TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**NHẬP MÔN HỌC MÁY**

**DỰ ÁN CUỐI KỲ**

*Người hướng dẫn*: **ThS Lê Anh Cường**

*Người thực hiện*: **NGUYỄN VINH ĐẠT – 52000642**

**VÕ QUANG VŨ – 51900696**

Lớp **: 20050201**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**NHẬP MÔN HỌC MÁY**

**DỰ ÁN CUỐI KỲ**

*Người hướng dẫn*: **ThS Lê Anh Cường**

*Người thực hiện*: **NGUYỄN VINH ĐẠT – 52000642**

**VÕ QUANG VŨ – 51900696**

Lớp **: 20050201**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Trước tiên, tôi xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc đến ThS. Lê Anh Cường. Thầy là người đã luôn hỗ trợ và hướng dẫn tận tình cho chúng tôi trong suốt quá trình nghiên cứu và hoàn thành bài báo cáo này.

Tiếp theo, tôi xin gửi lời cảm ơn đến khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng vì đã tạo điều kiện cho chúng tôi được học tập và nghiên cứu môn học này. Khoa đã luôn sẵn sàng chia sẻ các kiến thức bổ ích cũng như chia sẻ các kinh nghiệm tham khảo tài liệu, giúp ích không chỉ cho việc thực hiện và hoàn thành đề tài nghiên cứu mà còn giúp ích cho việc học tập và rèn luyện trong quá trình thực hành tại trường Đại học Tôn Đức Thắng nói chung.

Cuối cùng, sau khoảng thời gian học tập trên lớp tôi đã hoàn tất đề tài cuối kì nhờ vào sự hướng dẫn, giúp đỡ và những kiến thức học hỏi. Do giới hạn về mặt kiến thức và khả năng lý luận nên nhóm vẫn còn nhiều thiếu sót và hạn chế, kính mong sự chỉ dẫn và đóng góp của Quý cô cô giáo để bài Nghiên cứu của tôi được hoàn thiện hơn. Hơn nữa, nhờ những góp ý từ cô cô và các bạn hữu, tôi sẽ hoàn thành tốt hơn ở những bài nghiên cứu trong tương lai. Tôi mong Quý cô cô và các bạn bè – những người luôn quan tâm và hỗ trợ chúng tôi – luôn tràn đầy sức khỏe và vạn sự bình an.

**BÀI TẬP ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của cô Hồ Thị Linh;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 29 tháng 11 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Đạt*

*Nguyễn Vinh Đạt*

*Vũ*

*Lê Quang Vũ*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

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

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

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

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

ĐỀ BÀI

**Bài 1 (3 điểm): làm riêng từng người**

Trình bày một bài nghiên cứu, đánh giá của em về các vấn đề sau:

1. Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy;
2. Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.

**Bài 2: (7 điểm): làm chung trong nhóm**

Đưa ra một bài toán dự đoán có thể giải quyết bằng học máy (machine leanring) với các yêu cầu sau:

* Số Feature/Attribute gồm nhiều kiểu: categorial và numerical;
* Dữ liệu phải chưa được học, thực tập trên lớp và trong bài tập về nhà;

1. Phân tích thống kê trên dữ liệu, vẽ các đồ thị để hiểu bài toán, hiểu dữ liệu. Tìm hiểu các đặc trưng và đánh gía vai trò của các đặc trưng đối với mục tiêu bài toán;
2. Ứng dụng các mô hình học máy cơ bản để giải quyết bài toán, bao gồm cả các mô hình thuộc Ensemble Learing;
3. Sử dụng Feed Forward Neural Network và Reccurent Neural Network (hoặc mô thuộc loại này) để giải quyết bài toán;
4. Áp dụng các kỹ thuật tránh Overfiting trên các mô hình của câu (2) và câu (3) để giải quyết bài toán;
5. Sau khi huấn luyện xong mô hình thì muốn cải thiện độ chính xác, ta sẽ làm gì để giải quyết nó? Phân tích các trường hợp sai, đề ra giải pháp và thực hiện nó, sau đó đánh giá xem có cải tiến so với trước không.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc13476)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc26695)

[ĐỀ BÀI v](#_Toc19020)

[MỤC LỤC 1](#_Toc31974)

[CHƯƠNG 1 : BÀI TẬP 1 7](#_Toc11469)

[1.1 Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy 7](#_Toc4324)

[1.1.1 Optimizer là gì 7](#_Toc26725)

[1.1.2 Phân tích các phương pháp Optimizer trong huấn luyện mô hình học máy 7](#_Toc30978)

[1.1.2.1 Continual Learning 7](#_Toc23168)

[1.1.2.2 Gradient Descent 8](#_Toc21994)

[1.1.2.3 Stochasis Gradient Descent ( SGD) 9](#_Toc20596)

[1.1.2.4 Mini- batch Gradient Descent 11](#_Toc15716)

[1.1.2.5 Momentum 11](#_Toc29142)

[Công thức 11](#_Toc28652)

[1.1.2.6 Nesterov Accelearated Gradient ( NAG) 11](#_Toc15320)

[1.1.2.7 Adagrad 11](#_Toc31802)

[1.1.2.8 RMSprop 13](#_Toc10897)

[1.1.2.9 Adam 13](#_Toc32070)

[1.1.2.10 Adadelta 15](#_Toc31003)

[1.1.2.11 FTRL- Proximal 16](#_Toc3394)

[1.1.3 So sánh các phương pháp Optimizer trong mô hình học máy 17](#_Toc1553)

[1.1.4 Tổng kết 18](#_Toc24711)

[1.2.1 Continual Learning: 18](#_Toc19362)

[1.2.2 Test Production 19](#_Toc27531)

[1.2.3 Kết hợp Continual Learning và Test Production 19](#_Toc13840)

[CHƯƠNG 2 : BÀI TÂP 2 20](#_Toc11763)

[2.2 Phân tích thống kê trên dữ liệu, vẽ các đồ thị để hiểu bài toán, hiểu dữ liệu. Tìm hiểu các đặc trưng và đánh gía vai trò của các đặc trưng đối với mục tiêu bài toán 21](#_Toc32104)

[2.2.1 Loại hình dịch vụ 21](#_Toc29022)

[2.3 Ứng dụng các mô hình học máy cơ bản để giải quyết bài toán, bao gồm cả các mô hình thuộc Ensemble Learing 25](#_Toc5996)

[2.3.1 Giải quyết bài toán bằng Logistic Regression 25](#_Toc25763)

[2.3.2 K-Nearest Neighbors 26](#_Toc12107)

[2.3.3 Decision Tree 27](#_Toc10013)

[2.2.4 Random Forest 29](#_Toc11314)

[2.3.5 Ada Boost 30](#_Toc15234)

[2.3.6 Gradient Boosting 32](#_Toc17752)

[2.3.7 XgBoost 33](#_Toc1590)

[2.3.8 Tổng quang 34](#_Toc22383)

[2.4 Sử dụng Feed Forward Neural Network và Reccurent Neural Network (hoặc mô thuộc loại này) để giải quyết bài toán 35](#_Toc23418)

[1. Feedforward Neural Network (FNN): 35](#_Toc1008)

[2. Recurrent Neural Network (RNN): 35](#_Toc31962)

[3. Kết hợp FNN và RNN 35](#_Toc10103)

[36](#_Toc6267)

[2.5 Áp dụng các kỹ thuật tránh Overfiting trên các mô hình của câu (2) và câu (3) để giải quyết bài toán 37](#_Toc2031)

[2.6 Sau khi huấn luyện xong mô hình thì muốn cải thiện độ chính xác, ta sẽ làm gì để giải quyết nó? Phân tích các trường hợp sai, đề ra giải pháp và thực hiện nó, sau đó đánh giá xem có cải tiến so với trước không 38](#_Toc7481)

[2.6.1. Phân tích trường hợp sai: 38](#_Toc29724)

[2.6.2. Đề xuất giải pháp: 38](#_Toc9758)

[2.6.3. Thực hiện giải pháp: 39](#_Toc15178)

[2.6.4. Đánh giá hiệu suất sau cải thiện: 39](#_Toc19617)

[2.6.5. Lặp lại quy trình nếu cần thiết: 39](#_Toc28348)

[TÀI LIỆU THAM KHẢO 41](#_Toc27640)

CHƯƠNG 1 : BÀI TẬP 1

* 1. Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy
     1. Optimizer là gì

Optimizer còn được hiểu là “trình tối ưu hóa”, là một thành phần quan trọng trong quá trình training model maching leaning.Về cơ bản thuật toán tối ưu là cơ sở để xây dựng mô hình neural với mục đích học được các features của dữ liệu đầu vào, từ đó có thể tìm 1 cặp weights và bía phù hợp để tối ưu hóa model.Optimizer Đảm nhận nhiệm vụ điều chỉnh các trọng số của mô hình để giảm thiểu loss funcion (mất hàm) – một bộ đo của sự chênh lệch giữa các giá trị dự đoán vả giá trị thực tế. Khi ta đào tạo mô hình học máy, mỗi lần đưa dữ liệu vào mô hìn, mô hình sẽ tạo ra dự đoán và sau đó ta sẽ so sánh các dự đoán này với các giá trị thực tế bằng cách sử dụng các loss function và đo lường mức độ mà mô hình khi đã training so với dữ liệu thực tế.

