**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 

**IS353.P12 - MẠNG XÃ HỘI**

**GVHD: Th.S Nguyễn Thị Anh Thư**

# 

# BÁO CÁO ĐỒ ÁN

**ĐÁNH GIÁ ĐỘ HIỆU QUẢ CỦA KHÓA HỌC**

| Nhóm sinh viên thực hiện: Nhóm 2 | | |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Nguyễn Trần Vũ Quang | 21521347 |
| 2 | Nguyễn Hoàng Phúc | 21522473 |
| 3 | Nguyễn Thị Mai Trinh | 21522718 |
| 4 | Lê Công Hoài Nam | 22520910 |
| 5 | Lê Thanh Tài | 22521276 |
| 6 | Nguyễn Văn Thắng | 22520005 |

TP. HỒ CHÍ MINH, tháng 12 năm 2024

**NHẬN XÉT CỦA GIÁO VIÊN**

................................................................................................................................................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................

………...……,ngày….tháng….năm 20…

Người nhận xét

*(Ký tên và ghi rõ họ tên)*

# 

# MỤC LỤC

[**BÁO CÁO ĐỒ ÁN 1**](#_heading=h.1fob9te)

[**MỤC LỤC 3**](#_heading=h.os0l771vcdi4)

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 5**](#_heading=h.mk8jbh68m7jb)

[I. Giới thiệu 5](#_heading=h.j7ojgqy8cot4)

[II. Phát biểu bài toán 6](#_heading=h.j7ojgqy8cot4)

[III. Thách thức của bài toán 6](#_heading=h.j7ojgqy8cot4)

[IV. Kết quả dự kiến 7](#_heading=h.j7ojgqy8cot4)

[V. Đối tượng và phạm vi 7](#_heading=h.j7ojgqy8cot4)

[VI. Mục tiêu 8](#_heading=h.j7ojgqy8cot4)

[**CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU 8**](#_heading=h.w5wnbgyj62oy)

[I. Thông tin về dữ liệu 8](#_heading=h.wz780udq5k3e)

[1.1. Nguồn bộ dữ liệu 8](#_heading=h.8cgoihuzkyf8)

[1.2. Thông tin mô tả bộ dữ liệu 9](#_heading=h.g9r2pn4srwes)

[1.1 Bảng user 9](#_heading=h.cpe9mthc2xez)

[II. Tiền xử lý theo từng tiêu chí 13](#_heading=h.e4bmdllv31b6)

[2.2. Tiêu chí 2: Điểm trung bình các bài tập 13](#_heading=h.63j10090d6ux)

[2.2. Tiêu chí 2: Tần suất và thời gian học trên video 18](#_heading=h.kgwgfsub2b28)

[2.4. Tiêu chí 4: Số lần thử và nộp bài tập 19](#_heading=h.gmxv6z93u4r9)

[2.5. Tiêu chí 4: Mức độ hài lòng của học viên đối với khóa học 20](#_heading=h.q0fhqi43ysg8)

[2.5.1. Tiền xử lý dữ liệu (Data Preprocessing): 20](#_heading=h.uv3a3rqykp3v)

[2.5.2. Dịch bình luận (Translation): 23](#_heading=h.jqedh7g14ycd)

[**CHƯƠNG 3: PHƯƠNG PHÁP GÁN NHÃN 24**](#_heading=h.9frp9cp8qinv)

[I. Gán nhãn cho từng tiêu chí 24](#_heading=h.lh1d0nivw430)

[1.1 Tiêu chí 1: Điểm trung bình các bài tập 24](#_heading=h.y8p207osqh34)

[1.2. Tiêu chí 2: Tần suất và thời gian học trên video 28](#_heading=h.117p3jiuae2)

[1.4. Tiêu chí 4: Số lần thử và nộp bài tập 31](#_heading=h.cn55xhuvgagc)

[1.4.1. Xử lý và phân tích dữ liệu đầu vào 31](#_heading=h.vuli3rw03lzp)

[1.4.2. Xây dựng chỉ số đánh giá 31](#_heading=h.40bw4cm2p4ep)

[1.4.3. Gán nhãn thử thách (Challenge Label) 33](#_heading=h.8wgto79dpkke)

[1.4.4. Tạo sơ đồ phân phối nhãn 34](#_heading=h.6bjtzsuuefm8)

[1.4. Tiêu chí 4: Mức độ hài lòng của học viên đối với khóa học 34](#_heading=h.ue7c4b66sssl)

[1.4.1. Lựa chọn mô hình 34](#_heading=h.8bkgn6n8h0pz)

[1.5.2. Gán nhãn cho khóa học dựa trên tiêu chí 5 (Mức độ hài lòng của học viên) 36](#_heading=h.un7vm8a8c3re)

[II. Gán nhãn cho khóa học 38](#_heading=h.k8iky3xsiels)

[**CHƯƠNG 4: XÂY DỰNG MÔ HÌNH 40**](#_heading=h.jfhupexmrsv8)

[I. Dữ liệu sử dụng 40](#_heading=h.kny9ed8swyms)

[II. Đề xuất mô hình 40](#_heading=h.9b8s13bt0ud)

[1. Tổng quan bài toán 40](#_heading=h.9osphfvru6n6)

[2. Lựa chọn mô hình và thuật toán 41](#_heading=h.wtkaigp1y2be)

[2.1. Logistic Regression 41](#_heading=h.bnu1sk2tubag)

[2.2. Random Forest 41](#_heading=h.wtjoeqbcp52)

[2.3. SVM (Support Vector Machine) 41](#_heading=h.dux3pxx1d8m8)

[2.4. XGBoost (Extreme Gradient Boosting) 42](#_heading=h.7cje8goxrju)

[3. Giải pháp xử lý mất cân bằng dữ liệu 42](#_heading=h.nvidy5d7y16y)

[3.1. Tình trạng mất cân bằng trong dữ liệu 42](#_heading=h.d8fg381m3ii)

[3.2. Kỹ thuật SMOTE (Synthetic Minority Oversampling Technique) 42](#_heading=h.p6352q24mh2s)

[3.3. Kết hợp SMOTE với các mô hình máy học 43](#_heading=h.kdryl1dahtlw)

[III. Độ đo đánh giá 43](#_heading=h.811f3ta82brp)

[IV. Kết quả mô hình 46](#_heading=h.wsdpbpw5z9js)

[**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 48**](#_heading=h.6pafvqmyjpw3)

[**TÀI LIỆU THAM KHẢO 50**](#_heading=h.ibab512mfffn)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## Giới thiệu

Trong kỷ nguyên số, giáo dục trực tuyến đã và đang thay đổi cách con người tiếp cận tri thức, với hàng triệu khóa học được cung cấp trên các nền tảng MOOCs. Số lượng khóa học khổng lồ này mở ra cơ hội học tập không giới hạn nhưng cũng đặt ra bài toán đầy thách thức cho các nhà quản lý giáo dục: làm sao để kiểm soát và đảm bảo chất lượng giảng dạy trong khi vẫn đáp ứng nhu cầu cá nhân hóa trải nghiệm của từng học viên? Các phương pháp đánh giá truyền thống, như khảo sát ý kiến học viên, tổ chức phỏng vấn, hay thử nghiệm thực tế, dù mang lại hiệu quả nhất định nhưng lại tiêu tốn rất nhiều thời gian, tiền bạc, và nhân lực. Không chỉ vậy, chúng còn gặp khó khăn khi phải mở rộng quy mô để đáp ứng số lượng khóa học ngày một tăng cao.

Trong bối cảnh này, việc tận dụng sức mạnh của công nghệ hiện đại, đặc biệt là khai thác và phân tích dữ liệu, mở ra một hướng đi đầy triển vọng. Từ hành vi học tập của người dùng như thời gian xem video, số lần nộp bài tập, tỷ lệ hoàn thành khóa học cho đến phản hồi từ học viên, tất cả đều là nguồn dữ liệu quý giá để tự động hóa quy trình đánh giá chất lượng. Với sự hỗ trợ của các thuật toán học máy (Machine Learning) và trí tuệ nhân tạo (AI), hệ thống đánh giá không chỉ dừng lại ở việc đo lường các tiêu chí cơ bản mà còn có khả năng dự đoán chất lượng khóa học, phát hiện những vấn đề tiềm ẩn và thậm chí đưa ra các gợi ý cải thiện cụ thể.

Hãy thử tưởng tượng, thay vì mất hàng tuần để tổng hợp và phân tích kết quả khảo sát từ học viên, một hệ thống tự động sẽ cung cấp ngay lập tức cho nhà quản lý những báo cáo chi tiết và trực quan. Các khóa học sẽ được xếp hạng, đánh giá theo nhiều tiêu chí khác nhau, từ mức độ hoàn thành của học viên, chất lượng nội dung, đến mức độ tương tác của giảng viên. Nhờ vậy, các nhà quản lý không chỉ tiết kiệm thời gian và chi phí mà còn có thể đưa ra các quyết định kịp thời, chính xác để cải thiện chất lượng đào tạo. Hơn thế nữa, việc ứng dụng công nghệ vào đánh giá chất lượng không chỉ là giải pháp cho hiện tại mà còn là bước chuẩn bị chiến lược để đối mặt với những thách thức của giáo dục tương lai.

Hệ thống tự động hóa không chỉ là một công cụ, mà là chìa khóa giúp các nhà quản lý giáo dục khai thác tối đa tiềm năng của dữ liệu, từ đó mở ra cơ hội nâng cao trải nghiệm học tập và xây dựng một nền giáo dục trực tuyến chất lượng, bền vững hơn. Đây không chỉ là một bước tiến công nghệ, mà còn là hành trình đổi mới tư duy quản lý giáo dục.

## Phát biểu bài toán

Input của bài toán nghiên cứu bao gồm:

* Dữ liệu về khóa học như thông tin khóa học, các bình luận, video hay bài tập liên quan đến khóa học.
* Dữ liệu tương tác của học viên trong các khóa học, như xem video, làm bài tập.

Output của bài toán: Chất lượng của khóa học, từ 0 (thấp nhất) đến 2 (cao nhất).

Đầu vào của bài toán sẽ là các thông tin của khóa học, gồm thông tin nhiều bảng dữ liệu (course, user, user-problem, problem, video, user-video, comment, course- comment), được tiền xử lý, làm sạch, thống kê mô tả, sau đó thực hiện gán nhãn theo các quy tắc đặt ra để chấm điểm khóa học trên 5 tiêu chí. Đầu ra của bài toán chính là chất lượng của khóa học (kết quả tổng hợp từ điểm 5 tiêu chí).

## Thách thức của bài toán

Trong quá trình xây dựng hệ thống đánh giá chất lượng khóa học trực tuyến, nhóm chúng tôi đã xác định một số thách thức quan trọng cần giải quyết, cụ thể như sau:

* Xử lý dữ liệu lớn: Dữ liệu đa dạng từ nhiều nguồn, bao gồm video bài giảng, bài tập, bình luận, đánh giá học viên, và thông tin cá nhân, yêu cầu quy trình xử lý phức tạp để đảm bảo tính chính xác, đồng nhất và khả năng sử dụng.
* Thiếu nhãn dữ liệu: Dữ liệu đầu vào không đi kèm với nhãn (label) để phân loại chất lượng khóa học, khiến việc gán nhãn cần được thực hiện thủ công, bán tự động hoặc thông qua các thuật toán học máy tiên tiến.
* Tính phức tạp của hệ thống đánh giá: Hệ thống đánh giá cần kết hợp nhiều tiêu chí như tỷ lệ hoàn thành khóa học, mức độ tương tác, đánh giá của học viên và hiệu suất bài tập. Điều này đòi hỏi áp dụng các thuật toán phân tích phù hợp, từ phân loại đến phân cụm dữ liệu, cũng như xây dựng các mô hình học máy hiệu quả.
* Khả năng mở rộng và hiệu suất: Đối mặt với khối lượng dữ liệu lớn và đa dạng, hệ thống cần được thiết kế để mở rộng linh hoạt và xử lý dữ liệu nhanh chóng mà không ảnh hưởng đến độ chính xác của kết quả.
* Giải thích và ứng dụng kết quả: Sau khi phân tích, việc diễn giải kết quả để cung cấp thông tin rõ ràng, dễ hiểu cho người dùng là một thách thức. Hệ thống phải đảm bảo các kết quả phân tích có thể chuyển hóa thành gợi ý thực tiễn, hỗ trợ cải thiện chất lượng giảng dạy và trải nghiệm học tập.

Những thách thức này không chỉ yêu cầu giải pháp công nghệ mà còn đòi hỏi tầm nhìn chiến lược để tối ưu hóa nguồn lực và nâng cao hiệu quả của hệ thống đánh giá.

## Kết quả dự kiến

* Kết quả đánh giá khóa học dựa theo 5 tiêu chí, được đánh giá theo thang điểm 1-5 hoặc 1-3.
* Kết quả đánh giá tổng quan khóa học theo thang điểm từ 1-5.
* Mô hình gán nhãn và đánh giá khóa học.

## Đối tượng và phạm vi

Đối tượng của bài toán "Đánh giá chất lượng khóa học" là các khóa học nằm của nền tảng XueTangX được thu thập trong dataset MOOCCubeX từ năm 2019 - 2020. Phạm vi của bài toán bao gồm việc sử dụng 1 phần bộ dữ liệu này (986 khóa học) để phục vụ gán nhãn các khóa học dựa theo 05 tiêu chí đã đặt ra.

4 tiêu chí nhóm xây dựng để đánh giá độ hiệu quả khóa học là:

* Điểm trung bình khóa học: Đánh giá chất lượng và hiệu quả học tập thông qua điểm trung bình các bài tập.
* Thói quen học qua video: Theo dõi mức độ tương tác của học viên với nội dung video giảng dạy.
* Nỗ lực làm bài tập: Đánh giá sự nỗ lực và kiên trì của học viên trong việc hoàn thành các bài tập.
* Mức độ hài lòng của học viên: Tổng hợp đánh giá từ học viên để đo lường chất lượng khóa học từ góc nhìn người học.

## Mục tiêu

Mục tiêu chính của bài toán này gán nhãn và đánh giá các khóa học bằng các phương pháp học máy và thống kê theo các tiêu chí khác nhau mà nhóm đặt ra. Kết quả của bài toán có thể giúp các nhà quản lý giáo dục của các nền tảng MOOCs hiểu rõ hơn về các yếu tố ảnh hưởng đến chất lượng khóa học, từ đó đưa ra các giải pháp tối ưu hóa chất lượng, độ hiệu quả của giáo dục.

# CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU

## Thông tin về dữ liệu

### 1.1. Nguồn bộ dữ liệu

Bộ dữ liệu nhóm sử dụng cho bài toán: Đánh giá độ hiệu quả của khóa học là bộ MOOC CubeX. Bộ dữ liệu này được phát triển và cung cấp bởi AMiner, tập trung vào dữ liệu giáo dục từ các khóa học trực tuyến (MOOC - Massive Open Online Courses).

Mục tiêu của bộ dữ liệu: Phân tích hành vi và mô hình học tập của học viên thông qua dữ liệu từ khóa học trực tuyến, hỗ trợ phát triển các hệ thống khuyến nghị và cải tiến quá trình học tập cá nhân hóa.

Nhóm sử dụng nguồn dữ liệu được nhóm tác giả cung cấp tại đường dẫn: [github.com](https://github.com/THU-KEG/MOOCCubeX?tab=readme-ov-file)

### 1.2. Thông tin mô tả bộ dữ liệu

Mô tả: Bộ dữ liệu này thu thập thông tin chi tiết từ các khóa học, người dùng, các bài tập, và các bài giảng video, giúp phân tích hành vi học tập, tỷ lệ hoàn thành khóa học, và đánh giá hiệu quả học tập của học viên. MOOC CubeX cung cấp thông tin đa dạng từ nhiều nguồn như chi tiết khoá học, tương tác của người học với các bài tập và video, mối quan hệ giữa các bài tập và khóa học, v.v.

Để xây dựng và thực hiện đồ án, nhóm lựa chọn sử dụng **8** bảng dữ liệu, gồm:

| **STT** | **Tên bảng** | **Chi tiết** |
| --- | --- | --- |
| 1 | user | Chứa thông tin của học viên tham gia nền tảng |
| 2 | user\_problem | Lưu trữ thông tin về các bài tập hoặc vấn đề mà người dùng đã tương tác |
| 3 | course | Thông tin về các khóa học trong MOOC |
| 4 | problem | Thông tin về các bài tập hoặc vấn đề trong các khóa học |
| 5 | user\_video | Lưu thông tin về các video mà người dùng đã xem |
| 6 | video | Lưu trữ thông tin về các video trong khóa học |
| 7 | course\_comment | Bình luận của học sinh tại tài nguyên của khóa học đó, có thể là tại video hoặc bài tập |
| 8 | comment | Lưu trữ nội dung của các bình luận |

### 1.1 Bảng user

| **STT** | **Biến** | **Giải thích** | **Định dạng** |
| --- | --- | --- | --- |
| 1 | id | Id của người dùng | String |
| 2 | name | Tên | String |
| 3 | gender | Giới tính | Byte |
| 4 | school | Trường học | String |
| 5 | year\_of\_birth | Năm sinh | Integer |
| 6 | course\_order | Mảng số nguyên chứa id các khóa học đã đăng kí | Array |
| 7 | enroll\_time | Mảng chứa thời gian đăng ký khóa học | Array |

1.2. Bảng user\_problem

| **STT** | **Biến** | **Giải thích** | **Định dạng** |
| --- | --- | --- | --- |
| 1 | log\_id | Id của bản ghi | String |
| 2 | problem\_id | Id của câu hỏi | String |
| 3 | user\_id | Id của người dùng | String |
| 4 | is\_correct | - 'true' nếu người dùng đã đưa ra bài làm đúng hoàn toàn  - 'false' nếu ngược lại | Boolean |
| 5 | attempts | Số lần người dùng nộp bài tập | Integer |
| 6 | score | Điểm số của người dùng đạt được khi làm bài tập | Float |
| 7 | submit\_time | Thời điểm người dùng nộp bài | DateTime |

1.3 Bảng course

| **STT** | **Biến** | **Giải thích** | **Định dạng** | **Miền giá trị** |
| --- | --- | --- | --- | --- |
| 1 | id | Id khóa học | String | Bắt đầu bằng “C\_” |
| 2 | name | Tên khóa học | String |  |
| 3 | prerequisites | Những kiến thức, kĩ năng cần thiết mà người học cần phải nắm vững trước khi đăng kí khóa học mới | String |  |
| 4 | about | Giới thiệu về khóa học | String |  |
| 5 | resource | Danh sách tài nguyên của khóa học | Array |  |
| 6 | field | Mô tả phạm vi, chủ đề hoặc ngành học chính mà khóa học thuộc về | Array |  |

1.4. Bảng problem

| **STT** | **Biến** | **Giải thích** | **Định dạng** | **Miền giá trị** |
| --- | --- | --- | --- | --- |
| 1 | problem\_id | Id vấn đề | String | Bắt đầu bằng “Pm\_” |
| 2 | title | Tiêu đề của câu hỏi | String |  |
| 3 | content | Mô tả câu hỏi | String |  |
| 4 | option | Mảng chứa 4 đáp án | Array | A, B, C, D |
| 5 | answer | Câu trả lời đúng | Char | A, B, C, D |
| 6 | score | Số điểm của câu hỏi | Int |  |
| 7 | type | Loại câu hỏi | Int |  |
| 8 | typetext | Loại câu hỏi | String |  |
| 9 | location | Vị trí chương của câu hỏi | String |  |
| 10 | context\_id | Mảng chứa id liên quan đến vấn đề, cho biết vấn đề này thuộc về một khoá học, một bài giảng, hoặc một kỳ thi cụ thể nào đó. | Array |  |
| 11 | exercise\_id | Id bài tập | String | Bắt đầu bằng “Ex\_” |
| 12 | language | Ngôn ngữ mô tả câu hỏi | String | Chinese |

1.5. Bảng user\_video

| **STT** | **Biến** | **Giải thích** | **Định dạng** | **Miền giá trị** |
| --- | --- | --- | --- | --- |
| 1 | seq | Một mảng các dữ liệu có cấu trúc, mỗi phần tử bao gồm video\_id và một mảng segment. Mỗi segment bao gồm: start\_point, end\_point, speed, local\_start\_time | Array |  |
| 2 | user\_id | Id của người dùng | String | Bắt đầu bằng “U\_” |

1.6 Bảng video

| **STT** | **Biến** | **Giải thích** | **Định dạng** |
| --- | --- | --- | --- |
| 1 | ccid | Chứa mã phân loại dùng để phân loại video theo một tiêu chí cụ thể nào đó, chẳng hạn như thể loại, nguồn gốc, hoặc trạng thái | String |
| 2 | name | Tên của video | String |
| 3 | start | Mảng các số chỉ thời điểm bắt đầu của một sự kiện trong video | Array |
| 4 | end | Mảng các số chỉ thời điểm kết thúc của một sự kiện trong video | Array |
| 5 | text | Mảng chứa nội dung văn bản liên quan đến đoạn video, chẳng hạn như lời thoại, mô tả nội dung, hoặc chú thích cho một cảnh cụ thể | Array |

1.7 Bảng course\_comment

| **STT** | **Biến** | **Giải thích** | **Định dạng** |
| --- | --- | --- | --- |
| 1 | course\_id | Id khóa học | String |
| 2 | comment\_id | Id của bình luận | String |

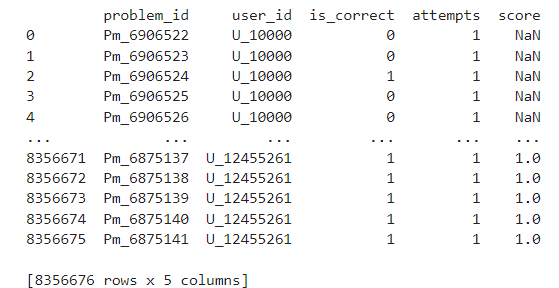
1.8 Bảng comment

| **STT** | **Biến** | **Giải thích** | **Định dạng** |
| --- | --- | --- | --- |
| 1 | id | Id của comment | String |
| 2 | user\_id | Id của người dùng | String |
| 3 | text | Nội dung của bình luận | String |
| 4 | create\_time | Thời điểm bình luận | DateTime |

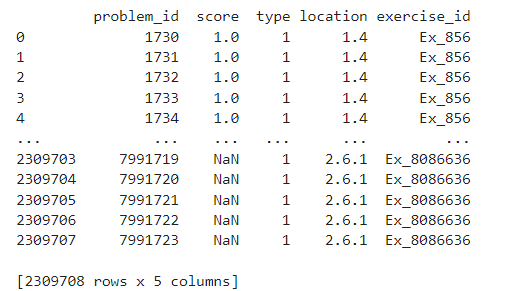
## Tiền xử lý theo từng tiêu chí

### 2.2. Tiêu chí 2: Điểm trung bình các bài tập

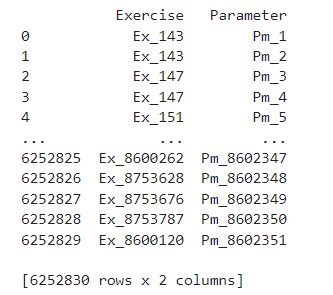
* **Bảng dữ liệu sử dụng**: user\_problem (lưu trữ thông tin của người dùng trong quá trình tham gia các khóa học), problem (lưu trữ thông tin của các bài tập, kết quả của người dùng, và phản hồi của người dùng).exercise\_problem(lưu trữ thông tin exercise id , problem \_id), course\_student(lưu trữ thông tin course\_id , student\_id )
* **Mục tiêu**: Tiêu chí này nhằm tính toán điểm trung bình đạt được của học viên trên tất cả các bài tập trong khóa học, từ đó đánh giá mức độ khó và hiệu quả của khóa học.
* **Quy trình tính toán**:
  + **Lấy dữ liệu từ bảng**: Trích xuất dữ liệu từ 3 bảng user problem và problem, exercise problem từ bộ dữ liệu MOOC CubeX.
  + **Các trường dữ liệu cần filter**: Dữ liệu sẽ được lọc theo các trường user\_id, is\_correct, attempts, và score để tính toán điểm trung bình của mỗi học viên cho tất cả các bài tập.
  + **Tính điểm trung bình của từng học viên**: Tính điểm trung bình của từng học viên trên tất cả bài tập để đánh giá kết quả học tập của họ.
  + **Đánh giá độ khó khóa học**: Tính điểm trung bình của tất cả học viên để làm mốc đánh giá độ khó của khóa học (dựa vào điểm trung bình của người học). Từ đó xác định khóa học có mức độ khó cao hay dễ.
  + **Đánh giá hiệu quả khóa học**: Căn cứ vào điểm trung bình của học viên để xác định hiệu quả của khóa học (được phân loại là hiệu quả hoặc không hiệu quả).
* **Kết quả đầu ra**:
  + **avg\_score\_exercise**: Điểm trung bình của từng exercise.
  + **avg\_score\_course**: Điểm trung bình của từng course.



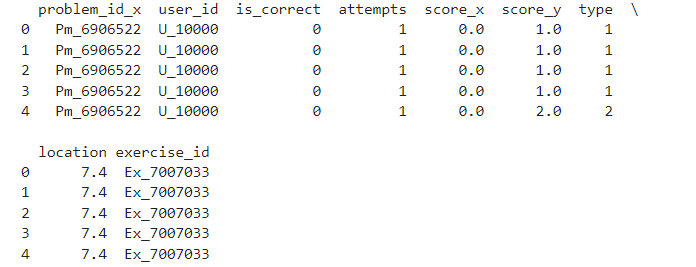
Hình 2.2-a user problem trước.



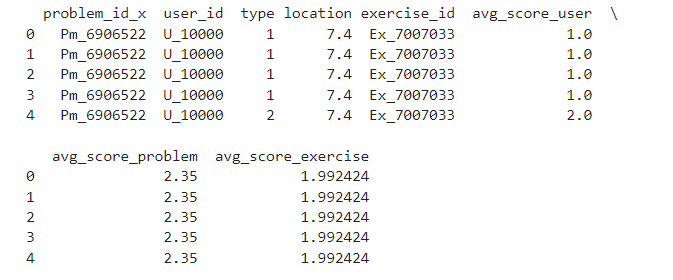
Hình 2.2-b problem trước.



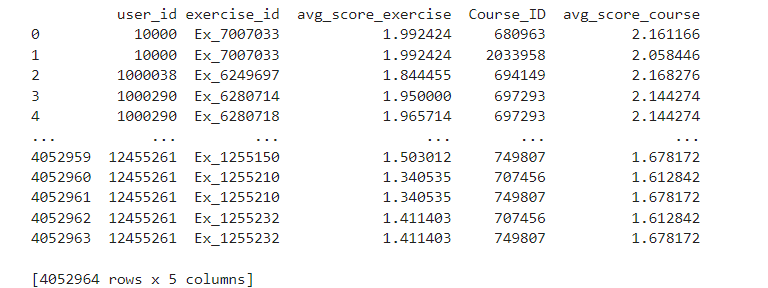
Hình 2.2-c exercise-problem trước



Hình 2.2-d merge-user-problem , problem ,exercise-problem



Hình 2.2-d merged data sau khi bổ sung các cột avg score(Điểm trung bình)

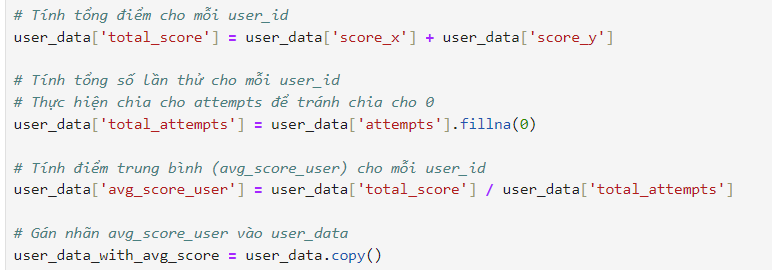


Hình 2.2-e bảng dữ liệu course-exercise (sau khi loại bỏ các cột không cần thiết )

Dữ liệu bảng user problem được chia thành 10 file dữ liệu nhỏ, vì vậy, để xử lý dữ liệu, chúng tôi tiến hành kết hợp các file này và lọc ra các trường dữ liệu cần thiết.

