**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------

Logo, company name

Description automatically generated

**PHẠM NGỌC HÙNG - 18133017**

**NGUYỄN ĐĂNG PHƯỚC TÍN - 18133056**

**LÂM HOÀNG VIỆT – 18133062**

**ĐỀ TÀI:**

**THEO DÕI VÀ DỰ ĐOÁN HIỆU SUẤT CỦA HỌC SINH QUA GAME GIÁO DỤC**

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ KỸ THUẬT DỮ LIỆU

**GIÁO VIÊN HƯỚNG DẪN**

**ThS. QUÁCH ĐÌNH HOÀNG**

**KHÓA 2018 –2022**

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**ĐH SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CNTT**

**\*\*\*\*\*\*\***

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**\*\*\*\*\*\*\*\*\***

Họ và tên Sinh viên 1: Phạm Ngọc Hùng MSSV: 18133017

Họ và tên Sinh viên 2: Nguyễn Đăng Phước Tín MSSV: 18133056

Họ và tên Sinh viên 3: Lâm Hoàng Việt MSSV: 18133062

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Theo dõi và dự đoán hiệu suất của học sinh qua game giáo dục.

Họ và tên giáo viên hướng dẫn: ThS. Quách Đình Hoàng

**NHẬN XÉT**

1. Về nội dung đề tài khối lượng thực hiện:

1. Ưu điểm:

1. Khuyết điểm:

1. Đề nghị cho bảo vệ hay không?
2. Đánh giá loại:
3. Điểm:

Tp*. Hồ Chí Minh, ngày tháng năm 2023*Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

**ĐH SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CNTT**

**\*\*\*\*\*\*\***

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**\*\*\*\*\*\*\*\*\***

Họ và tên Sinh viên 1: Phạm Ngọc Hùng MSSV: 18133017

Họ và tên Sinh viên 2: Nguyễn Đăng Phước Tín MSSV: 18133056

Họ và tên Sinh viên 3: Lâm Hoàng Việt MSSV: 18133062

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Theo dõi và dự đoán hiệu suất của học sinh qua game giáo dục.

Họ và tên giáo viên phản biện: TS. Nguyễn Thiên Bảo

**NHẬN XÉT**

1. Về nội dung đề tài khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

Tp*. Hồ Chí Minh, ngày tháng năm 2023*Giáo viên phản biện

*(Ký & ghi rõ họ tên)*

**LỜI CẢM ƠN**

Chúng em xin được gửi lời cảm ơn chân thành đến Thầy Quách Đình Hoàng. Thầy đã cung cấp tài liệu và hướng dẫn tận tình cho chúng em. Trong suốt quá trình thực hiện tiểu luận, Thầy luôn theo dõi tiến độ và giải đáp, chia sẻ giúp chúng em vượt qua những khó khăn. Chúng em rất trân quý sự tâm huyết và trách nhiệm của Thầy trong công việc giảng dạy và truyền đạt kiến thức.

Chúng em cũng xin gửi lời cảm ơn sâu sắc đến Thầy Cô khoa Công nghệ Thông tin - Đại học Sư phạm Kỹ thuật TP.HCM đã đồng hành và hỗ trợ chúng em trong suốt quá trình học tập và thực hiện khóa luận. Chúng em xin cảm ơn trường Sư phạm Kỹ thuật đã tạo nhiều điều kiện thuận lợi cho hoạt động phục vụ học tập của sinh viên chúng tôi, đặc biệt là thư viện số với nguồn tri thức vô tận.

Những giá trị cốt lõi nhà trường, Thầy Cô mang đến, chúng em sẽ luôn ghi nhớ để làm động lực thúc đẩy bản thân phát triển và hoàn thiện hơn nữa. Công việc nào chắc chắn cũng có khó khăn nhưng khổ luyện thành nhân, rõ ràng bản thân chúng em đã có những bước tiến cả về kiến thức, tư duy, kỹ năng, cách làm việc và mối quan hệ ứng xử với mọi người.

Trong phạm vi khả năng cho phép, chúng em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Song, chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em kính mong nhận được sự cảm thông và những ý kiến đóng góp của Quý Thầy Cô và các bạn.

**MỤC LỤC**

[1 Phần mở đầu 1](#_Toc140015340)

[1.1 Lý do chọn đề tài 1](#_Toc140015341)

[1.1.1 Tính cấp thiết đề tài 1](#_Toc140015342)

[1.2 Mục tiêu 2](#_Toc140015343)

[1.3 Cách tiếp cạnh 2](#_Toc140015344)

[1.4 Phương pháp nghiên cứu 3](#_Toc140015345)

[1.5 Phân tích những công trình liên quan 5](#_Toc140015346)

[1.6 Kết quả dự đoán 6](#_Toc140015347)

[2 Phần nội dung 6](#_Toc140015348)

[2.1 Tổng quan về Machine Learning 6](#_Toc140015349)

[2.1.1 Khái niệm 6](#_Toc140015350)

[2.1.2 Mối liên hệ 7](#_Toc140015351)

[2.1.3 Cách thức hoạt động 8](#_Toc140015352)

[2.1.4 Phương pháp 9](#_Toc140015353)

[2.1.5 Các thuật toán phổ biến 10](#_Toc140015354)

[2.1.6 Ứng dụng trong thực tế 12](#_Toc140015355)

[2.1.7 Quy trình 14](#_Toc140015356)

[2.2 Tổng quan về EDA 18](#_Toc140015357)

[2.2.1 Khái niệm: 18](#_Toc140015358)

[2.2.2 Vai trò: 18](#_Toc140015359)

[2.2.3 Phương pháp tiếp cận 18](#_Toc140015360)

[2.2.4 Kết luận 19](#_Toc140015361)

[2.3 Tổng quan về dữ liệu 19](#_Toc140015362)

[2.3.1 Giới thiệu bài toán 19](#_Toc140015363)

[2.3.2 Mô tả 20](#_Toc140015364)

[2.3.3 Đặc tả 21](#_Toc140015365)

[2.3.4 Cách đánh giá 22](#_Toc140015366)

[2.4 Tổng quan về thuật toán 25](#_Toc140015367)

[2.4.1 Tổng quan về Ensemble learning 25](#_Toc140015368)

[2.4.2 Bagging 26](#_Toc140015369)

[2.4.3 Bootstrapping 27](#_Toc140015370)

[2.4.4 Boosting 30](#_Toc140015371)

[2.4.5 Kỹ thuật 38](#_Toc140015372)

[2.5 Thực nghiệm thuật toán 39](#_Toc140015373)

[2.5.1 Xgboost 39](#_Toc140015374)

[2.5.2 CatBoost 49](#_Toc140015375)

[2.5.3 LightGBM 61](#_Toc140015376)

[2.5.4 Random Forest 67](#_Toc140015377)

[2.5.5 Điểm số và thứ hạng chung 72](#_Toc140015378)

[3 Phần kết luận 73](#_Toc140015379)

[3.1.1 Kết quả đạt được 73](#_Toc140015380)

[3.1.2 Ưu điểm 74](#_Toc140015381)

[3.1.3 Hạn chế 74](#_Toc140015382)

[3.1.4 Hướng phát triển 75](#_Toc140015383)

[3.1.5 Tài liệu tham khảo 76](#_Toc140015384)

**BẢNG HÌNH**

[Hình 1: Mối liên hệ Machine Learning 7](#_Toc139809346)

[Hình 2: Phương pháp Machine Learning 9](#_Toc139809347)

[Hình 3: Ứng dụng trong thực tế 12](#_Toc139809348)

[Hình 4: Quy trình Machine Learning 14](#_Toc139809349)

[Hình 5: Tổng quát Sơ đồ hoạt động của Bagging 26](#_Toc139809350)

[Hình 6: Sơ đồ hoạt động của Bootstrap Sampling 29](#_Toc139809351)

[Hình 7: Sơ đồ hoạt động của Adaptive Boosting 35](#_Toc139809352)

[Hình 8: Sơ đồ hoạt động của Gradient Boosting 37](#_Toc139809353)

[Hình 9: Train size of first piece 40](#_Toc139809354)

[Hình 10: Records of first piece 40](#_Toc139809355)

[Hình 11: Tập train\_labels sau khi đã xử lý 40](#_Toc139809356)

[Hình 12: Ý tưởng Feature Engineer 42](#_Toc139809357)

[Hình 13: Feature Engineer 42](#_Toc139809358)

[Hình 14: Feature Engineer in code 43](#_Toc139809359)

[Hình 15: Kích thước sau khi xử lý 44](#_Toc139809360)

[Hình 16: Áp dụng cho các phần còn lại của tập train 44](#_Toc139809361)

[Hình 17: Thuộc tính và users mà ta sẽ train 45](#_Toc139809362)

[Hình 18: Siêu tham số cho Xgboost 45](#_Toc139809363)

[Hình 19: Kết quả sau khi train mô hình 47](#_Toc139809364)

[Hình 20: Tìm Threshold tốt nhất 48](#_Toc139809365)

[Hình 21: Tương quan giữa Threshold và F1-Score tốt nhất 48](#_Toc139809366)

[Hình 22: Điểm F1-score cho từng câu hỏi và điểm tổng Xgboost 49](#_Toc139809367)

[Hình 23: Điểm số của XGBoost 49](#_Toc139809368)

[Hình 24: Xử lý data cho cột null 50](#_Toc139809369)

[Hình 25: Xử lý biến thời gian 50](#_Toc139809370)

[Hình 26: Tạo đặc trưng cho biến thời gian 51](#_Toc139809371)

[Hình 27: Lấy ra biến đặc trưng tốt nhất 53](#_Toc139809372)

[Hình 28: Xử lý biến câu hỏi 54](#_Toc139809373)

[Hình 29: Tạo tập dữ liệu mới dựa trên những đặc trưng mới 54](#_Toc139809374)

[Hình 30: Xử lý biến câu hỏi dựa trên tập dữ liệu mới 55](#_Toc139809375)

[Hình 31: Tinh chỉnh siêu tham số cho mô hình CatBoost 56](#_Toc139809376)

[Hình 32: Kết quả train câu hỏi từ 1 đến 3 57](#_Toc139809377)

[Hình 33: Kết quả train câu hỏi từ 4 đến 11 57](#_Toc139809378)

[Hình 34: Kết quả train câu hỏi từ 14 đến 17 57](#_Toc139809379)

[Hình 35: Kết quả đạt được 57](#_Toc139809380)

[Hình 36: Định nghĩa siêu tham số sử dụng trong thuật toán CatBoost 58](#_Toc139809381)

[Hình 37: Điểm số của bài dự thi CatBoost 60](#_Toc139809382)

[Hình 38: Feature Engineer 61](#_Toc139809383)

[Hình 39: Feature Engineer in code 62](#_Toc139809384)

[Hình 40: Kích thước sau khi xử lý 62](#_Toc139809385)

[Hình 41: Train model thuật toán LightGBM 63](#_Toc139809386)

[Hình 42: Tinh chỉnh siêu tham số LightGBM 63](#_Toc139809387)

[Hình 43: Tương quan giữa Threshold và F1-Score tốt nhất 65](#_Toc139809388)

[Hình 44: Điểm F1-score cho từng câu hỏi và điểm tổng LightGBM 66](#_Toc139809389)

[Hình 45: Điểm số LightGBM 67](#_Toc139809390)

[Hình 46: Feature Engineer 67](#_Toc139809391)

[Hình 47: Feature Engineer in code 68](#_Toc139809392)

[Hình 48: Kích thước sau khi xử lý 69](#_Toc139809393)

[Hình 49: Thuộc tính và users mà mô hình sẽ train 69](#_Toc139809394)

[Hình 50: Tương quan giữa Threshold và F1-Score tốt nhất 70](file:///C:\Users\phuoc\OneDrive\Desktop\KLTNNew%20(1).docx#_Toc139809395)

[Hình 51: Điểm F1-score cho từng câu hỏi và điểm tổng Random Forest 71](file:///C:\Users\phuoc\OneDrive\Desktop\KLTNNew%20(1).docx#_Toc139809396)

[Hình 52: Điểm số của Random Forest 72](#_Toc139809397)

[Hình 53: Điểm và thứ hạng ở Public Leaderboard 72](#_Toc139809398)

[Hình 54: Điểm và thứ hạng ở Private Leaderboard 73](#_Toc139809399)

**BẢNG BIỂU**

[Bảng 1: Mô tả tập dữ liệu 21](#_Toc139809400)

# Phần mở đầu

## Lý do chọn đề tài

### Tính cấp thiết đề tài

Đề tài "Predict Student Performance from Game Play" vươn lên với một tầm quan trọng tuyệt vời trong lĩnh vực khóa luận tốt nghiệp. Nó mang đến sự chất chứa sâu sắc cho lĩnh vực giáo dục, khám phá sự giao thoa tuyệt vời giữa trò chơi và hiệu suất học tập của học sinh. Với sự khai thác thông minh của trí tuệ nhân tạo, nghiên cứu này tạo nên sự mê hoặc trong việc tìm hiểu cách các yếu tố từ trò chơi gắn kết với thành tích học tập của học sinh.

Từ khóa là "cá nhân hóa" - một nguồn cảm hứng đầy tiềm năng. Phương pháp dự đoán hiệu suất học tập từ trò chơi mang đến khả năng khám phá một hành trình học tập tùy chỉnh cho từng cá nhân. Bằng cách xác định những điểm mấu chốt và mô hình học tập riêng, nghiên cứu này định hướng đến việc phát triển hệ thống hướng dẫn tối ưu, trợ giúp giáo viên và học sinh khám phá tiềm năng to lớn của trò chơi giáo dục và nâng cao kết quả học tập.

Ngoài ra, đề tài này còn đắm chìm trong tầm quan trọng của trí tuệ nhân tạo. Sự khả thi của việc phân tích dữ liệu từ trò chơi và dự đoán hiệu suất học tập mở ra một cánh cửa tuyệt vời đối với giáo dục. Không chỉ tạo ra mô hình dự đoán độc đáo cho từng cá nhân, mà còn phát hiện mẫu học tập và cung cấp phản hồi tức thì, tạo điều kiện thuận lợi để cải thiện chất lượng và hiệu quả của quá trình học tập.

Cuối cùng, đề tài này đặt nền móng cho sự phát triển công nghệ giáo dục. Việc kết hợp thông minh giữa giáo dục và công nghệ thông qua việc sử dụng dữ liệu từ trò chơi và kỹ thuật dự đoán không chỉ mang lại những cải tiến đáng kể, mà còn tạo ra những tiềm năng ứng dụng rộng rãi. Quá trình học tập không chỉ bị hạn chế trong giáo dục truyền thống, mà còn mở rộng đến các hệ thống học tập trực tuyến, trò chơi giáo dục và nền tảng giáo dục khác.

Trong tổng thể, đề tài "Predict Student Performance from Game Play" nổi bật với tính quan trọng to lớn, với sự kết hợp tuyệt vời giữa giáo dục và trí tuệ nhân tạo, và với sự tiềm năng tạo ra những bước tiến vượt bậc trong công nghệ giáo dục.

## Mục tiêu

Xây dựng một hệ thống dự đoán hiệu suất học tập của học sinh dựa trên dữ liệu từ trò chơi giáo dục. Đề tài tập trung nghiên cứu:

***Phân tích yếu tố ảnh hưởng***

* Nghiên cứu nhằm xác định các yếu tố từ trò chơi giáo dục có liên quan đến hiệu suất học tập của học sinh. Điều này bao gồm việc nghiên cứu các chỉ số, thông số và hoạt động trong trò chơi mà có thể ảnh hưởng tích cực hoặc tiêu cực đến quá trình học tập.

***Xây dựng mô hình dự đoán***

* Sử dụng dữ liệu thu thập được từ trò chơi, đề tài nhằm xây dựng một mô hình dự đoán hiệu suất học tập của học sinh. Mô hình này có thể áp dụng các kỹ thuật và phương pháp của trí tuệ nhân tạo, học máy, hay khai phá dữ liệu để dự đoán kết quả học tập một cách chính xác và đáng tin cậy.

***Đánh giá độ chính xác và hiệu quả***

* Mục tiêu này nhằm đánh giá độ chính xác và hiệu quả của mô hình dự đoán được xây dựng. Nghiên cứu sẽ thực hiện các thử nghiệm và kiểm tra mô hình trên dữ liệu thực tế để đánh giá khả năng dự đoán và tính ứng dụng của nó trong việc phân loại và đánh giá hiệu suất học tập của học sinh.

***Ứng dụng trong giáo dục và cải thiện quá trình học tập***

* Mục tiêu cuối cùng là ứng dụng kết quả nghiên cứu để cải thiện quá trình giảng dạy và hỗ trợ cá nhân hóa học tập. Các phân tích và dự đoán từ mô hình có thể cung cấp thông tin hữu ích cho giáo viên, nhà quản lý giáo dục và học sinh để tối ưu hóa quá trình học tập, cung cấp phản hồi và hướng dẫn cá nhân hóa. Mục tiêu này đóng góp vào sự cải thiện chất lượng giáo dục và đạt được kết quả học tập tốt hơn cho học sinh.

## Cách tiếp cạnh

Để tiếp cận đề tài thì ta có thể thực hiện các bước cụ thể sau:

***Thu thập dữ liệu***

* Đầu tiên, cần thu thập dữ liệu từ các trò chơi giáo dục hoặc nền tảng học tập trực tuyến. Dữ liệu này bao gồm thông tin về các hoạt động, thành tích, hành vi và các chỉ số liên quan đến quá trình học tập của học sinh trong trò chơi.

***Phân tích dữ liệu***

* Tiếp theo, sử dụng các phương pháp phân tích dữ liệu và khai phá dữ liệu để xác định các yếu tố ảnh hưởng đến hiệu suất học tập. Các phương pháp có thể bao gồm phân tích đa biến, khai thác quy tắc kết hợp, hoặc các kỹ thuật khác của khoa học dữ liệu.

***Xây dựng mô hình dự đoán***

* Dựa trên các yếu tố được xác định, tiến hành xây dựng một mô hình dự đoán hiệu suất học tập của học sinh từ dữ liệu trò chơi. Có thể áp dụng các kỹ thuật học máy như học tăng cường (reinforcement learning), mạng nơ-ron (neural networks), hoặc các phương pháp khác tùy thuộc vào đặc điểm của dữ liệu và mục tiêu dự đoán.

***Đánh giá mô hình***

* Tiến hành đánh giá độ chính xác và hiệu quả của mô hình dự đoán. Sử dụng các phương pháp đánh giá như cross-validation, đo lường chỉ số đánh giá như độ chính xác (accuracy), độ phủ (recall), độ chính xác dương tính (precision), và độ chính xác âm tính (negative precision) để đánh giá khả năng dự đoán và sự phù hợp của mô hình.

***Ứng dụng và cải thiện quá trình học tập***

* Cuối cùng, sử dụng kết quả từ mô hình dự đoán để cải thiện quá trình học tập và hỗ trợ cá nhân hóa. Cung cấp phản hồi cho học sinh về khả năng và điểm mạnh của họ, hướng dẫn giáo viên về cách tối ưu hóa quá trình giảng dạy và hỗ trợ cá nhân hóa học tập để nâng cao hiệu suất học tập và đạt được kết quả tốt hơn.