Optimizer có nhiệm vụ điều chỉnh các tham số (hoặc trọng số) của mô hình để giảm thiểu già trị của loss function. Quá trình này thường được thực hiện thông qua việc sử dụng các phương pháp tối ưu hóa như Gradient Descent (hạ gradient) và các biến thể của nó.

* + 1. Phân tích các phương pháp Optimizer trong huấn luyện mô hình học máy

1.1.2.1 Continual Learning

Continual Learning (còn được biết đến là Incremental Learning, Life-long Learning) là một khái niệm để học một mô hình cho một lượng lớn các nhiệm vụ theo thứ tự mà không quên đi kiến thức thu được từ các nhiệm vụ trước đó, trong đó dữ liệu của các nhiệm vụ cũ không còn khả dụng trong quá trình huấn luyện các nhiệm vụ mới. Nếu không được đề cập, các điểm chuẩn ở đây là Task-CL, trong đó task-id được cung cấp trong quá trình kiểm tra.

1.1.2.2 Gradient Descent

Không giống như đạo hàm gián đoạn, Gradient Descent giải quyết vấn đề tối ưu bằng cách thực hiện tìm kiếm giá trị tối ưu qua việc di chuyển từ điểm xuất phát (nghiệm khởi tạo) theo hướng ngược với đạo hàm. Trong mỗi vòng lặp, nghiệm di chuyển gần hơn đến điểm cực tiểu.

Trong bài toán tối ưu, mục tiêu là tìm giá trị nhỏ nhất của một hàm số, thường là hàm mất mát. Khi đạo hàm của hàm mất mát bằng 0, chúng ta có thể có điểm cực tiểu hoặc cực đại. Tuy nhiên, đối với hàm số phức tạp hoặc nhiều biến, việc tính toán đạo hàm có thể trở nên phức tạp hoặc thậm chí không khả thi. Trong trường hợp này, ông thức cập nhật trong Gradient Descent trở thành công cụ hiệu quả, giúp mô phỏng hướng và kích thước của bước cập nhật mỗi lần lặp.

Công thức

*Trong đó α là learning rate, là gradient của hàm mất mát theo các tham số θ.*

Ví dụ: Cho hàm số và đạo hàm

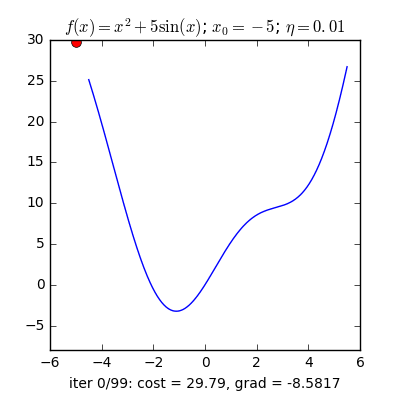
Nếu thì nghĩa là các gradian âm. Do đó, và nghiệm sẽ đi dần về phía bên phải, tiến dần đến cực tiểu.

A graph of a function

Description automatically generated

Hình: Gradian descent

Nếu thì nghĩa là các gradian dương. Do đó, và nghiệm sẽ đi dần về phía bên trái, tiến dần đến cực tiểu.



Hình : Gradian descent

Khi áp dụng cho hàm nhiều biến, công thức cập nhật mỗi tham số sẽ giống nhau. Công thức tổng quát:

không khả thi.

1.1.2.3 Stochasis Gradient Descent ( SGD)

Stochastic Gradient Descent (SGD) là một biến thể của Gradient Descent, được thiết kế để xử lý dữ liệu lớn một cách hiệu quả hơn. Việc sử dụng mẫu dữ liệu ngẫu nhiên trong mỗi vòng lặp giúp giảm độ phức tạp tính toán và tăng tốc quá trình cập nhật tham số trong quá trình đào tạo mô hình. Bằng cách này, SGD giúp tối ưu hóa mô hình một cách hiệu quả, đặc biệt là khi làm việc với tập dữ liệu lớn, giúp giảm thiểu gánh nặng tính toán và tăng tốc quá trình học..

SGD cũng giống như Gradient Descent nhằm tối ưu hóa hàm mất mát để điều chỉnh các tham số của mô hình máy học. Tuy nhiên, SGD sử dụng một phiên bản nhỏ ngẫu nhiên của dữ liệu trong mỗi vòng lặp, giảm thiểu mức độ tính toán so với Gradient Descent. Điều này giúp tăng tốc quá trình cập nhật tham số và làm cho quá trình đào tạo mô hình trở nên hiệu quả hơn, đặc biệt là khi làm việc với tập dữ liệu lớn..

Công thức cập nhật tương tự như GD:

*Với là một mẫu dữ liệu ngẫu nhiên.*

**Ví dụ:**

Nếu có 10,000 điểm dữ liệu, SGD có thể đạt được nghiệm tốt sau vài epoch, trong khi GD có thể cần đến nhiều epoch hơn để đạt được kết quả tương tự. Điều này là do SGD cập nhật trọng số sau mỗi điểm dữ liệu, giảm độ phức tạp tính toán.

A diagram of a solar system

Description automatically generated

Hình 1.1.3 1: Stochastic Dradient Descent

1.1.2.4 Mini- batch Gradient Descent

1.1.2.5 Momentum

Momentum là một thuật toán tối ưu hóa Gradient Descent nhằm giảm nguy cơ dừng ở local minimum và khuyến khích quá trình tiến tới global minimum. Nếu xem xét từ góc độ vật lý, Momentum có thể được mô tả như việc thả hai viên bi từ hai điểm khác nhau. Một viên bi với đà và vận tốc đủ lớn để vượt qua local minimum và tiến tới global minimum, giúp thuật toán cập nhật tham số mô hình một cách hiệu quả hơn.

**Công thức**

(Trong đó, ​ là tọa độ mới, ​ là tọa độ cũ, γ là hệ số momentum, learning rate learning rate là tốc độ học, và gradient gradient là đạo hàm của hàm .)

1.1.2.6 Nesterov Accelearated Gradient ( NAG)

1.1.2.7 Adagrad

Adagrad là một thuật toán tối ưu hóa tỷ lệ học (learning rate) trong quá trình huấn luyện mà không đòi hỏi sự điều chỉnh thủ công của learning rate. Thuật toán này tự động điều chỉnh tỷ lệ học tương ứng với mỗi tham số của mô hình dựa trên lịch sử của các đạo hàm trước đó.

Adagrad là lựa chọn phù hợp cho các bài toán trong trường hợp một số tham số có đạo hàm lớn và mô hình đòi hỏi learning rate tương ứng. Tuy nhiên, cần thận trọng khi sử dụng Adagrad cho các mô hình có đạo hàm biến động lớn và không ổn định.

Mỗi tham số của mô hình có một learning rate riêng, được cập nhật dựa trên bình phương của đạo hàm tương ứng ​ theo thời gian:

*(Ở đây, là phần tử (i, i) trên đường chéo của ma trận ​ tại thời điểm t.)*

*(Trong đó, η là learning rate mặc định, ϵ là một hằng số tránh lỗi để tránh chia cho 0.)*

1.1.2.8 RMSprop

RMSprop (Root Mean Square Propagation) là một thuật toán tối ưu hóa tỷ lệ học dựa trên việc điều chỉnh tỷ lệ học của Adagrad để giảm vấn đề của sự giảm dần của tốc độ học. Tuy nhiên, có khả năng thuật toán này rơi vào local minimum. Sự kết hợp giữa RMSprop và Momentum trong thuật toán Adam đã giải quyết một cách hiệu quả vấn đề này, mang lại hiệu suất tối ưu hơn trong quá trình huấn luyện.

RMSprop tính toán bình phương của đạo hàm theo thời gian và chia tỷ lệ học cho căn bậc hai của trung bình độ lớn của bình phương đạo hàm.

*(Ở đây, ​ là bình phương độ lớn của đạo hàm tại thời điểm t, η là learning rate, và ϵ là hằng số tránh lỗi.)*

*(Trong đó, η là learning rate mặc định, ϵ là một hằng số tránh lỗi để tránh chia cho 0.)*

1.1.2.9 Adam

Adam (Adaptive Moment Estimation) là một thuật toán tối ưu hóa kết hợp cả Momentum và RMSprop để tận dụng ưu điểm của cả hai.

