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.

BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Họ và tên | MSSV | Công việc | Mức độ đóng góp | Mức độ hoàn thành |
|  |  |  |  |  |
|  |  |  |  |  |

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc154220628)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc154220629)

[ĐỀ BÀI v](#_Toc154220630)

[BẢNG PHÂN CÔNG CÔNG VIỆC vi](#_Toc154220631)

[MỤC LỤC 1](#_Toc154220632)

[CHƯƠNG 1 : BÀI TẬP 1 4](#_Toc154220633)

[1.1 Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy 4](#_Toc154220634)

[1.1.1 Optimizer là gì 4](#_Toc154220635)

[1.1.2 Gradient Descent 5](#_Toc154220636)

[1.1.2.1 Công thức 5](#_Toc154220637)

[1.1.2.2 Ưu điểm và Nhược điểm 7](#_Toc154220638)

[1.1.3 Stochastic Dradient Descent 8](#_Toc154220639)

[1.1.3.1 Công thức 8](#_Toc154220640)

[1.1.3.2 Ưu điểm và Nhược điểm: 9](#_Toc154220641)

[1.1.4 Adagrad 9](#_Toc154220642)

[1.1.4.1 Công thức 9](#_Toc154220643)

[1.1.4.2 Ưu điểm và Nhược điểm: 10](#_Toc154220644)

[1.1.5 RMSprop 10](#_Toc154220645)

[1.1.5.1 Công thức 10](#_Toc154220646)

[1.1.5.2 Ưu điểm và Nhược điểm: 11](#_Toc154220647)

[1.1.6 Momentum 12](#_Toc154220648)

[1.1.6.1 Công thức 13](#_Toc154220649)

[1.1.6.2 Ưu và nhược điểm 13](#_Toc154220650)

[1.1.7 Adam 13](#_Toc154220651)

[1.1.7.1 Công thức 14](#_Toc154220652)

[1.1.7.2 Ưu và nhược điểm 14](#_Toc154220653)

[1.1.8 So sánh các phương pháp Optimizer 15](#_Toc154220654)

[1.2 Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó 16](#_Toc154220655)

[1.2.1 Continual Learning 16](#_Toc154220656)

[1.2.2 Test Production 16](#_Toc154220657)

[1.2.3 Vai trò của Continual Learning và Test Production trong việc xây dựng một giải pháp học máy 17](#_Toc154220658)

[1.2.3.1 Một số lưu ý khi áp dụng Continual Learning và Test Production 17](#_Toc154220659)

[1.2.4 Tổng kết 18](#_Toc154220660)

[CHƯƠNG 2 : BÀI TÂP 2 19](#_Toc154220661)

[2.1 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 20](#_Toc154220662)

[2.2 Ứng dụng các mô hình học máy cơ bản để giải quyết bài toán, bao gồm cả các mô hình thuộc Ensemble Learing 21](#_Toc154220663)

[2.2.1 Giải quyết bài toán bằng Logistic Regression 21](#_Toc154220664)

[2.2.2 K-Nearest Neighbors 22](#_Toc154220665)

[2.2.3 Decision Tree 23](#_Toc154220666)

[2.2.4 Random Forest 24](#_Toc154220667)

[2.2.5 Ada Boost 25](#_Toc154220668)

[2.2.6 Gradient Boosting 27](#_Toc154220669)

[2.2.7 XgBoost 28](#_Toc154220670)

[2.2.8 Tổng quang 29](#_Toc154220671)

[2.3 Sử dụng Feed Forward Neural Network và Reccurent Neural Network (hoặc mô thuộc loại này) để giải quyết bài toán 30](#_Toc154220672)

[2.4 Áp dụng các kỹ thuật tránh Overfiting trên các mô hình của câu (2) và câu (3) để giải quyết bài toán 31](#_Toc154220673)

[2.5 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 32](#_Toc154220674)

[2.5.1. Phân tích trường hợp sai: 32](#_Toc154220675)

[2.5.2. Đề xuất giải pháp: 32](#_Toc154220676)

[2.5.3. Thực hiện giải pháp: 33](#_Toc154220677)

[2.5.4. Đánh giá hiệu suất sau cải thiện: 33](#_Toc154220678)

[2.5.5. Lặp lại quy trình nếu cần thiết: 33](#_Toc154220679)

[TÀI LIỆU THAM KHẢO 35](#_Toc154220680)

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. 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.