## Phương pháp nghiên cứu

Cụ thể, trong khuôn khổ luận văn, để tiếp cận được đề tài và phương pháp nghiên cứu, sẽ lần lượt nghiên cứu các nội dung sau:

***Xác định mục tiêu nghiên cứu***

* Đầu tiên, cần xác định rõ mục tiêu và câu hỏi nghiên cứu mà đề tài muốn trả lời. Ví dụ: "Có thể dự đoán hiệu suất học tập của học sinh dựa trên dữ liệu từ trò chơi giáo dục không?" hoặc "Các yếu tố nào trong trò chơi giáo dục ảnh hưởng đến hiệu suất học tập của học sinh?"

***Lựa chọn phương pháp nghiên cứu***

* Dựa trên mục tiêu và câu hỏi nghiên cứu, chọn phương pháp nghiên cứu phù hợp. Có thể áp dụng phương pháp phân tích dữ liệu số liệu, phương pháp thực nghiệm trên một nhóm học sinh, hoặc phương pháp kết hợp của cả hai.

***Thu thập và tiền xử lý dữ liệu***

* Thu thập dữ liệu từ các trò chơi giáo dục hoặc nền tảng học tập trực tuyến. Tiền xử lý dữ liệu bao gồm làm sạch, chuẩn hóa và biểu diễn dữ liệu dưới dạng phù hợp cho việc phân tích và xây dựng mô hình.

***Phân tích dữ liệu***

* Áp dụng các phương pháp phân tích dữ liệu để tìm hiểu các mô hình, quy luật và xu hướng trong dữ liệu. Các phương pháp có thể bao gồm phân tích thống kê, khai phá dữ liệu và trí tuệ nhân tạo.

***Xây dựng mô hình dự đoán***

* Sử dụng dữ liệu đã được tiền xử lý, xây dựng một mô hình dự đoán hiệu suất học tập của học sinh từ dữ liệu trò chơi. Áp dụng các kỹ thuật học máy, như mạng nơ-ron, học tăng cường, hoặc cây quyết định để xây dựng mô hình.

***Đánh giá mô hình***

* Đánh giá độ chính xác và hiệu quả của mô hình dự đoán sử dụng các phương pháp đánh giá như cross-validation, so sánh với mô hình khác hoặc sử dụng các chỉ số đánh giá như độ chính xác, độ phủ, và độ chính xác dương tính.

***Hiện thực và ứng dụng***

* Áp dụng kết quả từ mô hình để cải thiện quá trình học tập và hỗ trợ cá nhân hóa. Cung cấp phản hồi cho học sinh và giáo viên, tối ưu hóa quá trình giảng dạy và đạt được kết quả học tập tốt hơn.

## Phân tích những công trình liên quan

Với sự phát triển mạnh mẽ của hệ thống dự đoán và đặc biệt là Deep Learning đã mang lợi ích và giải quyết rất nhiều vấn đề trong cuộc sống, hơn nữa bất kỳ doanh nghiệp nào cũng có thể hưởng lợi từ một hệ thống dự đoán. Hệ thống dự đoán đang được áp dụng và khai thác triệt để trong các lĩnh vực như: giáo dục, trò chơi giáo dục, khoa học dữ liệu và khai phá, trí tuệ nhân tạo và học máy,…

***Giáo dục***

* Đề tài này đặt trong bối cảnh giáo dục và mục tiêu cải thiện hiệu suất học tập của học sinh. Các khía cạnh liên quan đến quá trình giảng dạy, phương pháp học tập, đánh giá và cá nhân hóa giáo dục đóng vai trò quan trọng trong nghiên cứu này.

***Trò chơi giáo dục***

* Nghiên cứu liên quan đến sử dụng trò chơi giáo dục như một công cụ để tăng cường quá trình học tập. Các yếu tố trong trò chơi như giao diện, nhiệm vụ, cấu trúc, độ khó và phản hồi đóng vai trò quan trọng trong việc ảnh hưởng đến hiệu suất học tập của học sinh.

***Khoa học dữ liệu và khai phá dữ liệu***

* Sử dụng phương pháp phân tích dữ liệu và khai phá dữ liệu để xác định mối quan hệ và mô hình hóa dữ liệu từ trò chơi giáo dục. Các kỹ thuật trong khoa học dữ liệu như phân tích thống kê, học máy và khai thác dữ liệu sẽ được áp dụng để tìm ra các mẫu và thông tin quan trọng từ dữ liệu.

***Trí tuệ nhân tạo và học máy***

* Đề tài sẽ sử dụng các phương pháp của trí tuệ nhân tạo và học máy để xây dựng mô hình dự đoán hiệu suất học tập từ dữ liệu trò chơi. Các thuật toán học máy, mạng nơ-ron, học tăng cường và phân loại dữ liệu sẽ được áp dụng để phân tích và dự đoán kết quả.

***Công nghệ giáo dục và học trực tuyến***

* Với sự phát triển của công nghệ, học trực tuyến và các nền tảng giáo dục trực tuyến đã trở thành xu hướng phổ biến. Đề tài này có liên quan đến sử dụng dữ liệu từ các nền tảng giáo dục trực tuyến và các công nghệ giáo dục để phân tích và dự đoán hiệu suất học tập.

Tổng quan, đề tài "Predict Student Performance from Game Play" kết hợp các lĩnh vực giáo dục, trò chơi giáo dục, khoa học dữ liệu, trí tuệ nhân tạo và công nghệ giáo dục để nghiên cứu về cách dự đoán hiệu suất học tập của học sinh dựa trên dữ liệu từ trò chơi giáo dục.

## Kết quả dự đoán

Dự đoán hiệu suất học tập: Kết quả nghiên cứu có thể là một mô hình hoặc hệ thống dự đoán hiệu suất học tập của học sinh dựa trên dữ liệu từ trò chơi giáo dục. Mô hình này có thể sử dụng thông tin về hoạt động, thành tích và hành vi của học sinh để đưa ra dự đoán về thành tích học tập.

Phân tích yếu tố ảnh hưởng: Kết quả nghiên cứu cung cấp cái nhìn sâu hơn về các yếu tố trong trò chơi giáo dục có ảnh hưởng đến hiệu suất học tập của học sinh. Các yếu tố như độ khó của nhiệm vụ, tương tác học sinh-giảng viên, phản hồi từ trò chơi, hoặc các chiến lược học tập có thể được xác định và đánh giá về mức độ ảnh hưởng của chúng.

Cải thiện quá trình học tập: Kết quả nghiên cứu có thể được áp dụng để cải thiện quá trình học tập và hỗ trợ cá nhân hóa. Dựa trên kết quả dự đoán, học sinh và giáo viên có thể nhận được phản hồi và hướng dẫn để tối ưu hóa quá trình học tập. Điều này có thể giúp học sinh nắm bắt được điểm mạnh, điểm yếu và phát triển chiến lược học tập phù hợp.

Nâng cao hiệu suất giảng dạy: Kết quả nghiên cứu cung cấp thông tin hữu ích cho giáo viên và người thiết kế trò chơi giáo dục để cải thiện quá trình giảng dạy. Đánh giá hiệu suất học tập dựa trên dữ liệu từ trò chơi giáo dục có thể giúp giáo viên hiểu rõ hơn về cách tối ưu hóa phương pháp giảng dạy, nâng cao tương tác với học sinh và tạo ra môi trường học tập tích cực.

# Phần nội dung

## Tổng quan về Machine Learning

### Khái niệm

Machine learning (ML) hay máy học là một nhánh của trí tuệ nhân tạo (AI), là một lĩnh vực nghiên cứu cho phép máy tính có khả năng cải thiện chính bản thân chúng dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học). Machine learning có thể tự dự đoán hoặc đưa ra quyết định mà không cần được lập trình cụ thể.

Là một thành phần quan trọng trong lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được đào tạo để phân loại hoặc dự đoán và khám phá những hiểu biết chính trong các dự án khai thác dữ liệu. Những thông tin chi tiết này sau đó sẽ thúc đẩy quá trình ra quyết định trong các ứng dụng và doanh nghiệp, tác động lớn đến các chỉ số tăng trưởng chính.

Khi dữ liệu lớn (big data) tiếp tục mở rộng và phát triển, nhu cầu thị trường đối với các nhà khoa học dữ liệu sẽ tăng lên. Họ sẽ được yêu cầu giúp xác định các câu hỏi kinh doanh phù hợp nhất và dữ liệu để trả lời chúng.

### Mối liên hệ

A picture containing text, screenshot, font, rectangle

Description automatically generated

Hình : Mối liên hệ Machine Learning

Deep learning (học sâu) và Machine learning (ML) có xu hướng được sử dụng thay thế cho nhau, nên cần lưu ý đến các sắc thái giữa hai loại này. Machine learning, deep learning và neural network (mạng Nơ Ron nhân tạo) đều là các lĩnh vực phụ của trí tuệ nhân tạo.

Tuy nhiên, Deep learning là một lĩnh vực phụ của neural networks. Và neural networks thực sự là một lĩnh vực phụ của machine learning.

Deep learning và machine learning khác nhau ở cách mỗi thuật toán học. “Deep” machine learning có thể sử dụng tập dữ liệu được gắn nhãn, còn được gọi là học có giám sát, để cung cấp thông tin cho thuật toán của nó nhưng không nhất thiết phải có tập dữ liệu được gắn nhãn. Còn Deep learning có thể nhập dữ liệu phi cấu trúc ở dạng thô (ví dụ: văn bản hoặc hình ảnh) và nó có thể tự động xác định tập hợp các tính năng phân biệt các loại dữ liệu khác nhau với nhau. Điều này giúp loại bỏ một số sự can thiệp bắt buộc của con người và cho phép sử dụng các tập dữ liệu lớn hơn

Classical, hoặc “non-deep”, machine learning cơ bản phụ thuộc nhiều hơn vào sự can thiệp của con người để học. Các chuyên gia về con người xác định tập hợp các tính năng để hiểu sự khác biệt giữa các đầu vào dữ liệu, thường yêu cầu dữ liệu có cấu trúc hơn để tìm hiểu.

Mạng nơ-ron hoặc mạng nơ-ron nhân tạo (ANN) bao gồm các lớp nút, chứa lớp đầu vào, một hoặc nhiều lớp ẩn và lớp đầu ra. Mỗi nút, hoặc nơ-ron nhân tạo, kết nối với nút khác, có trọng số và ngưỡng liên quan. Nếu đầu ra của bất kỳ nút riêng lẻ nào cao hơn giá trị ngưỡng đã chỉ định, nút đó sẽ được kích hoạt, gửi dữ liệu đến lớp tiếp theo của mạng. Mặt khác, nếu thấp hơn thì sẽ không có dữ liệu nào được chuyển đến lớp tiếp theo của mạng bởi nút đó. “Deep” in deep learning chỉ đề cập đến số lớp trong neural network (mạng nơ-ron). Một mạng nơ-ron bao gồm nhiều hơn ba lớp (bao gồm đầu vào và đầu ra) có thể được coi là một thuật toán deep learning hoặc deep neural network. Một neural network có ba lớp chỉ là một neural network cơ bản.

Deep learning và neural network được cho là đã đẩy nhanh tiến độ trong các lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên và nhận dạng giọng nói.

### Cách thức hoạt động

Hệ thống học của thuật toán học máy gồm thành ba phần chính:

***Quy trình quyết định***

* Các thuật toán máy học được sử dụng để đưa ra dự đoán hoặc phân loại. Dựa trên một số dữ liệu đầu vào, có thể được gắn nhãn hoặc không gắn nhãn, thuật toán sẽ đưa ra ước tính về một mẫu trong dữ liệu.

***Hàm lỗi***

* Hàm lỗi đánh giá dự đoán của mô hình. Nếu có các ví dụ đã biết, một hàm lỗi có thể so sánh để đánh giá độ chính xác của mô hình.

***Quy trình tối ưu hóa mô hình***

* Nếu mô hình có thể phù hợp hơn với các điểm dữ liệu trong tập huấn luyện, thì các trọng số sẽ được điều chỉnh để giảm sự khác biệt giữa ví dụ đã biết và ước tính mô hình. Thuật toán sẽ lặp lại quy trình “đánh giá và tối ưu hóa” này, cập nhật trọng số một cách tự động cho đến khi đạt đến ngưỡng chính xác.

### Phương pháp

**A picture containing text, cartoon, circle

Description automatically generated**

Hình : Phương pháp Machine Learning

Mô hình machine learning cơ bản được chia thành các loại:

***Supervised machine learning***

Supervised learning, còn được gọi là Supervised machine learning (học máy có giám sát), được xác định bằng cách sử dụng các bộ dữ liệu được gắn nhãn để huấn luyện các thuật toán nhằm phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác.

Khi dữ liệu đầu vào được đưa vào mô hình, mô hình sẽ điều chỉnh trọng số của nó cho đến khi nó được điều chỉnh phù hợp. Điều này xảy ra như một phần của quy trình xác thực chéo để đảm bảo rằng mô hình tránh trang bị thừa hoặc thiếu. Learning có giám sát giúp các tổ chức giải quyết nhiều vấn đề trong thế giới thực ở quy mô lớn, chẳng hạn như phân loại thư rác trong một thư mục riêng biệt với hộp thư đến của bạn. Một số phương pháp được sử dụng trong học có giám sát bao gồm neural networks, naive bayes, hồi quy tuyến tính, hồi quy logistic, random forest và máy vector hỗ trợ (SVM).

***Unsupervised machine learning***

Unsupervised learning còn được gọi là Unsupervised machine learning (học máy không giám sát), sử dụng các thuật toán machine learning để phân tích và phân cụm các bộ dữ liệu không được gắn nhãn. Các thuật toán này khám phá các mẫu hoặc nhóm dữ liệu ẩn mà không cần sự can thiệp của con người.

Khả năng khám phá những điểm tương đồng và khác biệt trong thông tin của phương pháp này khiến nó trở nên lý tưởng cho việc phân tích dữ liệu khám phá, chiến lược cross-selling, phân khúc khách hàng cũng như nhận dạng hình ảnh và mẫu. Nó cũng được sử dụng để giảm số lượng các tính năng trong một mô hình thông qua quá trình giảm kích thước. Phân tích thành phần chính (PCA) và phân tích giá trị đơn lẻ (SVD) là hai cách tiếp cận phổ biến cho việc này. Các thuật toán khác được sử dụng trong học tập không giám sát bao gồm neural networks, phương pháp phân cụm k-means và phương pháp phân cụm xác suất.

***Semi-supervised learning***

Semi-supervised learning (học máy bán giám sát) cung cấp một phương tiện vui vẻ giữa học có giám sát và không giám sát. Trong quá trình đào tạo, nó sử dụng tập dữ liệu được gắn nhãn nhỏ hơn để hướng dẫn phân loại và trích xuất tính năng từ tập dữ liệu lớn hơn, không được gắn nhãn.

Học máy bán giám sát có thể giải quyết vấn đề không có đủ dữ liệu được gán nhãn cho thuật toán học có giám sát. Nó cũng hữu ích nếu việc dán nhãn đủ dữ liệu quá tốn kém.

***Reinforcement machine learning***

Reinforcement machine learning (Machine learning tăng cường) là một mô hình machine learning tương tự như learning có giám sát, nhưng thuật toán không được đào tạo bằng dữ liệu mẫu. Mô hình này học hỏi bằng cách sử dụng thử và sai. Một chuỗi các kết quả thành công sẽ được củng cố để phát triển khuyến nghị hoặc chính sách tốt nhất cho một vấn đề nhất định.

### Các thuật toán phổ biến

***Neural networks***

Mô phỏng cách thức hoạt động của bộ não con người, với một số lượng lớn các nút xử lý được liên kết. Mạng nơ-ron rất tốt trong việc nhận dạng các mẫu và đóng vai trò quan trọng trong các ứng dụng bao gồm dịch ngôn ngữ tự nhiên, nhận dạng hình ảnh, nhận dạng giọng nói và tạo hình ảnh.

***Hồi quy tuyến tính***

Thuật toán này được sử dụng để dự đoán các giá trị số, dựa trên mối quan hệ tuyến tính giữa các giá trị khác nhau. Ví dụ, kỹ thuật này có thể được sử dụng để dự đoán giá nhà dựa trên dữ liệu lịch sử của khu vực.

***Hồi quy logistic***

Thuật toán học có giám sát này đưa ra dự đoán cho các biến phản hồi phân loại, chẳng hạn như câu trả lời “có/không” cho các câu hỏi. Nó có thể được sử dụng cho các ứng dụng như phân loại thư rác và kiểm soát chất lượng trên dây chuyền sản xuất.

***Phân cụm***

Sử dụng phương pháp học không giám sát, các thuật toán phân cụm có thể xác định các mẫu trong dữ liệu để có thể nhóm lại. Máy tính có thể giúp các nhà khoa học dữ liệu bằng cách xác định sự khác biệt giữa các mục dữ liệu mà con người đã bỏ qua.

***Decision trees(cây quyết định)***

Có thể được sử dụng cho cả dự đoán giá trị số (hồi quy) và phân loại dữ liệu thành các danh mục. Cây quyết định sử dụng một chuỗi phân nhánh của các quyết định được liên kết có thể được biểu diễn bằng sơ đồ cây. Một trong những ưu điểm của cây quyết định là chúng dễ dàng xác thực và kiểm toán, không giống như hộp đen của neural networks.

***Random forests (rừng ngẫu nhiên)***

Trong một random forests, thuật toán machine learning dự đoán một giá trị hoặc danh mục bằng cách kết hợp các kết quả từ một số cây quyết định.

Bên cạnh đó, cũng cần quan tâm đến machine learning workflow (quy trình làm việc với Machine learning) để hoàn thành công việc nhanh chóng, đạt hiệu quả cao, hạn chế sai sót, tiết kiệm thời gian và nâng cao chất lượng toàn diện.

### Ứng dụng trong thực tế

**A picture containing text, font, screenshot, graphics

Description automatically generated**

Hình : Ứng dụng trong thực tế

***Nhận dạng giọng nói***

Nhận dạng giọng nói có 2 thuật ngữ là Voice recognition và Speech recognition

* Speech recognition tập trung vào việc nhận dạng từ ngữ và chuyển các từ ngữ này sang dạng văn bản. Đây là khả năng sử dụng quá trình xử lý ngôn ngữ tự nhiên (NLP) để dịch lời nói của con người sang định dạng viết.
* Voice recognition có khả năng nhận dạng và định danh giọng nói của từng người dùng, nhờ sự ra đời của Deep Learning.

Nhận diện giọng nói được ứng dụng trong:

* Tối ưu việc nhập liệu: tiết kiệm được thời gian nhập văn bản
* Điều khiển nhà thông minh (smarthome): chỉ bằng giọng nói có thể tắt/bật điện, tắt/mở điều hoà hoặc kéo/đóng rèm cửa)
* Lập trình trợ lý ảo: nhiều thiết bị di động kết hợp nhận dạng giọng nói vào hệ thống của họ, để tiến hành tìm kiếm bằng giọng nói, ví dụ: trợ lý ảo Siri của Apple, Google Assistant – trợ lý ảo của Google, Alexa của Amazon

***Dịch vụ khách hàng (Customer service)***

Các chatbot trực tuyến đang thay thế các đại lý con người trong suốt hành trình của khách hàng, thay đổi cách chúng ta nghĩ về sự tương tác của khách hàng trên các trang web và nền tảng truyền thông xã hội. Chatbot trả lời các câu hỏi thường gặp (FAQ) về các chủ đề như vận chuyển hoặc cung cấp lời khuyên được cá nhân hóa, bán chéo sản phẩm hoặc đề xuất kích thước cho người dùng.