A graph of a function

Description automatically generated

Hình 1.1.7 1 Adam

**Công thức**

Adam tính toán các giá trị động cho mean (momentum) và standard deviation (độ biến thiên) của đạo hàm theo thời gian.

*(Trong đó, ​ và ​ lần lượt là mean và standard deviation tại thời điểm t.)*

Adam cập nhật tham số theo công thức:

*(trong đó, η là learning rate mặc định, ϵ là hằng số tránh lỗi.*

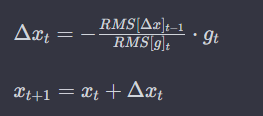
1.1.2.10 Adadelta

Adadelta là một thuật toán tối ưu hóa được sử dụng trong máy học, đặc biệt là trong quá trình huấn luyện các mạng neural. Nó là một sự mở rộng của thuật toán Adagrad, được thiết kế để giải quyết một số vấn đề của nó. Adadelta được giới thiệu bởi Matthew D. Zeiler trong bài báo "ADADELTA: An Adaptive Learning Rate Method" vào năm 2012.

Vấn đề chính của Adagrad là nó tích tụ gradient bình phương của tất cả các tham số theo thời gian, dẫn đến việc giảm nhanh chóng learning rate đối với các đặc trưng thường xuyên xuất hiện. Adadelta nhằm vượt qua vấn đề này bằng cách giới thiệu một chiến lược phức tạp hơn để thích ứng learning rates.

Ý tưởng chính của Adadelta là sử dụng một trung bình chạy của gradient bình phương trong quá khứ để tính toán learning rate động cho mỗi tham số. Thuật toán duy trì hai biến trạng thái cho mỗi tham số: \(E[g^2]\_t\), là một trung bình chạy mức giảm số bình phương của gradient, và \(E[\Delta x^2]\_t\), là một trung bình chạy mức giảm số bình phương của cập nhật tham số.

Công thức cập nhật cho tham số \(x\) tại bước thời gian \(t\) trong Adadelta được cho bởi:



Trong đó:

- \(g\_t\) là gradient tại bước thời gian \(t\),

- \(RMS[g]\_t\) là căn bậc hai trung bình của gradient trong quá khứ,

- \(RMS[\Delta x]\_{t-1}\) là căn bậc hai trung bình của cập nhật tham số trong quá khứ.

Thuật toán tránh được việc cần phải chọn một learning rate ban đầu và điều chỉnh learning rates trên cơ sở từng tham số. Chiến lược thích ứng này giúp Adadelta đối mặt với vấn đề giảm learning rate gặp phải trong Adagrad.

Adadelta thường được sử dụng kết hợp với các mô hình học sâu và có thể là một thuật toán tối ưu hóa hiệu quả, đặc biệt là khi xử lý dữ liệu thưa và nhiễu.

1.1.2.11 FTRL- Proximal

Thuật toán FTRL-Proximal (Follow-the-Regularized-Leader Proximal) là một thuật toán tối ưu hóa được sử dụng chủ yếu trong bài toán tối ưu hóa đối với các mô hình dự đoán, đặc biệt là trong bối cảnh học máy với các mô hình tuyến tính như logistic regression hoặc linear regression. FTRL-Proximal thường được sử dụng khi có sự thưa thớt trong dữ liệu, tức là chỉ một số lượng nhỏ các đặc trưng (features) quan trọng và cần được tối ưu hóa.

- Dùng các \(w\_i\) để cập nhật \(w\): \(w\_i\) được tính toán như đã mô tả ở trên.

Thuật toán FTRL-Proximal là một sự kết hợp của FTRL (Follow-the-Regularized-Leader) và thuật toán Proximal Gradient Descent, nơi Proximal là để áp dụng một hàm rút ngắn (proximal function) lên các tham số. Thuật toán này đặc biệt hiệu quả trong việc xử lý dữ liệu thưa thớt và giảm hiện tượng overfitting. Các tham số như \(\lambda\) và \(\alpha\) thường được điều chỉnh để đạt được hiệu suất tối ưu cho từng bài toán cụ thể.

1.1.3 So sánh các phương pháp Optimizer trong mô hình học máy

|  |  |  |
| --- | --- | --- |
| Phương pháp | Ưu điểm | Nhược đi |
| Gradient Descent ( GD) | - Dễ hiểu và triển khai.  - Hoạt động tốt trên các mô hình đơn giản.  - Giải quyết vấn đề tối ưu model neural network | - Có thể chậm và không hiệu quả trên các mô hình lớn hoặc phức tạp.  - Phụ thuộc vào điểm khởi tạo ban đầu và learning rate.  - Tốc độ học quá lớn hoặc quá nhỏ có thể ảnh hưởng đến quá trình training. |
| Stochasis Gradient Descent ( SGD) | - Phù hợp cho dữ liệu lớn và phức tạp.  - GIảm độ độ phức tạp tính toán.  - Cập nhập trọng số sau mỗi mẫu dữ liệu, thích hợp cho dữ liệu lớn.  - Cập nhập tham số nhanh chóng. | - Có thể dao động nhiều hơn do tính ngẫu nhiên.  - Đô chính xác có thể bị giảm vì sự ngẫu nhiên trong việc chọn mẫu. |
| Mini- batch Gradient Descent | - Kết hợp lợi ích của GD và SGD  - Phù hợp với nhiều loại dữ liệu | - Cần điều chỉnh kích thước mini- batch |
| Momentum | - Vượt qua Local Mometum  - Giảm độ dao động và giúp vượt qua các đồng lõi  - Thuật toán giảm thiểu đảo hàm và giúp cập nhật nhanh chóng | - Cần chọn thêm hyperparameter. |
| Nesterov Accelearated Gradient ( NAG) | - Cải thiện trên Momentum bằng cách ước lượng gradient trước khi thực hiện cập nhật | - Cần điều chỉnh thêm hyperparameter |
| Adagrad | - Điều chỉnh tỷ học tự động cho từng tham số  - Tự điều chỉnh learning rate.  - Không cần điều chỉnh thủ công giảm công sức và độ rủi ro. | - Tăng dần learning rate có thể dẫn đến việc giảm độ chính xác. |
| RMSprop | - Giải quyết vấn đề tốc độ học giảm dần.  - Cải thiện Adagrad bằng cách giảm độ chậm của learning rate | - Cân điều chỉnh thêm hyperparameter.  - Khả năng cao rơi vào Local Minimum  - Có thể dẫn đến tốc độ học cực kỳ nhỏ và làm cho quá trình training trở nên đóng băng |
| Adam | - Kết hợp Momemtum và RMSprop, hiệu quả trên nhiều loại dữ liệu. | - Cần điều chỉnh thêm hyperparameter.  - Phưc tạp hơn so với ác thuật toán đơn giản như GD và SGD |
| Adadelta | - Không yêu cầu thiết lập learning rate ban đầu | - Cần điều chỉnh hyperparameter |
| FTRL- Proximal | - Phù hợp cho các mô hình với lượng lớn các tính năng | - Cần điều chỉnh hyper pameter |

1.1.4 Tổng kết

Trong quá trình xây dựng giải pháp học máy, việc tích hợp cả hai khái niệm Continual Learning và Test Production có thể đồng thời giúp mô hình duy trì khả năng học liên tục và đảm bảo hiệu suất ổn định và đánh giá chính xác trên các tập dữ liệu thử nghiệm. Test Production đóng vai trò quan trọng trong việc chuẩn hóa quá trình đánh giá mô hình khi áp dụng continual learning bằng cách tạo ra các tập dữ liệu thử nghiệm đa dạng. Đồng thời, cả Continual Learning và Test Production đều đóng góp vào việc đảm bảo rằng mô hình không chỉ học được từ dữ liệu mới mà còn duy trì hiệu suất trên cả dạng dữ liệu đã biết và mới.Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.

1.2.1 Continual Learning:

Continual Learning là một phương pháp trong học máy mà mô hình học được liên tục cập nhật và nâng cấp khi có thêm dữ liệu mới. Thay vì huấn luyện một mô hình một lần và sử dụng nó với tất cả dữ liệu có sẵn, continual learning giúp mô hình học từ các tập từ dữ liệu mới liên tục mà không quên đi kiến thức đã học trước đó.

Các thách thức của continual learning bao gồm:

+ Quên thông tin cũ: Khi mô hình học từ dữ liệu mới, có nguy cơ mất đi kiến thức đã học trước đó.

+ Hiệu suất giảm đột ngột: Mô hình có thể đối mặt với hiện tượng “castatrophic forgetting” nghĩa là nó quên đi hiệu suất trước đó khi học thêm một nhiệm vụ mới.