Chúng tôi tiến hành tính toán theo từng bước như sau:

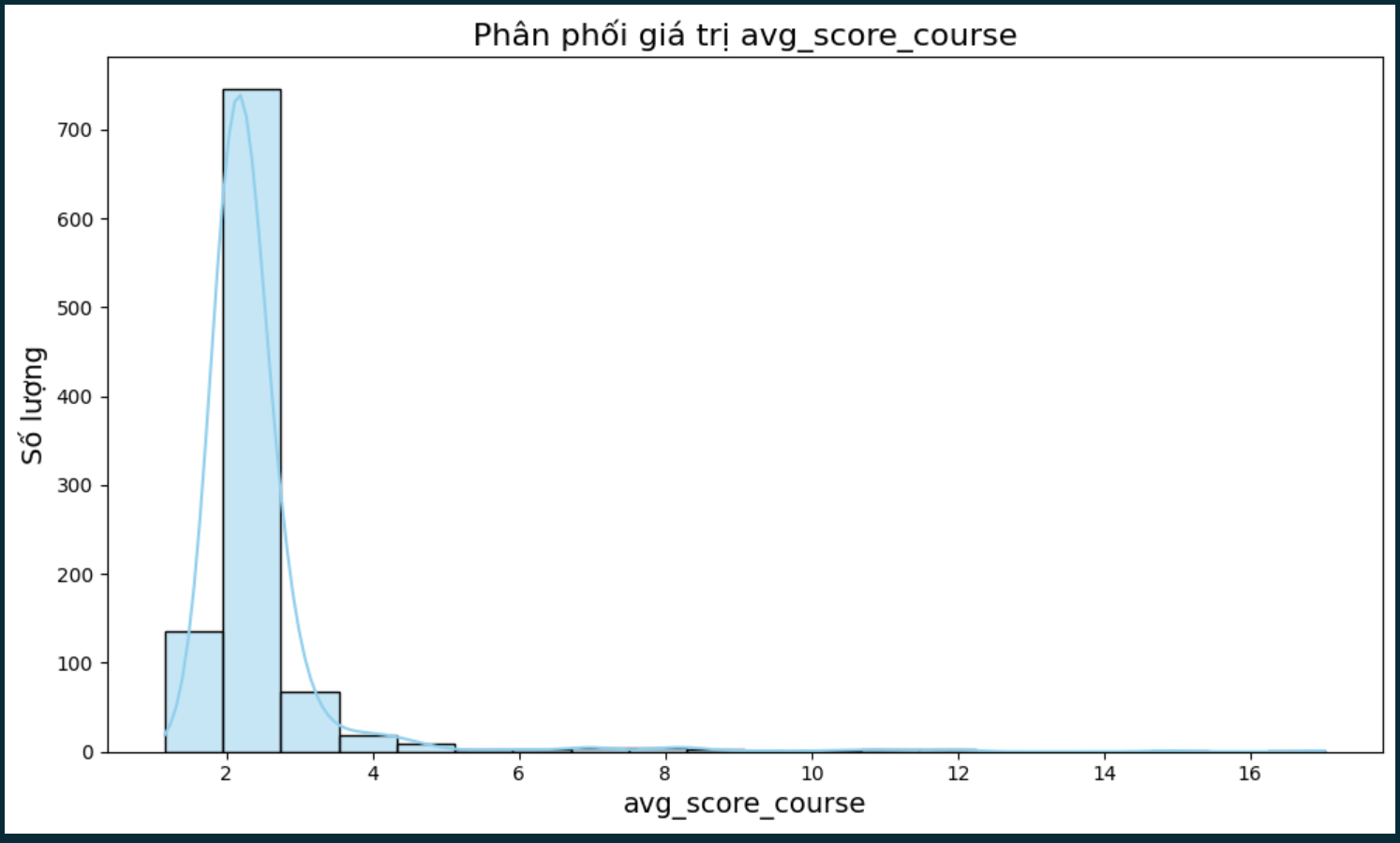
1. **Tính toán avg\_score\_user**:  
   Đầu tiên, chúng tôi tính toán avg score user dựa trên các trường is correct, attempt, score theo từng user\_id.
2. **Tính toán avg score problem**:  
   Tiếp theo, chúng tôi tính toán avg score problem dựa trên avg score user cho từng problem id.
3. **Tính toán avg score exercise**:  
   Tiếp theo, chúng tôi tính toán avg score exercise dựa trên avg score problem cho từng exercise id.
4. **Tính toán avg score course**:  
   Cuối cùng, chúng tôi tính toán avg score course dựa trên avg\_score\_exercise cho từng course\_id.



Hình 2.2-f source code add avg\_score\_user

Việc xử lý và lọc dữ liệu được thực hiện thủ công, tuần tự từng file user problem từ file 1 đến file 10. Sau khi kết hợp và lọc các dữ liệu cần thiết, chúng tôi tính toán giá trị điểm trung bình **unique** cho từng user\_id. Tiếp theo, tính toán điểm trung bình của toàn bộ học viên.

Dựa vào điểm trung bình của từng học viên (avg\_score), chúng tôi sẽ đánh giá độ khó của khóa học. Thông qua việc tính được giá trị avg\_score của từng course và exercise.



Hình 2.2-g biểu đồ phân phối giá trị avg score course

### 2.2. Tiêu chí 2: Tần suất và thời gian học trên video

Bảng dữ liệu sử dụng: “user-video.json”

Tiêu chí 2 dùng để đánh giá hành vi người dùng trong việc học các video trực tuyến và tạo ra các phân tích sâu về cách người dùng tương tác với các video này. Quá trình tiền xử lý dữ liệu để gán nhãn tần suất và thời gian học bao gồm các bước chính nhằm chuẩn hóa, làm sạch và trích xuất thông tin giá trị từ tập dữ liệu thô của bảng user\_video

Quá trình tiền xử lý dữ liệu cho tiêu chí 2 từ bảng user\_video bao gồm các bước sau:

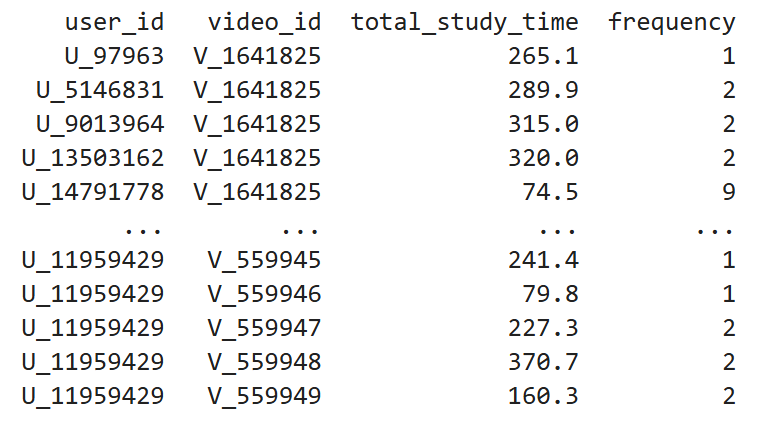
- Kiểm tra và làm sạch dữ liệu: Để đảm bảo dữ liệu đầu vào đáng tin cậy và không chứa các giá trị bất thường, bước đầu tiên trong quá trình tiền xử lý là kiểm tra và làm sạch dữ liệu. Điều này bao gồm việc loại bỏ các bản ghi trùng lặp, xử lý các giá trị bị thiếu và chuyển đổi dữ liệu từ định dạng thô sang cấu trúc dễ sử dụng. Những thao tác này giúp chuẩn bị dữ liệu để tiến hành tính toán các chỉ số một cách chính xác và hiệu quả.

* Xóa dữ liệu trùng lặp: Sử dụng drop\_duplicates() để loại bỏ các bản ghi bị lặp trong bảng user\_video.
* Kiểm tra giá trị rỗng: Loại bỏ hoặc xử lý các hàng có giá trị NaN trong cột seq vì đây là cột chính chứa thông tin về thời gian học.

- Tính toán các chỉ số học tập: Sau khi làm sạch dữ liệu, bước tiếp theo là trích xuất và tính toán các chỉ số học tập quan trọng. Hai tiêu chí chính, bao gồm tổng thời gian học (total\_study\_time) và tần suất học (frequency), sẽ được tính toán từ dữ liệu đã cấu trúc. Những chỉ số này không chỉ phản ánh hành vi học tập của người dùng mà còn cung cấp cơ sở để phân tích chuyên sâu và gán nhãn dữ liệu.

* Tổng thời gian học (total\_study\_time): Duyệt qua từng phần tử trong danh sách seq để trích xuất start point và endpoint.
* Tính tổng thời gian học bằng công thức: study time = end point − start point để loại bỏ các giá trị bất hợp lý (thời gian âm).
* Tần suất học (frequency): Đếm số lượng đoạn video đã được học (số phần tử trong danh sách seq).

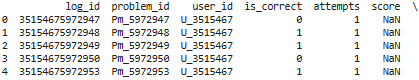
- Kết quả đầu ra:



Hình X:

### 2.4. Tiêu chí 4: Số lần thử và nộp bài tập

* **Bảng dữ liệu sử dụng**: user\_problem (lưu trữ thông tin của người dùng trong quá trình tham gia các khóa học)
* **Mục tiêu**:
* **Đánh giá mức độ thử thách của bài tập và khóa học:** Xác định độ khó của bài tập thông qua **số lần thử và nộp bài**. Bài tập có **số lần thử cao** có thể phản ánh nội dung khó hiểu, đòi hỏi học viên phải nỗ lực hơn. Bài tập có **số lần thử và nộp thấp** có thể chỉ ra rằng bài tập chưa đủ thử thách hoặc học viên không đủ động lực thực hiện.
* **Đánh giá sự phù hợp của nội dung giảng dạy:** Nếu **số lần nộp cao nhưng tỷ lệ thành công thấp**, có thể nội dung giảng dạy hoặc cách đánh giá chưa phù hợp, cần điều chỉnh để cải thiện.
* **Hỗ trợ cải thiện khóa học:** Xây dựng một hệ thống gán nhãn từ 1-5 để phân loại các bài tập theo độ thử thách. Từ đó, giảng viên hoặc nhà thiết kế khóa học có thể xác định bài tập nào cần điều chỉnh, sửa đổi để phù hợp với năng lực học viên.
* **Phân tích hành vi học tập của học viên: Số lần thử cao:** Học viên gặp khó khăn hoặc đang rất nỗ lực để hoàn thành. **Số lần thử thấp:** Có thể do bài tập dễ hoặc học viên chưa tập trung vào bài tập. Thông qua đó, khóa học có thể cá nhân hóa hơn, hỗ trợ từng học viên theo mức độ phù hợp.
* Đầu tiên tiến hành loại bỏ các cột dữ liệu không cần thiết ở các bảng dữ liệu có kích thước lớn. Mục đích của việc làm này giúp cải thiện tốc độ xử lý.

****

* Tương tự với tiêu chí 2. dữ liệu bảng user problem được chia thành 10 file dữ liệu nhỏ, vì vậy, để xử lý dữ liệu, chúng tôi tiến hành kết hợp các file này và lọc ra các trường dữ liệu cần thiết, bao gồm user\_id, is\_correct, attempts, và score, để phục vụ cho việc đánh giá.

### 2.5. Tiêu chí 4: Mức độ hài lòng của học viên đối với khóa học

Bảng dữ liệu sử dụng: course.json (lưu trữ thông tin khóa học), course-comments.txt (lưu trữ thông tin các bình luận tương ứng với khóa học), comments.json (lưu trữ thông tin bình luận)

Tiêu chí 4 nhằm thể hiện mức độ hài lòng của học viên đối với khóa học. Để có thể triển khai đánh giá tiêu chí này, chúng tôi cần phải biết được nguồn thể hiện cảm xúc của học viên đối với khóa học; chính là những tương tác (interaction) dạng bình luận (comment). Những bình luận khác nhau mang lại những sắc thái nghĩa khác nhau – vui vẻ, buồn chán, ngạc nhiên, hoảng hốt, điềm tĩnh… Do đó, ta có thể tận dụng nguồn dữ liệu này để phân tích mức độ hài lòng của người học.

Bộ dữ liệu MOOC Cube X có bảng dữ liệu comment.json lưu trữ thông tin bình luận của học viên, đây sẽ là bảng dữ liệu chính để chúng tôi thực hiện việc xử lý ngôn ngữ tự nhiên (NLP) phục vụ mục đích ban đầu. Ý tưởng là sẽ đi phân loại các bình luận (comment) thành 5 loại cảm xúc chính: Từ 1 đến 5, theo cấp độ từ ‘rất tệ’ đến ‘rất tốt’.

Sau khi có nhãn tương ứng với các bình luận, ta thực hiện tính mức độ hài lòng của học viên cho các khóa học (course). Để có thể nối dữ liệu của khóa học với bình luận, ta cần có sự hỗ trợ của 2 bảng dữ liệu course.json và course comment.txt.

#### 2.5.1. Tiền xử lý dữ liệu (Data Preprocessing):

* **Chỉ lấy lượng thông tin từ 1000 khóa học (course) có nhiều người học nhất trong bộ dữ liệu Mooc CubeX**

Do hạn chế về thời gian và lượng dữ liệu nguyên bản quá lớn, nhóm lựa chọn lấy ra thông tin 1000 khóa học để phục vụ cho bài toán. Sau thao tác này, nhóm có một bộ dữ liệu bình luận mới với 5971791 hàng dữ liệu.

* **Thực hiện nhiều thao tác tiền xử lý và làm sạch dữ liệu:**
* **Loại bỏ thuộc tính không cần thiết:** Để quá trình tính toán diễn ra suôn sẻ và hiệu quả hơn, loại bỏ dữ liệu không cần thiết là một bước rất quan trọng. Bảng comment.json có đến 4 cột (id, user\_id, text, create\_time), trong khi nhóm chỉ cần thực hiện NLP task đối với các comment, do đó chỉ giữ lại 2 thuộc tính là ‘comment\_id’ (khóa chính, đã đổi tên, ban đầu là ‘id’) và ‘text’. Tương tự, bảng course.json có đến 6 cột (‘about’, ‘id’, ‘field’, ‘name’, ‘prerequisites’, ‘resource’), mà bảng course comment.txt chỉ chứa 2 cột tương ứng với id của khóa học (‘course\_id’) và id của bình luận (‘comment\_id’), do đó ở bảng course.json chỉ cần giữ lấy ‘course\_id’ (khóa chính, đã đổi tên, ban đầu là ‘id) là phù hợp.
* **Kết hợp 3 bảng:** Các dữ liệu bị thừa sẽ được loại bỏ khi ta thực hiện kết với các bảng lại với nhau. Các khóa học (course) có thể có nhiều bình luận (comment), tuy nhiên những khóa học xuất hiện trong bảng course comment.txt mà không tồn tại trong course.json sẽ bị xem xét là thông tin thừa và bị loại bỏ. Tương tự cũng có những bình luận bị thừa (xuất hiện trong bảng comment.json nhưng không xuất hiện trong bảng course comment.txt), nhưng ta sẽ không loại bỏ, do những bình luận đó sẽ đóng góp trong việc xây dựng mô hình (model) để phân loại nhãn các bình luận. Sau bước này ta thu được file ‘top\_1000\_courses\_comments.csv’ với 5,971,791 dòng dữ liệu.
* **Loại bỏ dữ liệu thừa, tiền xử lý ‘text’:**
* Loại bỏ các dòng ‘text’ chứa giá trị rỗng, chỉ chứa 1 chữ cái tiếng anh, chỉ chứa chữ số, chỉ chứa dấu câu...
* Xử lý thuộc tính ‘text’: Loại bỏ dấu câu, khoảng trắng thừa, ký tự đặc biệt, tab, xuống dòng, loại bỏ khoảng trắng thừa,..