Ví dụ như đại lý ảo trên các trang web thương mại điện tử; bot nhắn tin, sử dụng Slack và Facebook Messenger; và các tác vụ thường được thực hiện bởi trợ lý ảo và trợ lý giọng nói.

***Thị giác máy tính (Computer vision)***

Công nghệ AI này cho phép máy tính lấy thông tin có ý nghĩa từ hình ảnh kỹ thuật số, video và các đầu vào trực quan khác, sau đó thực hiện hành động thích hợp.

Được hỗ trợ bởi convolutional neural networks, thị giác máy tính có các ứng dụng trong việc:

* Gắn thẻ ảnh trên phương tiện truyền thông xã hội như là Facebook
* Chụp ảnh X quang trong chăm sóc sức khỏe và
* Ô tô tự lái trong ngành công nghiệp ô tô

***Công cụ đề xuất (Recommendation engines)***

Sử dụng dữ liệu hành vi tiêu dùng trong quá khứ, thuật toán AI giúp khám phá các xu hướng dữ liệu có thể được sử dụng để phát triển các chiến lược bán kèm hiệu quả hơn. Phương pháp này được các nhà bán lẻ trực tuyến sử dụng để đưa ra các đề xuất sản phẩm phù hợp cho khách hàng trong quá trình thanh toán.

***Giao dịch chứng khoán tự động (Automated stock trading)***

Được thiết kế để tối ưu hóa danh mục đầu tư chứng khoán, các nền tảng giao dịch tần suất cao do AI điều khiển thực hiện hàng nghìn, thậm chí hàng triệu giao dịch mỗi ngày mà không cần sự can thiệp của con người.

***Phát hiện gian lận (Fraud detection)***

Các ngân hàng và tổ chức tài chính khác đã sử dụng machine learning để phát hiện các giao dịch đáng ngờ. Learning có giám sát có thể đào tạo một mô hình bằng cách sử dụng thông tin về các giao dịch gian lận đã biết. Tính năng phát hiện bất thường có thể xác định các giao dịch có vẻ không điển hình và cần được điều tra thêm.

### Quy trình

**A diagram of a machine learning workflow

Description automatically generated with low confidence**

Hình : Quy trình Machine Learning

***Gathering machine learning data (Thu thập dữ liệu)***

Quá trình này phụ thuộc vào dự án và loại dữ liệu bạn cần. Có thể là dữ liệu thời gian thực hoặc là dữ liệu tĩnh từ cơ sở dữ liệu hiện có.

Lập trình viên cần cung cấp bộ dữ liệu (dataset) cho máy để máy tính có thể học, phân tính và đưa ra phán đoán phục vụ cho mục đích sử dụng.

Chú ý lựa chọn bộ dữ liệu cần có tính xác thực, từ các nguồn chính thống thì máy tính mới có thể học một cách chính xác, đưa ra được kết quả đúng và tăng tính hiệu quả cho dự án.

***Data pre-processing (Tiền xử lý dữ liệu)***

Bước này giúp bạn chuẩn hoá bộ dữ liệu đã thu thập được. Sau đó, loại bỏ các thuộc tính không cần thiết, loại bỏ các yếu tố thiếu hoặc nhiễu. Đồng thời, thực hiện gán nhãn dữ liệu, mã hóa một số đặc trưng, trích xuất các đặc trưng, rút gọn dữ liệu mà vẫn đảm bảo kết quả đầu ra.

Đây là bước tốn nhiều thời gian nhất, bước 1 và bước 2 chiếm đến khoảng 70% thời gian của cả quy trình. Do đó, bước này rất quan trọng vì nó quyết định đến sự thành công và tính hiệu quả của cả quy trình.

***Training model (Huấn luyện mô hình)***

Ở bước này, máy tính sẽ học và tiến hành xử lý từ dữ liệu đã cung cấp ở 2 bước trên, bằng cách kết nối dataset với một thuật toán. Tiếp đến, thuật toán sẽ tận dụng mô hình toán học phức tạp để tìm hiểu và phát triển các dự đoán.

Thuật toán được sử dụng sẽ rơi vào 3 loại:

* Nhị phân
* Phân loại
* Hồi quy

***Evaluating model (đánh giá mô hình)***

Bước này sẽ đánh giá, kiểm tra độ chính xác của mô hình vừa tạo ra. Quá trình này sẽ dựa vào từng loại độ đo khác nhau để đánh giá độ tốt xấu nên sẽ không có quy chuẩn cụ thể. Nhưng về cơ bản, độ chính xác của mô hình vừa huấn luyện đạt trên 80% thì được coi là có hiệu quả.

***Improve (cải thiện)***

Nếu kết quả từ bước đánh giá mô hình không khả quan thì machine learning phải được huấn luyện lại. Ta sẽ lặp lại bước 3, bước 4 cho đến khi đạt được độ chính xác như kỳ vọng.

***Thách thức***

Khi công nghệ machine learning phát triển, chắc chắn nó đã làm cho cuộc sống của chúng ta dễ dàng hơn. Tuy nhiên, việc triển khai machine learning trong các doanh nghiệp cũng làm dấy lên một số lo ngại về đạo đức đối với công nghệ AI.

***Điểm kỳ dị công nghệ (Technological singularity)***

Trong khi chủ đề này thu hút rất nhiều sự chú ý của công chúng, nhiều nhà nghiên cứu không quan tâm đến ý tưởng AI sẽ vượt qua trí thông minh của con người trong tương lai gần.

Điểm kỳ dị công nghệ còn được gọi là trí tuệ nhân tạo mạnh mẽ hoặc siêu trí tuệ. Triết gia Nick Bostrum định nghĩa siêu trí tuệ là “bất kỳ trí tuệ nào vượt trội hơn rất nhiều so với bộ não tốt nhất của con người trong thực tế mọi lĩnh vực, bao gồm khả năng sáng tạo khoa học, trí tuệ chung và kỹ năng xã hội”.

Mặc dù thực tế là trí tuệ siêu việt chưa xuất hiện trong xã hội, nhưng ý tưởng về nó đặt ra một số câu hỏi thú vị khi chúng ta xem xét việc sử dụng các hệ thống tự trị, chẳng hạn như ô tô tự lái.

Thật không thực tế khi nghĩ rằng một chiếc xe không người lái sẽ không bao giờ gặp tai nạn, nhưng ai chịu trách nhiệm và chịu trách nhiệm pháp lý trong những trường hợp đó? Chúng ta có nên tiếp tục phát triển các phương tiện tự trị hay chỉ giới hạn công nghệ này ở các phương tiện bán tự trị giúp mọi người lái xe an toàn? Vẫn chưa thể có câu trả lời chính xác vấn đề này, nhưng nó gây tranh luận về đạo đức và trách nhiệm xã hội khi mà công nghệ AI đang phát triển, đổi mới không ngừng.

***AI tác động đến việc làm (AI impact on jobs)***

Trong khi chúng ta đang lo ngại rằng trí tuệ nhân tạo sẽ gây ra tình trạng mất việc làm, thì có lẽ nên điều chỉnh lại suy nghĩ này.

Chỉ những công việc có tính lặp đi lặp lại cao hoặc dựa trên các hướng dẫn, quy định cụ thể sẽ là nhưng vị trí gặp nhiều rủi ro. Còn những công việc thường xuyên thay đổi, không ngừng cập nhập, đổi mới, yêu cầu sự linh hoạt và khả năng thích ứng cao thì sẽ rất khó bị thay thế.

Với mỗi công nghệ mới, mang tính đột phá, chúng ta thấy rằng nhu cầu thị trường đối với các vai trò công việc cụ thể sẽ thay đổi. Ví dụ, khi chúng ta xem xét ngành công nghiệp ô tô, nhiều nhà sản xuất, như GM, đang chuyển sang tập trung vào sản xuất xe điện để phù hợp với các sáng kiến xanh. Ngành năng lượng sẽ không biến mất, nhưng nguồn năng lượng đang chuyển từ tiết kiệm nhiên liệu sang sử dụng điện.

Theo cách tương tự, trí tuệ nhân tạo sẽ chuyển nhu cầu việc làm sang các lĩnh vực khác. Sẽ cần có những cá nhân giúp quản lý hệ thống AI. Sẽ vẫn cần có người giải quyết các vấn đề phức tạp hơn trong các ngành có nhiều khả năng bị ảnh hưởng bởi sự thay đổi nhu cầu việc làm, chẳng hạn như dịch vụ khách hàng. Thách thức lớn nhất với trí tuệ nhân tạo và ảnh hưởng của nó đối với thị trường việc làm sẽ là giúp mọi người chuyển đổi sang những vai trò mới đang được yêu cầu.

***Sự riêng tư (Privacy)***

Quyền riêng tư đề cập đến ở đây là quyền riêng tư dữ liệu, bảo vệ dữ liệu và bảo mật dữ liệu. Những mối quan tâm này đã cho phép các nhà hoạch định chính sách đạt được nhiều bước tiến hơn trong những năm gần đây.

Ví dụ: Vào năm 2016, luật GDPR đã được tạo ra để bảo vệ dữ liệu cá nhân của những người ở Liên minh Châu Âu và Khu vực Kinh tế Châu Âu, giúp các cá nhân có nhiều quyền kiểm soát hơn đối với dữ liệu của họ.

Tại Hoa Kỳ, các tiểu bang riêng lẻ đang xây dựng các chính sách, chẳng hạn như Đạo luật về quyền riêng tư của người tiêu dùng California (CCPA), được ban hành vào năm 2018 và yêu cầu các doanh nghiệp thông báo cho người tiêu dùng về việc thu thập dữ liệu của họ.

Những luật như thế này đã buộc các công ty phải suy nghĩ lại về cách họ lưu trữ và sử dụng thông tin nhận dạng cá nhân (PII). Do đó, các khoản đầu tư vào bảo mật ngày càng trở thành ưu tiên hàng đầu của các doanh nghiệp khi họ tìm cách loại bỏ mọi lỗ hổng và cơ hội để giám sát, hack và tấn công mạng.

***Thiên vị và phân biệt đối xử (Bias and discrimination)***

Các trường hợp thiên vị và phân biệt đối xử trong một số hệ thống máy học đã đặt ra nhiều câu hỏi về đạo đức liên quan đến việc sử dụng trí tuệ nhân tạo. Làm cách nào chúng ta có thể bảo vệ chống lại sự thiên vị và phân biệt đối xử khi chính dữ liệu đào tạo có thể được tạo ra bởi các quy trình thiên vị của con người?

Mặc dù các công ty thường có ý định tốt cho các nỗ lực tự động hóa của họ, nhưng Reuters (liên kết nằm bên ngoài IBM)) nêu bật một số hậu quả không lường trước được của việc kết hợp AI vào các hoạt động tuyển dụng. Trong nỗ lực tự động hóa và đơn giản hóa một quy trình, Amazon đã vô tình phân biệt đối xử với các ứng viên theo giới tính đối với các vị trí kỹ thuật và cuối cùng công ty đã phải hủy bỏ dự án. Harvard Business Review (liên kết nằm bên ngoài IBM) đã đặt ra những câu hỏi quan trọng khác về việc sử dụng AI trong thực tiễn tuyển dụng, chẳng hạn như dữ liệu nào bạn có thể sử dụng khi đánh giá ứng viên cho một vai trò.

Sự thiên vị và phân biệt đối xử cũng không giới hạn trong chức năng nguồn nhân lực; chúng có thể được tìm thấy trong một số ứng dụng từ phần mềm nhận dạng khuôn mặt đến các thuật toán truyền thông xã hội.

Khi các doanh nghiệp nhận thức rõ hơn về những rủi ro với AI, họ cũng trở nên tích cực hơn trong cuộc thảo luận này về các giá trị và đạo đức của AI. Ví dụ, IBM đã ngừng sản xuất các sản phẩm phân tích và nhận dạng khuôn mặt cho mục đích chung. Giám đốc điều hành IBM Arvind Krishna đã viết: “IBM kiên quyết phản đối và sẽ không tha thứ cho việc sử dụng bất kỳ công nghệ nào, kể cả công nghệ nhận dạng khuôn mặt do các nhà cung cấp khác cung cấp, để giám sát hàng loạt, lập hồ sơ chủng tộc, vi phạm các quyền và tự do cơ bản của con người hoặc bất kỳ mục đích nào không nhất quán. với các giá trị và Nguyên tắc Tin cậy và Minh bạch của chúng ta.”

***Trách nhiệm giải trình (Accountability)***

Vì không có luật quan trọng để điều chỉnh các hoạt động của AI, nên không có cơ chế thực thi nào thực sự để đảm bảo rằng AI có đạo đức được thực hành. Các khuyến khích hiện tại để các công ty trở nên có đạo đức là những hậu quả tiêu cực của một hệ thống AI phi đạo đức ở điểm mấu chốt. Để lấp đầy khoảng trống, các khuôn khổ đạo đức đã xuất hiện như một phần của sự hợp tác giữa các nhà đạo đức và nhà nghiên cứu để quản lý việc xây dựng và phân phối các mô hình AI trong xã hội.

Tuy nhiên, tại thời điểm này, những điều này chỉ phục vụ để hướng dẫn. Một số nghiên cứu cho thấy rằng sự kết hợp giữa trách nhiệm phân tán và việc thiếu tầm nhìn xa đối với các hậu quả có thể xảy ra không có lợi cho việc ngăn chặn tác hại đối với xã hội.

## Tổng quan về EDA

### Khái niệm:

EDA là viết tắt của "Exploratory Data Analysis" (Phân tích dữ liệu khám phá) và là một kỹ thuật trong khoa học dữ liệu được sử dụng để khám phá dữ liệu một cách đầy đủ trước khi bắt đầu xây dựng mô hình hoặc thực hiện các phân tích chi tiết.

Trong EDA, người phân tích sử dụng nhiều phương pháp trực quan và thống kê để khám phá một số tính chất chính của dữ liệu, bao gồm phân bố, tương quan, giá trị ngoại lai, và các đặc tính khác của dữ liệu. EDA giúp tạo ra một cái nhìn tổng quan về dữ liệu, cung cấp các thông tin quan trọng cho việc lựa chọn và xây dựng mô hình dữ liệu phù hợp và các phân tích thống kê phù hợp.

### Vai trò:

***Khám phá dữ liệu***

* EDA giúp nhà phân tích dữ liệu hiểu rõ hơn về dữ liệu của mình, bao gồm cách dữ liệu được thu thập, đặc tính của dữ liệu, các giá trị ngoại lai, độ phân bố của dữ liệu và tương quan giữa các biến.

***Phát hiện giá trị ngoại lai***

* EDA giúp phát hiện các giá trị ngoại lai và dữ liệu bất thường trong tập dữ liệu của bạn. Điều này rất quan trọng trong việc xác định liệu các giá trị này có ảnh hưởng đến mô hình của bạn hay không.

***Lựa chọn và xây dựng mô hình***

* EDA cung cấp thông tin để lựa chọn mô hình phù hợp nhất với dữ liệu của bạn. Nó cũng giúp định hình chiến lược để xây dựng mô hình.

***Tối ưu hóa quy trình làm việc***

* EDA cung cấp thông tin để tối ưu hóa các quy trình làm việc, như cách xử lý dữ liệu, cách chọn biến đầu vào, cách tiền xử lý dữ liệu, cách chọn mô hình, và cách đánh giá mô hình.

***Trình bày kết quả***

* EDA cung cấp cho nhà phân tích dữ liệu các công cụ để trình bày kết quả một cách đầy đủ và trực quan, giúp đưa ra những kết luận hợp lý từ dữ liệu.

### Phương pháp tiếp cận

***Phân tích mô tả***

* Phân tích mô tả là một phương pháp đơn giản và trực quan để khám phá dữ liệu bằng cách sử dụng các biểu đồ, biểu đồ tần số và các đại lượng thống kê mô tả như trung bình, độ lệch chuẩn, trung vị và phạm vi.

***Biểu đồ***

* Biểu đồ là một công cụ quan trọng để trực quan hóa dữ liệu trong EDA. Các loại biểu đồ thông dụng bao gồm biểu đồ đường, biểu đồ cột, biểu đồ hộp, biểu đồ tần số, biểu đồ phân tán, biểu đồ đường cong.

***Độ tương quan***

* Độ tương quan đo lường mức độ tương quan giữa hai biến. Các loại độ tương quan thông dụng bao gồm hệ số tương quan Pearson và Spearman.

***Phát hiện giá trị ngoại lai***

* Giá trị ngoại lai là các giá trị bất thường hoặc sai lệch so với các giá trị khác trong tập dữ liệu. Phương pháp thường được sử dụng để phát hiện giá trị ngoại lai bao gồm biểu đồ hộp và phát hiện giá trị ngoại lai bằng z-score.

***Chuẩn hóa dữ liệu***

* Chuẩn hóa dữ liệu là quá trình chuyển đổi dữ liệu thành các giá trị có cùng đơn vị và phạm vi để có thể so sánh và phân tích dữ liệu một cách hiệu quả. Phương pháp chuẩn hóa dữ liệu thông dụng bao gồm chuẩn hóa Min-Max, chuẩn hóa z-score và chuẩn hóa lề.

***Kiểm định phân phối***

* Kiểm định phân phối được sử dụng để kiểm tra xem dữ liệu có tuân theo phân phối chuẩn hay không. Các kiểm định phân phối thông dụng bao gồm kiểm định Shapiro-Wilk và kiểm định Kolmogorov-Smirnov.

### Kết luận

EDA là bước rất quan trọng và không thể thiếu trong quá trình xử lý và phân tích dữ liệu. Bằng cách sử dụng nhiều công cụ và phương pháp khác nhau để khám phá giúp người phân tích có cái nhìn tổng quan về dữ liệu tạo tiền đề để dự đoán kết quả về bài toán.

## Tổng quan về dữ liệu

### Giới thiệu bài toán

Nhóm tham gia cuộc thi "Predict Student Performance from Game Play" được diễn ra trên [kaggle](https://www.kaggle.com). Khoảng thời gian diễn ra cuộc thi từ 26/02/2023 - 28/06/2023, trùng với khoảng thời gian làm khóa luận của nhóm vào Học kỳ II năm học 2022 - 2023.

Mục tiêu của cuộc thi hướng tới là dự đoán hiệu suất của học sinh trong quá trình học tập dựa trên trò chơi trong thời gian thực. Nhóm sẽ phát triển một mô hình được đào tạo trên một bộ dữ liệu mở lớn về nhật ký trò chơi để đánh giá hiệu suất.

### Mô tả

Tập dữ liệu được sử dụng để dự đoán kết quả các câu hỏi trong các phiên của mỗi học sinh dựa trên thông tin về các sự kiện trong mỗi phiên. Các học sinh chơi trò chơi trực tuyến và thực hiện các hoạt động như click chuột, hover, trả lời câu hỏi và xem thông tin. Mục tiêu cuộc thi là phải dự đoán câu trả lời đúng của các câu hỏi dựa trên thông tin về các hoạt động được cung cấp, để có thể giúp các giáo viên và nhà nghiên cứu đánh giá và cải thiện hiệu quả giảng dạy.