+ Đối mặt với phân phối dữ liệu thay đổi: Dữ liệu mới có thể phân phối khác biệt so với dữ liệu cũ, điều này đặt ra thách thức về sự thích ứng của mô hình

1.2.2 Test Production

Test Production là quá trình tạo ra các bài kiểm tra (test cases) để đánh giá hiệu suất của mô hình học máy. Khi xây dựng giải pháp học máy, việc đảm bảo rằng mô hình hoạt động hiệu quả và có thể tổng quát hóa cho các tính huống mới là quan trọng. Test Production giúp đảm bảo rằng mô hình chỉ học thuật toán mà còn có khả năng áp dụng nó cho các dữ liệu thực tế.

Các bước quan trọng trong quá trình Test Production bao gồm:

+ Xác định các bài kiểm tra đại diện: Chọn ra các tình huống mà mô hình cần phải xử lý, đảm bảo rằng chúng phản ánh đa dạng của dữ liệu thực tế

+ Tích hợp bài kiểm tra vào quy trình phát triển: Bài kiểm tra nên được tích hợp sớm trong quy trình phát triển để kiểm soát và đảm bảo chất lượng của mô hình

+ Thực hiện đánh giá định kỳ: Đánh giá hiệu suất của mô hình thường xuyên để đảm bảo rằng nó vẫn hoạt động tốt trên các bài kiểm tra mới và cũ.

1.2.3 Kết hợp Continual Learning và Test Production

Khi xây dựng giải pháp học máy, kết hợp continual learning và test production là quan trọng để đảm bảo mô hình không chỉ linh hoạt mà con giữ được hiệu suất cao qua thời gian. Bằng cách liên tục cập nhật mô hình với dữ liệu mới và kiểm ra hiệu suất của nó trên bài kiểm tra đại diện, ta có thể đảm bảo tính tổng quát và ổn định của giải pháp học máy.

CHƯƠNG 2 : BÀI TÂP 2

2.1Giới thiệu đề tài

Du lịch luôn là vấn đề được đặt ra trong nhiều năm gần đây và cả trong tương lai, nhất là trong việc kinh doanh khách sạn trong mùa du lịch.

Để có sự chuẩn bị kịp thời và đạt được lợi nhuận cao, dự án của chúng em giúp đưa ra được giải pháp và giá cả phù hợp với từng thời điểm.

2.2 Đặc tả đề tài

Đề tài Phân tích nhu cầu đặt khách sạn của chúng em dựa trên bộ dữ liệu của hai khách sạn với hai loại hình khác nhau cùng với các thông tin về thời gian, dịch vụ, lượng khách và một số thông tin khác để đưa ra giải pháp kinh doanh khách sạn hợp lý.

Đề tài sẽ phân tích và dựa trên các mô hình học máy để dự đoán xu hướng kinh doanh dành cho năm/tháng tiếp theo, từ đó đưa ra được chính sách kinh doanh phù hợp và giá cả hợp lý có thể mang về lợi nhuận cao.

2.3 Mục tiêu đề tài và ý nghĩa thực tiễn

Mục tiêu của đề tài là phân tích các đặc trưng trong nhu cầu đặt phòng khách sạn, dự đoán các tiềm năng phát triển của khách sạn trong tương lai, từ đó đưa ra chính sách phù hợp cho việc phát triển kinh doanh.

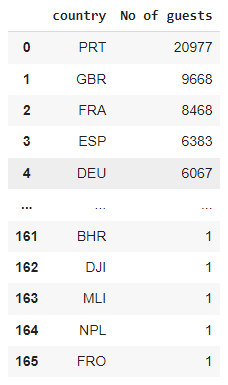
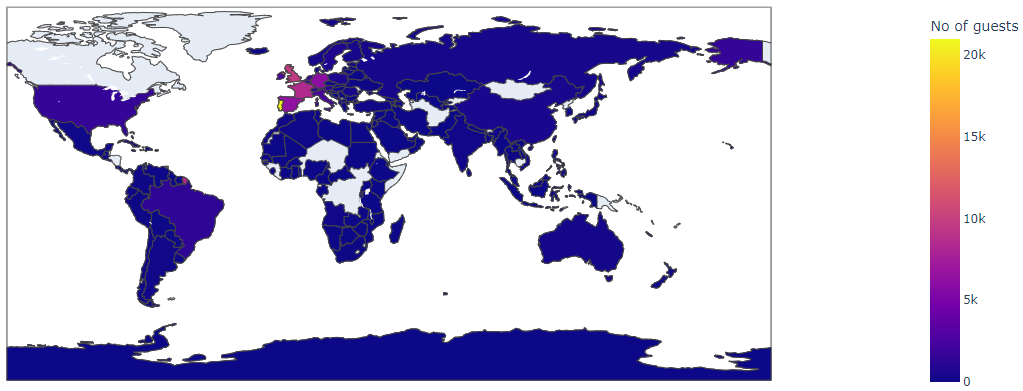
Từ việc phân tích và dự đoán, ta có thể đưa ra các loại hình phục vụ phù hợp để thu hút khách hàng cũng như đưa ra chi phí hợp lý theo từng mùa, từng tháng để nâng cao doanh thu.

2.2 Phân tích thống kê trên dữ liệu, vẽ các đồ thị để hiểu bài toán, hiểu dữ liệu. Tìm hiểu các đặc trưng và đánh gía vai trò của các đặc trưng đối với mục tiêu bài toán

2.2.1 Loại hình dịch vụ

Dựa trên bộ dữ liệu đã được lọc, ta có thể phân tích để đưa ra dịch vụ phù hợp cũng như giá cả hợp lý vào từng tháng.

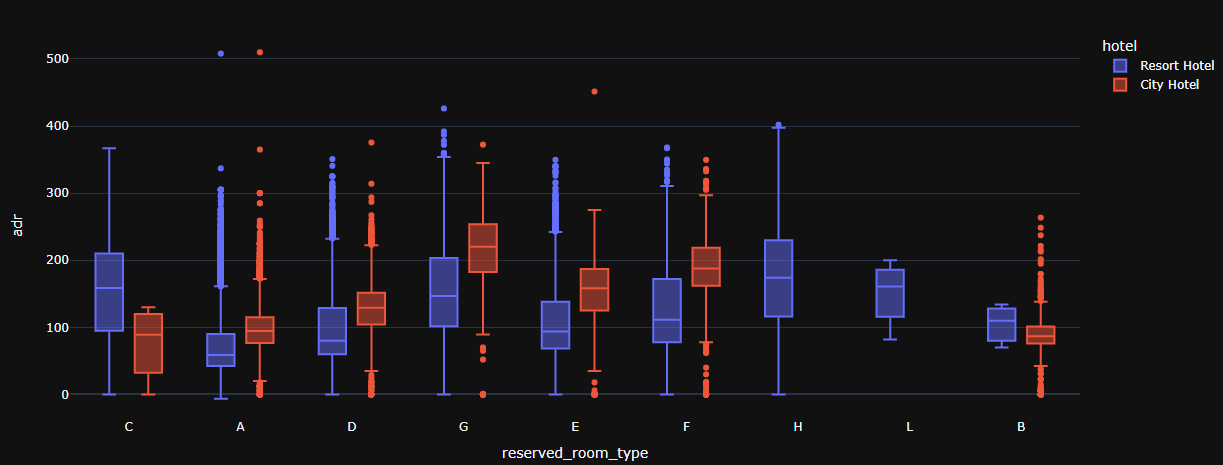
Qua hình bên dưới, ta có thể thấy được khách hàng chủ yếu đến từ Bồ Đào Nha và một số nước khác ở Châu Âu. Từ đó có thể đưa ra các loại hình và phong cách phục vụ phù hợp với thị hiếu của khách hàng.



