô hình ổn định và đáng tin cậy.

Đánh giá Relevance (Mức độ liên quan của đặc trưng đến kết quả dự đoán)

R^2 Score:

- $R^2 = 0.9996$ (gần như tuyệt đối)

=> Gần như toàn bộ phương sai của biến mục tiêu được giải thích bởi dữ liệu đầu vào → mô hình rất phù hợp.

MAPE (Mean Asolute Percentage Error):

- MAPE = 3b.83% (thấp hơn nhiều so với 20%)

=> Sai lệch phần trăm thấp → mô hình có độ chính xác cao ngay cả khi xét về tỷ lệ phần trăm dự đoán.

Feature Importance:

Top đặc trưng quan trọng:

- problem_ratio: 45.0%
- alpha: 41.3%
- video_completion_ratio: 6.2%
- avg_problem_attempts_per_student: 3.1%

Một số đặc trưng có độ quan trọng thấp:

- num_teacher, num_videos, num_exercises,...

=> Những đặc trưng như num_teacher, num_videos có độ quan trọng < 1% → có thể cân nhắc loại bỏ để đơn giản mô hình mà không ảnh hưởng đáng kể đến kết quả.

Kết luận

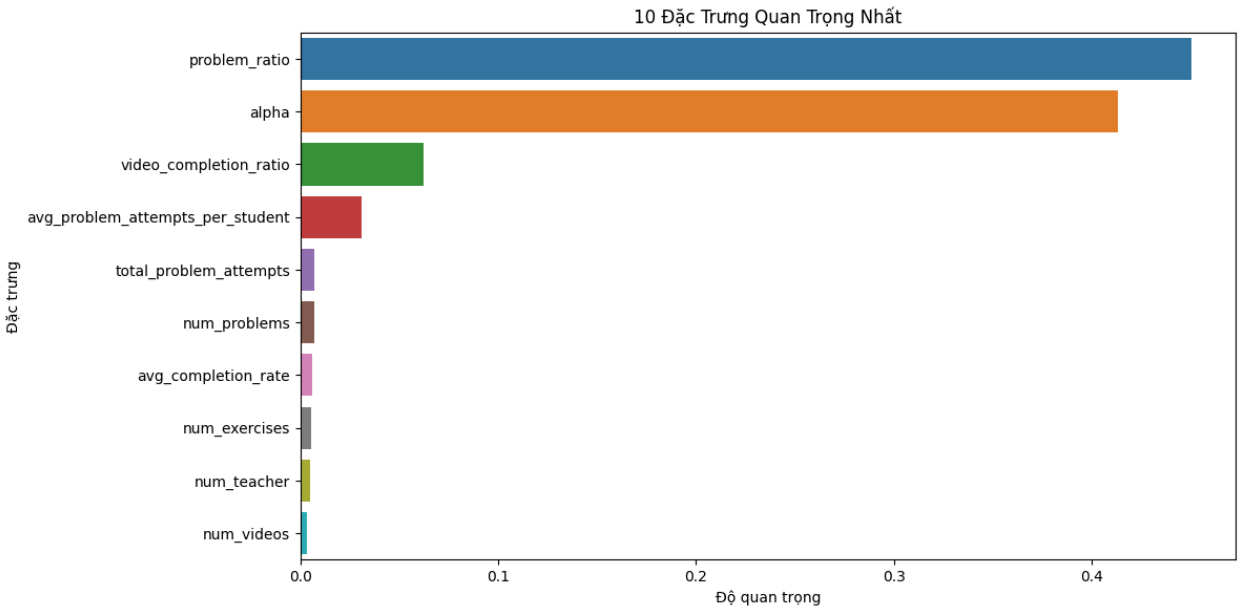
Reliability: Rất cao. Mô hình cho sai số rất thấp và ổn định giữa các tập → đáng tin cậy.

Relevance: Xuất sắc. R^2 gần bằng 1, MAPE rất thấp → mô hình sử dụng các đặc trưng hiệu quả.

```
[Validation]
Best params: {'learning_rate': 0.1, 'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 20
0, 'subsample': 1.0}
Validation RMSE: 0.0033
Validation MAE : 0.0015

[Test Final Evaluation]
Test RMSE: 0.0031
Test MAE : 0.0014
R2 Score  : 0.9996
MAPE      : 3.83%

Top 10 đặc trưng quan trọng:
                         Đặc trưng  Độ quan trọng
26                    problem_ratio       0.450366
30                            alpha       0.413466
0             video_completion_ratio       0.062204
15  avg_problem_attempts_per_student       0.031045
14            total_problem_attempts       0.006945
7                      num_problems       0.006821
17              avg_completion_rate       0.006108
6                    num_exercises       0.005321
1                      num_teacher       0.004879
5                      num_videos       0.003297
```