**Hình 2.5.1.1:** Bộ dữ liệu sau tiền xử lý ‘text’

Sau bước này, ta thu được một bộ dữ liệu với 4,868,960 dòng dữ liệu

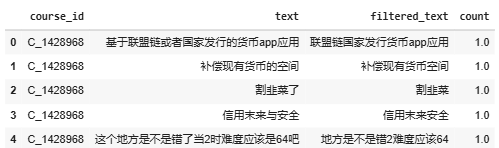
* **Chuyển kiểu dữ liệu:** Thực hiện chuyển ‘text’ về kiểu dữ liệu string.
* **Đánh trọng số các bình luận:** Nhóm tạo thêm thuộc tính ‘count’, chỉ số lần lặp lại của bình luận đó tại từng khóa học (group\_by theo ‘course\_id’)



**Hình 2.5.1.2:** Dữ liệu bổ sung thuộc tính ‘count’

* **Giảm số lượng bình luận mỗi khóa học**: Do 2 nguyên nhân chính là: Mất cân bằng dữ liệu về số lượng bình luận tại mỗi khóa học (có course chỉ có 1 - 2 thậm chí không có comment nào, nhưng có course lại có tới 1998514 comment như khóa học C\_936971), Quá tải trong việc dịch thuật cùng gán nhãn dữ liệu; nhóm quyết định đối với các khóa học có trên 300 dòng bình luận, chỉ lấy random 300 dòng bình luận. Giữ nguyên các khóa học số lượng các bình luận có dưới 300 dòng.

Sau tất cả các bước tiền xử lý trên, bộ dữ liệu ‘cleaned\_comments\_3.csv’ được ghi nhận có 147,754 dòng bình luận với 4 thuộc tính: ‘course\_id’, ‘text’, ‘filtered\_text’, ‘count’.



**Hình 2.5.1.3**: Bộ dữ liệu ‘cleaned\_comments\_3.csv’

* **Chia thành 20 file nhỏ:** Do giới hạn về cơ sở vật chất, nhóm chia nhỏ file ‘cleaned\_comments\_3.csv’ để thực hiện các giai đoạn sau mà không lo ngại bị gián đoạn.

#### 2.5.2. Dịch bình luận (Translation):

Các bình luận trong bảng ‘cleaned\_comments\_3.csv’ hầu hết bao gồm tiếng Trung. Việc phân tích ngữ nghĩa sẽ trở nên khó khăn nếu dựa vào tiếng Trung (Nhóm không rõ về tiếng Trung), do đó việc dịch các bình luận về tiếng Việt là cần thiết.

Nhóm tạo thêm thuộc tính ‘text vi’ là các bình luận được dịch sang tiếng Anh tương ứng với ‘text’ với công cụ GoogleTranslator() của thư viện ‘deep translator’. Sau đó, chúng tôi cũng thực hiện một số tiền xử lý với dữ liệu Tiếng Việt ‘text\_vi’ như: loại bỏ khoảng trắng, dấu câu, kí tự thừa, …. Các file nhỏ ở giai đoạn trước sẽ được dịch và cho ra các file ‘translated\_ comments\_{i}.csv’ tương ứng.

Sau các giai đoạn tiền xử lý cùng với dịch thuật, ta gộp các file nhỏ và thu được một bộ dữ liệu gồm 144,787 dòng dữ liệu với 5 thuộc tính chính như sau:



**Hình 2.5.2**: Bộ dữ liệu bình luận sau khi tiền xử lý và dịch thuật

Do hạn chế về hiểu biết Tiếng Trung, nhóm chỉ thực hiện phân tích cảm xúc trên bình luận tiếng Việt, từ đó đưa ra nhãn về mức độ hài lòng của học viên đối với khóa học.

# CHƯƠNG 3: PHƯƠNG PHÁP GÁN NHÃN

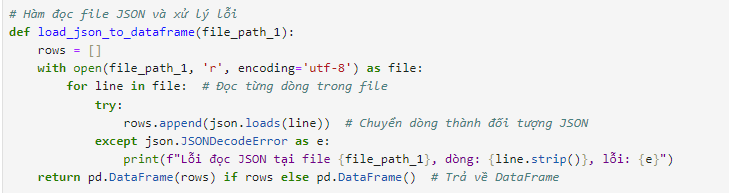
## Gán nhãn cho từng tiêu chí

### 1.1 Tiêu chí 1: Điểm trung bình các bài tập

Quá trình gán nhãn điểm trung bình giúp phân loại điểm trung bình của từng học viên qua các bài tập, đồng thời đánh giá độ khó của khóa học bằng cách so sánh với điểm trung bình của tất cả học viên. Đánh giá hiệu quả của khóa học được thực hiện thông qua trường avg\_score\_couse, trong đó giá trị được tính thông qua điểm trung bình của từng học viên avg\_score\_user , avg score problem và cuối cùng là avg score exercise.

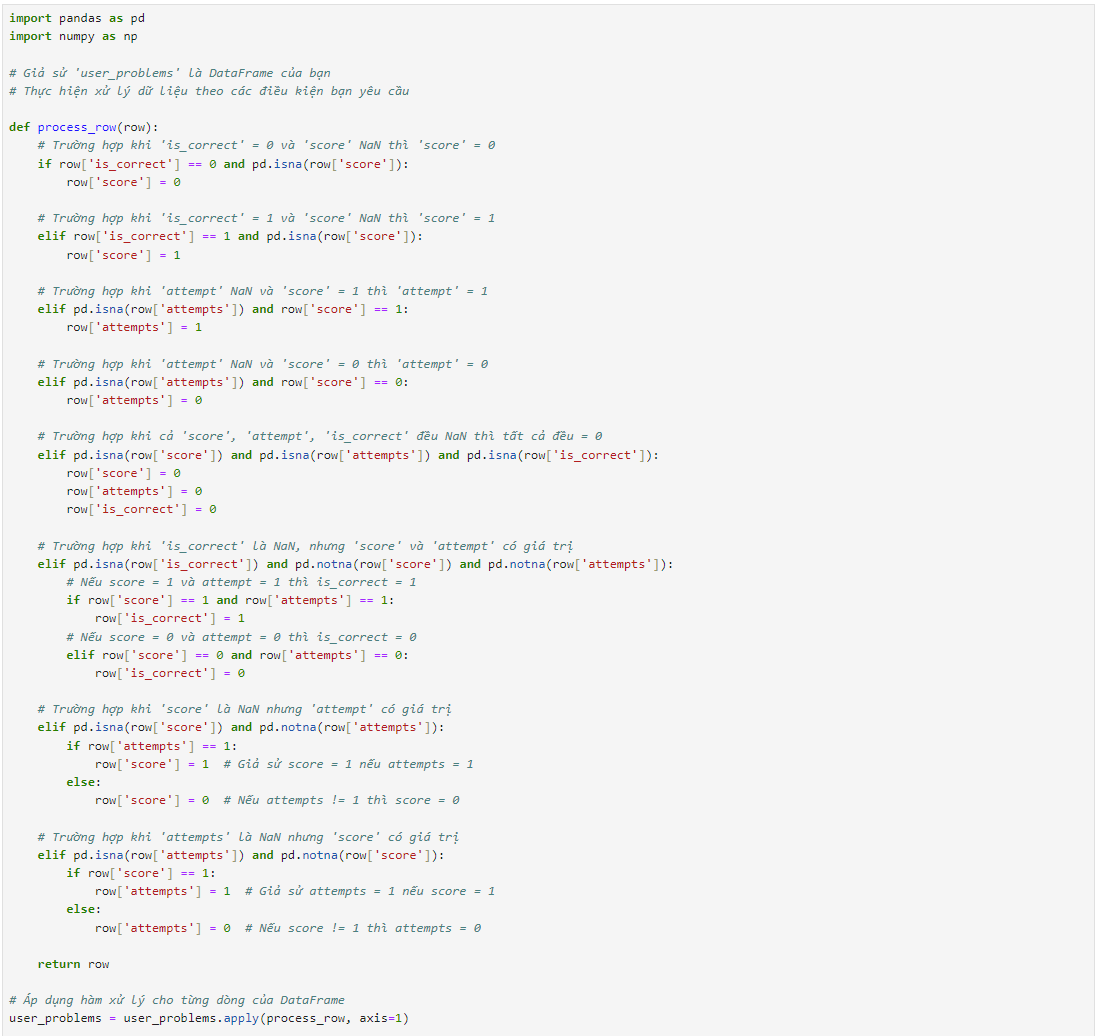
Để tính được điểm trung bình của tất cả học viên và đánh giá hiệu quả, trước tiên cần lọc các giá trị của các trường user id, score, is correct, và attempts.

Vì dữ liệu được chia thành 10 file nhỏ, bước đầu tiên là quy ước một hàm để đọc và xử lý các file dữ liệu, đồng thời xử lý lỗi khi gặp phải vấn đề trong quá trình đọc dữ liệu.



Hình 1.2.- a hàm đọc file json và xử lý lỗi.

Sau đó, chúng ta sẽ lần lượt import các file dữ liệu từ bảng user problem và xử lý các lỗi trong dữ liệu. Cụ thể, chúng tôi sẽ kiểm tra và loại bỏ các dòng dữ liệu thiếu user id hoặc có giá trị score là null. Đồng thời, nếu các cột is correct và attempts, score không tồn tại, chúng tôi sẽ fill các giá trị 0, 1 tương ứng cho từng case mà các giá trị kể trên là null.



Hình 1.2-b kiểm tra dữ liệu và fill các trường null (trước khi tính giá trị avg score)

Sau khi hoàn thành kiểm tra và xử lý dữ liệu, chúng ta tiến hành tính điểm trung bình cho từng học viên. Đồng thời, tính toán tổng điểm trung bình và tổng số học viên để xác định điểm trung bình của tất cả học viên.Thông qua đó lần lượt tính được cá giá trị trung bình avg\_score theo từng đơn vị là user, problem , exercise và cuối cùng là course

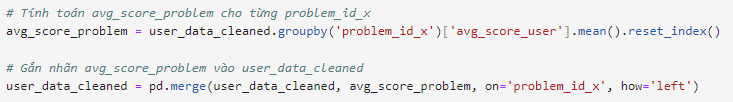


Hình 1.2-c vd về việc lọc các cột không cần thiết trong tiến trình

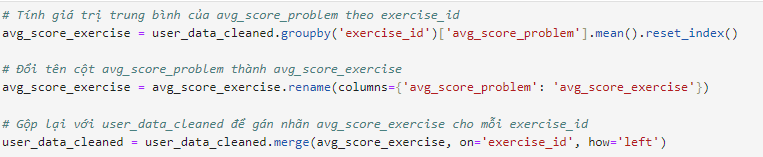
Trong quá trình đó để đảm báo output đạt chuẩn , chung tôi tiến hành lọc các cột dữ liệu không cần thiết sau khi sử dụng. Tiến trình được tiến hành theo dạng cuốn chiếu và các dữ liệu sẽ được lọc dần đến khi output đạt yêu cầu của nhãn.



Hình 1.2-d tính điểm trung bình của từng user



Hình 1.2-e tính điểm trung bình của từng problem

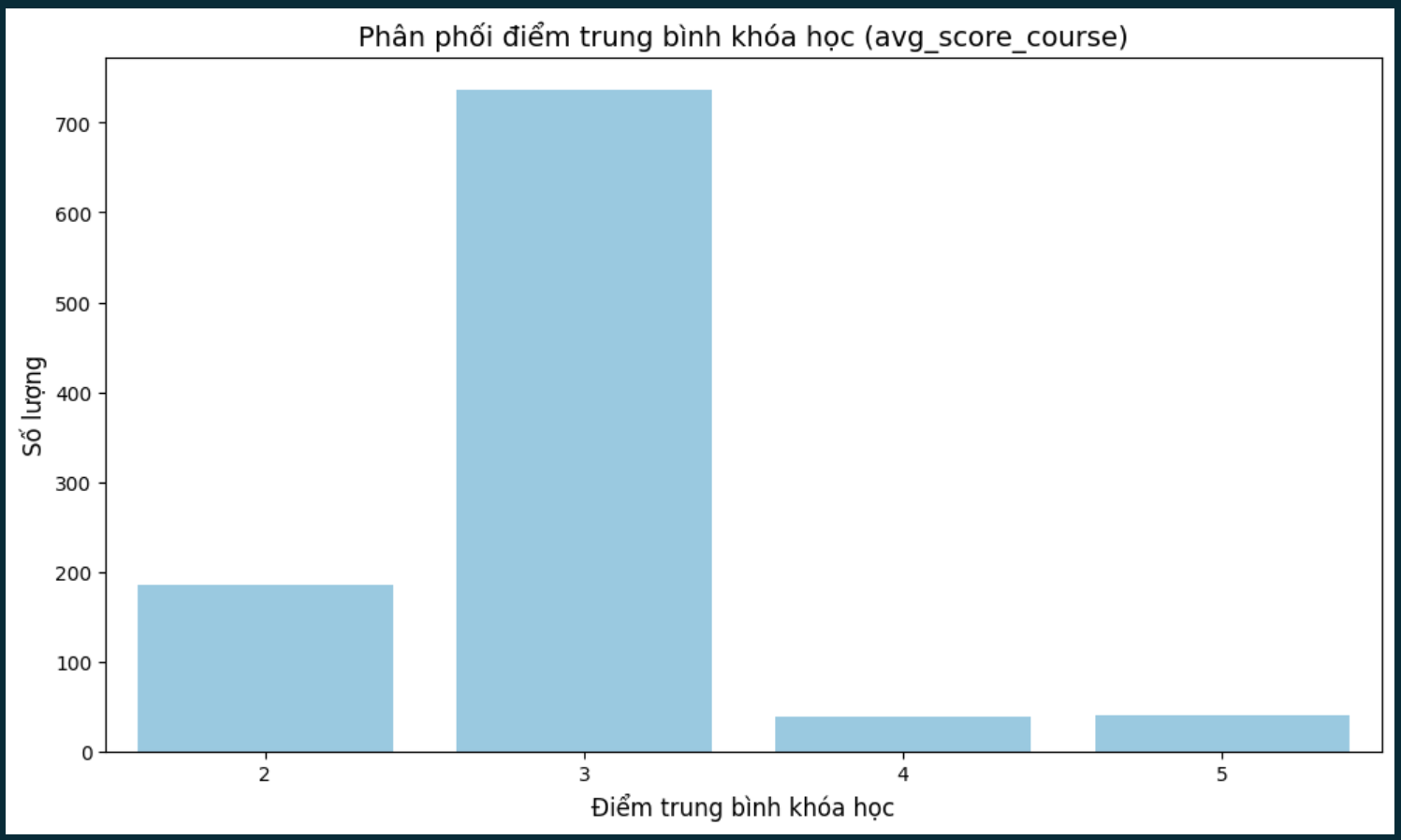


Hình 1.2-f tính điểm trung bình của từng exercise



Hình 1.2-g tính điểm trung bình của từng course

Quá trình này sẽ lặp qua từng file user problem từ file 1 đến file 10. Sau khi hoàn thành việc tính toán, chúng ta sẽ có được điểm trung bình của từng exercise và điểm trung bình của từng course.