Một số Optimizers phổ biến trong maching leaning gồm:

* Gradient Descent (GD): là phương pháp cơ bản, cập nhật trong số theo ngược chiều gradient của loss function.
* Stochastic Dradient Descent (SGD): Một biến thể của GD, nó chỉ sử dụng một mẫu dữ liệu ngẫu nhiên trong mỗi lần cập nhật thay vì sử dụng toàn bộ tập dữ liệu.
* Adagrad (Adaptive Gradient Algorithm): Tự động điều chỉnh tỷ lệ học cho từng tham số dựa trên lịch sử của gradient.
* RMSprop (Root Mean Square Propagation): Giải quyết vấn đề tỷ lệ học (leaning rate) giảm dần của Adagrad bằng cách chia tỷ lệ học cho trung bình của bình phương gradient.
* Momentum: Kỹ thuật được sử dụng trong quá trình tối ưu hóa (optimizer) để cải thiện khả năng hội tụ của mô hình trong quá trình đào tạo. Kỹ thuật này giúp tăng tốc quá trình hội tụ và giảm đà hoặc dao động khi mô hình đang tiến hành cập nhật trọng số.
* Adam (Adaptive Moment Estimation): Adam là sự kết hợp của Momentum và RMSprop
  + 1. Gradient Descent

Gradient Descent giải quyết vấn đề này bằng cách tìm kiếm giá trị tối ưu thông qua việc di chuyển từ điểm xuất phát (nghiệm khởi tạo) theo hướng ngược với đạo hàm. Cứ mỗi vòng lặp, nghiệm tiến gần hơn đến điểm cực tiểu.

Trong bài toán tối ưu, chúng ta muốn tìm giá trị nhỏ nhất của một hàm số, thường là hàm mất mát . Nếu 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 phức tạp hoặc thậm chí không khả thi.

* + - 1. **Công thức**

Công thức cập nhật trong Gradient Descent:

*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

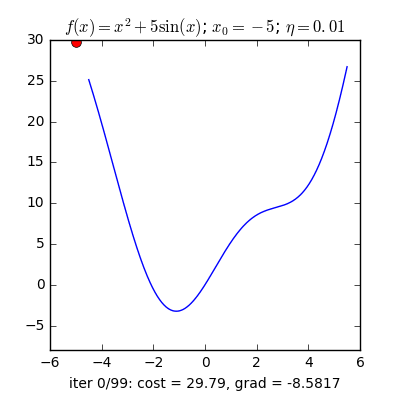
Nếu thì nghĩa là các gradian âm. Do đó, và nghiệm sẽ đi dần về phía bên phải, tiến dần đến cực tiểu.

A graph of a function

Description automatically generated

Hình 1.1.2 1: 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 1.1.2 2: Gradian descent

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

* + - 1. **Ưu điểm và Nhược điểm**

Ưu điểm:

* Thuật toán đơn giản và dễ hiểu.
* Giải quyết vấn đề tối ưu model neural network.

Nhược điểm:

* Phụ thuộc vào điểm khởi tạo ban đầu và learning rate.
* Có thể dẫn đến nghiệm cực tiểu cục bộ.
* Tốc độ học quá lớn hoặc quá nhỏ có thể ảnh hưởng đến quá trình training.
  + 1. Stochastic Dradient Descent

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 bớt độ phức tạp tính toán và tăng tốc quá trình cập nhật tham số trong quá trình đào tạo mô hình.

SGD cũng 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.

* + - 1. **Công thức**

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. **Ưu điểm và Nhược điểm:**

Ưu điểm:

* Phù hợp cho dữ liệu lớn và phức tạp.
* Giảm độ phức tạp tính toán.
* Cập nhật tham số nhanh chóng.

Nhược điểm:

* Đôi khi có thể làm tăng độ dao động và giảm độ chính xác vì sự ngẫu nhiên trong việc chọn mẫu.
  + 1. Adagrad

Adagrad là một thuật toán tối ưu hóa tỷ lệ học (learning rate) trong quá trình training mà không yêu cầu sự điều chỉnh thủ công của learning rate. Nó 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 thích hợp cho các bài toán khi một số tham số có đạo hàm lớn, và mô hình yêu cầu learning rate tương ứng. Cân nhắc cẩn thận khi áp dụng Adagrad cho các mô hình có đạo hàm biến động lớn và không ổn định.

* + - 1. **Công thức**

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. **Ưu điểm và Nhược điểm:**

Ưu Điểm:

* Tự điều chỉnh Learning Rate: Adagrad tự điều chỉnh learning rate tương ứng với mỗi tham số dựa trên lịch sử của đạo hàm của tham số đó. Điều này giúp đảm bảo rằng các tham số có đạo hàm lớn sẽ có learning rate nhỏ hơn, và ngược lại.
* Không cần điều chỉnh thủ công: Không cần phải điều chỉnh learning rate thủ công, giảm công sức và độ rủi ro.