Mọi người tham gia cuộc thi đều được truy cập dữ liệu đào tạo và nhãn. Có 18 câu hỏi cho mỗi phiên. Người tham gia không được cung cấp câu trả lời mà chỉ được cho biết liệu người dùng trong một phiên cụ thể có trả lời đúng từng câu hỏi hay không. Nhiệm vụ của người tham gia là đưa ra dự đoán cho từng nhóm câu hỏi.

Người tham gia sẽ dựa trên mẫu file nộp (***Sample\_submission.csv***) để dự đoán các hãng nhãn <session\_id>\_<question #> với mỗi session có 18 dòng tương ứng với 18 câu hỏi. Đối với mỗi <session\_id>\_<question #>, người tham gia sẽ phải điền dữ liệu dự đoán vào cột correct, xác định xem người dùng trong phiên cụ thể này sẽ trả lời chính xác câu hỏi này hay không, chỉ sử dụng thông tin trước đó cho phiên. API chuỗi thời gian trình bày các câu hỏi và dữ liệu cho mọi người theo thứ tự cấp độ, mỗi phân đoạn cấp độ 0-4, 5-12 và 13-22 được cung cấp theo trình tự và mọi người sẽ dự đoán tính chính xác của các câu hỏi của từng phân đoạn khi chúng được trình bày.

Tập dữ liệu của cuộc thi "Predict Student Performance from Game Play" bao gồm các file sau

* ***train.csv:*** tập dữ liệu huấn luyện, chứa thông tin về các sự kiện trong các phiên chơi của các học sinh.
* ***test.csv:*** tập dữ liệu kiểm tra,  chứa thông tin về các sự kiện trong các phiên chơi của các học sinh.
* ***Sample\_submission.csv:*** là file mẫu cho kết quả dự đoán. Yêu cầu của cuộc thi là phải dự đoán câu trả lời đúng của các câu hỏi trong các phiên chơi của các học sinh trong tập test.csv.
* ***Train\_labels.csv:*** Chứa thông tin về kết quả đúng của 18 câu hỏi trong các phiên chơi của các học sinh. Mỗi dòng trong file này tương ứng với một phiên chơi và có thông tin về session\_id và các trường cho biết câu trả lời đúng của các câu hỏi trong phiên chơi đó.

### Đặc tả

Bảng : Mô tả tập dữ liệu

|  |  |
| --- | --- |
| Tên cột | Mô tả |
| Session\_ID | ID của phiên mà sự kiện diễn ra trong đó. |
| Index | Chỉ mục của sự kiện cho phiên đó. |
| Elapsed\_time | thời gian đã trôi qua (tính bằng mili giây) giữa thời điểm bắt đầu phiên và khi sự kiện được ghi lại. |
| Event\_name | Tên của loại sự kiện. |
| Name | Tên sự kiện |
| Level | Cấp độ của trò chơi mà sự kiện diễn ra trong đó (từ 0 đến 22) |
| Page | Số trang của sự kiện (chỉ liên quan đến sự kiện NoteBook). |
| Room\_coor\_x | Tọa độ của nhấp chuột liên quan đến phòng trong trò chơi (chỉ cho các sự kiện nhấp chuột) |
| Room\_coor\_y | Tọa độ của nhấp chuột liên quan đến phòng trong trò chơi (chỉ cho các sự kiện nhấp chuột) |
| Screen\_coor\_x | Tọa độ của nhấp chuột liên quan đến màn hình của người chơi (chỉ cho các sự kiện nhấp chuột) |
| Screen\_coor\_y | Tọa độ của nhấp chuột liên quan đến màn hình của người chơi (chỉ cho các sự kiện nhấp chuột) |
| Hover\_duration | Thời gian được tính bằng (ms) khi bắt đầu giữ chuột đến khi dừng lại (chỉ cho các sự kiện "hover") |
| Text | Văn bản người chơi thấy trong sự kiện này |
| fqid | ID đầy đủ của sự kiện |
| Room\_fqid | ID đầy đủ của phòng mà sự kiện diễn ra trong đó |
| Text\_fqid | ID đầy đủ của văn bản trong sự kiện này |
| Fullscreen | Liệu người chơi có đang ở chế độ toàn màn hình hay không? |
| Hq | Liệu trò chơi có ở chất lượng cao hay không? |
| Music | Liệu âm nhạc trong trò chơi có bật hay không? |
| Level\_group | Nhóm cấp độ của câu hỏi. |

### Cách đánh giá

#### Phương pháp đánh giá

##### Định nghĩa

Phương pháp đánh giá được sử dụng là Group K-fold cross-validation. Đây là một phương pháp chia dữ liệu thành các fold (phân nhóm) sao cho các mẫu trong cùng một nhóm (group) không bị chia ra khỏi nhóm đó.

##### Quá trình

Quá trình đánh giá hiệu suất mô hình được thực hiện như sau:

***Chia dữ liệu thành các fold***

Dữ liệu được chia thành các fold bằng phương pháp Group K-fold cross-validation, với số lượng fold là 5. Các fold được xác định bằng cách phân nhóm dựa trên chỉ số của dữ liệu (index) và nhãn (group).

***Lặp qua các fold***

Quá trình huấn luyện và đánh giá mô hình được thực hiện cho mỗi fold.

***Huấn luyện mô hình***

Mô hình XGBoost được huấn luyện trên dữ liệu huấn luyện của từng fold, với các hyperparameters đã được cài đặt trước.

***Đánh giá mô hình***

Mô hình được đánh giá bằng cách dự đoán kết quả trên tập kiểm tra của từng fold. Kết quả dự đoán được lưu vào DataFrame "oof" (out-of-fold predictions) tương ứng với các chỉ số của dữ liệu kiểm tra.

***Lưu trữ mô hình***

Mô hình XGBoost được lưu trữ trong dictionary "models" với key là kết hợp của nhóm (group) và câu hỏi (question).

Hơn nữa, để đo lường hiệu suất của mô hình, ta dùng chỉ số F1-Score

Quá trình này được lặp lại cho mỗi fold, trong đó mỗi fold sẽ có dữ liệu huấn luyện và kiểm tra khác nhau, nhưng không có mẫu trong cùng một nhóm (group) bị chia ra khỏi nhóm đó. Kết quả đánh giá từ mỗi fold được tổng hợp và sử dụng để đánh giá tổng thể hiệu suất của mô hình.

#### Độ đo

##### Định nghĩa

F1-Score là một độ đo sử dụng trong bài toán phân loại để đánh giá sự kết hợp giữa độ chính xác (precision) và độ bao phủ (recall) của mô hình.

F1-Score được tính dựa trên hai thành phần chính: precision và recall. Precision đo lường tỷ lệ các dự đoán dương tính (positive) được xác định đúng so với tổng số các dự đoán dương tính, trong khi recall đo lường tỷ lệ các mẫu dương tính được tìm ra đúng so với tổng số mẫu dương tính trong tập dữ liệu.

F1-Score là một số từ 0 đến 1, và giá trị càng cao càng tốt. Nó được tính bằng công thức:

F1-Score = 2 \* (precision \* recall) / (precision + recall)

F1-Score kết hợp cả precision và recall, và thường được sử dụng trong các tình huống mà cả precision và recall đều quan trọng. Nếu chỉ sử dụng precision hoặc recall mà không quan tâm đến yếu tố còn lại, có thể dẫn đến đánh giá sai lệch về hiệu suất của mô hình phân loại.

F1-Score thường được sử dụng khi dữ liệu mất cân bằng, tức là số lượng mẫu thuộc mỗi lớp không đồng đều. Trong trường hợp này, F1-Score đảm bảo rằng cả precision và recall đều được cân nhắc và không bị ảnh hưởng bởi mất cân bằng dữ liệu.

##### Quá trình

***Chuẩn bị dữ liệu:***

* Chuẩn bị dữ liệu dùng để đánh giá mô hình, bao gồm dữ liệu thực tế và dữ liệu dự đoán.
* Dữ liệu thực tế là các nhãn thực tế của các điểm dữ liệu.
* Dữ liệu dự đoán là các giá trị dự đoán được từ mô hình.

***Thiết lập threshold***

* Thiết lập một danh sách các giá trị threshold để thử nghiệm.
* Thông thường, danh sách threshold được chọn trong một khoảng giá trị nhất định.

***Tính toán F1-score và chọn threshold tốt nhất***

* Duyệt qua từng giá trị threshold trong danh sách đã thiết lập.
* Áp dụng threshold lên dữ liệu dự đoán để phân loại các điểm dữ liệu thành nhãn dương tính hoặc âm tính.
* Tính toán F1-score sử dụng nhãn thực tế và nhãn dự đoán với threshold hiện tại.
* Lưu trữ F1-score và threshold tương ứng.

***Chọn threshold tốt nhất***

* Chọn threshold tương ứng với F1-score cao nhất làm threshold tốt nhất.

***Đánh giá hiệu suất mô hình***

* Áp dụng threshold tốt nhất lên dữ liệu dự đoán để phân loại các điểm dữ liệu.
* Tính toán F1-score sử dụng nhãn thực tế và nhãn dự đoán với threshold tốt nhất.
* Đánh giá hiệu suất của mô hình dựa trên F1-score thu được.

Quá trình này giúp tìm ra threshold tối ưu để phân loại các điểm dữ liệu và đánh giá hiệu suất mô hình dựa trên F1-score, một phép đo kết hợp giữa precision và recall. Threshold tốt nhất được chọn dựa trên F1-score cao nhất, cho phép tối ưu hóa hiệu suất mô hình trong việc phân loại các điểm dữ liệu.

## Tổng quan về thuật toán

### Tổng quan về Ensemble learning

#### Khái niệm

Là một phương pháp trong Machine Learning, trong đó nhiều mô hình học máy (models) được kết hợp lại để tạo thành một mô hình dự đoán mạnh mẽ hơn. Mục tiêu của Ensemble Learning là tận dụng sự đa dạng và sự khác biệt giữa các mô hình để cải thiện khả năng dự đoán.

Cách thức hoạt động của Ensemble Learning là kết hợp các mô hình độc lập hoặc giống nhau nhưng được huấn luyện trên các tập dữ liệu khác nhau. Có ba phương pháp chính để kết hợp các mô hình trong Ensemble Learning:

***Bagging (Bootstrap Aggregating)***

* Các mô hình độc lập được huấn luyện trên các tập dữ liệu con được tạo ra bằng phương pháp tái chọn mẫu với khả năng tái chọn. Kết quả dự đoán cuối cùng được tính bằng cách lấy trung bình hoặc đa số phiếu bầu từ các mô hình con.

***Boosting***

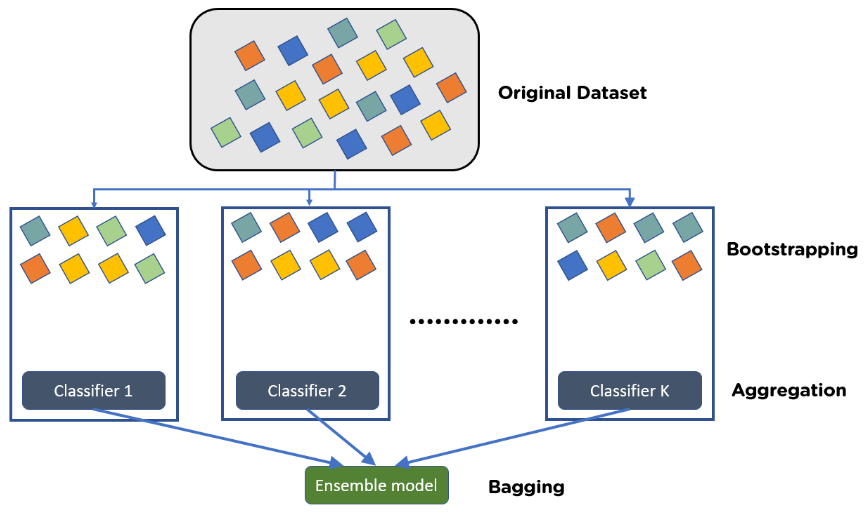
* Các mô hình được huấn luyện theo chuỗi liên tiếp, trong đó mỗi mô hình cố gắng sửa đổi dự đoán của các mô hình trước đó. Dữ liệu được tạo ra bằng cách tập trung vào các mẫu khó phân loại hoặc các mẫu bị dự đoán sai. Kết quả dự đoán cuối cùng được tính bằng cách kết hợp các dự đoán từ tất cả các mô hình con.

***Stacking***

* Xây dựng một số model (thường là khác loại) và một meta model (supervisor model), train những model này độc lập, sau đó meta model sẽ học cách kết hợp kết quả dự báo của một số mô hình một cách tốt nhất.

Ensemble Learning thường cho phép cải thiện độ chính xác và khả năng tổng quát hóa của mô hình dự đoán. Nó được sử dụng rộng rãi trong các bài toán phân loại, hồi quy và xếp hạng trong Machine Learning.

### Bagging



Hình : Tổng quát Sơ đồ hoạt động của Bagging

Bagging, còn được gọi là Bootstrap aggregating, là một kỹ thuật học tập đồng bộ giúp cải thiện hiệu suất và độ chính xác của các thuật toán học máy. Nó được sử dụng để đối phó với sự đánh đổi sai lệch-phương sai và giảm phương sai của một mô hình dự đoán. Bagging giúp tránh khớp dữ liệu quá mức và được sử dụng cho cả mô hình hồi quy và mô hình phân loại, đặc biệt cho các thuật toán cây quyết định.

#### Các bước thực hiện

* Giả sử có n quan sát và m thuộc tính trong tập huấn luyện. Chọn một mẫu ngẫu nhiên từ tập dữ liệu huấn luyện mà không cần thay thế.
* Một tập hợp con của m tính năng được chọn ngẫu nhiên để tạo mô hình bằng cách sử dụng các quan sát mẫu.
* Tính năng cung cấp sự phân chia lô tốt nhất được sử dụng để phân chia các nút
* Cây được trồng, vì vậy có các nút gốc tốt nhất.
* Các bước trên được lặp lại n lần. Nó tổng hợp đầu ra của các cây quyết định riêng lẻ để đưa ra dự đoán tốt nhất.

#### Ưu điểm

* Bagging giảm thiểu việc lắp quá nhiều dữ liệu.
* Cải thiện độ chính xác của mô hình.
* Xử lý dữ liệu chiều cao hơn một cách hiệu quả.

#### Sơ lược dưới dạng toán học

Ta có L bootstrap samples (tương ứng với L bộ dữ liệu) có kích thước B.

-th observation of the -th bootstrap sample

Tương ứng với L bộ dữ liệu là L model "yếu".

*w*1​(.),*w*2​(.),...,*wL*​(.)​

Kết hợp các model này lại, ta được một model mới mạnh hơn. Với những vấn đề khác nhau, như regression, đầu ra của các model "yếu" sẽ được trung bình cộng, kết quả này sẽ là đầu ra của model "mạnh". Còn với classification, class đầu ra của mỗi một model "yếu" sẽ được coi là 1 vote và class mà nhận được số vote nhiều nhất sẽ là đầu ra của model "mạnh" (cách này gọi là hard-voting). Trong trường hợp model "yếu" dự đoán xác suất của tất cả class thì ta sẽ tính trung bình cộng của xác suất của từng class rồi lấy xác suất có giá trị lớn nhất (cách này gọi là soft-voting).

(simple average, for regression problem)

(simple majority vote, for classification problem)

Một trong những lợi ích mà bagging mang lại, đó là tính song song. Như hình dưới, bạn sẽ thấy phần core của bagging đều là tiến trình song song nên nếu bạn có con máy khỏe, bạn có thể train từng model song song với nhau và cuối cùng tổng hợp đầu ra của các model này lại.

### Bootstrapping

#### Các bước thực hiện

***Chuẩn bị tập dữ liệu ban đầu***

Đầu tiên,cần có một tập dữ liệu ban đầu để tạo các bootstrap samples từ đó.

***Xác định kích thước bootstrap sample***

Xác định kích thước mỗi bootstrap sample mà bạn muốn tạo. Thông thường, kích thước của mỗi bootstrap sample bằng kích thước của tập dữ liệu ban đầu.

***Lặp lại quá trình bootstrap***

Các bước để tạo ra bootstrap samples

* Chọn một mẫu ngẫu nhiên từ tập dữ liệu ban đầu với khả năng chọn lại (replacement). Mẫu này có thể được chọn nhiều lần hoặc không được chọn.
* Thêm mẫu đã chọn vào bootstrap sample.
* Lặp lại các bước trên cho đến khi kích thước của bootstrap sample đạt được đúng kích thước mong muốn.

Có thể được lặp lại nhiều lần để tạo ra nhiều bootstrap samples. Số lần lặp lại này thường được chọn trước hoặc theo một số tiêu chí cụ thể.

#### Ưu điểm

***Sử dụng toàn bộ dữ liệu***

Kỹ thuật Bootstrapping cho phép sử dụng toàn bộ tập dữ liệu ban đầu để tạo ra các bootstrap samples. Điều này đảm bảo rằng không có mẫu nào bị bỏ lỡ và mọi thông tin từ tập dữ liệu gốc đều được sử dụng.

***Độc lập và ngẫu nhiên***

Các bootstrap samples được tạo ra từ dữ liệu gốc thông qua quá trình chọn mẫu ngẫu nhiên lặp lại. Điều này đảm bảo tính độc lập và ngẫu nhiên giữa các mẫu trong các bootstrap samples. Điều này quan trọng để đảm bảo tính khách quan và đại diện của các bootstrap samples.

***Ước lượng phương sai***

Bootstrapping được sử dụng rộng rãi để ước lượng phương sai của các ước lượng thống kê. Bằng cách tạo ra nhiều bootstrap samples và tính toán các ước lượng trên mỗi bootstrap sample, ta có thể xây dựng một phân phối ước lượng và ước lượng phương sai dựa trên phân phối này. Điều này giúp đánh giá độ tin cậy của ước lượng thống kê và xác định khoảng tin cậy.

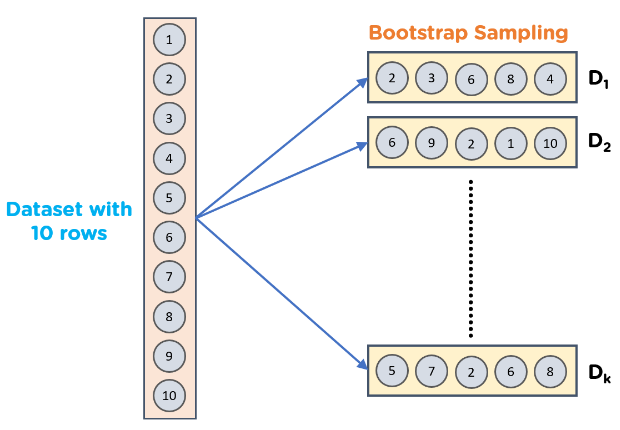
***Phân phối mẫu***

Các bootstrap samples có xu hướng có phân phối tương tự với phân phối của tập dữ liệu ban đầu. Điều này cho phép mô phỏng và kiểm tra các giả thuyết thống kê một cách linh hoạt và hiệu quả.