Hình 1.2-k Biểu đồ phân phối điểm trung bình khóa học(avg\_score\_course)

### 1.2. Tiêu chí 2: Tần suất và thời gian học trên video

Quá trình gán nhãn tần suất và thời gian học giúp phân loại các video học tập theo hai tiêu chí chính: Tần suất học và Thời gian học. Điều này giúp hiểu rõ hơn về hành vi người dùng trong việc học các video trực tuyến và tạo ra các phân tích sâu về cách người dùng tương tác với các video này.

Đầu tiên ta cần tính thời gian học và số lần học, sau đó lưu vào cột ‘total\_study\_time’ và ‘frequency’



Hình X: Tính toán tần suất và thời gian học

Tiếp theo, để gán nhãn tần suất học (frequency) và thời gian học (total\_study\_time), ta phân loại mỗi video vào một trong ba nhóm: thấp, trung bình, cao, tùy thuộc vào số lần người dùng đã xem video đó và thời gian học. Quy tắc gán nhãn được quy định cụ thể như sau:

- Nhãn ‘time\_label’:

* Low: Tổng thời gian học trên 30 phút.
* Medium: Tổng thời gian học trong khoảng từ 30 phút đến 60 phút.
* High: Tổng thời gian học trên 60 phút.

- Nhãn ‘freq\_label’:

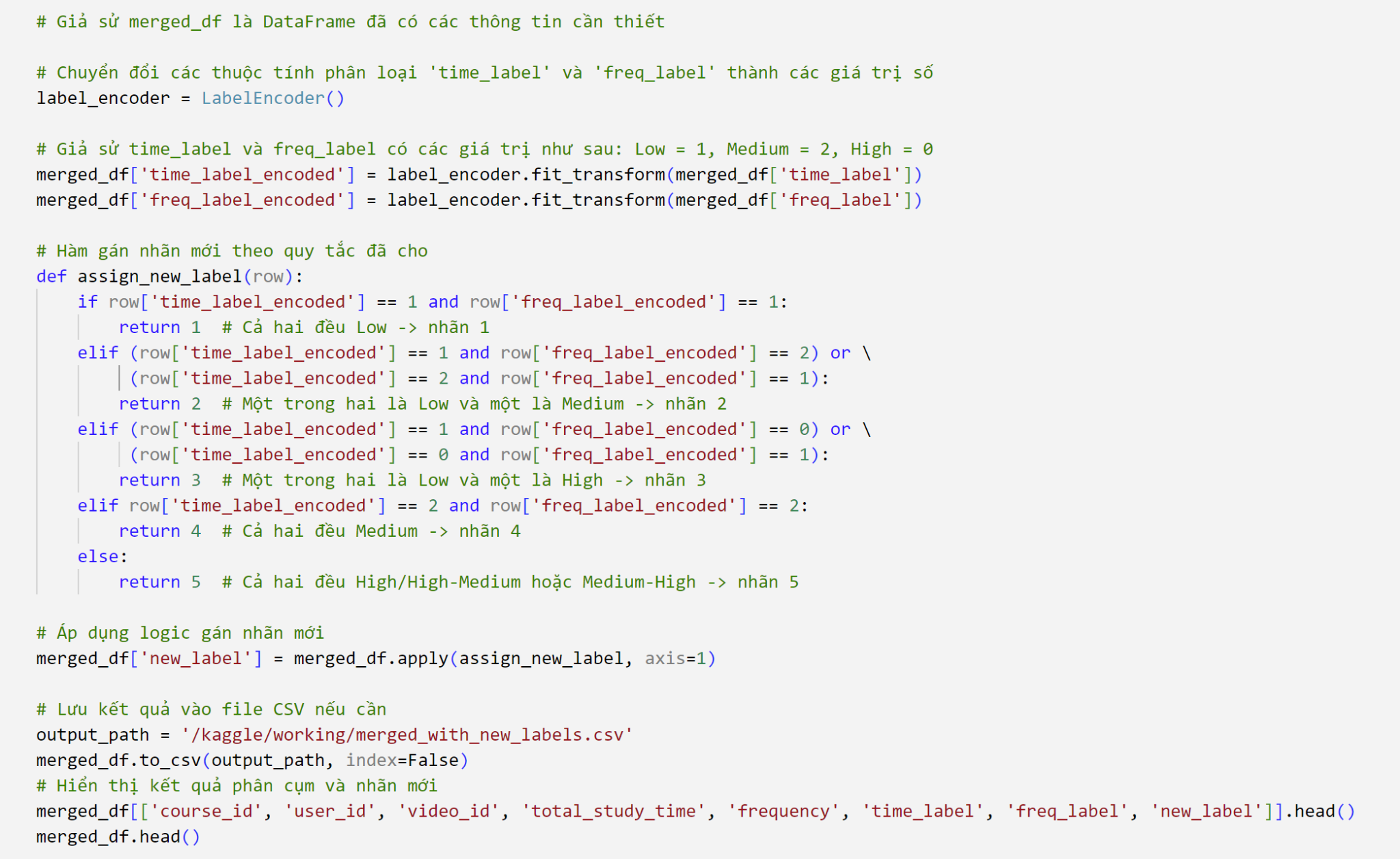
* Low: Số lần học dưới 2 lần
* Medium: Số lần học từ 2 đến 5 lần
* High : Số lần học trên 5 lần



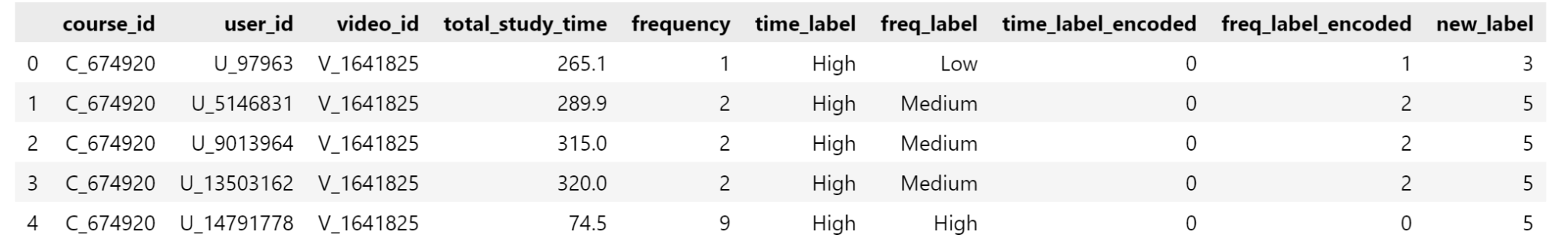
Hình X: Gán nhãn tần suất và thời gian học

Để gán nhãn tổng hợp, nhóm áp dụng logic sau:

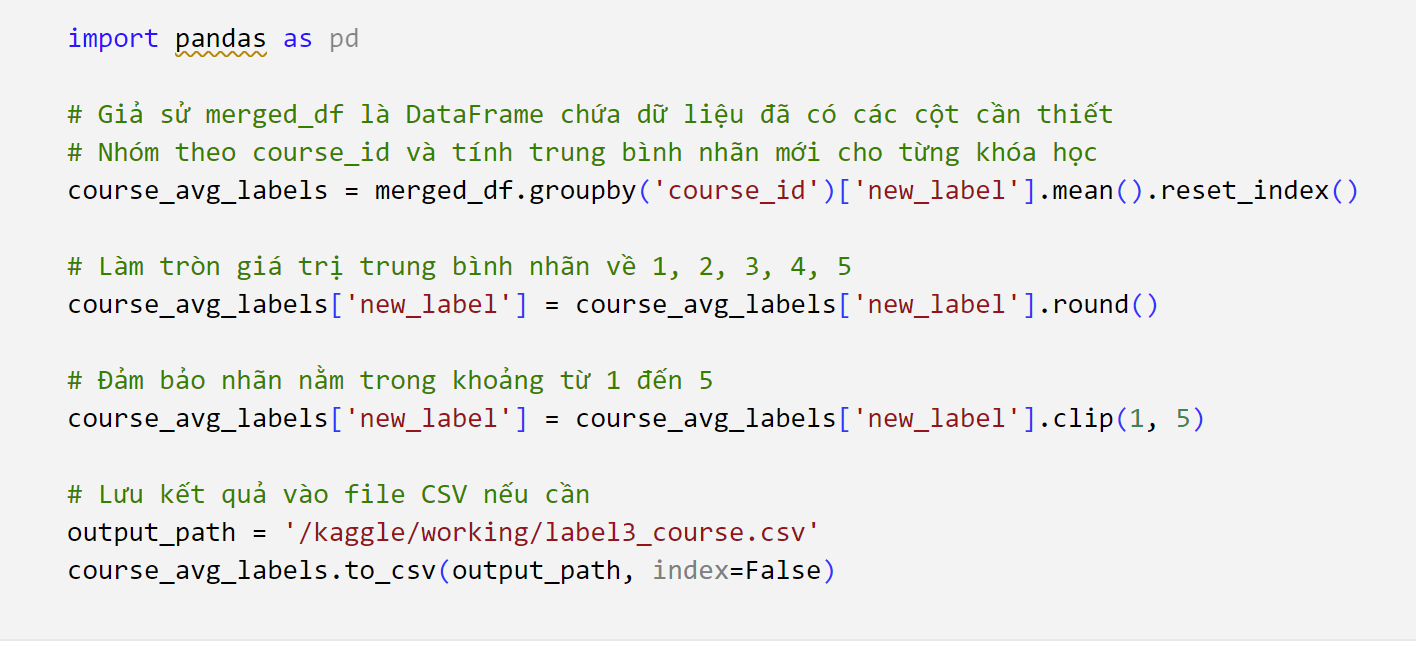
* Nếu cả time\_label và freq\_label đều thấp (Low), thì nhãn mới sẽ là 1.
* Nếu một trong hai nhãn time label hoặc freq\_label có giá trị Medium và còn lại là Low, thì nhãn mới sẽ là 2.
* Nếu một trong hai nhãn time\_label hoặc freq label có giá trị High và còn lại là Low hoặc Medium, thì nhãn mới sẽ là 3.
* Nếu cả hai nhãn time\_label và freq\_label đều có giá trị Medium, nhãn mới sẽ là 4.
* Nếu cả hai nhãn đều có giá trị High, nhãn mới sẽ là 5.



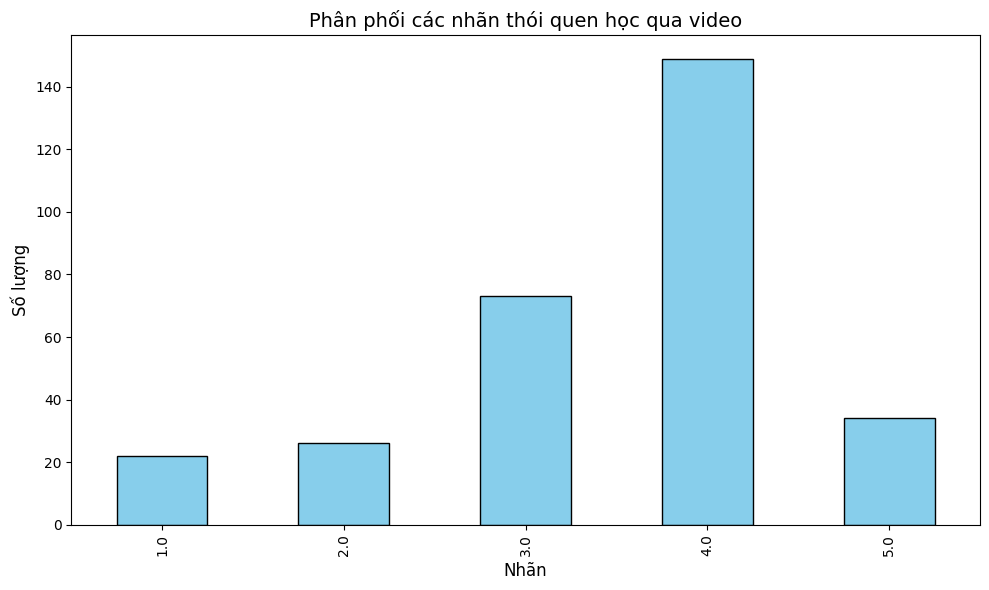
Kết quả thu được:



Cuối cùng, nhóm sẽ gán nhãn vừa thu được cho các khóa học:



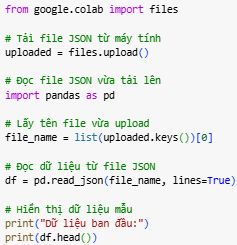
Sau đó ta thu được một bộ dữ liệu đã gán nhãn. Dưới đây là biểu đồ phân phối nhãn của khóa học theo tiêu chí 2: Tần suất và thời gian học trên video.



Hình X: Biểu đồ phân phối nhãn thói quen học qua video

### 1.3. Tiêu chí 3: Số lần thử và nộp bài tập

#### 1.3.1. Xử lý và phân tích dữ liệu đầu vào

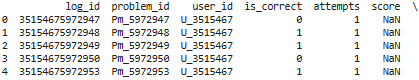
****

Hình 1.4a: Upload và đọc dữ liệu từ file JSON

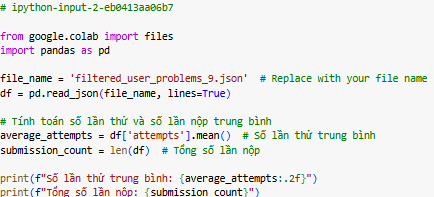
● **Dữ liệu đầu vào** bao gồm các thông tin về số lần thử (attempts), thời gian nộp bài (submit\_time), điểm số (score), và kết quả đúng/sai (is\_correct) cho từng bài tập của học viên.

● **Mục tiêu:** Đánh giá mức độ thử thách của khóa học dựa trên số lần thử và nộp bài của học viên.