Nhược Điểm:

* Đóng băng Learning Rate: Do bình phương các đạo hàm được tích lũy theo thời gian, tổng bình phương biến thiên sẽ tăng, làm cho learning rate giảm đi nhanh chóng. Điều này có thể dẫn đến tốc độ học cực kỳ nhỏ và làm cho quá trình training trở nên đóng băng.
  + 1. RMSprop

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

* + - 1. **Công thức**

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. **Ưu điểm và Nhược điểm:**

Ưu Điểm:

* Giải quyết vấn đề tốc độ học giảm dần: RMSprop giải quyết vấn đề của Adagrad, trong đó tốc độ học giảm dần theo thời gian có thể làm chậm quá trình training.

Nhược Điểm:

* Khả năng rơi vào Local Minimum: RMSprop có thể rơi vào local minimum thay vì global minimum, giống như vấn đề của Adagrad. Điều này đã thúc đẩy sự phát triển của thuật toán tối ưu hóa kết hợp như Adam. Đóng băng Learning Rate: Do bình phương các đạo hàm được tích lũy theo thời gian, tổng bình phương biến thiên sẽ tăng, làm cho learning rate giảm đi nhanh chóng. Điều này có thể dẫn đến tốc độ học cực kỳ nhỏ và làm cho quá trình training trở nên đóng băng.
  + 1. Momentum

Momentum là một thuật toán tối ưu hóa Gradient Descent nhằm giảm tình trạng dừng ở local minimum và thúc đẩy việc tiến tới global minimum. Từ góc độ vật lý, Momentum được giải thích như việc thả hai viên bi từ hai điểm khác nhau. Một viên bi có đà và vận tốc đủ lớn để vượt qua local minimum và tiến tới global minimum.

A screenshot of a graph

Description automatically generated

Hình 1.1.6 1: Momentum

* + - 1. **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. **Ưu và nhược điểm**

Ưu Điểm:

* Vượt Qua Local Minimum: Momentum giúp thuật toán vượt qua local minimum và tiếp tục tiến tới global minimum.
* Hiệu Quả Tính Toán: Thuật toán giảm thiểu đào hàm và giúp việc cập nhật nhanh chóng.

Nhược Điểm:

* Giao Động Khi Gần Đích: Khi tiến gần đến đích, viên bi vẫn có thể dao động qua lại và mất thời gian để dừng lại. Điều này xuất phát từ việc viên bi vẫn còn đà.
  + 1. 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

* + - 1. **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. **Ưu và nhược điểm**

Ưu Điểm:

* Tận Dụng Ưu Điểm của Cả Momentum và RMSprop: Adam kết hợp cả hai thành phần Momentum và RMSprop, giúp tận dụng ưu điểm của cả hai thuật toán.

Nhược Điểm:

* Phức Tạp Hơn: Adam có thể phức tạp hơn so với các thuật toán đơn giản hơn như Gradient Descent và SGD.

1.1.8 So sánh các phương pháp Optimizer

Mỗi thuật toán đều có ưu và nhược điểm riêng. Gradient Descent và Stochastic Gradient Descent đơn giản và hiệu quả cho dữ liệu nhỏ, trong khi Adagrad và RMSprop tự động điều chỉnh learning rate và phù hợp cho dữ liệu có biến động lớn. Momentum giúp vượt qua local minimum và cập nhật nhanh chóng. Adam kết hợp cả Momentum và RMSprop, tận dụng ưu điểm của cả hai, tuy nhiên có độ phức tạp cao hơn.

Lựa chọn phương pháp tối ưu hóa phụ thuộc vào loại dữ liệu, kích thước mô hình, và yêu cầu về hiệu suất tính toán. Việc thử nghiệm và điều chỉnh các optimizer là quan trọng để đạt được hiệu suất tối ưu trong quá trình đào tạo mô hình học máy.

* 1. 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. Continual Learning

Continual Learning là một lĩnh vực nghiên cứu của học máy tập trung vào việc phát triển các mô hình học máy có thể học hỏi và cải thiện hiệu suất của chúng khi được cung cấp dữ liệu mới. Điều này trái ngược với cách tiếp cận truyền thống trong học máy, trong đó mô hình được đào tạo trên một tập dữ liệu cố định và sau đó được triển khai mà không được cập nhật thêm.