***Sử dụng trong học máy***

Bootstrapping được sử dụng trong các phương pháp ensemble như Bagging và Random Forest, nơi việc tạo ra các bootstrap samples giúp tăng tính đa dạng và ổn định của mô hình. Ngoài ra, nó cũng được sử dụng để đánh giá hiệu suất của mô hình thông qua phân chia bootstrap và kiểm tra thống kê.

#### Sơ lược về dạng toán học



Hình : Sơ đồ hoạt động của Bootstrap Sampling

observation of the -th bootstrap sample

Bootstrapping là phương pháp tạo ngẫu nhiên các mẫu dữ liệu từ một tập hợp với sự thay thế để ước tính một tham số tập hợp.

### Boosting

#### Các bước thực hiện

***Chuẩn bị tập dữ liệu***

Đầu tiên, cần chuẩn bị tập dữ liệu để sử dụng trong quá trình Boosting. Tập dữ liệu này bao gồm các biến đầu vào và biến mục tiêu (đối với bài toán phân loại hoặc hồi quy).

***Chọn mô hình cơ bản***

Chọn một mô hình cơ bản để sử dụng trong quá trình Boosting. Mô hình cơ bản thường là một mô hình yếu (weak learner), ví dụ như cây quyết định đơn giản (decision stump) hoặc hồi quy tuyến tính.

***Xây dựng mô hình ban đầu***

Xây dựng mô hình cơ bản đầu tiên trên toàn bộ tập dữ liệu huấn luyện. Đây sẽ là mô hình ban đầu để bắt đầu quá trình Boosting.

***Đánh giá hiệu suất***

Đánh giá hiệu suất của mô hình ban đầu trên tập dữ liệu huấn luyện. Điều này giúp đánh giá mức độ hoạt động của mô hình và xác định những mẫu dữ liệu mà mô hình đang dự đoán sai.

***Tạo lỗi dự đoán***

Tính toán lỗi dự đoán cho mỗi mẫu dữ liệu trong tập huấn luyện. Lỗi dự đoán thường được xác định dựa trên sự khác biệt giữa giá trị dự đoán và giá trị thực tế của biến mục tiêu.

***Tạo mô hình mới***

Dựa trên lỗi dự đoán, tạo một mô hình mới (mô hình yếu) để tập trung vào những mẫu dữ liệu bị dự đoán sai. Mô hình mới sẽ cố gắng cải thiện việc dự đoán trên các mẫu này.

***Tích hợp mô hình mới***

Kết hợp mô hình mới với mô hình ban đầu để tạo thành một mô hình tổng quát. Quá trình này có thể bao gồm việc tính toán trọng số cho mỗi mô hình và cách tính toán đầu ra cuối cùng.

***Cập nhật lỗi dự đoán***

Cập nhật lại lỗi dự đoán dựa trên mô hình mới được thêm vào. Điều này giúp tập trung vào những mẫu dữ liệu mà mô hình mới dự đoán sai.

***Lặp lại quá trình***

Lặp lại các bước từ bước 4 đến bước 8 cho đến khi đạt được số lượng mô hình yếu đủ hoặc khi điều kiện dừng được đáp ứng. Thông thường, Boosting sẽ tạo ra một tập hợp các mô hình yếu được kết hợp thành một mô hình mạnh (strong learner).

***Kết hợp dự đoán***

Kết hợp dự đoán từ tất cả các mô hình yếu để tạo ra dự đoán cuối cùng. Quá trình này có thể bao gồm việc tính toán trọng số cho từng dự đoán và cách tính toán đầu ra cuối cùng, tùy thuộc vào thuật toán Boosting cụ thể được sử dụng.

#### Ưu điểm

***Tính mạnh mẽ và hiệu quả***

Boosting có khả năng xây dựng mô hình mạnh (strong learner) từ các mô hình yếu (weak learner). Quá trình lặp lại trong Boosting giúp tập trung vào những mẫu dữ liệu khó dự đoán, cải thiện hiệu suất dự đoán và giảm lỗi.

***Tính linh hoạt***

Boosting có khả năng thích ứng và linh hoạt trong việc xử lý các bài toán phức tạp. Nó có thể được áp dụng cho cả bài toán phân loại và hồi quy, và có thể sử dụng với nhiều loại mô hình cơ bản khác nhau.

***Độ chính xác cao***

Với khả năng tập trung vào các mẫu dữ liệu khó dự đoán, Boosting có khả năng cung cấp độ chính xác cao hơn so với một mô hình đơn lẻ. Nó có thể giảm thiểu lỗi và cải thiện khả năng dự đoán trên tập dữ liệu kiểm tra.

***Khả năng xử lý nhiễu và overfitting***

Boosting có khả năng xử lý tốt vấn đề nhiễu trong dữ liệu và giảm thiểu tình trạng overfitting. Quá trình Boosting tập trung vào những mẫu dữ liệu khó dự đoán và tạo ra mô hình tổng quát hơn.

***Học tập tổng hợp***

Quá trình Boosting tạo ra một mô hình tổng hợp bằng cách kết hợp nhiều mô hình yếu. Mô hình tổng quát này có khả năng khai thác sự đa dạng của các mô hình yếu và tổng hợp thông tin từ tất cả các mô hình.

***Tăng tốc độ học tập***

Boosting thường được thực hiện theo cách song song, tức là các mô hình yếu có thể được huấn luyện đồng thời. Điều này giúp tăng tốc độ học tập và giảm thời gian huấn luyện so với một mô hình lớn đơn lẻ.

***Tích hợp công cụ***

Boosting được tích hợp sẵn trong các thư viện và framework học máy phổ biến như XGBoost và AdaBoost. Điều này giúp việc triển khai và sử dụng Boosting trở nên dễ dàng và tiện lợi.

#### Sơ lược về dạng toán học

where , and are the regularization parameters and residuals computed with the tree respectfully, and is a function that is trained to predict residuals, using for the tree. To compute we use the residuals computed, and compute the following: where is a differentiable loss function.

#### Ý tưởng

Boosting ra đời dựa trên việc mong muốn cải thiện những hạn chế của Bagging. Các model trong Bagging đều là học một cách riêng rẽ, không liên quan hay ảnh hưởng gì đến nhau, điều này trong một số trường hợp có thể dẫn đến kết quả tệ khi các model có thể học cùng ra 1 kết quả. Chúng ta không thể kiểm soát được hướng phát triển của các model con thêm vào bagging.

Chúng ta mong đợi các model yếu của thể hỗ trợ lẫn nhau, học được từ nhau để tránh đi vào các sai lầm của model trước đó. Đây là điều Bagging không làm được,

Ý tưởng cơ bản là Boosting sẽ tạo ra một loạt các model yếu, học bổ sung lẫn nhau. Nói cách khác, trong Boosting, các model sau sẽ cố gắng học để hạn chế lỗi lầm của các model trước.

Vậy làm thế nào để hạn chế được sai lầm từ các model trước? Boosting tiến hành đánh trọng số cho các mô hình mới được thêm vào dựa trên các cách tối ưu khác nhau. Tùy theo cách đánh trọng số (cách để các model được fit một cách tuần tự) và cách tổng hợp lại các model, từ đó hình thành nên 2 loại Boosting:

* Adaptive Boosting (AdaBoost)
* Gradient Boosting

Chúng ta sẽ phân tích sâu hơn về 2 dạng Boosting này ở phần sau. Để kết thúc phần này, có một vài nhận xét về Boosting như sau:

* Boosting là một quá trình tuần tự, không thể xử lí song song, do đó, thời gian train mô hình có thể tương đối lâu.
* Sau mỗi vòng lặp, Boosting có khả năng làm giảm error theo cấp số nhân.
* Boosting sẽ hoạt động tốt nếu base learner của nó không quá phức tạp cũng như error không thay đổi quá nhanh.
* Boosting giúp làm giảm giá trị bias cho các model base learner.

##### AdaBoost – Gradient Boosting

Cả AdaBoost và Gradient Boosting đều xây dựng thuật toán nhằm giải quyết bài toán tối ưu sau:

Trong đó:

* *L* : giá trị loss function
* *y* : label
* *cn* : confidence score của weak learner thứ n (hay còn gọi là trọng số)
* *wn* : weak learner thứ n

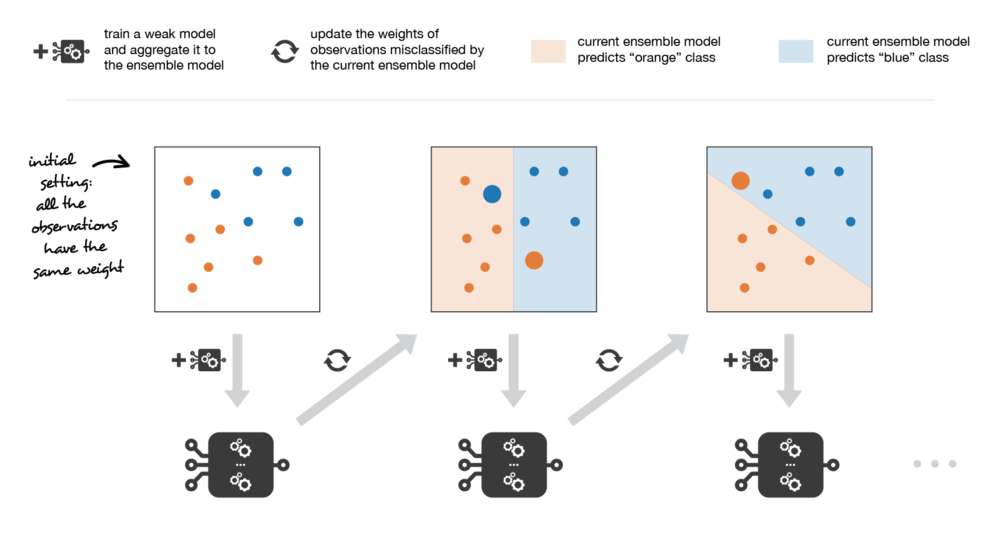
Thoạt nhìn, công thức trên có vẻ khá giống với Bagging, thế nhưng cách tính ra các giá trị confidence score kia lại làm nên sự khác biệt về hướng giải quyết của Boosting. Thay vì cố gắng quét tìm tất cả các giá trị  *cn,wn*​ để tìm nghiệm tối ưu toàn cục - một công việc tốn nhiều thời gian và tài nguyên, chúng ta sẽ cố gắng tìm các giá trị nghiệm cục bộ sau khi thêm mỗi một mô hình mới vào chuỗi mô hình với mong muốn dần đi đến nghiệm toàn cục.

với

##### Adaptive Boosting

AdaBoost tiến hành train các mô hình mới dựa trên việc đánh lại trọng số cho các điểm dữ liệu hiện tại, nhằm giúp các mô hình mới có thể tập trung hơn vào các mẫu dữ liệu đang bị học sai, từ đó làm giảm giá trị loss của mô hình. Cụ thể, các bước triển khai thuật toán như sau:

* Khởi tạo weight ban đầu là bằng nhau (bằng 1/*N*) cho mỗi điểm dữ liệu
* Tại vòng lặp thứ i
  + train model *wi*​ (weak learner) mới được thêm vào
  + tính toán giá trị loss (error), từ đó tính toán ra giá trị confidence score *ci*​ của model vừa train
  + Cập nhật model chính *W*=*W*+*ci*​∗*wi*
  + Cuối cùng, đánh lại trọng số cho các điểm dữ liệu (Các điểm dữ liệu bị đoán sai --> tăng trọng số, các điểm dữ liệu đoán đúng --> giảm trọng số).
* Sau đó lặp lại với vòng lặp thêm model tiếp theo i + 1.



Hình : Sơ đồ hoạt động của Adaptive Boosting

##### Gradient Boosting

Gradient Boosting là một dạng tổng quát hóa của AdaBoost. Cụ thể như sau, vẫn vấn đề tối ưu ban đầu

Phía trên là công thức cập nhật tham số mô hình theo hướng giảm của đạo hàm (Gradient Descent). Công thức này được sử dụng không gian tham số, tuy nhiên, để liên hệ với bài toán chúng ta đang xét, mình chuyển công thức sang góc nhìn của không gian hàm số.

Khá đơn giản thôi, nếu chúng ta coi chuỗi các model boosting là một hàm số *W*, thì mỗi hàm learner có thể coi là một tham số *w*. Đến đây, để cực tiểu hóa hàm loss *L*(*y*,*W*), chúng ta áp dụng Gradient Descent

Đến đây, ta có thể thấy mối quan hệ liên quan sau

Với wn làmodel được thêm vào tiếp theo. Khi đó, model mới cần học để fit để vào giá trị . (Giá trị còn có 1 tên gọi khác là **pseudo-residuals**

Tóm lại, chúng ta có thể tóm tắt quá trình triển khai thuật toán như sau:

* Khởi tạo giá trị pseudo-residuals là bằng nhau cho từng điểm dữ liệu
* Tại vòng lặp thứ i
  + Train model mới được thêm vào để fit vào giá trị của pseudo-residuals đã có
  + Tính toán giá trị confidence score *ci* của model vừa train
  + Cập nhật model chính *W*=*W*+*ci*​∗*wi*​
* Cuối cùng, tính toán giá trị pseudo-residuals để làm label cho model tiếp theo
* Sau đó lặp lại với vòng lặp i + 1.

Nếu bạn để ý thì phương pháp cập nhật lại trọng số của điểm dữ liệu của AdaBoost cũng là 1 trong các case của Gradient Boosting. Do đó, Gradient Boosting bao quát được nhiều trường hợp hơn.

A screenshot of a computer

Description automatically generated with medium confidence

Hình : Sơ đồ hoạt động của Gradient Boosting

##### Xgboost

***Regularization***

* XGBoost cho phép áp dụng regularization trên hàm ước lượng mục tiêu của mô hình, giúp giải quyết vấn đề quá khớp và giảm thiểu nhiễu trong dữ liệu.

***Parallel Processing***

* XGBoost hỗ trợ việc xử lý đa luồng và phân tán với nhiều CPU và máy tính khác nhau, giúp cho việc huấn luyện mô hình được thực hiện nhanh chóng.

***Handling Missing Values***

* XGBoost có khả năng xử lý giá trị thiếu sót trong dữ liệu, bằng cách xem chúng như một biến tạm thời và giải quyết chúng dựa trên cách mà các giá trị khác được sắp xếp

***Customization***

* XGBoost cho phép người dùng tùy chỉnh hàm mất mát và đánh giá mô hình để phù hợp với các vấn đề đặc biệt.

***Cross-validation***

* XGBoost cho phép người dùng sử dụng kỹ thuật cross-validation để kiểm tra hiệu quả của mô hình.

### Kỹ thuật

***Symmetric Decision Trees***

* CatBoost sử dụng các cây quyết định đối xứng trong quá trình huấn luyện mô hình, đảm bảo rằng các quyết định được chia sẻ đối xứng giữa các nhánh, giúp cải thiện độ chính xác của mô hình.

***CatBoost Encoding***

* Đây là một phương pháp mã hóa dữ liệu định danh mới được đưa ra bởi CatBoost. Nó cho phép mô hình xử lý các tính năng định danh một cách hiệu quả bằng cách tối ưu hóa giá trị đầu vào của chúng.

***Handling Categorical Features***

* CatBoost có khả năng xử lý các tính năng phân loại và có thể sử dụng dữ liệu định danh như là một đặc trưng đầu vào thông qua phương pháp mã hóa của riêng nó.

***Speed and Memory Optimization***

* CatBoost được tối ưu hóa để chạy nhanh và ít tốn tài nguyên bộ nhớ, giúp cho việc huấn luyện mô hình cũng như đưa ra các dự đoán được thực hiện trong thời gian ngắn.

***Cross-validation***

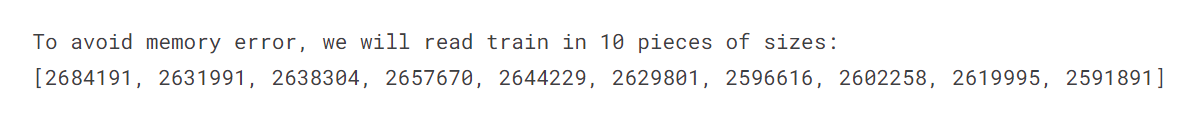
* CatBoost hỗ trợ các kỹ thuật cross-validation để đánh giá hiệu quả mô hình của nó.

## Thực nghiệm thuật toán

### Xgboost

#### Các bước thực hiện

***Bước 1: Giảm thiểu bộ nhớ để tránh lỗi***



**Hình 5. 13: Avoid memory error**

Lưu ý: Dữ liệu của cuộc thi đã thay đổi (cụ thể là gấp đôi kích thước) vào cuối tháng 3.

Do đó chúng tôi đã chia tập dữ liệu tập train thành 10 phần và xử lý từng phần để tránh việc bộ nhớ bị lỗi (vì cuộc thi chỉ cho phép 8GB RAM).

Sau khi chia tập dữ liệu train thành 10 phần thì chúng ta sẽ kiểm tra xem ở phần đầu tiên sẽ có kích thước là bao nhiêu:

A picture containing text, font, white, typography

Description automatically generated

Hình : Train size of first piece

A screenshot of a computer

Description automatically generated with medium confidence

Hình : Records of first piece

***Bước 2: Phân tích về tập train\_labels***

Tách phần đuôi của session\_id để lấy ra riêng câu hỏi và session để thuận tiện và dự đoán, sau khi xử lý kết quả nhận được như dưới.

A screenshot of a computer

Description automatically generated with low confidence

Hình : Tập train\_labels sau khi đã xử lý

***Bước 3: Feature engineering***

Feature engineering là quá trình tạo ra và chọn lọc các đặc trưng (features) từ dữ liệu gốc để cải thiện hiệu suất và kết quả của mô hình dự đoán. Điều này bao gồm việc áp dụng các phép biến đổi, trích xuất và chọn lọc đặc trưng để tạo ra các biến đầu vào tốt hơn cho mô hình học máy. Mục tiêu của feature engineering là tăng tính biểu diễn của dữ liệu, giảm độ nhiễu và cải thiện khả năng dự đoán của mô hình.

Do đó mà có thể nói Feature engineer là một trong những bước rất quan trọng trong việc quyết định độ thành công của mô hình.

***Ý tưởng:***

* Đây là dữ liệu nhật ký, nghĩa là mỗi phiên có nhiều sự kiện. Đối với việc lập mô hình, điều quan trọng là phải quyết định đơn vị phân tích.
* Đối với mỗi (session\_id) \_ (câu hỏi #), bạn đang dự đoán đúng cột, xác định xem bạn có tin rằng người dùng trong phiên cụ thể này sẽ trả lời chính xác câu hỏi này hay không, chỉ sử dụng thông tin trước đó cho phiên.
* Trong trường hợp này, bạn có thể coi mỗi session\_id là một người dùng. Nói một cách đơn giản, tôi sẽ xem xét tất cả dữ liệu nhật ký của mỗi phiên và dự đoán liệu người dùng có thể trả lời đúng từng câu hỏi hay không.
* Tôi cho rằng hành vi của người dùng ở mỗi cấp độ trò chơi là quan trọng. Do đó, tôi sẽ tổng hợp dữ liệu theo cấp độ trò chơi của từng phiên (người dùng). Ngoài ra, tôi sẽ tạo các biến giả của biến event\_name để mô hình có thể tìm hiểu thêm về từng sự kiện.
* Tóm lại, chúng tôi xây dựng ba mô hình. - Tìm hiểu hành vi đăng nhập của nhóm cấp độ 0-4, dự đoán câu hỏi 1-3. - tìm hiểu hành vi log của nhóm cấp độ 5-12, dự đoán câu hỏi 4~13 - tìm hiểu hành vi log của nhóm cấp độ 13-22, dự đoán câu hỏi 14~18.