#### 1.3.2. Xây dựng chỉ số đánh giá

****

Hình 1.4b: Dữ liệu ban đầu

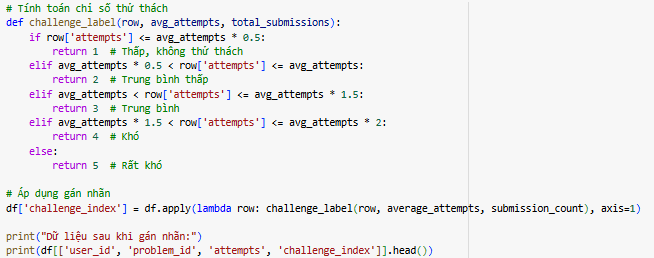
****

Hình 1.4c: Tính toán số lần nộp và thử trung bình

**Số lần thử trung bình (Average Attempts)**: Đánh giá mức độ khó khăn của bài tập thông qua việc học viên phải thử lại nhiều lần.

* **Đánh giá độ khó của bài tập**: Số lần thử trung bình phản ánh mức độ khó khăn của bài tập trong khóa học. Nếu học viên cần thử lại nhiều lần để hoàn thành bài tập, điều này có thể cho thấy bài tập không dễ dàng và yêu cầu sự nỗ lực để hiểu và giải quyết.
* **Hiệu quả giảng dạy và hướng dẫn**: Một số lần thử cao bất thường có thể cho thấy rằng phần hướng dẫn, bài giảng hoặc nội dung bài tập chưa rõ ràng, khiến học viên dễ mắc lỗi hoặc không hiểu cách tiếp cận đúng.
* **Hành vi học tập của học viên**: Số lần thử trung bình còn thể hiện sự kiên trì và cố gắng của học viên khi gặp khó khăn. Một số lần thử cao có thể chỉ ra rằng học viên quyết tâm hoàn thành bài tập dù gặp khó khăn ban đầu.
* **Cải thiện thiết kế bài tập**: Nếu nhiều bài tập có số lần thử trung bình cao hơn mức mong đợi, giảng viên có thể cần điều chỉnh thiết kế bài tập sao cho phù hợp với trình độ học viên.

#### 1.3.3. Gán nhãn thử thách (Challenge Label)

****

Hình 1.4d: Tính Chỉ số thử thách và gán nhãn

● **Chỉ số thử thách (Challenge Index)**:

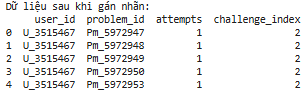
o Được tính dựa trên tỉ lệ số lần thử và nộp bài so với mức trung bình của khóa học.

o Nếu Challenge Index càng cao, bài tập càng thử thách.

● **Gán nhãn (1-5)**:

o Phân loại độ thử thách dựa trên giá trị của Challenge Index:

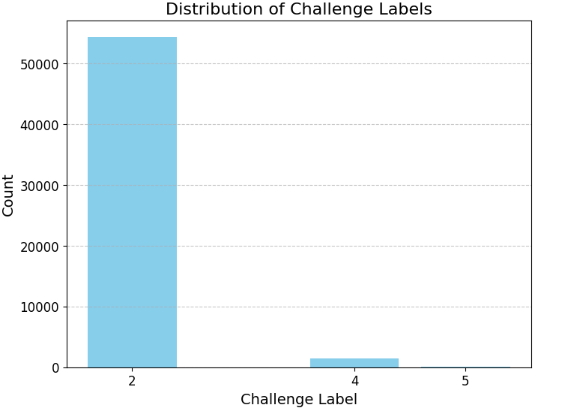
* **1 (Rất dễ)**: Nếu Challenge Index <= 0.5 \* average\_attempts
* **2 (Dễ)**: Nếu 00.5 \* average\_attempts < Challenge Index <= average\_attempts
* **3 (Trung bình)**: Nếu average\_attempts < Challenge Index <= 1.5 \* average\_attempts
* **4 (Khó)**: Nếu 1.5 \* average\_attempts < Challenge Index <= 2 \* average\_attempts
* **5 (Rất khó)**: Nếu Challenge Index > 2 \* average\_attempts



Hình 1.4e: Dữ liệu sau khi gán nhãn

#### 1.3.4. Tạo sơ đồ phân phối nhãn

Sau khi gán nhãn cho từng bài tập, tính toán phân phối của các nhãn (số lượng bài tập ở mỗi mức thử thách).



Hình 1.4f: Biểu đồ phân phối nhãn theo tiêu chí số lần thử và nộp bài tập

Từ biểu đồ trên, ta có thể nhận xét như sau: Phần lớn các bài tập được gán nhãn "2", chiếm áp đảo trong tổng số, phản ánh rằng các bài tập chủ yếu có độ thử thách vừa phải, không quá khó

### 1.4. Tiêu chí 4: Mức độ hài lòng của học viên đối với khóa học

Trước khi xây dựng mô hình phân loại ngữ nghĩa, ta cần phải xác định loại bài toán cần phải giải quyết. Nhận thấy, các bình luận (comment) đều chưa có nhãn, vì thế, ta có thể quy về một bài toán phân cụm (clustering).

#### 1.4.1. Lựa chọn mô hình

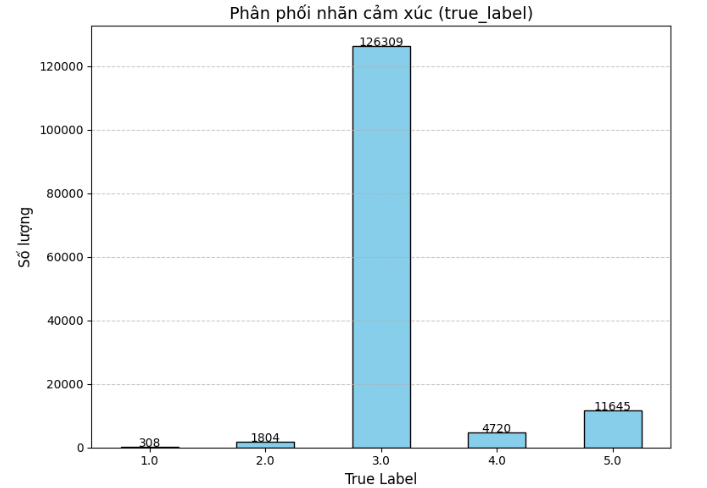
Để có thể phân cụm các bình luận vào 3 nhãn good, bad, neutral mà hoàn toàn không biết trước ground truth của các bình luận thực sự là một nan đề.

Một phương pháp đề xuất là sử dụng pretrained model đã được huấn luyện trên một bộ dữ liệu khác, để dự đoán nhãn tương ứng cho các bình luận. Tuy nhiên, phương pháp này gặp nhiều trở ngại do mỗi bộ dữ liệu có đặc điểm đặc thù, cho nên nếu có được pretrained model, ta cũng cần phải fine-tune mô hình để có thể nâng cao hiệu quả dự đoán – cần chuẩn bị trước một bộ dữ liệu có nhãn (labelled dataset) để huấn luyện. Một phương pháp khác chính là gán nhãn thủ công. Phương pháp này vô cùng hao tổn chi phí, ở nhiều phương diện. Bộ ‘translated\_comments.csv’ có hơn 140 nghìn dòng, gán nhãn thủ công cho toàn bộ trong thời gian ngắn mà đảm bảo chất lượng là điều gần như bất khả thi. Nên chúng tôi thực hiện hybrid – gán nhãn thủ công một phần nhỏ của bộ rồi thực hiện supervised learning để dự đoán phần còn lại thì mô hình được huấn luyện cũng không có tính bao quát dẫn đến hiệu quả dự đoán cũng sẽ kém.

Trước tiên, nhóm thực hiện gán nhãn thủ công với 10,000 dòng bình luận đầu tiên của bộ dữ liệu. Sau đó, chúng tôi lựa chọn sử dụng mô hình phân cụm KNN nhằm gán nhãn những bình luận còn lại của bộ dữ liệu. Cụ thể, quy tắc gán nhãn cảm xúc bình luận được nhóm quy định cụ thể như sau:

* Nhãn ‘1’: Chí những cảm xúc tiêu cực như chán ghét, thù hằn, tức giận, khó chịu của người dùng đối với khóa học. Ví dụ: 'Bị mắc kẹt hoàn hảo', 'Thật khó chịu khi cứ liên tục gặp sự cố.', Tôi không hiểu gì cả.', 'Thầy này cứ mắc lỗi hoài',...
* Nhãn ‘2’: Những bình luận có cảm xúc không mấy tích cực, không quá thích, không ủng hộ, gặp vấn đề đối với khóa học. Ví dụ: 'Không thể tham gia lớp học trong Rain Classroom', 'Đang làm mới hơi lag', …
* Nhãn ‘3’: Những bình luận có cảm xúc trung tính, không bày tỏ cảm xúc với khóa học, đơn giản trình bày kiến thức chuyên môn, hoặc lời chào, câu vô nghĩa,... Ví dụ: 'Mọi người đều đến từ Đại học Thanh Hoa', 'Đếm ngược đến 2 phút', 'Đăng nhập', 'Hãy bắt đầu.'
* Nhãn ‘4’: Những bình luận có chứa cảm xúc tích cực đối với khóa học, như việc công nhận, mong chờ đối với khóa học. Ví dụ: 'Tôi thực sự muốn xem chương trình phát sóng trực tiếp này', 'Mong được bắt đầu', 'Xem mượt mà trên điện thoại di động', 'Kakaka', 'Chất lượng hình ảnh ở đây tốt hơn Rain Classroom', …
* Nhãn ‘5’: Những bình luận giàu cảm xúc tích cực, thích thú, … đối với khóa học. Ví dụ: 'Tôi đã thử nó và trải nghiệm thật tuyệt vời.', 'quá đơn giản', 'Cơ sở dữ liệu rất tốt', 'Nói rất hay', 'Dễ hiểu',...

Theo nhóm đánh giá trên bộ dữ liệu được gán nhãn tự động bởi KNN, các nhãn cảm xúc được gán tương đối chính xác. Tổng kết, ta thu được một bộ dữ liệu bình luận đã gán nhãn gồm 144,787 dòng dữ liệu, nhưng ứng với 445,054 dòng bình luận thực tế (Tổng giá trị thuộc tính ‘count’). Nhãn cảm xúc của bộ dữ liệu phân bố chủ yếu tại nhãn 3 và 4. Điều này là phù hợp với thực tế, do các bình luận thường mang tính đóng góp, thảo luận chuyên môn, số lượng bình luận giàu cảm xúc như ‘cảm ơn’, ‘rất hay’, … chỉ xuất hiện 1 phần nhỏ so với tổng số bình luận của 1 khóa học.



**Hình 3.1.4.a.1:** Biểu đồ phân phối nhãn cảm xúc của các bình luận

Nếu tính số lượng nhãn mới = count \* số lượng nhãn hiện có (1), ta có bảng phân phối cảm xúc các bình luận như **Hình 3.1.4.a.2**. Để giảm ảnh hưởng của nhãn ‘3’, nhãn trung tính, không chứa cảm xúc, xuất hiện phần lớn trong các khóa học, ta đưa giá trị ‘count’ của các bình luận gán nhãn này thành 1. Tương tự, ta thực hiện vẽ bảng phân phối nhãn bình luận theo công thức (1), thu được **Hình 3.1.4.a.3**. Theo đó, ta thấy việc giảm giá trị của ‘count’ với nhãn 3 giúp giải quyết phần nào việc mất cân bằng nhãn trong dữ liệu. Và hơn hết, mục tiêu của tiêu chí 5 là gán nhãn mức độ hài lòng của học viên, tức các nhãn mang cảm xúc rõ rệt hơn như ‘1’, ‘2’, ‘4’, ‘5’ cần được ưu tiên hơn.

| **Hình 3.1.4.a.2:** Phân phối nhãn theo (1) | **Hình 3.1.4.a.3:** Phân phối nhãn theo  (1) đã giảm count của nhãn 3 |
| --- | --- |

#### 1.4.2. Gán nhãn cho khóa học dựa trên tiêu chí 5 (Mức độ hài lòng của học viên)

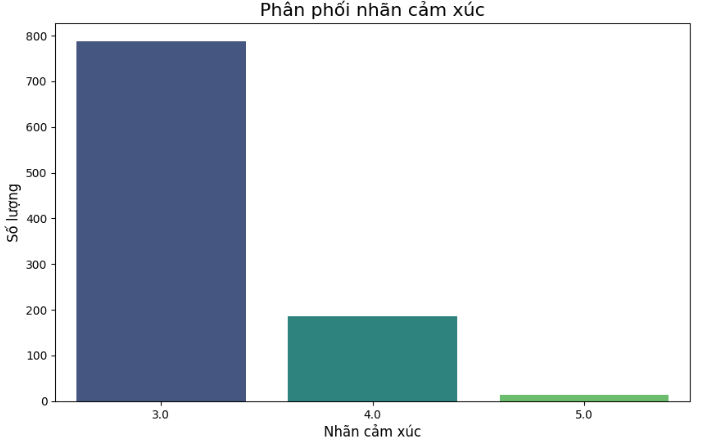
Đối với những dữ liệu bị thiếu (missing values) cụ thể là có các khóa học không có bình luận nào, ta sẽ thực hiện chọn chiến lược điền khuyết hợp lý (imputation): mean, median, most\_frequent hoặc fix value. Vì các khóa học đó không có bình luận nào, nên ta xem như chúng tương tự với khóa học mà chỉ có bình luận neutral, để không overrated hay underrated khóa học đó mà chưa có dữ liệu cơ sở để tính toán. Vì thế ta sẽ cho các khóa học đó với score là fix value = 0, và tương ứng xếp hạng theo quantile đã được định nghĩa trước.

Do các bình luận của khóa học đã được đánh nhãn cảm xúc trước đó, ta chỉ cần thực hiện tính trung bình giá trị cảm xúc theo từng khóa học. Lưu ý, ở đây để giảm ảnh hưởng của các bình luận được đánh giá trị nhãn ‘3’ (trung tính), ta chuyển các ‘count’ của các bình luận này về bằng 1. Đoạn code thực hiện cụ thể như sau:



**Hình 3.1.4.b.1**: Đoạn code thực hiện gán nhãn mức độ hài lòng cho khóa học

Sau đó, ta thu được một danh sách gồm 986 khóa học đã được gán nhãn cảm xúc. Dưới đây là biểu đồ phân phối các nhãn mức độ hài lòng của học viên đối với khóa học.