2.2.2 Chi phí và lợi nhuận

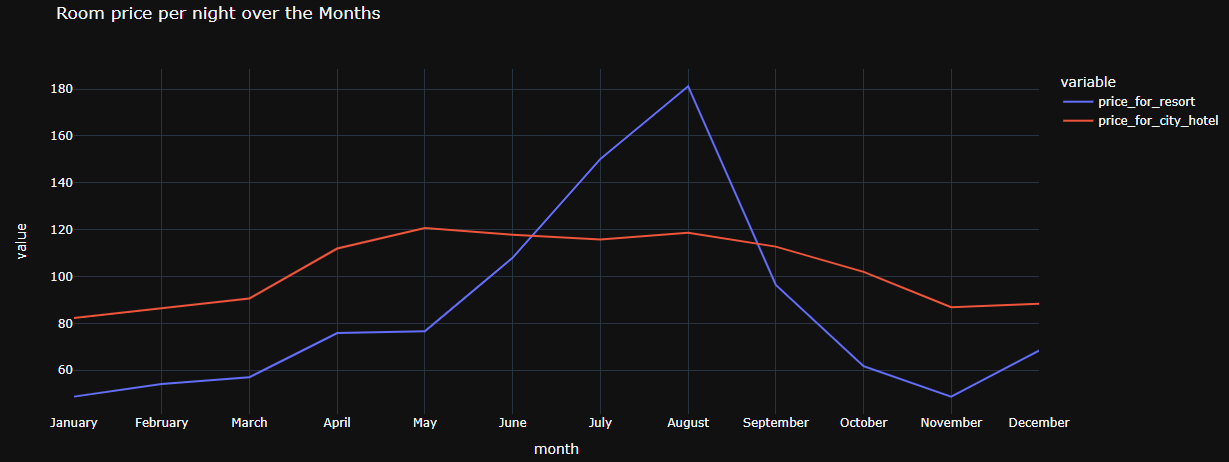
Cả hai khách sạn đều có loại phòng khác nhau và cách sắp xếp bữa ăn khác nhau, lượng khách trong các mùa khác nhau cũng rất quan trọng nên giá cả cũng khác nhau rất nhiều.

Thông qua độ lệch chuẩn về tỉ giá cũng như các loại phòng, ta thấy giá độ lệch giữa giá trung bình của từng loại phòng.



Dựa theo tỉ giá, ta có thể đưa ra giá theo từng tháng đối với hai khách sạn như sau

Dựa vào bảng giá theo từng tháng, qua biểu đồ dưới đây, ta có thể thấy được giá ở khách sạn nghỉ dưỡng cao hơn nhiều vào khoảng từ tháng 6 đến tháng 9 và giá khách sạn ở thành phố ít thay đổi hơn.

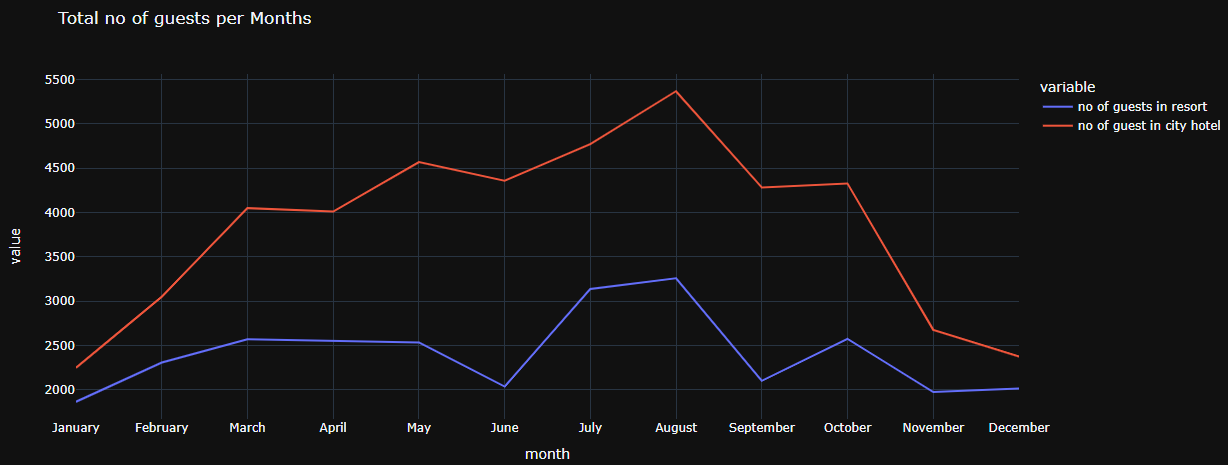


2.2.3 Lượng khách mỗi tháng

Tùy vào lượng khách đến vào mỗi tháng, khách sạn có thể đưa ra các chính sách phù hợp như thuê nhân viên bán thời gian vào những lúc đông khách để sắp xếp phục vụ, dịch vụ tốt hơn, từ đó thu được nhiều lợi nhuận hơn.

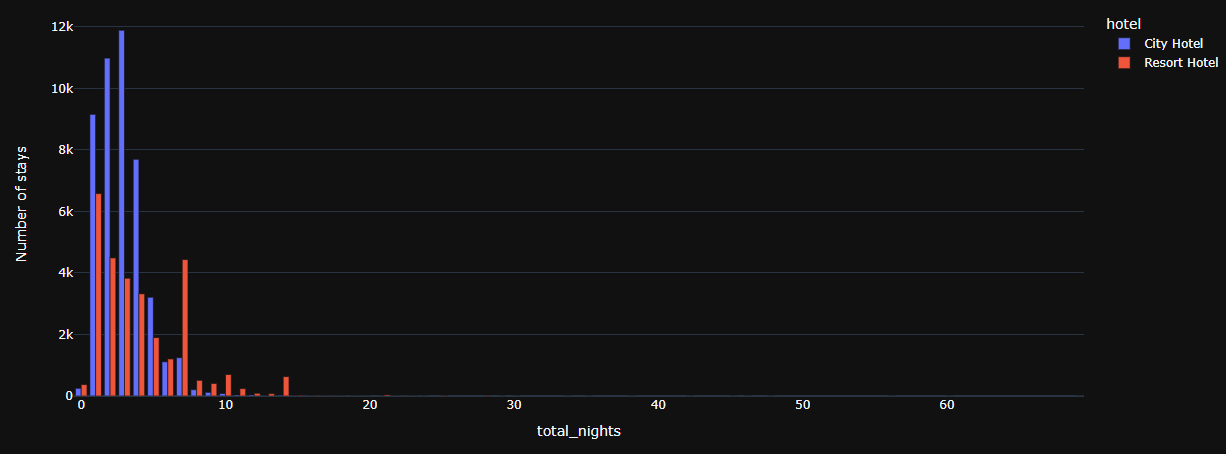
Qua số liệu và biểu đồ, ta có thể thấy khách sạn thành phố có nhiều khách hơn vào khoảng tháng 6 đến tháng 9, cũng là khi giá cao nhất. Lượng khách đến nghỉ tại khách sạn Resort tăng từ khoảng tháng 6 đến tháng 9, đây cũng là thời điểm giá cao nhất. Vào khoảng tháng 11 đến tháng 2 cũng là lúc khách đến ít nhất ở cả hai khách sạn, đây cũng là thời điểm có giá thấp trong năm.





Lượng khách ở lại qua đêm cũng ảnh hưởng đến chi phí phòng cũng như chi phí dịch vụ có thể cung cấp cho khách hàng như phục vụ bữa sáng, nơi đỗ xe,…

Thông qua biểu đồ dưới, ta thấy được khách hàng có xu hướng ở lại qua đêm trong khoảng từ 2 đến 5 ngày, từ đó có thể đưa ra các loại hình phục vụ để thu hút khách hàng cũng như tang them lợi nhuận.



2.2.4 Kết quả thu được

Dựa trên các dữ liệu trên, ta thu được bảng về lượng khách và giá cả trung bình của từng khách sạn theo mỗi tháng như sau:

Lượng khách và giá theo từng tháng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **month** | **no of guests in resort** | **no of guest in city hotel** | **price\_for\_resort** | **price\_for\_city\_hotel** |
| January | 1866 | 2249 | 48.7611254 | 82.33098266 |
| February | 2308 | 3051 | 54.14747834 | 86.52006227 |
| March | 2571 | 4049 | 57.05683781 | 90.65853297 |
| April | 2550 | 4010 | 75.86781569 | 111.9622668 |
| May | 2535 | 4568 | 76.65755819 | 120.6698271 |
| June | 2037 | 4358 | 107.9748503 | 117.8743598 |
| July | 3137 | 4770 | 150.1225279 | 115.8180189 |
| August | 3257 | 5367 | 181.2058919 | 118.6745985 |
| September | 2102 | 4283 | 96.41686013 | 112.7765818 |
| October | 2575 | 4326 | 61.77544854 | 102.0046718 |
| November | 1975 | 2676 | 48.70628861 | 86.94659193 |
| December | 2014 | 2377 | 68.41010427 | 88.40185528 |

2.3 Ứng dụng các mô hình học máy cơ bản để giải quyết bài toán, bao gồm cả các mô hình thuộc Ensemble Learing

2.3.1 Giải quyết bài toán bằng Logistic Regression

A screenshot of a computer

Description automatically generated

Hình 2.2.1 1: Logistic Regression

Đánh Giá Tổng Quan: Mô hình Logistic Regression đã đạt được độ chính xác (Accuracy) là 80.97%, tức là mô hình đúng dự đoán trên hơn 80% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 7610
* True Negative (TN): 21346
* False Positive (FP): 1305
* False Negative (FN): 5502

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 0.85
* Đối với Negative (0): 0.80