A picture containing text, diagram, plan, line

Description automatically generated

Hình : Ý tưởng Feature Engineer

***Bước 4: Triển khai Feature Engineer***

A picture containing text, screenshot, font, line

Description automatically generated

Hình : Feature Engineer

Ở đây, phân loại các biến thành 2 loại, CATS (Category hoặc là Categorical), bao gồm những những cột mang giá trị thuộc một tập hợp hữu hạn các nhãn hoặc danh mục. Các biến đầu vào này không có thứ tự hay mức độ đo được và thường biểu thị các thuộc tính không liên tục.

NUMS (Numbers) là bao gồm những cột mang giá trị số. (Các số có thể biểu diễn các thuộc tính định lượng như chiều cao, trọng lượng, tuổi, hoặc các đặc trưng số khác)

EVENTS là thuộc tính quan trọng mà chúng cần phân tích và sử dụng cho việc dự đoán, do đó ở trong biến này, ta liệt kê tất cả các giá trị có trong cột này.

A screenshot of a computer program

Description automatically generated with medium confidence

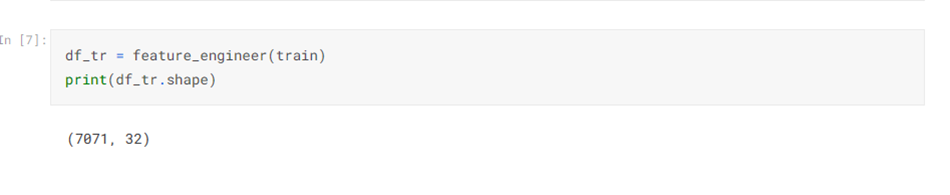
Hình : Feature Engineer in code

Việc xử lí diễn ra trong hàm này như sau:

Đối với CATS, Lấy ra những giá trị duy nhất có trong cột này, và những giá trị này được lấy theo session\_id và level\_group, nghĩa là lấy theo mã định danh của người dùng và nhóm cấp độ.

Đối với NUMS, tính giá trị trung bình và độ lệch chuẩn cho các giá trị của cột này, và cũng như CATS, những giá trị này cũng được lấy theo session\_id và level group.

Đối với EVENTS, chuyển các giá trị cụ thể có trong cột event\_name thành kiểu int8 để giảm bộ nhớ và cộng với cột elapsed\_time để chúng ta có thể lấy ra được chính xác người dùng mất bao lâu thời gian cho mỗi event, từ đó tính tổng cho cột elapsed\_time theo session\_id và level group để có thể biết được cụ thể người nào và thuộc ở nhóm cấp độ nào.



Hình : Kích thước sau khi xử lý

Áp dụng Feature Engineer cho các phần còn lại của tập train, bên cạnh đó để xem xét cả cải thiện về thời gian

A picture containing text, font, screenshot, algebra

Description automatically generated

Hình : Áp dụng cho các phần còn lại của tập train

***Bước 5: Tinh chỉnh siêu tham số và Train XGBoost model***

A picture containing text, font, white, screenshot

Description automatically generated

Hình : Thuộc tính và users mà ta sẽ train

Train model thuật toán Xgboost với 31 thuộc tính và 23562 bản ghi thông tin người dùng.

***Tinh chỉnh siêu tham số***

A screenshot of a computer code

Description automatically generated with medium confidence

Hình : Siêu tham số cho Xgboost

***Objective ‘binary: logistic’***

Thuật toán sử dụng hồi quy logistic để phân loại nhị phân, và output sẽ là xác suất.

***Eval\_metric: logloss***

Số liệu đánh giá cho dữ liệu xác thực, số liệu mặc định sẽ được chỉ định theo mục tiêu (rmse cho hồi quy và logloss cho phân loại, độ chính xác trung bình cho xếp hạng: bản đồ, v.v.)

**Learning\_rate: 0.05**

Thu hẹp kích thước bước được sử dụng trong bản cập nhật để ngăn quá mức. Sau mỗi bước tăng cường, có thể trực tiếp lấy trọng số của các tính năng mới và eta thu nhỏ trọng số của tính năng để làm cho quá trình tăng cường trở nên thận trọng hơn. phạm vi: [0,1], mặc định = 0.3

***Max\_depth: 5***

Độ sâu tối đa của cây. Việc tăng giá trị này sẽ làm cho mô hình trở nên phức tạp hơn và dễ bị overfit hơn. 0 cho biết không có giới hạn về độ sâu. XGBoost sẽ tiêu thụ bộ nhớ mạnh mẽ khi đào tạo một cây sâu. phương pháp cây chính xác yêu cầu giá trị khác không. phạm vi: [0,∞], mặc định = 6.

***N\_estimators: 1000***

Số lượng cây quyết định (Decision tree), tham số này sẽ xác định số lượng vòng lặp (iterations) mà thuật toán sẽ thực hiện để xây dựng các cây trong quá trình tối ưu hóa mục tiêu.

***Early\_stopping\_rounds: 50***

Tham số quan trọng trong việc cải thiện hiệu suất của mô hình. Khi sử dụng early\_stopping, quá trình train mô hình sẽ dừng lại nếu không có sự cải thiện đáng kể trong một số vòng lặp liên tiếp đã được chỉ định. Ngăn chặn overfitting và giảm đáng kể thời gian train.

***Tree\_method: hist***

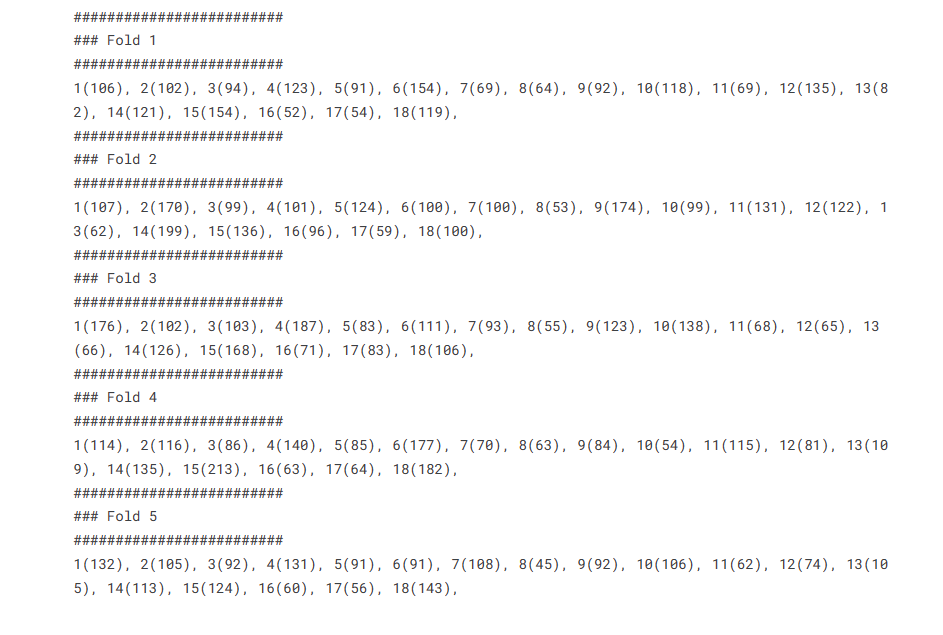
Xgboost sử dụng phương pháp xây dựng cây dựa trên histogram để tăng tốc độ huấn luyện và giảm bộ nhớ sử dụng. Phương pháp này tổng hợp các giá trị đặc trưng vào các histogram và sử dụng chúng để đưa ra quyết định tốt hơn về việc tách nút.

***Subsample: 0.8***

80% mẫu được chọn ngẫu nhiên sẽ được sử dụng trong mỗi vòng lặp huấn luyện. Tham số subsample giúp giảm khả năng overfitting và tăng tính tổng quát hóa của mô hình. Bằng cách sử dụng một tỷ lệ mẫu nhỏ hơn 100%, mô hình chỉ sử dụng một phần mẫu để huấn luyện, điều này có thể làm giảm độ chính xác trên tập huấn luyện nhưng có thể cải thiện khả năng tổng quát hóa và tránh overfitting.

***Colsample\_bytree:0.8***

80% số lượng đặc trưng sẽ được chọn ngẫu nhiên để sử dụng trong việc xây dựng cây. Tham số colsample\_bytree giúp giảm khả năng overfitting và tăng khả năng tổng quát hóa của mô hình. Bằng cách sử dụng một tỷ lệ nhỏ hơn 100%, mô hình chỉ sử dụng một phần đặc trưng để xây dựng cây, điều này có thể giúp giảm hiện tượng quá khớp (overfitting) và tăng cường khả năng tổng quát hóa của mô hình.



Hình : Kết quả sau khi train mô hình

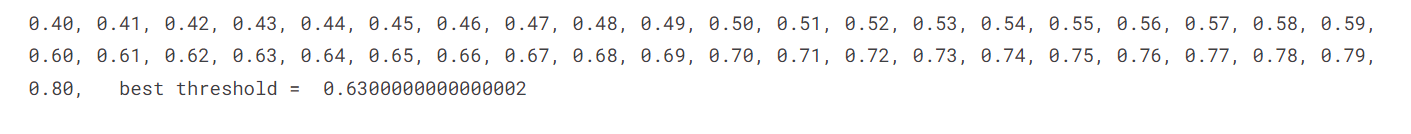
Bên cạnh đó, nhóm còn sử dụng phương pháp GroupKFold để chia tập dữ liệu thành 5 nhóm Fold vì nhóm chia tập ra train theo nhóm cấp độ của câu hỏi nên sẽ nâng cao hiệu suất của mô hình hơn.

***Bước 6: Tìm ra điểm Threshold (điểm ngưỡng)***

Threshold là một ngưỡng hoặc giá trị ngưỡng được sử dụng để đưa ra quyết định hoặc phân loại trong các bài toán liên quan đến dự đoán hoặc phân loại.

Trong bài toán dự đoán nhị phân (binary prediction), threshold được sử dụng để quyết định xem một mẫu nào đó thuộc vào lớp dương (positive) hay lớp âm (negative).

Lựa chọn ngưỡng có thể ảnh hưởng đến kết quả của mô hình và phụ thuộc vào yêu cầu và mục tiêu của bài toán cụ thể. Thông thường, ngưỡng được chọn dựa trên các yếu tố như tỷ lệ lỗi phân loại (misclassification rate), độ nhạy (sensitivity), độ đặc trưng (specificity), hoặc F1-score.



Hình : Tìm Threshold tốt nhất

A picture containing line, plot, diagram, slope

Description automatically generated

Hình : Tương quan giữa Threshold và F1-Score tốt nhất

#### Kết quả

Áp dụng điểm Threshold tốt nhất vào để tính điểm F1-Score cho từng câu hỏi, như vậy ta có thể điểm F1-Score tổng cho mô hình dự đoán của chúng ta.

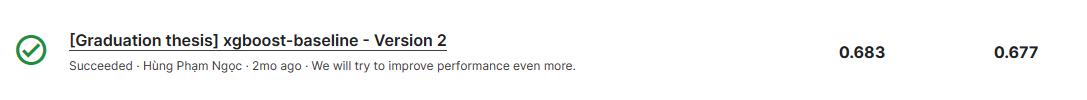
A screenshot of a computer

Description automatically generated with medium confidence

Hình : Điểm F1-score cho từng câu hỏi và điểm tổng Xgboost

Vậy là chúng ta đã có số điểm dự đoán là 67,85% số học sinh sẽ trả lời đúng từ lần đầu tiên trả lời câu hỏi trên tập train này.

#### Điểm số của XGBoost



Hình : Điểm số của XGBoost

Với điểm số công khai là 0.677 và điểm số bí mật được chấm lại là 0.683

### CatBoost

#### Các bước thực hiện

Ở thuật toán CatBoost, Xử lý data kỹ càng hơn để có thể đưa ra được kết quả tốt hơn cho các biến phân loại cũng như là tính đặc trưng của thuật toán.

Bước 1: Feature Engineering



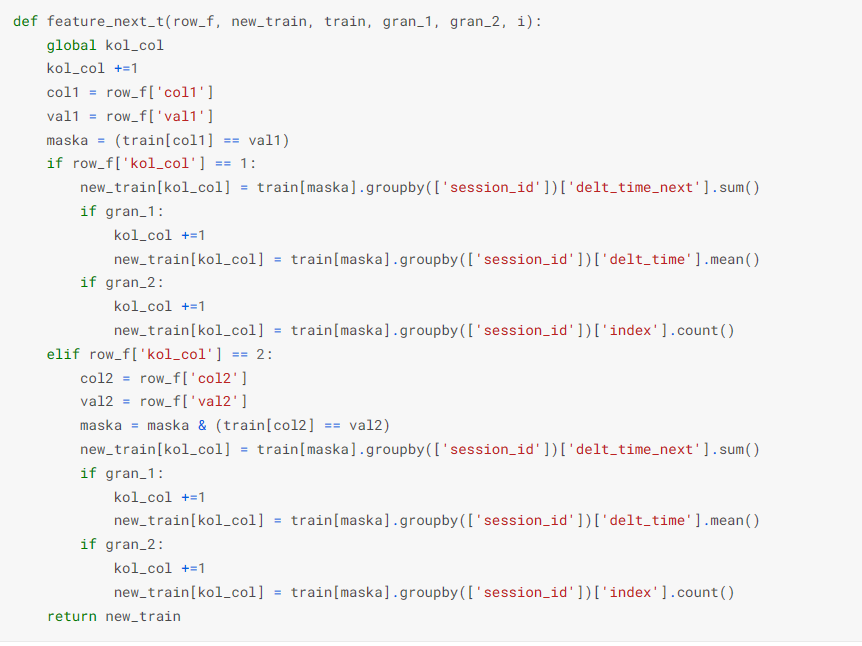
Hình : Xử lý data cho cột null

Trước hết, cần xử lý biến thời gian “elapsed\_time” theo “session id” để tăng hiệu suất xử lý của thuật toán, đơn giản hóa ý nghĩa của hàm là biến giá trị số thời gian thành biến phân loại, do đó thuật toán Catboost có thể xử lý tốt hơn.



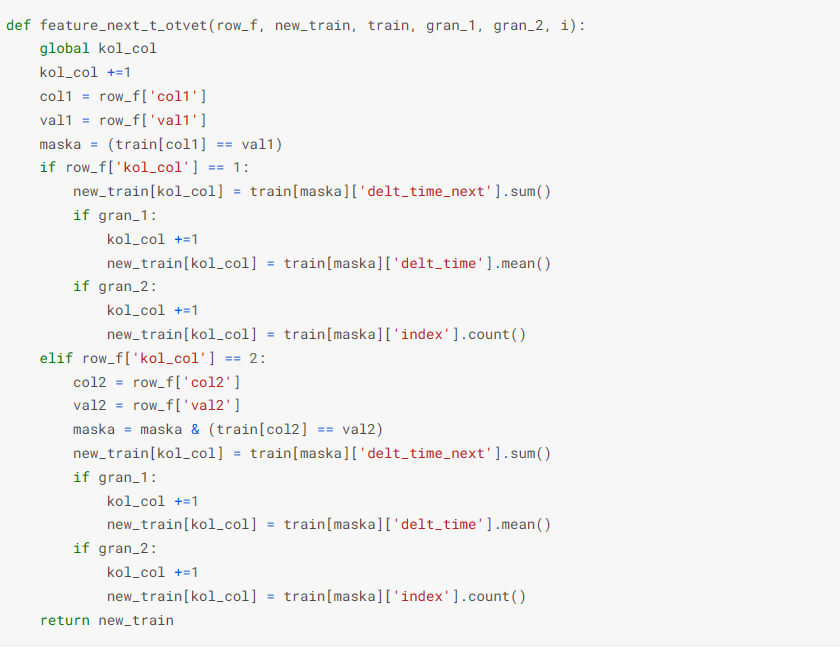
Hình : Xử lý biến thời gian

Ở đây, hàm sẽ tiếp tục xử lý tất cả các đặc trưng về thời gian (“d\_time”:trước đó đã xử lý lần 1, “hover\_duration”,..) như là tính toán giá trị trung bình, độ lệch chuẩn cũng như là biến những giá trị tháng, năm, ngày và cả thời gian thành biến phân loại.



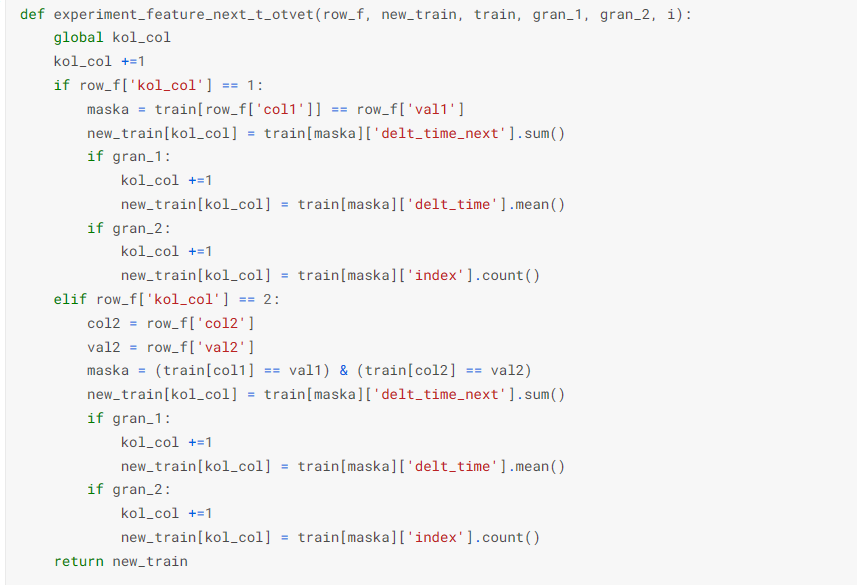
Hình : Tạo đặc trưng cho biến thời gian

Tiếp tục tạo thêm những đặc trưng cho việc xử lý dữ liệu của CatBoost càng thêm tốt hơn bằng cách xét từng giá trị cụ thể cho các biến thời gian. Tất cả những đặc trưng này đều được tính toán và tạo ra dựa trên “session\_id” để có thể tương quan và đưa ra dự đoán chính xác nhất về hành vi của người dùng.



Tiếp tục tạo thêm đặc trưng cho biến thời gian dựa trên biến đã tạo

Hàm “feature\_next\_t\_otvet” tương tự như hàm “feature\_next\_t” nhưng chỉ khác biệt một tí xíu về cách gán giá trị và cách truy cập.