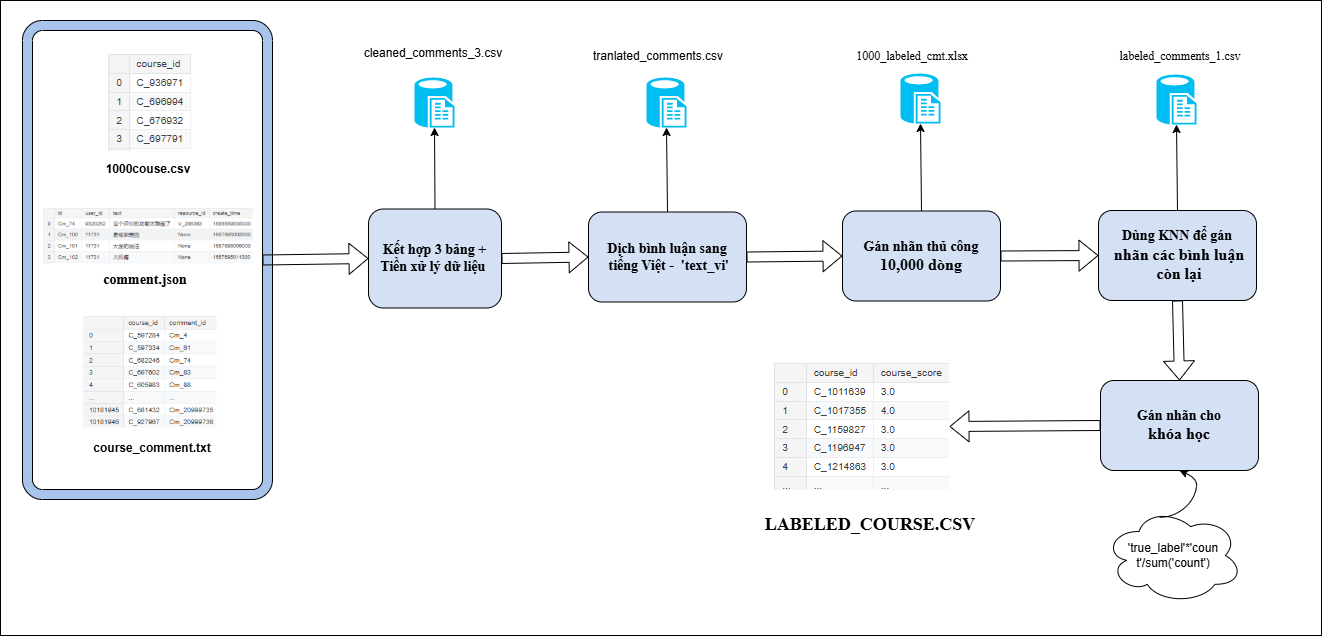


**Hình 3.1.4.b.2**: Biểu đồ phân phối nhãn mức độ hài lòng của học viên

Quan sát **Hình 3.1.4.b.2** - biểu đồ phân phối nhãn mức độ hài lòng của học viên - chúng tôi nhận thấy, các khóa học 3 sao chiếm phần lớn và các khóa học đạt 4 và 5 sao chỉ chiếm số ít. Điều này phù hợp vì:

* Các khóa học càng chất lượng thì càng ít biểu thị tiêu chí xếp hạng vô cùng chặt chẽ.
* Các khóa học 3 sao chiếm phần lớn bởi vì đa số các khóa học sẽ chủ yếu gồm các bình luận mang tính góp ý, và những bình luận này được gán nhãn 3, số lượng bình luận giàu cảm xúc như ‘cảm ơn’, ‘rất hay’, … chỉ xuất hiện 1 phần nhỏ so với tổng số bình luận của 1 khóa học nên điểm trung bình của khóa học có khả năng cao dao động xung quanh ngưỡng 3.

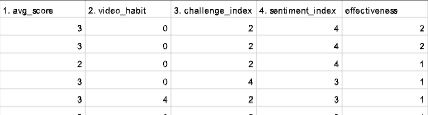
Tổng quan lại, ta sẽ có pipeline hoàn chỉnh cho quá trình gán nhãn theo tiêu chí 5 như sau:



**Hình 3.1.4.b.3**: Pipeline quá trình gán nhãn theo tiêu chí 5

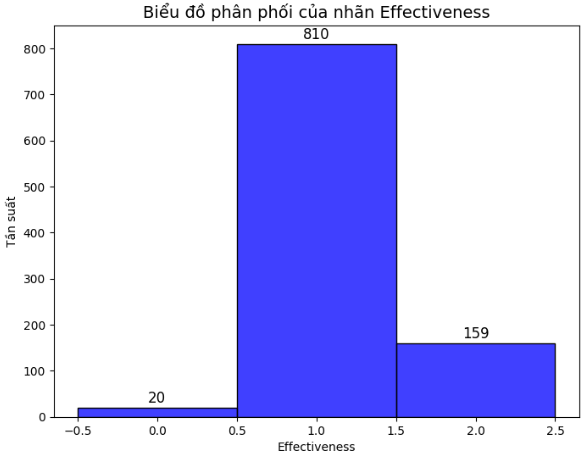
## Gán nhãn cho khóa học

Thực hiện gán nhãn cho khóa học bằng việc đánh giá khách quan các yếu tố (4 tiêu chí mà nhóm đề cập trong bài). Theo đó, nhóm đánh giá cao yếu tố ‘Tiêu chí 4: Mức độ hài lòng của học viên đối với khóa học) với việc thể hiện được quan điểm, góc nhìn, thái độ của học viên đối với khóa học…. Sau khi đã thống nhất quy tắc gán nhãn độ hiệu quả của khóa học, ta thực hiện và thu được một bộ dữ liệu gồm 1000 khóa học đã được gán nhãn như sau:



**Hình 3.2.1.** Các nhãn đánh giá trong bài toán

Nhãn Độ hiệu quả của khóa học hay ‘effectiveness’ được phân phối như hình X. Ta nhận thấy rõ ràng việc mất cân bằng ở các nhãn. Việc này ảnh hưởng 1 phần do việc mất cân bằng ở nhãn ‘4. Sentiment Index’ - nhãn có khả năng ảnh hưởng đến 50% giá trị nhãn ‘effectiveness’. Tuy nhiên, nhìn nhận thực tế thì kết quả như hình X là hoàn toàn hợp lý, trong 1000 khóa học đang xét, có đến 980 khóa học đạt độ hiệu quả từ mức trung bình đến cao, chỉ có 20 khóa học đạt độ hiệu quả thấp. Rất có khả năng đây là một trong những khóa học bị gán nhiều nhãn cảm xúc tiêu cực ở bình luận (giá trị ‘4.sentiment index’ thấp).



**Hình 3.2.2.**.Biểu đồ phân phối nhãn Effectiveness

# CHƯƠNG 4: XÂY DỰNG MÔ HÌNH

## Dữ liệu sử dụng

Sau khi thu thập và gán nhãn dữ liệu, chúng tôi sẽ tiến hành xây dựng một mô hình để dự đoán sớm chất lượng của khóa học ngay từ thời điểm khóa học được tạo. Mục tiêu của chúng tôi là giúp giảng viên và nhà quản lý khóa học (các trường đại học) có thể nắm bắt được những vấn đề tiềm ẩn ngay từ giai đoạn đầu, từ đó giúp họ có thể điều chỉnh và cải thiện chất lượng khóa học một cách kịp thời.

Khi một khóa học mới được tạo, khóa học sẽ có các thông tin cơ bản như sau: tên khóa học, danh sách các ngành liên quan, kiến thức tiên quyết, trường và giảng viên mà khóa học đó thuộc về. Từ những thông tin trên, ta cần phải lựa chọn những đặc trưng phù hợp, có ý nghĩa và có độ tương quan cao với nhãn chất lượng khóa học đã được gán. Tên khóa học, kiến thức tiên quyết và các ngành liên quan thường sẽ là các thuật ngữ chuyên ngành nên không mang nhiều ý nghĩa đối với chất lượng khóa học. Trong khi đó, chất lượng của trường và giảng viên giảng dạy sẽ là những đặc trưng tiềm năng, và chúng tôi cũng đã sử dụng hai đặc trưng này để thực hiện bước gán nhãn trước đó. Bên cạnh đó, chúng tôi sẽ sử dụng thêm thông tin mô tả của khóa học để dự đoán. Theo quan sát được, một số khóa học có thông tin mô tả rất đầy đủ và chi tiết, một số khóa học bỏ trống thông tin này hoặc chỉ viết một cách đối phó với các kí tự ngẫu nhiên… Vậy nên dựa vào thông tin mô tả, ta hy vọng nó có thể phần nào dự đoán được độ chỉn chu của thiết kế khóa học sau này, một yếu tố ảnh hưởng lớn đến chất lượng khóa học.

## Đề xuất mô hình

### Tổng quan bài toán

Nhóm xây dựng một số mô hình máy học cơ bản nhằm dự đoán nhanh độ hiệu quả của khóa học khi chỉ có các thông tin cơ bản của khóa học. Do thông tin khóa học (course.json) chỉ gồm 5 thuộc tính, ta lựa chọn 2 thuộc tính là ‘name’ và ‘about’ làm biến đầu vào, do các thuộc tính còn lại nhiều giá trị NaN. Hướng phát triển theo dạng bài toán:

* Input: Thông tin cơ bản của khóa học, cụ thể ở đây là tổ hợp ‘name’ + ‘about’
* Output: Độ hiệu quả của khóa học (nhãn 0 -> 2)

Chúng tôi nhận thấy bài toán không quá phức tạp nên lựa chọn sử dụng 4 mô hình máy học cơ bản (Logistic Regression, Random Forest, SVM, XGBoost) để dự đoán độ hiệu quả của khóa học. Ngoài ra, để giải quyết yếu tố mất cân bằng của bộ dữ liệu, chúng tôi có sử dụng kết hợp các mô hình này với SMOTE, với mong muốn cải thiện được hiệu suất dự đoán.

### 2. Lựa chọn mô hình và thuật toán

Để dự đoán độ hiệu quả của khóa học dựa trên dữ liệu hạn chế, nhóm đã sử dụng 4 thuật toán máy học cơ bản nhằm nhanh chóng đánh giá khả năng dự đoán của bài toán. Các thuật toán này được lựa chọn vì chúng phù hợp với quy mô dữ liệu nhỏ và cấu trúc đơn giản của bài toán.

#### 2.1. Logistic Regression

* **Đặc điểm**: Logistic Regression là một thuật toán tuyến tính, sử dụng hàm sigmoid để dự đoán xác suất của các lớp đầu ra. Đây là một trong những mô hình phổ biến nhất trong phân loại nhị phân và phân loại đa lớp.
* **Ưu điểm**:
  + Đơn giản, dễ triển khai và dễ hiểu.
  + Thích hợp cho việc thiết lập baseline nhanh chóng.
  + Hiệu quả trong các bài toán mà mối quan hệ giữa biến đầu vào và đầu ra là tuyến tính.
* **Hạn chế**:
  + Không phù hợp với dữ liệu phi tuyến tính hoặc khi có sự tương tác phức tạp giữa các biến.
  + Nhạy cảm với dữ liệu mất cân bằng, dễ bị lệch về lớp chiếm đa số.

#### 2.2. Random Forest

* **Đặc điểm**: Random Forest là một thuật toán học ensemble, được xây dựng bằng cách tạo ra nhiều cây quyết định (decision trees) độc lập thông qua kỹ thuật bagging (Bootstrap Aggregating). Dự đoán cuối cùng được lấy bằng cách trung bình hoặc bỏ phiếu từ các cây.
* **Ưu điểm**:
  + Có khả năng xử lý dữ liệu mất cân bằng tốt hơn Logistic Regression.
  + Giảm thiểu overfitting bằng cách kết hợp nhiều cây độc lập.
  + Hiệu quả với dữ liệu đa chiều và có nhiễu.
* **Hạn chế**:
  + Tốn tài nguyên tính toán nếu số lượng cây quá lớn.
  + Khó giải thích kết quả so với Logistic Regression.

#### 2.3. SVM (Support Vector Machine)

* **Đặc điểm**: SVM là một thuật toán mạnh mẽ dựa trên việc tìm siêu phẳng tối ưu (optimal hyperplane) để phân tách dữ liệu. Khi dữ liệu không thể phân tách tuyến tính, SVM sử dụng **kernel trick** để chuyển đổi dữ liệu sang không gian chiều cao hơn, nơi dữ liệu có thể được phân tách tuyến tính.
* **Ưu điểm**:
  + Hiệu quả với dữ liệu có biên phân tách rõ ràng.
  + Tương thích tốt với dữ liệu phi tuyến tính thông qua kernel trick.
  + Bền vững trước nhiễu trong dữ liệu.
* **Hạn chế**:
  + Không phù hợp với dữ liệu lớn vì chi phí tính toán cao.
  + Đòi hỏi lựa chọn kernel và tham số tinh chỉnh cẩn thận.

#### 2.4. XGBoost (Extreme Gradient Boosting)

* **Đặc điểm**: XGBoost là một thuật toán boosting mạnh mẽ, sử dụng các cây quyết định được xây dựng tuần tự để giảm lỗi dự đoán. Mỗi cây mới được tạo ra nhằm sửa lỗi của cây trước đó.
* **Ưu điểm**:
  + Hiệu suất cao, tối ưu hóa tốc độ tính toán và sử dụng bộ nhớ hiệu quả.
  + Khả năng xử lý tốt dữ liệu mất cân bằng và các bài toán phức tạp.
  + Được sử dụng rộng rãi trong các bài toán thực tế và cuộc thi khoa học dữ liệu.
* **Hạn chế**:
  + Cần tinh chỉnh nhiều tham số để đạt hiệu suất tốt nhất.
  + Dễ bị overfitting nếu không điều chỉnh tham số cẩn thận.

### 3. Giải pháp xử lý mất cân bằng dữ liệu

#### 3.1. Tình trạng mất cân bằng trong dữ liệu

Dữ liệu trong bài toán có sự phân bố không đồng đều giữa các lớp nhãn, với một lớp chiếm ưu thế rõ rệt so với các lớp còn lại như **Hình 3.2.2**. Điều này khiến các mô hình dễ nghiêng về lớp chiếm đa số và bỏ qua lớp thiểu số, dẫn đến hiệu suất dự đoán thấp trên lớp nhỏ.

#### 3.2. Kỹ thuật SMOTE (Synthetic Minority Oversampling Technique)

* **Phương pháp**:  
  SMOTE là một kỹ thuật oversampling được sử dụng để tăng cường số lượng mẫu của lớp thiểu số bằng cách tạo ra dữ liệu tổng hợp. Các mẫu mới được sinh ra dựa trên khoảng cách giữa các điểm dữ liệu hiện có của lớp thiểu số trong không gian đặc trưng.
* **Ưu điểm**:
  + Cân bằng số lượng mẫu giữa các lớp, giúp cải thiện khả năng học của mô hình trên lớp thiểu số.
  + Không tạo ra dữ liệu lặp lại như các kỹ thuật oversampling truyền thống.
  + Dễ dàng tích hợp vào pipeline xử lý dữ liệu.
* **Ứng dụng**:  
  SMOTE được áp dụng trong giai đoạn tiền xử lý, sau đó dữ liệu được đưa vào huấn luyện với các mô hình Logistic Regression, Random Forest, SVM, và XGBoost.