1.3. Train mô hình với thuật toán Support Vector Regression

- Loại mô hình:
  - Học máy có giám sát.
  - Biến thể của Support Vector Machines (SVM) dùng cho bài toán hồi quy thay vì phân loại.
- Nguyên lý hoạt động:
  - SVR cố gắng tìm một siêu phẳng (hyperplane) sao cho các điểm dữ liệu nằm trong một "khoảng chấp nhận được" (epsilon tube) quanh siêu phẳng này.
  - Mô hình chỉ quan tâm đến các điểm nằm ngoài epsilon – gọi là các "vector hỗ trợ" (support vectors).
  - Mục tiêu là tối thiểu hóa sai số của các điểm nằm ngoài epsilon, đồng thời giữ cho siêu phẳng càng đơn giản càng tốt (minimize độ phức tạp mô hình).

- Ưu điểm:
  - Hoạt động tốt với dữ liệu có chiều cao, không tuyến tính (với kernel phi tuyến).
  - Khả năng khái quát tốt, chống overfitting.
  - Có thể kiểm soát sai số chấp nhận được (epsilon).

- Nhược điểm:
  - Khó chọn được kernel và tham số tối ưu.
  - Tốn tài nguyên tính toán nếu dữ liệu lớn.
  - Nhạy cảm với dữ liệu nhiễu nếu không chọn epsilon và C phù hợp.
- Các bước thực hiện:

  Bước 1: Tiền xử lý dữ liệu

  - Đọc dữ liệu từ file CSV user_course_manager_v0_update.csv.
  - Kiểm tra thông tin cơ bản của dữ liệu với df.info() và df.describe().
  - Kiểm tra giá trị khuyết trong dữ liệu bằng df.isna().sum().
  - Tách đặc trưng và nhãn:
    + Biến mục tiêu (y): composite_completion
    + Đặc trưng (X): tất cả các cột trừ composite_completion, user_id và course_id
  - Xử lý giá trị khuyết bằng cách điền giá trị trung bình vào các cột có giá trị khuyết.
  - Kiểm tra lại để đảm bảo không còn giá trị khuyết sau khi xử lý.

  Bước 2: Tạo tập huấn luyện, validation và kiểm thử

- Chia dữ liệu thành tập huấn luyện (60%), validation (20%) và kiểm thử (20%).
- Thực hiện chia theo 2 bước:
    + Đầu tiên: chia 80% cho train+val và 20% cho test
    + Tiếp theo: chia 75% cho train và 25% cho val từ tập train+val (tương đương 60% và 20% của toàn bộ dữ liệu)
- In kích thước của từng tập dữ liệu và tỷ lệ so với tổng số.

Bước 3: Huấn luyện mô hình
- Sử dụng mô hình SVR (Support Vector Regression) từ thư viện sklearn.
- Tạo pipeline kết hợp StandardScaler và SVR để chuẩn hóa dữ liệu trước khi huấn luyện.
- Thực hiện tìm kiếm siêu tham số tối ưu bằng GridSearchCV với các tham số:
    + C: [1, 10] (hệ số điều chỉnh mức độ phạt)
    + epsilon: [0.01, 0.1] (sai số cho phép trong hàm mất mát)
    + kernel: ['rbf'] (hàm nhân phi tuyến)
- Sử dụng KFold với 3 fold cho cross-validation.
- In thông số tốt nhất và điểm cross-validation tốt nhất.

Bước 4: Đánh giá mô hình
- Định nghĩa hàm evaluate_model để tính toán và in các chỉ số đánh giá:
    + RMSE (Root Mean Squared Error): Đánh giá sai số trung bình theo phương sai
    + MAE (Mean Absolute Error): Đánh giá sai số trung bình tuyệt đối
    + R^2 (Hệ số xác định): Đánh giá mức độ giải thích của mô hình
    + MAPE (Mean Absolute Percentage Error): Đánh giá sai số trung bình theo tỷ lệ phần trăm
- Đánh giá mô hình trên cả 3 tập: huấn luyện, validation và kiểm thử.

Bước 5: Đánh giá độ ổn định của mô hình
- Thực hiện cross-validation để đánh giá độ ổn định của mô hình.
- Tính toán các độ đo RMSE, MAE và R^2 trên các fold khác nhau.
- Tính và in giá trị trung bình và độ lệch chuẩn cho mỗi độ đo để đánh giá độ ổn định.