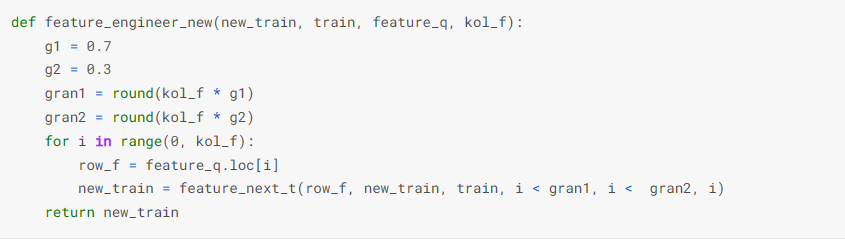
Hình : Lấy ra biến đặc trưng tốt nhất

Nếu chỉ nhìn sơ qua hàm “experiment\_feature\_next\_t\_otvet” không khác gì 2 hàm ở trên, biến maska ở đây được xử lý khác đi một tí xíu, đó là ở 2 hàm trước, mở rộng biến maska sau khi nhận thấy đặc trưng đó đã có, nhưng biến maska ở hàm này chỉ là so sánh lại xem với những đặc trưng trước đó đã được cải thiện hay chưa.



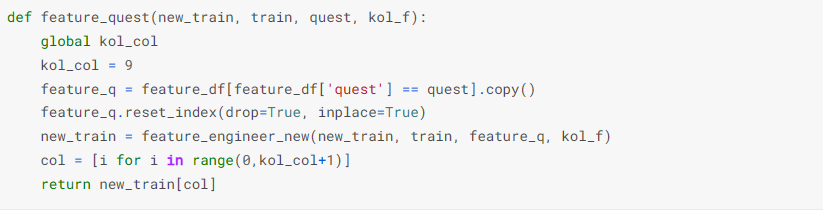
Hình : Xử lý biến câu hỏi

Trước đó, việc sắp xếp và tạo một tập dữ liệu dành riêng cho các đặc trưng quan trọng cần xử lý có tên là “feature\_df” đã được diễn ra, do đó ở hàm này, tạo những đặc trưng mới cho biến câu hỏi (“quest”). Ở đây, gọi lại hàm “feature\_next\_t\_otvet”, là để tính toán các đặc trưng mới của biến “quest” dựa trên biến thời gian trả lời các câu hỏi của người dùng.



Hình : Tạo tập dữ liệu mới dựa trên những đặc trưng mới

Ở hàm “feature\_engineer” khác này, sẽ xử lý thêm những đặc trưng về thời gian, nhưng mà là dựa trên tập “feature\_df” như đã đề cập ở hàm trên. Từ đó, tạo thêm những đặc trưng mới.

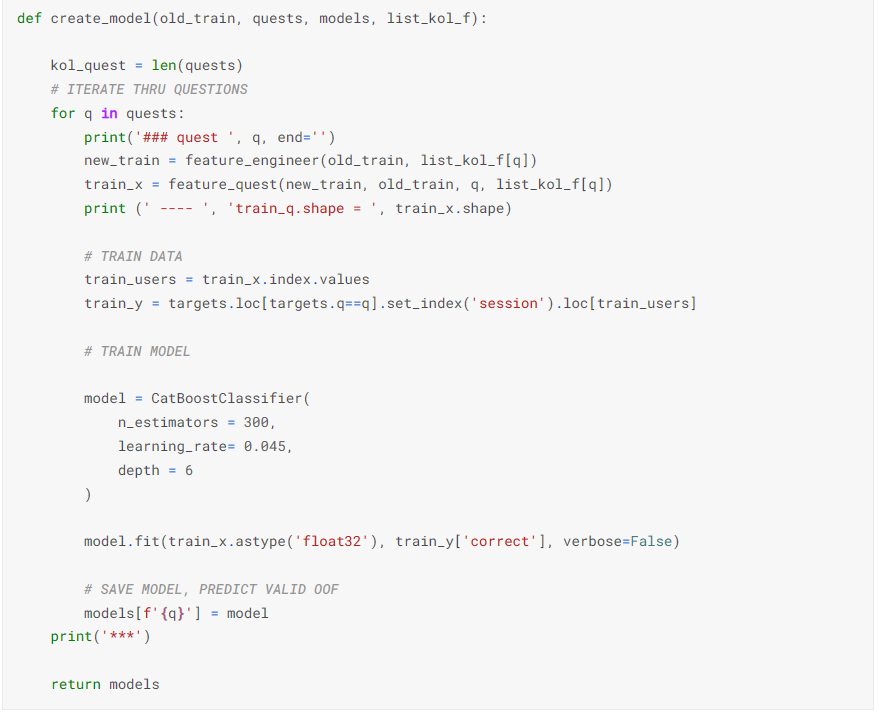


Hình : Xử lý biến câu hỏi dựa trên tập dữ liệu mới

Hàm này sẽ là hàm xử lý các đặc trưng về biến quest dựa trên tập data train và tập mà hàm “feature\_engineer\_new” đã tạo ra. Từ đó tạo ra những đặc trưng mới, góp phần tăng hiệu suất cho thuật toán CatBoost

Tổng kết phần Feature Engineer: Qua một bước xử lý các đặc trưng cho tập dữ liệu thì cũng sẽ đoán được là ở mô hình CatBoost này, Đặc trưng quan trọng mà mô hình muốn đánh mạnh vào đó chính là thời gian trả lời câu hỏi hay còn có thể nói là thời gian đưa ra câu trả lời của từng câu hỏi dựa trên từng người dùng.

***Bước 2: Tinh chỉnh siêu tham số và train model CatBoost***



Hình : Tinh chỉnh siêu tham số cho mô hình CatBoost

Huấn luyện mô hình CatBoost với các siêu tham số. Ở mô hình CatBoost thì chỉ tinh chỉnh 3 siêu tham số, còn những tham số khác thì mặc định.

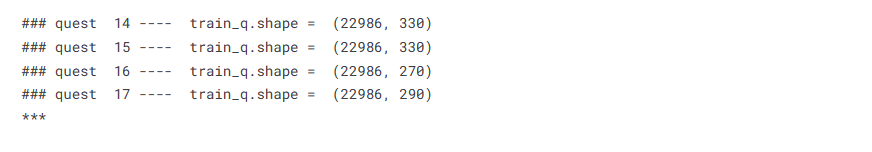
Bên cạnh đó thì như ở XGBoost, chỉ số ngưỡng (Threshold) tốt nhất là 0.63 nên ở CatBoost, thì sử dụng lại chỉ số ngưỡng mà không cần tính toán nữa.



Hình : Kết quả train câu hỏi từ 1 đến 3



Hình : Kết quả train câu hỏi từ 4 đến 11



Hình : Kết quả train câu hỏi từ 14 đến 17

***Bước 3: Kết quả trên tập test***



Hình : Kết quả đạt được

***Tinh chỉnh siêu tham số***

A screen shot of a computer code

Description automatically generated with low confidence

Hình : Định nghĩa siêu tham số sử dụng trong thuật toán CatBoost

***Iterations: 1000***

Trong quá trình huấn luyện, mô hình sẽ huấn luyện 1000 cây tăng cường để tạo ra mô hình cuối cùng.

***Early\_stopping\_rounds: 50***

Tham số quan trọng trong việc cải thiện hiệu suất của mô hình. Khi sử dụng early\_stopping, quá trình train mô hình sẽ dừng lại nếu không có sự cải thiện đáng kể trong một số vòng lặp liên tiếp đã được chỉ định. Ngăn chặn overfitting và giảm đáng kể thời gian train.

***Depth: 4***

Các cây tăng cường trong quá trình huấn luyện sẽ có độ sâu tối đa là 4.

***Learning\_rate: 0.05***

Thu hẹp kích thước bước được sử dụng trong bản cập nhật để ngăn quá mức. Sau mỗi bước tăng cường, chúng tôi có thể trực tiếp lấy trọng số của các tính năng mới và eta thu nhỏ trọng số của tính năng để làm cho quá trình tăng cường trở nên thận trọng hơn. phạm vi: [0,1], mặc định = 0.3

***Loss\_function:***

Logloss Nghĩa là hàm mất mát được sử dụng trong quá trình huấn luyện mô hình là Logarithmic Loss.

***Random\_seed: 0***

Điều này đảm bảo rằng mô hình sẽ sử dụng hạt giống ngẫu nhiên là 0 cho các quá trình ngẫu nhiên, giúp đạt được tính nhất quán và tái lặp khi chạy lại mô hình trên cùng một cấu hình.

***Metric\_period:1***

Điều này đảm bảo rằng mô hình sẽ tính toán và báo cáo kết quả độ đo sau mỗi bước huấn luyện.

***Subsample: 0.8***

80% mẫu được chọn ngẫu nhiên sẽ được sử dụng trong mỗi vòng lặp huấn luyện. Tham số subsample giúp giảm khả năng overfitting và tăng tính tổng quát hóa của mô hình. Bằng cách sử dụng một tỷ lệ mẫu nhỏ hơn 100%, mô hình chỉ sử dụng một phần mẫu để huấn luyện, điều này có thể làm giảm độ chính xác trên tập huấn luyện nhưng có thể cải thiện khả năng tổng quát hóa và tránh overfitting.

***Colsample\_bytree:0.4***

Mỗi cấp của các cây tăng cường sẽ sử dụng ngẫu nhiên 40% số lượng đặc trưng có sẵn để xây dựng cây.

***Verbose:0***

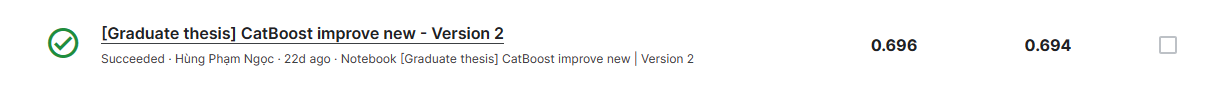
Điều này đảm bảo rằng không có thông tin hiển thị được in ra trong quá trình huấn luyện mô hình.

#### Kết quả

Về mặt cơ bản thì CatBoost cũng được xây dựng dựa trên Gradient Boosting, CatBoost được thiết kế để xử lý dữ liệu có biến đổi dạng danh mục (categorical features) mà không cần mã hóa trước. Thay vì mã hóa dữ liệu, CatBoost tự động xử lý biến đổi dạng danh mục bằng cách sử dụng mã hóa đặc trưng thẳng (direct feature hashing) và học mẫu đặc trưng (feature embedding). Điều này giúp giảm khó khăn và thời gian tiền xử lý dữ liệu và cho phép sử dụng dữ liệu dạng danh mục trực tiếp trong mô hình.

So với kết quả là 0.6 ở thuật toán XGBoost thì thật sự CatBoost làm tốt hơn về bài toán phân loại này.

#### Điểm số của CatBoost



Hình : Điểm số của bài dự thi CatBoost

Với điểm số công khai là 0.694 và điểm số bí mật được chấm lại sau khi cuộc thi kết thúc là 0.696

#### Kết luận

Có thể thấy hiệu suất đạt ngưỡng 68% tỉ lệ trả lời đúng câu hỏi lần đầu tiên của học sinh ở tập train.

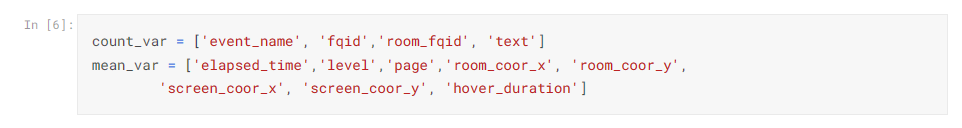
Nghĩa là với nhiều biến đầu vào là categorical thì thuật toán CatBoost có vẻ như làm tốt hơn so với Xgboost, thông qua hiệu suất có thể thấy điều đó, mặc dù Catboost mất gian xử lý dữ liệu hơn Xgboost. Nhưng tổng quan về hiệu suất mà mô hình mang lại thì CatBoost đang thực sự chiếm ưu thế hơn hẳn so với XGboost.

### LightGBM

#### Các bước thực hiện

Với thuật toán LightGBM, ta cũng sẽ có các bước giảm thiểu bộ nhớ để tránh lỗi như Xgboost. Vì thế ta sẽ không nói lại ở phần này.

***Bước 1: Triển khai Feature Engineer***



Hình : Feature Engineer

Phân loại các biến thành 2 loại:

count\_var (biến đếm) là biến bao gồm những những cột mang giá trị thuộc một tập hợp hữu hạn các nhãn hoặc danh mục. Các biến đầu vào này không có thứ tự hay mức độ đo được và thường biểu thị các thuộc tính không liên tục.

mean\_var (biến trung bình) là biến bao gồm những cột mang giá trị số (Các số có thể biểu diễn các thuộc tính định lượng như chiều cao, trọng lượng, tuổi, hoặc các đặc trưng số khác)

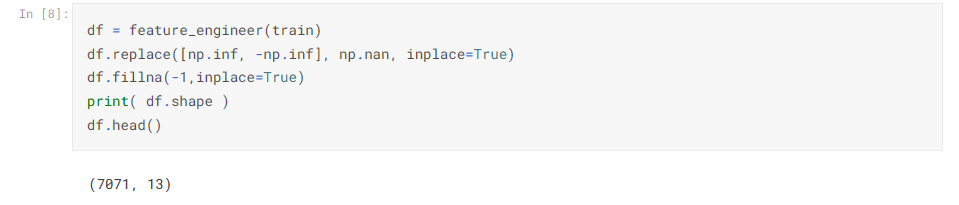


Hình : Feature Engineer in code

Việc xử lí diễn ra trong hàm này như sau:

Đối với count\_var, Lấy ra những giá trị duy nhất có trong cột này, và những giá trị này được lấy theo session\_id và level\_group, nghĩa là lấy theo mã định danh của người dùng và nhóm cấp độ.

Đối với mean\_var, tính giá trị trung bình cho các giá trị của cột này, những giá trị này cũng được lấy theo session\_id và level group.



Hình : Kích thước sau khi xử lý

***Bước 2: Tinh chỉnh siêu tham số và train LightGBM model***

****

Hình : Train model thuật toán LightGBM

***Tinh chỉnh siêu tham số***

A screen shot of a computer code

Description automatically generated with low confidence

Hình : Tinh chỉnh siêu tham số LightGBM

***'objective': 'binary'***

Trong tham số của LightGBM, 'objective': 'binary' được sử dụng để chỉ định mục tiêu của bài toán huấn luyện là phân loại nhị phân (binary classification). Điều này có nghĩa là LightGBM sẽ xây dựng một mô hình dự đoán để phân loại các điểm dữ liệu vào hai nhóm khác nhau, thường được đại diện bởi các nhãn nhị phân, ví dụ: 0 và 1, đúng và sai, hay positive và negative.

Khi sử dụng 'objective': 'binary', LightGBM sẽ tối ưu hóa một hàm mất mát (loss function) phù hợp cho bài toán phân loại nhị phân. Mục tiêu của thuật toán là tìm ra một cách phân loại tốt nhất các điểm dữ liệu thành hai nhóm dựa trên các đặc trưng có sẵn.

Các thuật toán Gradient Boosting Decision Tree (GBDT) như LightGBM thường được sử dụng rộng rãi cho bài toán phân loại nhị phân, và 'binary' là một trong những giá trị thông thường được sử dụng cho tham số 'objective' trong LightGBM khi ta đang làm việc với bài toán này.

***‘metric’: ‘auc’***

Trong tham số của LightGBM, 'metric': 'auc' được sử dụng để chỉ định phương pháp đánh giá hiệu suất (performance) của mô hình dự đoán. 'auc' là viết tắt của "Area Under the ROC Curve" (diện tích dưới đường cong ROC).

ROC Curve là một biểu đồ thể hiện mối quan hệ giữa tỷ lệ True Positive (TPR) và tỷ lệ False Positive (FPR) của mô hình dự đoán tại các ngưỡng quyết định khác nhau. Diện tích dưới đường cong ROC (AUC) được sử dụng làm độ đo cho khả năng phân loại của mô hình. Giá trị AUC nằm trong khoảng từ 0 đến 1, với giá trị cao hơn cho thấy mô hình có khả năng phân loại tốt hơn.

Khi sử dụng 'metric': 'auc' trong LightGBM, mô hình sẽ được đào tạo để tối đa hóa diện tích dưới đường cong ROC. Điều này có nghĩa là LightGBM sẽ điều chỉnh các tham số của mô hình để tạo ra dự đoán tốt nhất có thể với đánh giá dựa trên AUC.

'AUC' là một phương pháp đánh giá thông thường được sử dụng trong bài toán phân loại, đặc biệt khi dữ liệu không cân bằng (imbalanced data) hoặc khi quan tâm đến hiệu suất cả hai lớp dương và âm của mô hình dự đoán.

***learning\_rate: 0.02:***

Trong tham số của LightGBM, 'learning\_rate' là một siêu tham số quan trọng. Nó xác định tỷ lệ học (learning rate) được sử dụng trong quá trình huấn luyện mô hình.

'learning\_rate' là một giá trị thực dương nằm trong khoảng từ 0 đến 1. Giá trị này quy định mức độ cập nhật của các trọng số trong mô hình trong mỗi bước tối ưu. Một learning rate nhỏ (ví dụ: 0.01, 0.001) sẽ làm cho mô hình hội tụ chậm hơn nhưng có thể đạt được kết quả tốt hơn. Ngược lại, một learning rate lớn (ví dụ: 0.1, 0.2) sẽ làm cho mô hình hội tụ nhanh hơn nhưng có thể gặp khó khăn trong việc đạt được kết quả tối ưu.

Khi thiết lập 'learning\_rate': 0.02, LightGBM sẽ sử dụng tỷ lệ học là 0.02 để cập nhật các trọng số trong quá trình huấn luyện. Đây là giá trị thường được lựa chọn dựa trên kinh nghiệm hoặc tùy chỉnh thích nghi với bài toán cụ thể. Việc điều chỉnh 'learning\_rate' cùng với các siêu tham số khác như 'num\_iterations' (số lượng bước tối ưu) là cách điều chỉnh hiệu suất và tốc độ hội tụ của mô hình LightGBM

Ý nghĩa của max\_depth: 6

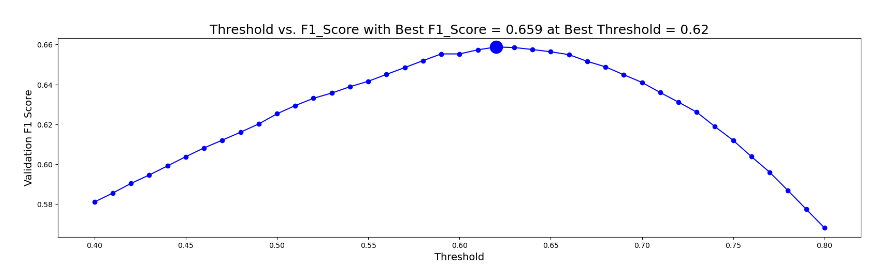
* Trong tham số của LightGBM, 'max\_depth' là một siêu tham số quan trọng. Nó xác định độ sâu tối đa của cây quyết định trong mô hình.
* 'max\_depth' là một giá trị nguyên dương, quy định độ sâu tối đa mà mỗi cây quyết định trong mô hình có thể đạt được. Khi giá trị 'max\_depth' được thiết lập, các cây sẽ ngừng phát triển khi đạt đến độ sâu này.
* Khi thiết lập 'max\_depth': 6, LightGBM sẽ xây dựng các cây quyết định có độ sâu tối đa là 6. Điều này có nghĩa là mỗi cây chỉ có thể phân chia dữ liệu tối đa đến 6 lần, và đạt đến độ sâu 6 sau khi xây dựng.
* Giá trị 'max\_depth' có thể ảnh hưởng đến khả năng mô hình khớp với dữ liệu huấn luyện và khả năng tổng quát hóa của mô hình. Một 'max\_depth' lớn có thể dẫn đến overfitting (quá khớp), trong đó mô hình học "quá nhớ" dữ liệu huấn luyện và không tổng quát hóa tốt cho dữ liệu mới. Ngược lại, 'max\_depth' nhỏ hơn có thể làm mất đi khả năng phân loại chi tiết và phức tạp của mô hình.
* Thường thì giá trị 'max\_depth' được điều chỉnh và tinh chỉnh thông qua quá trình tìm kiếm siêu tham số để đạt được hiệu suất và tốc độ học tốt nhất cho bài toán cụ thể.