#### 3.3. Kết hợp SMOTE với các mô hình máy học

* Các mô hình Logistic Regression, Random Forest, SVM, và XGBoost khi kết hợp với SMOTE sẽ có thể cải thiện đáng kể hiệu suất trên lớp thiểu số. Tuy nhiên, mức độ cải thiện phụ thuộc vào bản chất của từng thuật toán và cách xử lý dữ liệu đầu vào.
* Mặc dù vậy, hạn chế của dữ liệu nhỏ và đơn giản vẫn ảnh hưởng đến hiệu quả tổng thể của bài toán.

## Độ đo đánh giá

Với bộ dữ liệu có sự mất cân bằng đáng kể giữa các lớp, chúng tôi lựa chọn chỉ số đánh giá chính là Weighted F1, kết hợp với các chỉ số Accuracy, Precision, và Recall để phân tích thêm về độ chính xác và khả năng bao phủ của mô hình. Ngoài ra, chúng tôi sử dụng Confusion Matrix để cung cấp minh họa chi tiết hơn về hiệu suất phân loại của mô hình.

* Accuracy (độ chính xác) đo lường tỷ lệ dự đoán đúng so với tổng số dự đoán, với công thức:

Giá trị Accuracy thể hiện mức độ tổng quát của mô hình trong việc dự đoán chính xác tất cả các lớp. Với bài toán phân loại nhị phân, giá trị Accuracy nên lớn hơn 0.5 (tương đương với dự đoán ngẫu nhiên). Với bài toán phân loại đa lớp, Accuracy kỳ vọng phải lớn hơn tần suất xuất hiện của lớp chiếm đa số.

Accuracy thích hợp để dùng cho những tập dữ liệu cân bằng tốt giữa các lớp (well-balanced) và không lệch chuẩn.

Đối với tập dữ liệu bị mất cân bằng, accuracy không phản ánh rõ hiệu suất của từng lớp, bởi kết quả accuracy là tổng quát của các lớp nên sẽ nghiêng về lớp chiếm đa số trong tập dữ liệu. Kết quả Accuracy cao có thể chỉ do mô hình dự đoán tốt trên lớp chiếm đa số, nhưng lại kém hiệu quả với lớp thiểu số. Vậy nên chúng tôi đã sử dụng thêm các độ đo khác như weighted F1, Precision và Recall để đánh giá.

* Precision: hay Positive Predictive Value (PPV) đánh giá mức độ chính xác khi mô hình dự đoán một lớp cụ thể. Công thức tính Precision:

Precision phản ánh tỷ lệ các dự đoán đúng trong số các dự đoán thuộc lớp Positive, đặc biệt hữu ích trong trường hợp số lượng FP cao hơn FN. Nhờ việc tối ưu Precision, số lượng FP sẽ giảm đi và sẽ cho những quyết định đúng và chuẩn hơn.

Tuy nhiên, nếu chỉ sử dụng precision chưa thể biết được model có thực sự tốt hay không. Nếu precision quá cao và recall quá thấp, model sẽ có xu hướng bỏ qua nhiều trường hợp FN hoặc tệ hơn là nó chỉ cần dự đoán tất cả là negative thì khi đó precision sẽ vô cùng cao.

* Recall

Recall (hay Sensitivity, Hit Rate, hoặc True Positive Rate) đo lường khả năng mô hình bắt được các mẫu thực sự thuộc lớp Positive. Recall được tính theo công thức:

Recall tập trung vào việc giảm thiểu False Negative (FN) và phù hợp với các bài toán ưu tiên tránh bỏ sót mẫu Positive. Ví dụ: Trong bài toán phát hiện bệnh, Recall cao đảm bảo rằng hầu hết các bệnh nhân thực sự mắc bệnh được phát hiện.

Hiểu đơn giản, recall đang cố thực hiện chiến lược đánh lầm còn hơn bỏ sót. Tuy nhiên, cũng vì thế mà nó có một nhược điểm lớn đó là để recall cao thì nó chỉ cần đơn giản cho tất cả mọi mẫu là positive, khi đó thì recall sẽ bằng

* Weighted-F1

Weighted F1 là một chỉ số đánh giá hiệu suất mô hình phân loại đa lớp, tính toán trung bình độ F1 trọng số trên tất cả các lớp. Đây là chỉ số tổng hợp cân bằng giữa Precision và Recall, được tính toán có trọng số dựa trên kích thước của từng lớp. Công thức tính Weighted F1:

Với là F1 cho lớp i, là trọng số cho lớp i.

Giá trị Weighted-F1 cao (thường từ 0.7 đến 1.0) thể hiện mô hình có hiệu suất tốt trong việc phân loại tất cả các lớp, đặc biệt chú trọng đến các lớp có số lượng mẫu lớn.

Giá trị Weighted-F1 thấp (thường dưới 0.5) cho thấy mô hình cần cải thiện khả năng phân loại một hoặc nhiều lớp đặc biệt là các lớp có số lượng mẫu lớn

Weighted F1 phụ thuộc vào cách chọn trọng số cho các lớp.

* Confusion Matrix

Ma trận lỗi (confusion matrix) là một bố cục bảng cụ thể cho phép hình dung hiệu suất thuật toán Machine Learning. Đây là một trong những kỹ thuật đo hiệu suất phổ biến cho các mô hình phân loại.

Confusion matrix có kích thước nxn tương ứng với n lớp. Trong đó, các hàng tượng trưng cho kết quả dự đoán còn các cột tượng trưng cho kết quả trong thực tế.

* True Positive (TP): Dự đoán đúng và thực tế đúng.
* True Negative (TN): Dự đoán sai và thực tế sai.
* False Positive (FP): Dự đoán sai nhưng thực tế đúng.
* False Negative (FN): Dự đoán đúng nhưng thực tế sai.

Recall và Accuracy, đặc biệt hữu ích trong việc đánh giá mô hình trên tập dữ liệu mất cân bằng. Một model tốt (good model) sẽ có tỉ lệ TP và TN cao, trong đó tỉ lệ FN, FP thấp. Việc sử dụng confusion matrix cho những tập dữ liệu bị mất cân bằng (imbalanced dataset) cũng sẽ cho ta biết chi tiết hơn về hiệu suất của model. Các metrics như accuracy, precision… thực chất chỉ là phiên bản mở rộng của Confusion Matrix.

## Kết quả mô hình

Chúng tôi chia bộ dữ liệu ra làm 2 phần train và test với tỉ lệ 80:20, và đảm bảo rằng mỗi tệp tỉ lệ cáp nhãn là như nhau. Bộ train sẽ được sử dụng để huấn luyện mô hình. Bộ test sẽ được giữ để thực hiện dự đoán và đánh giá hiệu suất mô hình. Sau khi thực hiện huấn luyện bộ dữ liệu với mô hình đề xuất, chúng tôi thu được kết quả như sau:

| **Mô hình** | **Accuracy** | **Weighted precision avg** | **Weighted recall avg** | **Weighted f1-score avg** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.8332 | 0.83 | 0.82 | 0.75 |
| Random Forest | 0.7980 | 0.70 | 0.80 | 0.73 |
| SVM | 0.8131 | 0.73 | 0.81 | 0.74 |
| XGBoost | 0.8182 | 0,76 | 0,82 | 0.76 |
| LogisticRegression…class\_weight='balanced'...) + SMOTE | 0.7626 | 0.70 | 0.76 | 0.72 |
| RandomForestClassifier(class\_weight='balanced'...) + SMOTE | 0.7879 | 0.70 | 0.79 | 0.73 |
| SVC(class\_weight='balanced’...) + SMOTE | 0.7626 | 0.69 | 0.76 | 0.72 |
| XGBClassifier(scale\_pos\_weight=1…) + SMOTE | 0.7828 | 0.71 | 0.78 | 0.74 |

HÌnh 4.1-a Kết quả các mô hình dự đoán

Trong các mô hình mà nhóm thực hiện, mô hình cho kết quả tốt nhất là SVM có sử dụng SMOTE. Kết quả này được đánh giá dựa trên độ chính xác (accuracy), và việc gán chính xác các nhãn (precision, f1-score, recall ở từng nhãn).

|  |  |
| --- | --- |

Hình 4.1-b: kết quả mô hình tốt nhất - SVM có sử dụng SMOTE

Với kỳ vọng Accuracy lớn hơn 0.35 nếu sử dụng random guessing, có thể thấy kết quả đạt được với Weighted F1-score khoảng 0.74 và Accuracy 78,28% cho thấy mô hình cũng đạt được hiệu quả tương đối tốt. Tuy nhiên, mô hình còn dự đoán sai khá nhiều ở nhãn 0 và 2 (giá trị độ đo recall, f1-score đều rất thấp), điều đó cho thấy rằng mô hình vẫn chưa xử lý tốt trên bộ dữ liệu mất cân bằng. Bằng chứng cho việc này chính là Macro-F1 score tương đối thấp, chỉ đạt 0.34, và F1-score trên các lớp 0, 2 cũng khá thấp.

Về nguyên nhân mô hình hoạt động chưa thực sự hiệu quả là do:

* Hạn chế dữ liệu: Dữ liệu đầu vào nhỏ (1000 dòng) và mất cân bằng nghiêm trọng, dù đã áp dụng SMOTE nhưng chưa khắc phục hoàn toàn.
* Hạn chế đặc trưng: Các biến đầu vào chưa đủ đa dạng để phản ánh đầy đủ đặc điểm bài toán. Bài toán này chỉ mới lấy thông tin cơ bản của khóa học gồm ‘name’, và ‘about’ để dự đoán biến mục tiêu 'effectiveness'; trong khi biến này được xây dựng trên nhiều tiêu chí, lần lượt là: Điểm trung bình khóa học; Thói quen học qua video; Nỗ lực làm bài tập; Mức độ hài lòng của học viên.

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi đã thành công xây dựng bài toán Đánh giá độ hiệu quả của khóa học. Chúng tôi đánh giá các khóa học trong bộ dữ liệu Mooc CubeX với 4 tiêu chí, cụ thể là: Điểm trung bình khóa học; Thói quen học qua video; Nỗ lực làm bài tập; Mức độ hài lòng của học viên, từ số điểm mỗi tiêu chí ta biết được kết quả học tập, thói quen học tập qua video, mức độ hài lòng của học viên và độ khó, độ hiệu quả của các khóa học. Kết quả chúng tôi thu được một bộ dữ liệu với 1,000 khóa học đã được gán nhãn ở cả 4 tiêu chí và đã gán nhãn độ hiệu quả 'effectiveness'.

Sau đó, chúng ta xây dựng mô hình dự đoán cơ bản với là 4 mô hình máy học (Logistic Regression, Random Forest, SVM, XGBoost) để dự đoán hiệu quả của các khóa học khi mới biết một vài yếu tố cơ bản của khóa học (‘name’ + ‘about’ - thông tin ở bảng course.json). Mô hình dự đoán cũng cho kết quả dự đoán tương đối ổn, với độ chính xác accuracy luôn đạt trên 0.76. Tuy nhiên, dù có sử dụng phương pháp khắc phục mất cân bằng dữ liệu là SMOTE, thì các mô hình vẫn còn dự đoán sai khá nhiều ở nhãn 0 và 2 (giá trị độ đo recall, f1-score đều rất thấp), điều đó cho thấy rằng mô hình vẫn chưa xử lý tốt trên bộ dữ liệu mất cân bằng. Mô hình tốt nhất là SVM có sử dụng SMOTE, cho Weighted F1-score khoảng 0.74 và Accuracy 78,28%, tuy nhiên mô hình này vẫn chưa thể dự đoán chính xác ở nhãn ‘0’, ‘2’.

Trong bài toán này, chúng tôi đã thành công xây dựng và gán nhãn bộ dữ liệu theo các tiêu chí đánh giá Độ hiệu quả khóa học mà nhóm đề ra (4 tiêu chí), cùng xây dựng mô hình dự đoán nhanh độ hiệu quả khóa học khi chỉ biết thông tin cơ bản của khóa học.

Từ kết quả đó, các nhà phát hành khóa học, giáo viên, … dễ dàng hơn trong việc tìm và khắc phục những khó khăn của học viên trong quá trình học tập, rút ra được vấn đề còn tồn động và đưa ra hướng giải quyết. Ví dụ: từ kết quả nộp bài tập của các học viên có thấy những vướng mắc và khó khăn của học viên để đưa ra hướng giải quyết để giúp học viên của các khóa được cải thiện.

Trong quá trình xây dựng và thực hiện bài toán, chúng tôi đã phải đối mặt với nhiều thách thức như: Bộ dữ liệu Mooc CubeX quá lớn (dữ liệu bảng comments có đến hơn 8 triệu dòng, bảng ….) cần tiền xử lý, giảm kích thước dữ liệu mà đảm bảo chất lượng dữ liệu không ảnh hưởng quá nhiều kết quả bài toán; cần chia thành nhiều file nhỏ để chạy; cần định nghĩa đúng tiêu chí và phương pháp gán nhãn,...

Bài toán còn tiềm năng phát triển trong tương lai. Đối với nhà cung cấp khóa học, mở rộng bài toán này có thể cho ra kết quả là hệ thống tính điểm khóa học tự động. Ví dụ, mỗi khi có một bình luận mới, bình luận sẽ được hệ thống tự động tính điểm cảm xúc, sau đó cập nhật điểm cảm xúc mới cho khóa học (Tiêu chí 5). Hệ thống này có tiềm năng giúp các nhà quản lý giáo dục nhận được thông tin chân thực, khách quan, và nhanh nhất từ học viên.

# TÀI LIỆU THAM KHẢO

1. Bộ dữ liệu MOOC CubeX. Đường dẫn:<https://github.com/THU-KEG/MOOCCubeX?tab=readme-ov-file>
2. Breiman, L, “*Random Forests*. *Machine Learning”*, Volume 45, pages 5–32, October 2001. Đường dẫn trực tuyến:<https://doi.org/10.1023/A:1010933404324>
3. scikit-learn.org, “**1.4. Support Vector Machines**”, 2024. Đường dẫn:<https://scikit-learn.org/stable/modules/svm.html>
4. xgboost.readthedocs.io, “Introduction to Boosted Trees”, 2024. Đường dẫn:<https://xgboost.readthedocs.io/en/stable/tutorials/model.html>
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., *“SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research”*, 2011. Đường dẫn:<https://doi.org/10.1613/jair.953>
6. scikit-learn.org, “3.4. Metrics and scoring: quantifying the quality of predictions”, 2024. Đường dẫn:<https://scikit-learn.org/stable/modules/model_evaluation.html>