Recall:

* Đối với Positive (1): 0.58
* Đối với Negative (0): 0.94

F1-Score:

* Đối với Positive (1): 0.69
* Đối với Negative (0): 0.86

2.3.2 K-Nearest Neighbors

A screenshot of a computer

Description automatically generated

Hình 2.2.3: KNN

Đánh Giá Tổng Quan: Mô hình KNN đã đạt được độ chính xác (Accuracy) là 89.24%, tức là mô hình đúng dự đoán trên hơn 89% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 10076
* True Negative (TN): 21839
* False Positive (FP): 812
* False Negative (FN): 3036

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.88

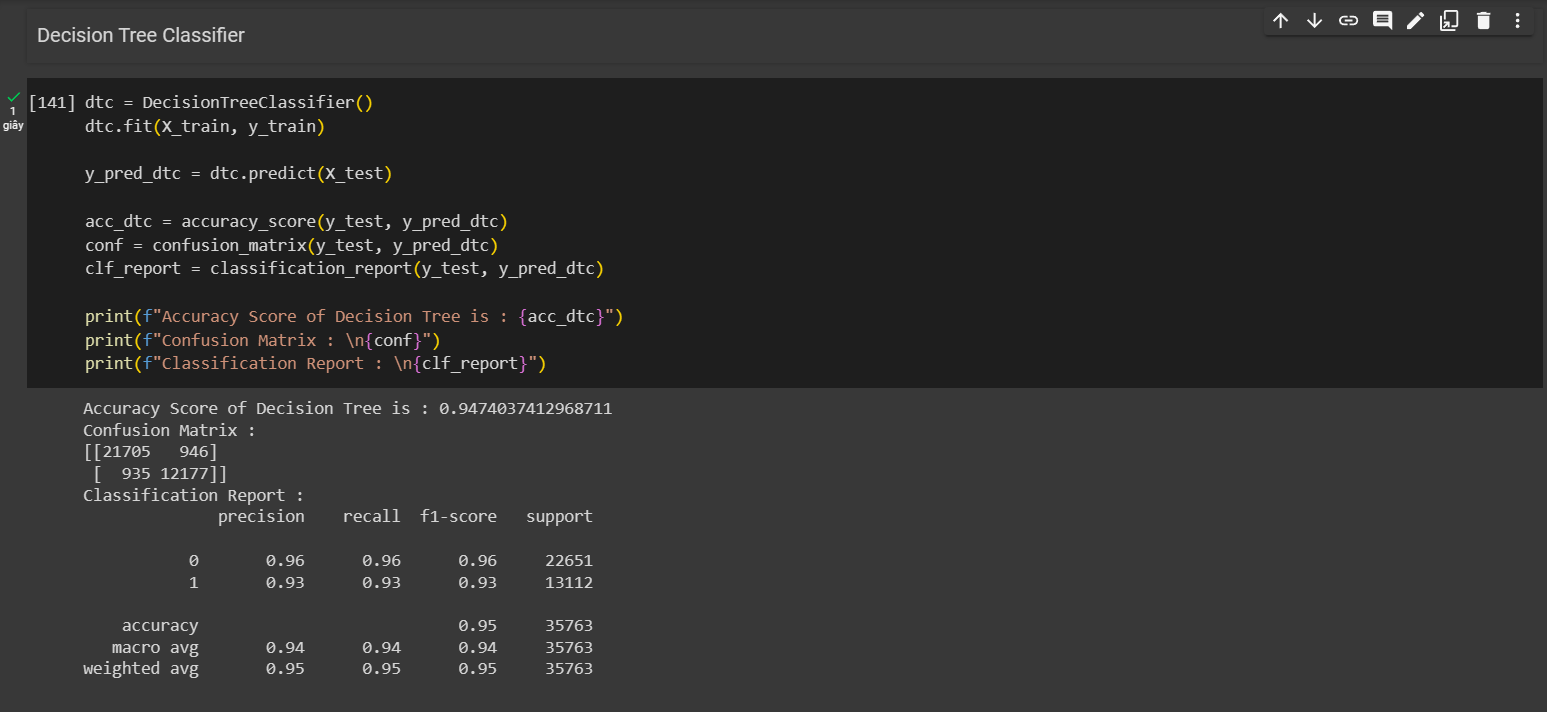
Recall:

* Đối với Positive (1): 0.77
* Đối với Negative (0): 0.96

F1-Score:

* Đối với Positive (1): 0.84
* Đối với Negative (0): 0.92

2.3.3 Decision Tree



Hình 2.2.3 : Decision Tree

Đánh Giá Tổng Quan: Mô hình Decision Tree đã đạt được độ chính xác (Accuracy) là 94.74%, tức là mô hình đúng dự đoán trên hơn 94% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 12177
* True Negative (TN): 21705
* False Positive (FP): 946
* False Negative (FN): 935

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.96

Recall:

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.96

F1-Score:

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.96

2.2.4 Random Forest

A screenshot of a computer

Description automatically generated

Hình 2.2 4: Random Forest

Đánh Giá Tổng Quan: Mô hình Random Forest đã đạt được độ chính xác (Accuracy) là 95.49%, tức là mô hình đúng dự đoán trên hơn 95% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 11683
* True Negative (TN): 22468
* False Positive (FP): 183
* False Negative (FN): 1429

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 0.98
* Đối với Negative (0): 0.94

Recall:

* Đối với Positive (1): 0.89
* Đối với Negative (0): 0.99

F1-Score:

* Đối với Positive (1): 0.94
* Đối với Negative (0): 0.97

2.3.5 Ada Boost



Hình 2.2 5: Ada Boost

Đánh Giá Tổng Quan: Mô hình Ada Boost Classifier đã đạt được độ chính xác (Accuracy) là 94.71%, tức là mô hình đúng dự đoán trên hơn 94% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 12179
* True Negative (TN): 21692
* False Positive (FP): 959
* False Negative (FN): 933

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.96

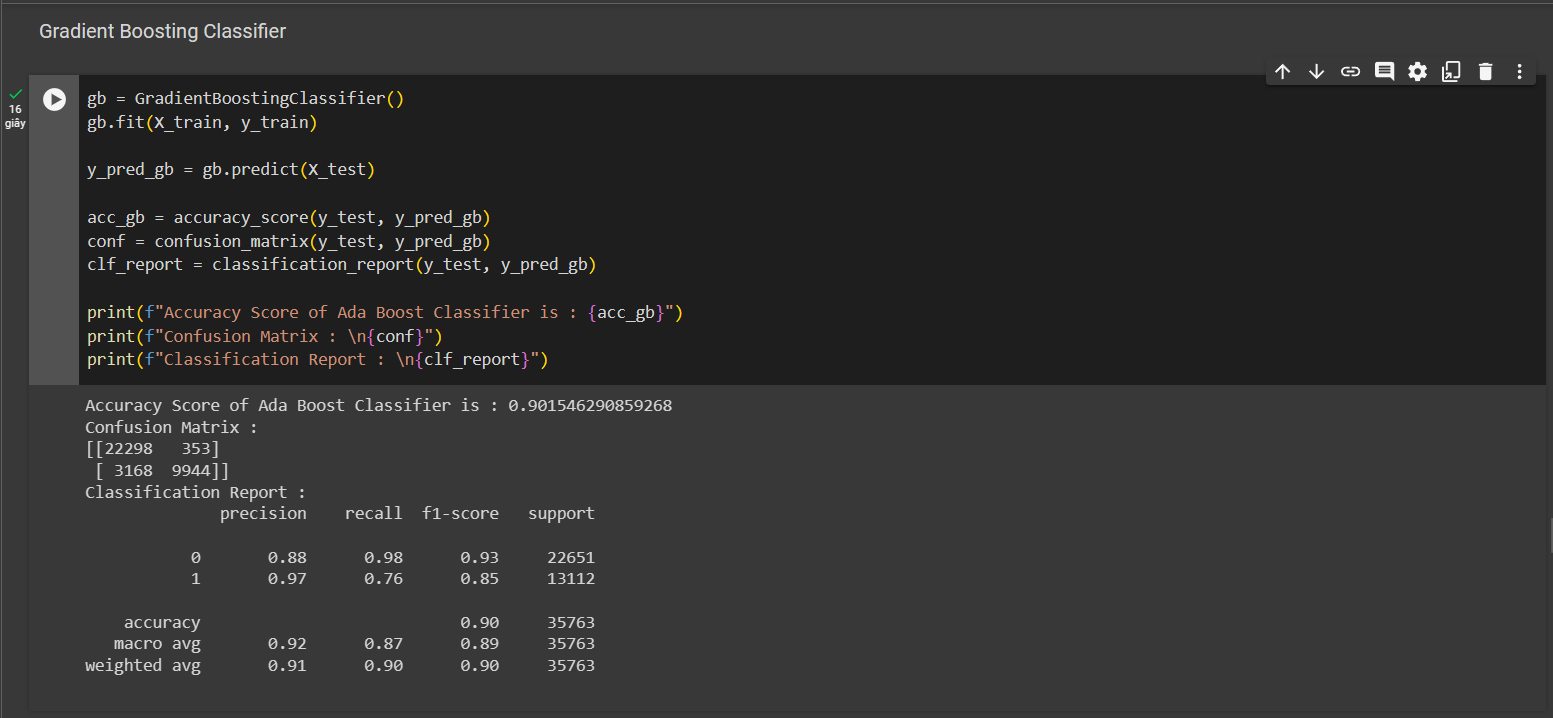
Recall:

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.96

F1-Score :

* Đối với Positive (1): 0.93
* Đối với Negative (0): 0.96

2.3.6 Gradient Boosting



Hình 2.2 6: Gradient Boosting

Đánh Giá Tổng Quan: Mô hình Ada Boost Classifier đã đạt được độ chính xác (Accuracy) là 90.15%, tức là mô hình đúng dự đoán trên hơn 90% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 9944
* True Negative (TN): 22298
* False Positive (FP): 353
* False Negative (FN): 3168

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 0.97
* Đối với Negative (0): 0.88

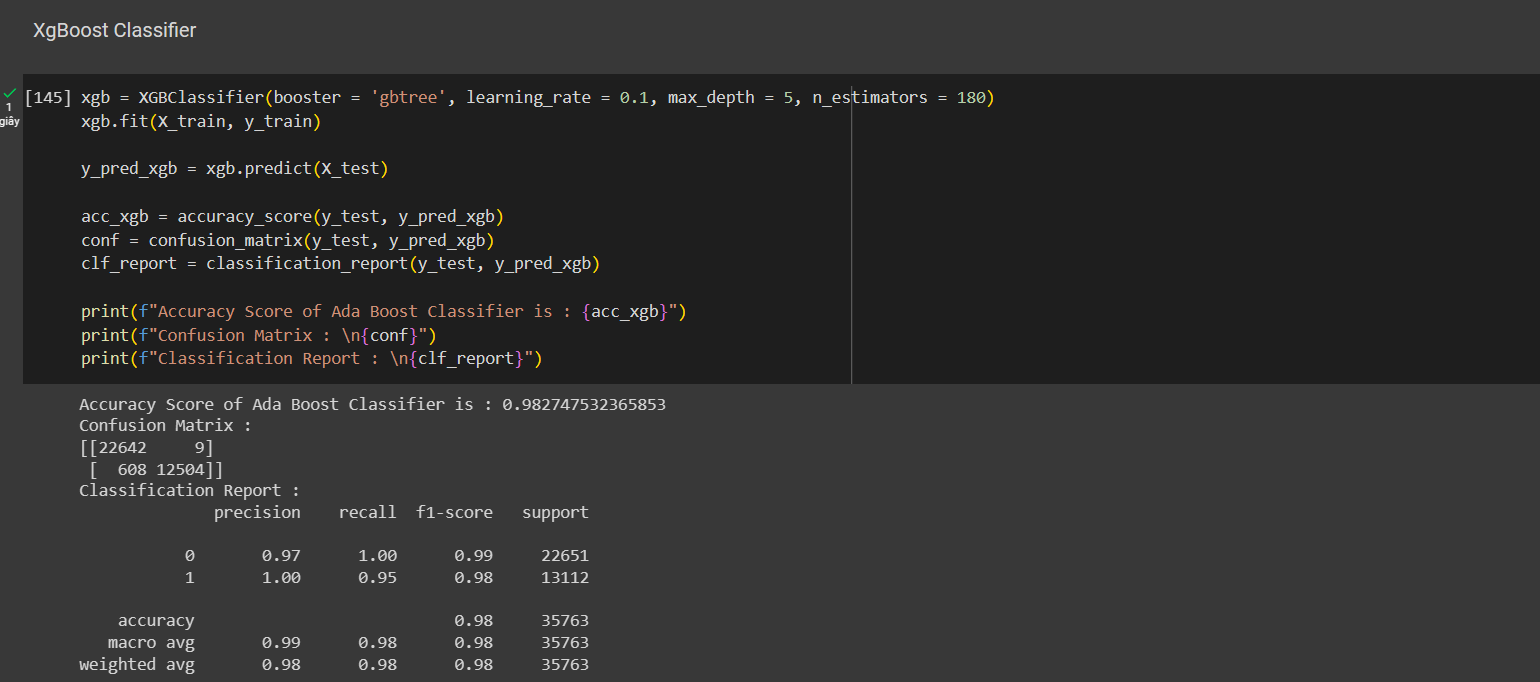
Recal:

* Đối với Positive (1): 0.76
* Đối với Negative (0): 0.98

F1-Score:

* Đối với Positive (1): 0.85
* Đối với Negative (0): 0.93

2.3.7 XgBoost



Hình 2.2 7: XgBoost

Đánh Giá Tổng Quan: Mô hình Ada Boost Classifier đã đạt được độ chính xác (Accuracy) là 98.27%, tức là mô hình đúng dự đoán trên hơn 98% tổng số trường hợp.

**Confusion Matrix:**

* True Positive (TP): 12504
* True Negative (TN): 22642
* False Positive (FP): 9
* False Negative (FN): 608

**Báo Cáo Phân Loại (Classification Report):**

Precision:

* Đối với Positive (1): 1.00
* Đối với Negative (0): 0.97

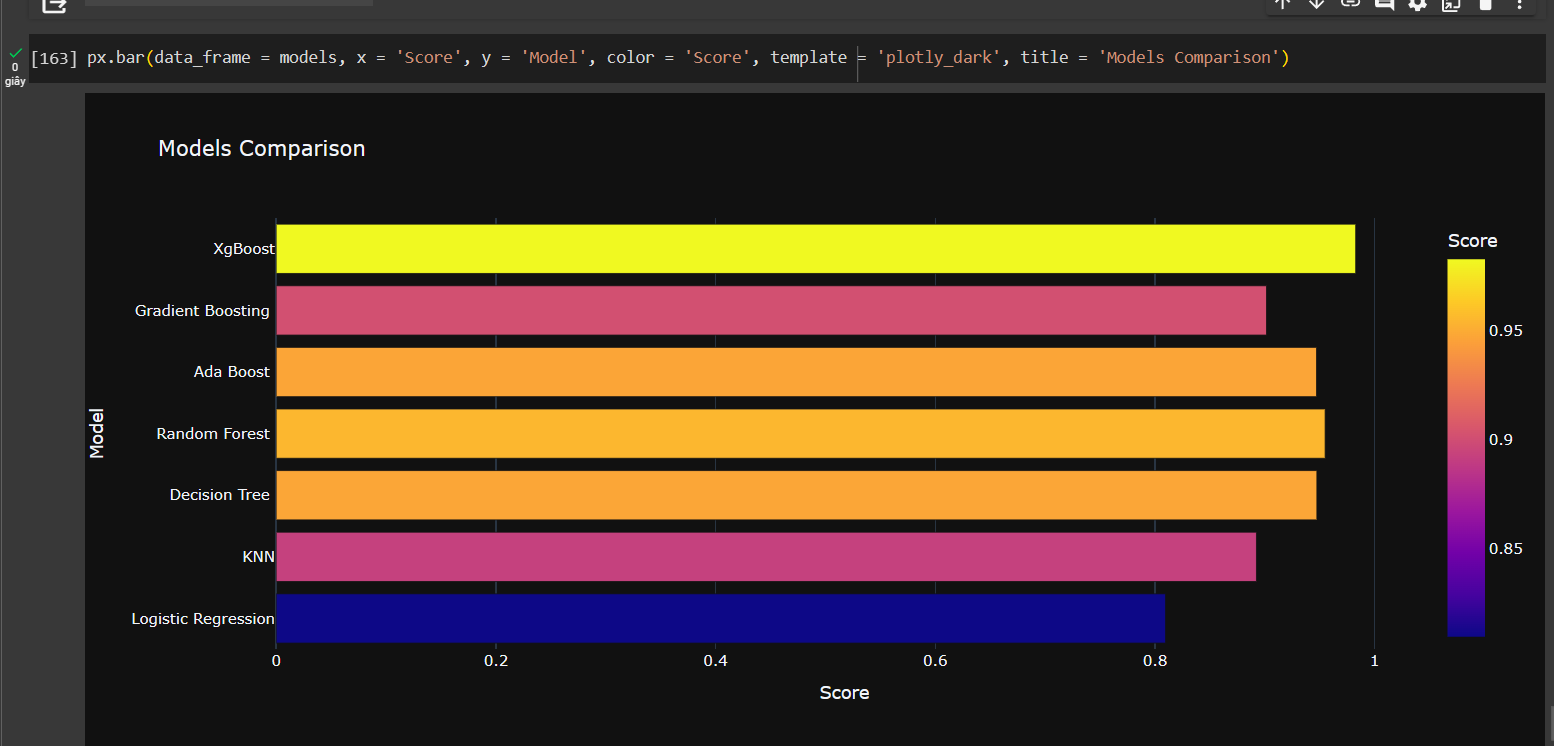
Recall:

* Đối với Positive (1): 0.95
* Đối với Negative (0): 1.00

F1-Score:

* Đối với Positive (1): 0.98
* Đối với Negative (0): 0.99

2.3.8 Tổng quang



Hình 2.2 7: XgBoost

Trong bài toán “dự đoán đặt và hủy đặt phòng khách sạn”, độ chính xác (Accuracy) của các mô hình đã cho thấy những kết quả như sau:

* XgBoost (98.27%)
* Random Forest (95.49%)
* Decision Tree (94.74%):
* Ada Boost (94.71%)
* K-Nearest Neighbors (89.24%)
* Logistic Regression (80.97%)

Kết quả cho thấy XgBoost là mô hình có độ chính xác cao nhất, dự đoán chính xác trên hơn 98% tổng số trường hợp.

2.4 Sử dụng Feed Forward Neural Network và Reccurent Neural Network (hoặc mô thuộc loại này) để giải quyết bài toán

Bài toán đặt phòng khách sạn (hotel booking) có thể được giải quyết bằng cả Feedforward Neural Network (FNN) và Recurrent Neural Network (RNN) tùy thuộc vào đặc điểm và yêu cầu cụ thể của bài toán. Dưới đây là một số cách tiếp cận:

1. Feedforward Neural Network (FNN):

- \*\*Mục tiêu:\*\* Dự đoán xác suất hoặc quyết định việc đặt phòng dựa trên đầu vào như thông tin người dùng, thông tin về khách sạn, giá cả, thời gian đặt phòng, và các yếu tố khác.

- Kiến trúc mạng: Mạng FNN có thể được xây dựng với các tầng ẩn để học các đặc trưng phức tạp từ dữ liệu đầu vào. Các đặc trưng này sau đó được sử dụng để đưa ra quyết định đặt phòng.

2. Recurrent Neural Network (RNN):

- Mục tiêu: Xử lý dữ liệu dạng chuỗi và mối quan hệ thời gian trong quá trình đặt phòng. Điều này có thể hữu ích khi cần phải lấy ý kiến người dùng từ các đặt phòng trước đó hoặc dự đoán xu hướng đặt phòng theo thời gian.

- Kiến trúc mạng: Sử dụng các tầng RNN để mô hình hóa thông tin liên tục qua các bước thời gian. Các trạng thái ẩn của RNN có thể giữ lại thông tin về các đặt phòng trước đó và đưa ra dự đoán cho các đặt phòng tiếp theo.

3. Kết hợp FNN và RNN

- Mục tiêu:Kết hợp sức mạnh của cả hai mô hình để đạt được hiệu suất tốt hơn. FNN có thể xử lý thông tin tĩnh và đặc trưng từ dữ liệu, trong khi RNN có thể xử lý thông tin dạng chuỗi và quan hệ thời gian.

- Kiến trúc mạng: Sử dụng các tầng FNN cho thông tin tĩnh và các tầng RNN để xử lý thông tin dạng chuỗi hoặc theo thời gian.

Lựa chọn giữa FNN và RNN phụ thuộc vào đặc tính cụ thể của dữ liệu và yêu cầu của bài toán. Ngoài ra, có thể cần thực hiện tiền xử lý dữ liệu để chúng phù hợp với mô hình được chọn.

2.5 Áp dụng các kỹ thuật tránh Overfiting trên các mô hình của câu (2) và câu (3) để giải quyết bài toán

2.6 Sau khi huấn luyện xong mô hình thì muốn cải thiện độ chính xác, ta sẽ làm gì để giải quyết nó? Phân tích các trường hợp sai, đề ra giải pháp và thực hiện nó, sau đó đánh giá xem có cải tiến so với trước không

2.6.1. Phân tích trường hợp sai:

Cần phân tích các trường hợp sai của mô hình. Điều này giúp ta hiểu rõ hơn về những lỗi mà mô hình mắc phải, từ đó có thể đưa ra các giải pháp phù hợp.

Có nhiều cách để phân tích các trường hợp sai của mô hình, chẳng hạn như:

* Xem xét các kết quả dự đoán của mô hình trên tập dữ liệu kiểm tra.
* Sử dụng các kỹ thuật phân tích lỗi, chẳng hạn như confusion matrix, ROC curve, precision-recall curve.
* Trực quan hóa các dữ liệu đầu vào và đầu ra của mô hình.
* False Positive (FP): Xác định các trường hợp mà mô hình dự đoán là tích cực nhưng thực tế là tiêu cực.
* False Negative (FN): Xác định các trường hợp mà mô hình dự đoán là tiêu cực nhưng thực tế là tích cực.
* Tổ chức các thông tin để hiểu rõ ngữ cảnh xung quanh các dự đoán sai.

2.6.2. Đề xuất giải pháp:

Sau khi đã phân tích các trường hợp sai, ta cần đề ra các giải pháp phù hợp để cải thiện độ chính xác của mô hình.

Đối với FP:

* Tăng ngưỡng quyết định để giảm số lượng dự đoán tích cực giả mạo.
* Kiểm tra và cân nhắc sử dụng các thuật toán tinh chỉnh để giảm bias dương.

Đối với FN:

* Giảm ngưỡng quyết định để giảm số lượng dự đoán tiêu cực giả mạo.
* Cân nhắc sử dụng các thuật toán tinh chỉnh để giảm bias âm.

2.6.3. Thực hiện giải pháp:

Sau khi đã đề ra các giải pháp, ta cần thực hiện các giải pháp đó để xem có cải thiện được độ chính xác của mô hình hay không.

* Thay đổi các hyperparameters như ngưỡng quyết định, hệ số học (nếu áp dụng cho mô hình), số lượng cây (đối với ensemble models), vv.
* Thêm dữ liệu mới hoặc thực hiện các kỹ thuật chẩn đoán dữ liệu để giảm noise.
* Cân nhắc sử dụng các kỹ thuật resampling nếu tỷ lệ giữa các lớp trong dữ liệu không cân bằng.

2.6.4. Đánh giá hiệu suất sau cải thiện:

Đánh giá mô hình trên tập kiểm tra độc lập để đảm bảo tính tổng quát. So sánh các thước đo độ chính xác, precision, recall, và F1-score với mô hình trước cải thiện.

2.6.5. Lặp lại quy trình nếu cần thiết:

Nếu độ chính xác của mô hình được cải thiện, thì ta có thể tiếp tục sử dụng các giải pháp đó. Nếu độ chính xác không được cải thiện, thì ta cần quay lại bước phân tích các trường hợp sai để tìm ra nguyên nhân và đề ra các giải pháp mới.

Cung cấp thêm dữ liệu huấn luyện: Nếu mô hình mắc lỗi do thiếu dữ liệu, thì việc cung cấp thêm dữ liệu huấn luyện có thể giúp mô hình học được những đặc trưng quan trọng hơn.

* + Chọn lựa lại tập dữ liệu huấn luyện: Nếu mô hình mắc lỗi do tập dữ liệu huấn luyện không đại diện cho dữ liệu thực tế, thì việc chọn lựa lại tập dữ liệu huấn luyện có thể giúp mô hình học được những đặc trưng phù hợp hơn với dữ liệu thực tế.
  + Thay đổi kiến trúc của mô hình: Nếu mô hình mắc lỗi do kiến trúc không phù hợp, thì việc thay đổi kiến trúc của mô hình có thể giúp mô hình học được những đặc trưng hiệu quả hơn.
  + Điều chỉnh các siêu tham số của mô hình: Các siêu tham số của mô hình có thể ảnh hưởng đến hiệu suất của mô hình. Việc điều chỉnh các siêu tham số của mô hình có thể giúp mô hình hoạt động tốt hơn.

TÀI LIỆU THAM KHẢO

[1]. Trực, T. T. (2023, December 21). *Optimizer- Hiểu sâu về các thuật toán tối ưu ( GD,SGD,Adam,..)*. Viblo. https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8\_3-momentum-4

[2]. Gupta, A. (2023, September 13). *A comprehensive guide on optimizers in deep learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/

[3]. *Papers with Code - Continual Learning*. (n.d.). https://paperswithcode.com/task/continual-learning

[4].

[5].