Bước 6: Phân tích tầm quan trọng của đặc trưng
- Sử dụng kỹ thuật permutation importance để đánh giá tầm quan trọng của các đặc trưng.

- Sắp xếp các đặc trưng theo thứ tự tầm quan trọng giảm dần.
- Trực quan hóa tầm quan trọng của đặc trưng bằng biểu đồ thanh ngang.
- Hiển thị và lưu trữ kết quả phân tích tầm quan trọng của đặc trưng.

Demo

- Giải thích tham số:

- C: Tham số điều chỉnh mức độ phạt đối với sai số. Giá trị C càng lớn, mô hình càng cố gắng khớp với dữ liệu huấn luyện (có thể dẫn đến overfitting). Ngược lại, C nhỏ sẽ tạo mô hình đơn giản hơn.

- Trong code đang thử nghiệm với C = 1 và C = 10
- epsilon: Định nghĩa độ rộng của "vùng không nhạy cảm" xung quanh hàm dự đoán. Chỉ những điểm dữ liệu có sai số lớn hơn epsilon mới bị "phạt".

- Trong code đang thử nghiệm với epsilon = 0.01 và 0.1

kernel: Hàm kernel chuyển đổi dữ liệu sang không gian có chiều cao hơn.

'- rbf': Radial Basis Function, phù hợp với dữ liệu phi tuyến tính

- Kết quả mô hình:

ĐÁNH GIÁ ĐỘ TIN CẬY (RELIABILITY)

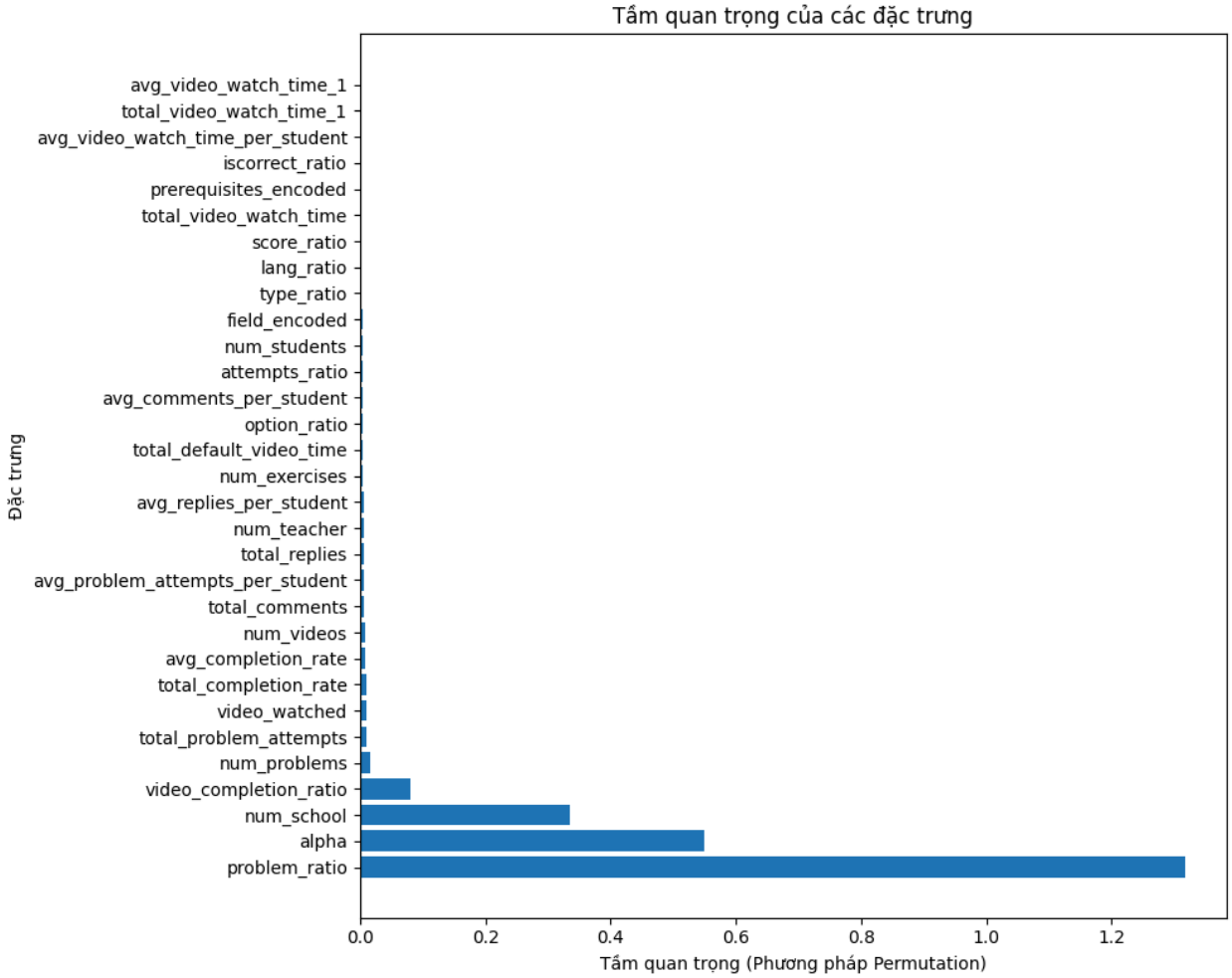
RMSE (Tập kiểm thử): 0.0068
MAE (Tập kiểm thử): 0.0050
Độ ổn định qua Cross-Validation:
- RMSE CV: 0.0083 ± 0.0002
- MAE CV: 0.0050 ± 0.0001