Ý nghĩa của 'num\_iterations': 1000

* Trong tham số của LightGBM, 'num\_iterations' là một siêu tham số quan trọng. Nó xác định số lượng bước tối ưu (iterations) được thực hiện trong quá trình huấn luyện mô hình.

**Bước 3: Tìm ra điểm Threshold (điểm ngưỡng)**

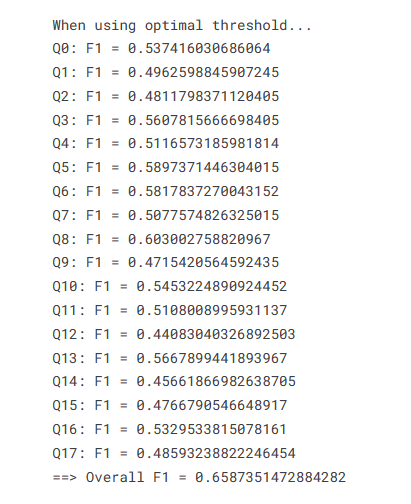
Tinh chỉnh threshold và tương quan với F1-score



Hình : Tương quan giữa Threshold và F1-Score tốt nhất

#### Kết quả

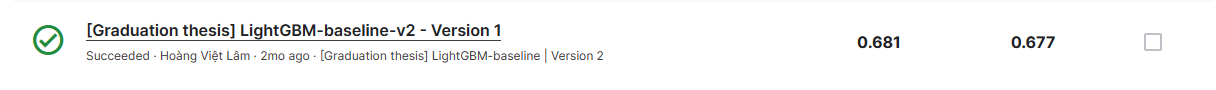
Áp dụng điểm Threshold tốt nhất vào để tính điểm F1-Score cho từng câu hỏi, như vậy ta có thể điểm F1-Score tổng cho mô hình dự đoán của chúng ta.



Hình : Điểm F1-score cho từng câu hỏi và điểm tổng LightGBM

Vậy là chúng ta đã có số điểm dự đoán là 65,58% số học sinh sẽ trả lời đúng từ lần đầu tiên trả lời câu hỏi trên tập train này.

#### Điểm số của LightGBM

****

Hình : Điểm số LightGBM

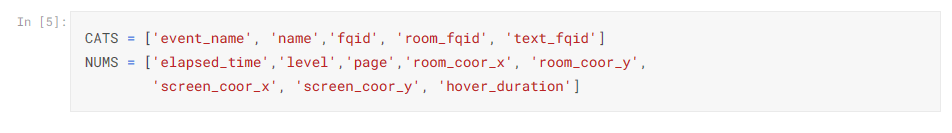
Với điểm số công khai là 0.677 và điểm số bí mật được chấm lại sau khi cuộc thi kết thúc là 0.681

### Random Forest

#### Các bước thực hiện

Với thuật toán Random Forest, ta cũng sẽ có các bước giảm thiểu bộ nhớ để tránh lỗi như Xgboost. Vì thế ta sẽ không nói lại ở phần này.

***Bước 1: Triển khai Feature Engineer***



Hình : Feature Engineer

Phân loại các biến thành 2 loại:

CATS là biến bao gồm những những cột mang giá trị thuộc một tập hợp hữu hạn các nhãn hoặc danh mục. Các biến đầu vào này không có thứ tự hay mức độ đo được và thường biểu thị các thuộc tính không liên tục.

NUMS là biến bao gồm những cột mang giá trị số (Các số có thể biểu diễn các thuộc tính định lượng như chiều cao, trọng lượng, tuổi, hoặc các đặc trưng số khác).

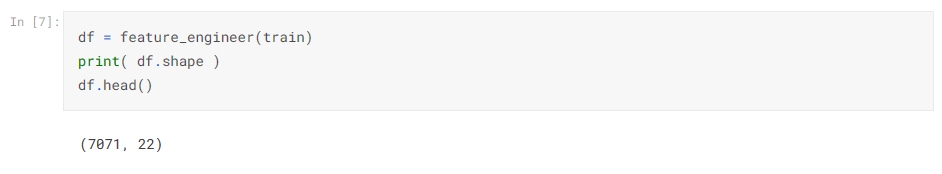


Hình : Feature Engineer in code

Việc xử lí diễn ra trong hàm này như sau:

Đối với CATS, Lấy ra những giá trị duy nhất có trong cột này, và những giá trị này được lấy theo session\_id và level\_group, nghĩa là lấy theo mã định danh của người dùng và nhóm cấp độ.

Đối với NUMS, tính giá trị trung bình và độ lệch chuẩn cho các giá trị của cột này, những giá trị này cũng được lấy theo session\_id và level group.

****

Hình : Kích thước sau khi xử lý

**Bước 2: Tinh chỉnh siêu tham số và train Random Forest model**

****

Hình : Thuộc tính và users mà mô hình sẽ train

Train model thuật toán Random Forest với 21 thuộc tính và 2357 bản ghi thông tin người dùng.

**Tinh chỉnh siêu tham số**

Các siêu tham số đang để mặc định theo thư viện sklearn.ensemble.RandomForestClassifier.

***'n\_estimators', default=100***

Số lượng cây của rừng

***'criterion', default='gini'***

Hàm đo lường chất lượng của phép chia

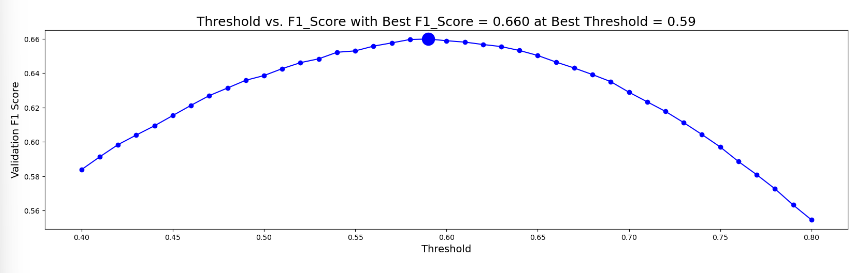
***'min\_samples\_split', default=2***

Số lượng mẫu tối thiểu cần được chia có trong 1 node

***'min\_samples\_leaf', default=1***

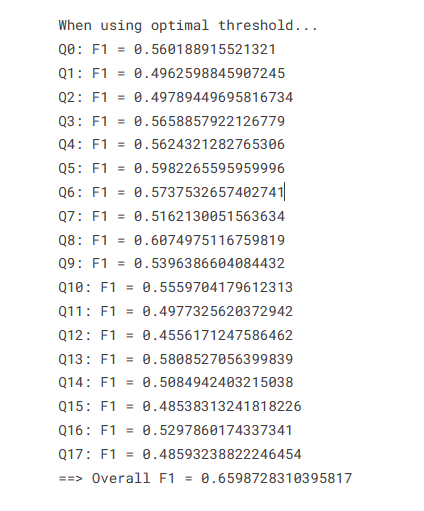
Số lượng mẫu tối thiểu cần được chia trong mỗi node lá

***Bước 3: Tìm ra điểm Threshold (điểm ngưỡng)***

Tinh chỉnh threshold và tương quan với F1-score

Hình : Tương quan giữa Threshold và F1-Score tốt nhất

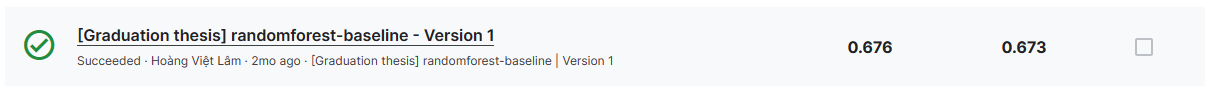
#### Kết quả

Áp dụng điểm Threshold tốt nhất vào để tính điểm F1-Score cho từng câu hỏi, như vậy 

Hình : Điểm F1-score cho từng câu hỏi và điểm tổng Random Forest

Vậy là chúng ta đã có số điểm dự đoán là 65,98% số học sinh sẽ trả lời đúng từ lần đầu tiên trả lời câu hỏi trên tập train này.

#### Điểm số của Random Forest



Hình : Điểm số của Random Forest

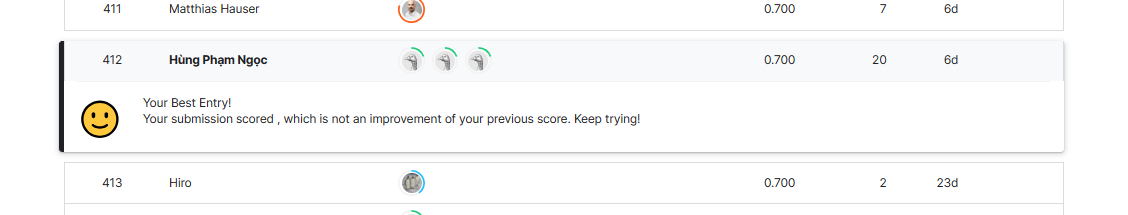
Với điểm số công khai là 0.673 và điểm số bí mật được chấm lại sau khi cuộc thi kết thúc là 0.676.

#### Kết luận

Với thuật toán Random Forest, qua thực nghiệm ta thấy thuật toán có tốc độ chạy nhanh hơn so với LightGBM. Vì Random Forest không dùng tất cả dữ liệu để huấn luyện, nó chỉ lấy ngẫu nhiên dữ liệu để huấn luyện sau đó tổng hợp lại và đưa ra kết quả cuối cùng. Tuy nhiên điểm số của LightGBM(60%) lại nhỉnh hơn Random Forest (55%) vì Random Forest là một thuật toán tương đối đơn giản, mặc khác LightGBM lại là một thuật toán kết hợp nhiều mô hình yếu thành một hình mạnh hơn nên nó có nhiều ưu thế hơn so với Random Forest.

### Điểm số và thứ hạng chung

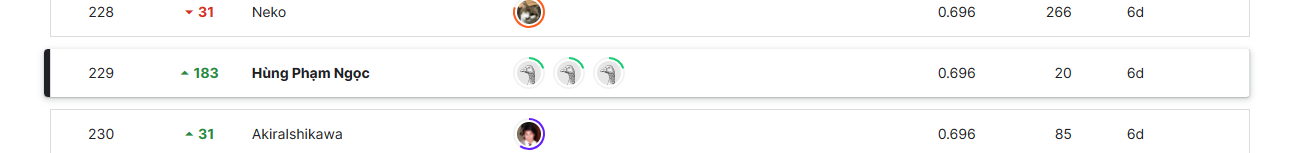
#### Công khai

****

Hình : Điểm và thứ hạng ở Public Leaderboard

Đây là thứ hạng và điểm số chung của nhóm ở bảng xếp hạng công khai, điểm số ở đây sẽ được lấy điểm cao nhất trong các bài nộp của nhóm và điểm số này được chấm ngay sau khi nộp bài.

#### Bí mật

****

Hình : Điểm và thứ hạng ở Private Leaderboard

Đây là thứ hạng và điểm số chung của nhóm ở bảng xếp hạng bí mật, điểm số ở đây sẽ được lấy điểm cao nhất trong các bài nộp của nhóm, được chấm lại một cách kỹ càng trực tiếp từ giám khảo ngay sau khi cuộc thi kết thúc. Và đây cũng là điểm số và thứ hạng để quyết định đội thi đạt giải.

# Phần kết luận

### Kết quả đạt được

***Dự đoán hiệu suất học tập***

Sử dụng các phương pháp và mô hình học máy, đã thành công trong việc dự đoán hiệu suất học tập của học sinh dựa trên dữ liệu từ game giáo dục. Mô hình này có khả năng đưa ra dự đoán về thành tích học tập của học sinh và cung cấp thông tin hữu ích để đánh giá và quản lý hiệu suất học tập.

***Phân tích yếu tố ảnh hưởng***

Thông qua việc phân tích dữ liệu từ game giáo dục, đã xác định và đánh giá các yếu tố ảnh hưởng đến hiệu suất học tập của học sinh. Các yếu tố như khả năng giải quyết vấn đề, tương tác học sinh-giảng viên, mức độ thử thách trong nhiệm vụ và sự tương tác xã hội có thể được phân tích để cung cấp thông tin giá trị cho việc cải thiện quá trình học tập.

***Hỗ trợ quyết định giảng dạy***

Kết quả nghiên cứu cung cấp thông tin hữu ích để hỗ trợ quyết định giảng dạy. Giáo viên có thể sử dụng thông tin từ hệ thống để tùy chỉnh phương pháp giảng dạy, tạo ra phản hồi cá nhân và cung cấp hướng dẫn tùy chỉnh để nâng cao hiệu quả giảng dạy và đạt được kết quả học tập tốt hơn cho học sinh.

***Cải thiện hiệu suất học tập***

Thông qua việc theo dõi và đánh giá hiệu suất học tập, đề tài đã góp phần cải thiện hiệu suất học tập của học sinh. Nhờ vào việc cung cấp phản hồi và hướng dẫn cá nhân, học sinh có thể nhận biết và nắm bắt được điểm mạnh và điểm yếu của mình, từ đó nỗ lực cải thiện kỹ năng và kiến thức.

### Ưu điểm

***Cung cấp cái nhìn tổng quan về hiệu suất học tập***

Đề tài này cho phép theo dõi và đánh giá hiệu suất học tập của học sinh thông qua game giáo dục, giúp cung cấp một cái nhìn toàn diện về khả năng và tiến bộ của học sinh trong quá trình học tập.

***Tăng cường tính tương tác và thúc đẩy học tập***

Việc sử dụng game giáo dục trong việc theo dõi và dự đoán hiệu suất học tập có thể tạo ra môi trường học tập tích cực và thúc đẩy sự tương tác giữa học sinh và nội dung giáo dục. Điều này có thể làm tăng động lực học tập và tạo ra trải nghiệm học tập thú vị.

***Cung cấp phản hồi và hướng dẫn cá nhân***

Hệ thống giúp xác định điểm mạnh và điểm yếu của học sinh trong quá trình chơi game giáo dục, từ đó cung cấp phản hồi và hướng dẫn cá nhân để học sinh có thể cải thiện kỹ năng và kiến thức của mình.

***Hỗ trợ quyết định giảng dạy***

Kết quả đề tài có thể cung cấp thông tin giá trị cho giáo viên trong việc tùy chỉnh phương pháp giảng dạy và cung cấp hướng dẫn tùy chỉnh dựa trên nhu cầu và khả năng của từng học sinh. Điều này giúp tăng cường hiệu quả giảng dạy và đáp ứng được các yêu cầu đa dạng của học sinh.

### Hạn chế

***Độ tin cậy của dữ liệu***

Một trong những thách thức đối với đề tài này là đảm bảo tính chính xác và đáng tin cậy của dữ liệu thu thập từ game giáo dục. Có thể xảy ra các sai sót hoặc nhiễu trong dữ liệu, ảnh hưởng đến độ chính xác của việc theo dõi và dự đoán hiệu suất học tập.

***Định nghĩa và đo lường hiệu suất học tập***

Việc định nghĩa và đo lường hiệu suất học tập là một vấn đề phức tạp. Mỗi game giáo dục có thể có các tiêu chí và mục tiêu học tập khác nhau, gây khó khăn trong việc xác định và đo lường hiệu suất học tập một cách chính xác và công bằng.

### Hướng phát triển

***Mở rộng lĩnh vực áp dụng***

Đề tài có thể mở rộng để áp dụng vào các lĩnh vực giáo dục khác như ngôn ngữ, toán học, khoa học, và nghệ thuật. Điều này sẽ mang lại những ứng dụng đa dạng và mở ra cơ hội phát triển trong các lĩnh vực học tập khác nhau.

***Tích hợp công nghệ học máy tiên tiến***

Để cải thiện độ chính xác và khả năng dự đoán, có thể nghiên cứu và áp dụng các phương pháp học máy tiên tiến như học sâu (deep learning) và học tăng cường (reinforcement learning) để xây dựng mô hình mạnh mẽ hơn.

***Nghiên cứu về tương tác xã hội trong game giáo dục***

Để tăng cường tính tương tác và tạo động lực học tập, có thể khám phá và nghiên cứu về tương tác xã hội trong game giáo dục, bao gồm khả năng tương tác giữa học sinh và học sinh, học sinh và giáo viên, và cộng đồng học tập trực tuyến.

***Đánh giá và phân tích kết quả học tập***

Nghiên cứu có thể tập trung vào việc đánh giá và phân tích kết quả học tập của học sinh thông qua game giáo dục, từ đó đưa ra những thông tin quan trọng và hữu ích để cải thiện quá trình học tập và nâng cao chất lượng giáo dục.

### Tài liệu tham khảo

###### [George Seif,A Beginner’s guide to XGBoost, 29/05/2019](https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7)

###### [Matt Przybyla, 4 Easy Steps for Implementing CatBoost, 20/03/2021](https://towardsdatascience.com/4-easy-steps-for-implementing-catboost-c196fd82274b)

###### [DataTechNotes, LightGBM Classification Example in Python, 23/03/2022](https://www.datatechnotes.com/2022/03/lightgbm-classification-example-in.html)

###### [Sruthi E R, Understand Random Forest Algorithms With Examples, 17/06/2021](https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest)

###### [Anshul Saini, Gradient Boosting Algorithm: A Complete Guide for Beginners, 20/09/2021](https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners)

###### [Piyush Agarwal, Dealing with Groups in Cross-Validation, 28/02/2022](https://towardsdatascience.com/cross-validate-on-data-containing-groups-correctly-ffa7173a37e6)

###### [Rohit Kundu, F1 Score in Machine Learning: Intro & Calculation, 16/12/2022](https://www.v7labs.com/blog/f1-score-guide)

###### [Jason Brownlee, A Gentle Introduction to Threshold-Moving for Imbalanced Classification, 10/02/2020](https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification)

###### [Jason Brownlee, How to Tune the Number and Size of Decision Trees with XGBoost in Python, 07/09/2016](https://machinelearningmastery.com/tune-number-size-decision-trees-xgboost-python)

###### [Susan Li, Ad Demand Forecast with Catboost & LightGBM, 31/12/2018](https://towardsdatascience.com/ad-demand-forecast-with-catboost-lightgbm-819e5073cd3e)

###### [Great Learning Team, What is Machine Learning? Defination, Types, Applications, and more, 11/05/2023](https://www.mygreatlearning.com/blog/what-is-machine-learning/)

###### [IBM Team, “What is exploratory data analysis?](https://www.ibm.com/topics/exploratory-data-analysis)

###### [IBM Team, What is machine learning?](https://www.ibm.com/topics/machine-learning)