Kết luận về độ tin cậy: Độ ổn định cao qua các fold, cho thấy dữ liệu đáng tin cậy

ĐÁNH GIÁ ĐỘ LIÊN QUAN (RELEVANCE)

Hệ số xác định R^2 (Tập kiểm thử): 0.9983
MAPE (Tập kiểm thử): 390038933.23%
Top 5 đặc trưng quan trọng nhất:
- problem_ratio: 1.3176
- alpha: 0.5491
- num_school: 0.3357

- video_completion_ratio: 0.0794
- num_problems: 0.0168



Đánh giá từ R^2: Độ liên quan cao - các đặc trưng giải thích tốt biến mục tiêu
Đánh giá từ MAPE: Dữ liệu có thể không liên quan (MAPE > 20%)

ĐÁNH GIÁ TỔNG THỂ

Mô hình có hiệu suất yếu, cho thấy có thể có vấn đề với độ liên quan hoặc độ tin cậy của dữ liệu

1.4. Train mô hình với thuật toán Decision Trees Regression

- Loại mô hình:
  - Học máy dựa trên quy tắc
  - Học mô hình dạng cây để phân chia dữ liệu.Dựa trên các câu hỏi để đưa ra dự đoán

- Nguyên lý hoạt động:
    - Xây dựng một cây quyết định để phân chia dữ liệu thành các lớp khác nhau. Cây bao gồm các nút (node) và nhánh (branch). Mỗi nút đại diện cho một câu hỏi về một thuộc tính của dữ liệu. Mỗi nhánh đại diện cho một câu trả lời cho câu hỏi. Dữ liệu được phân chia theo các nhánh cho đến khi đến lá (leaf). Lớp của lá được chọn là dự đoán cho dữ liệu
- Ưu điểm:
    - Đơn giản, dễ hiểu. Dễ giải thích.
    - Hiệu quả với tập dữ liệu nhỏ. Ít yêu cầu về dữ liệu.
- Nhược điểm:
    - Khả năng khái quát hạn chế.
    - Dễ bị quá khớp (overfitting).
    - Khó lựa chọn điểm cắt (threshold).
    - Kích thước cây có thể lớn
- Các bước thực hiện:

Demo

- Giải thích tham số:

- Kết quả mô hình:

1.5. Train mô hình với thuật toán K-Nearest Neighbors Regression

- Loại mô hình: K-Nearest Neighbors Regression

- Nguyên lý hoạt động:

- KNN Regression là mô hình dự đoán giá trị của một mẫu dựa trên giá trị trung bình của k mẫu gần nhất.
- Mô hình tính toán khoảng cách từ mẫu cần dự đoán đến tất cả các mẫu trong tập huấn luyện.
- K mẫu có khoảng cách gần nhất được chọn làm 'hàng xóm'.
- Giá trị dự đoán là giá trị trung bình (có thể có trọng số) của các giá trị mục tiêu từ k hàng xóm đó.

- Ưu điểm:

- Mô hình đơn giản, dễ hiểu và triển khai.
- Hoạt động tốt với dữ liệu không nhiễu và có cấu trúc rõ ràng.
- Tự nhiên thích ứng với dữ liệu mới mà không cần huấn luyện lại.

- Nhược điểm:

- Hiệu suất giảm khi số lượng đặc trưng tăng (curse of dimensionality).
- Cần lưu trữ toàn bộ tập dữ liệu huấn luyện, tốn bộ nhớ.
- Thời gian dự đoán chậm với dữ liệu lớn, vì phải tính khoảng cách đến tất cả các mẫu.
- Nhạy cảm với tỷ lệ thang đo của các đặc trưng, đòi hỏi chuẩn hóa dữ liệu.

- Các bước thực hiện:

1. Đọc và khám phá dữ liệu

2. Tiền xử lý dữ liệu: loại bỏ cột định danh, chuẩn hóa đặc trưng, xử lí giá trị ngoại lai, nhiễu, thiếu,...

3. Chia dữ liệu thành tập train, validation và test (60:20:20)

```
Kích thước tập huấn luyện (train): 95051 mẫu (60.0%)
Kích thước tập kiểm định (validation): 31684 mẫu (20.0%)
Kích thước tập kiểm tra (test): 31684 mẫu (20.0%)
```

4. Xây dựng mô hình KNN cơ bản (n_neighbors = 5) và đánh giá hiệu suất ban đầu

```
Đánh giá mô hình cơ bản trên tập validation:
RMSE: 0.0167
MAE: 0.0051
R² Score: 0.9894
MAPE: 21.78%
```

5. Tinh chỉnh siêu tham số với GridSearchCV

```
param_grid = {
    'n_neighbors': [3, 5, 7, 9, 11, 15, 20],
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan', 'minkowski']
}
```

```
Siêu tham số tốt nhất:
metric: euclidean
n_neighbors: 3
weights: distance
```

6. Đánh giá mô hình với siêu tham số tốt nhất trên tập validation

```
Đánh giá mô hình tốt nhất trên tập validation:
RMSE: 0.0147
MAE: 0.0039
R² Score: 0.9918
MAPE: 14.98%
```

c. Triển khai ứng dụng

Mục tiêu

- Quản lý sinh viên hiệu quả: tổ chức và hiển thị danh sách sinh viên dựa trên các khóa học mà họ đang theo học, quản lý thông tin chi tiết của sinh viên.
- Dự đoán khả năng bỏ học: sử dụng các mô hình dự đoán để ước tính mức độ hoàn thành của sinh viên, từ đó giúp nhà quản lý can thiệp kịp thời và hiệu quả.

Phương pháp thực hiện

VI. Kết quả dự kiến, sản phẩm đề tài

Kết quả dự kiến

Nhóm đặt ra các mục tiêu cụ thể đồ án đạt được kết quả nhất định như sau:

- Xây dựng mô hình dự đoán mức độ hoàn thành khóa học: Mục tiêu là phát triển một mô hình dự đoán chính xác mức độ bỏ học của sinh viên dựa trên dữ liệu lịch

sử hoạt động học tập của họ. Mô hình này cần được huấn luyện và đánh giá kỹ lưỡng để đảm bảo tính chính xác và ứng dụng được trong thực tế.

- Tìm hiểu được các yếu tố ảnh hưởng đến việc hoàn thành khóa học giúp ích cho các nhà quản lý: Mục tiêu là phân tích và hiểu rõ các yếu tố tác động đến quyết định của sinh viên khi họ quyết định bỏ học. Các yếu tố này có thể bao gồm mức độ tham gia vào các hoạt động học tập, mức độ hoàn thành nhiệm vụ, độ khó của khóa học, và các yếu tố tâm lý hoặc xã hội khác.

Sản phẩm đề tài

Sản phẩm cuối cùng của đồ án là một ứng dụng web quản lý sinh viên và dự đoán mức độ hoàn thành khóa học của họ. Sản phẩm này sẽ có các tính năng chính sau:

- Quản lý sinh viên hiệu quả: Ứng dụng sẽ cung cấp cho nhà quản lý giáo dục một công cụ để tổ chức và hiển thị dashboard danh sách sinh viên dựa trên các khóa học mà họ đang theo học, cũng như quản lý thông tin chi tiết của từng sinh viên.
- Dự đoán mức độ hoàn thành khóa học: Sản phẩm sẽ sử dụng một mô hình dự đoán đã được huấn luyện kỹ lưỡng để ước tính mức độ hoàn thành khóa học của sinh viên. Điều này giúp nhà quản lý can thiệp kịp thời và hiệu quả để giảm thiểu tỷ lệ bỏ học.
- Sản phẩm có thể ưu trữ dữ liệu an toàn và nhất quán chứa các thông tin về sinh viên và các kết quả dự đoán và có thể cung cấp các phân tích báo cáo cho các nhà quản lý hiểu được các thông tin quan trọng.

VII. Tài liệu tham khảo

Bổ sung sau