Continual Learning có thể được sử dụng trong nhiều ứng dụng khác nhau, chẳng hạn như:

* Các ứng dụng cần phải thích nghi với các thay đổi trong dữ liệu đầu vào, chẳng hạn như các hệ thống phát hiện gian lận hoặc các hệ thống nhận dạng giọng nói.
* Các ứng dụng cần phải học hỏi từ dữ liệu mới được thu thập theo thời gian, chẳng hạn như các hệ thống khuyến nghị hoặc các hệ thống dịch tự động.
* Các ứng dụng cần phải duy trì hiệu suất cao trong một môi trường thay đổi, chẳng hạn như các hệ thống điều khiển tự động hoặc các hệ thống giao thông thông minh.
  + 1. Test Production

Test Production là một quá trình trong đó một giải pháp học máy được triển khai trong môi trường sản xuất với quy mô nhỏ, để đánh giá hiệu suất và ổn định của giải pháp trước khi triển khai toàn diện.

Test Production là một bước quan trọng trong quá trình phát triển và triển khai giải pháp học máy. Nó giúp phát hiện các lỗi và vấn đề tiềm ẩn trong giải pháp trước khi giải pháp được triển khai trên quy mô lớn. Sản xuất thử cũng giúp đánh giá hiệu suất của giải pháp trong môi trường thực tế.

* + 1. Vai trò của Continual Learning và Test Production trong việc xây dựng một giải pháp học máy

Continual Learning giúp giải pháp học máy có thể thích nghi với các thay đổi trong dữ liệu đầu vào và môi trường hoạt động. Điều này giúp giải pháp duy trì hiệu suất cao trong một thời gian dài.

Test Production giúp phát hiện các lỗi và vấn đề tiềm ẩn trong giải pháp trước khi giải pháp được triển khai trên quy mô lớn. Điều này giúp giảm thiểu rủi ro khi triển khai giải pháp học máy.

**1.2.3.1 Một số lưu ý khi áp dụng Continual Learning và Test Production**

Khi áp dụng Continual Learning:

* Chọn mô hình học máy phù hợp: Không phải tất cả các mô hình học máy đều phù hợp với Continual Learning. Một số mô hình học máy, chẳng hạn như mô hình mạng thần kinh có thể học hỏi và thích nghi với dữ liệu mới một cách hiệu quả.
* Sử dụng các kỹ thuật học tập liên tục hiệu quả: Có nhiều kỹ thuật học tập liên tục khác nhau. Cần chọn kỹ thuật phù hợp với ứng dụng cụ thể.
* Theo dõi hiệu suất của mô hình: Cần theo dõi hiệu suất của mô hình trong quá trình học tập liên tục. Điều này giúp phát hiện các vấn đề tiềm ẩn và điều chỉnh các tham số học tập.

Khi áp dụng Test Production:

* Chọn quy mô sản xuất thử phù hợp: Quy mô sản xuất thử không nên quá nhỏ, vì điều này sẽ không thể đánh giá hiệu suất của giải pháp trong môi trường thực tế. Tuy nhiên, quy mô sản xuất thử cũng không nên quá lớn, vì điều này có thể gây ra các vấn đề về hiệu suất hoặc ổn định.
* Chọn thời gian triển khai sản xuất thử phù hợp: Thời gian triển khai sản xuất thử nên đủ dài để đánh giá hiệu suất của giải pháp trong môi trường thực tế.
* Theo dõi hiệu suất của giải pháp trong sản xuất thử: Cần theo dõi hiệu suất của giải pháp trong sản xuất thử. Điều này giúp phát hiện các vấn đề tiềm ẩn và điều chỉnh giải pháp nếu cần.
  + 1. Tổng kết

Khi xây dựng giải pháp học máy, việc tích hợp cả hai khái niệm này có thể giúp mô hình không chỉ học liên tục mà còn đảm bảo hiệu suất ổn định và đánh giá chính xác trên các tập dữ liệu thử nghiệm. Test Production có thể giúp 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. Ngoài ra Continual Learning và Test Production đều đóng vai trò trong việc đảm bảo rằng mô hình không chỉ học được từ dữ liệu mới mà còn duy trì hiệu suất trên các dạng dữ liệu đã biết và mới.

CHƯƠNG 2 : BÀI TÂP 2

Bài toán "Dự Đoán Đặt Phòng và Hủy Đặt Phòng Khách Sạn" nhằm ước lượng khả năng đặt và hủy đặt phòng dựa trên dữ liệu liên quan. Qua việc phân tích thống kê và sử dụng mô hình học máy, nó giúp hiểu rõ hơn về hành vi đặt phòng và cung cấp dự đoán về khả năng hủy đặt, đồng thời hỗ trợ quyết định trong quản lý khách sạn.

2.1 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

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

2.2.1 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

2.2.2 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

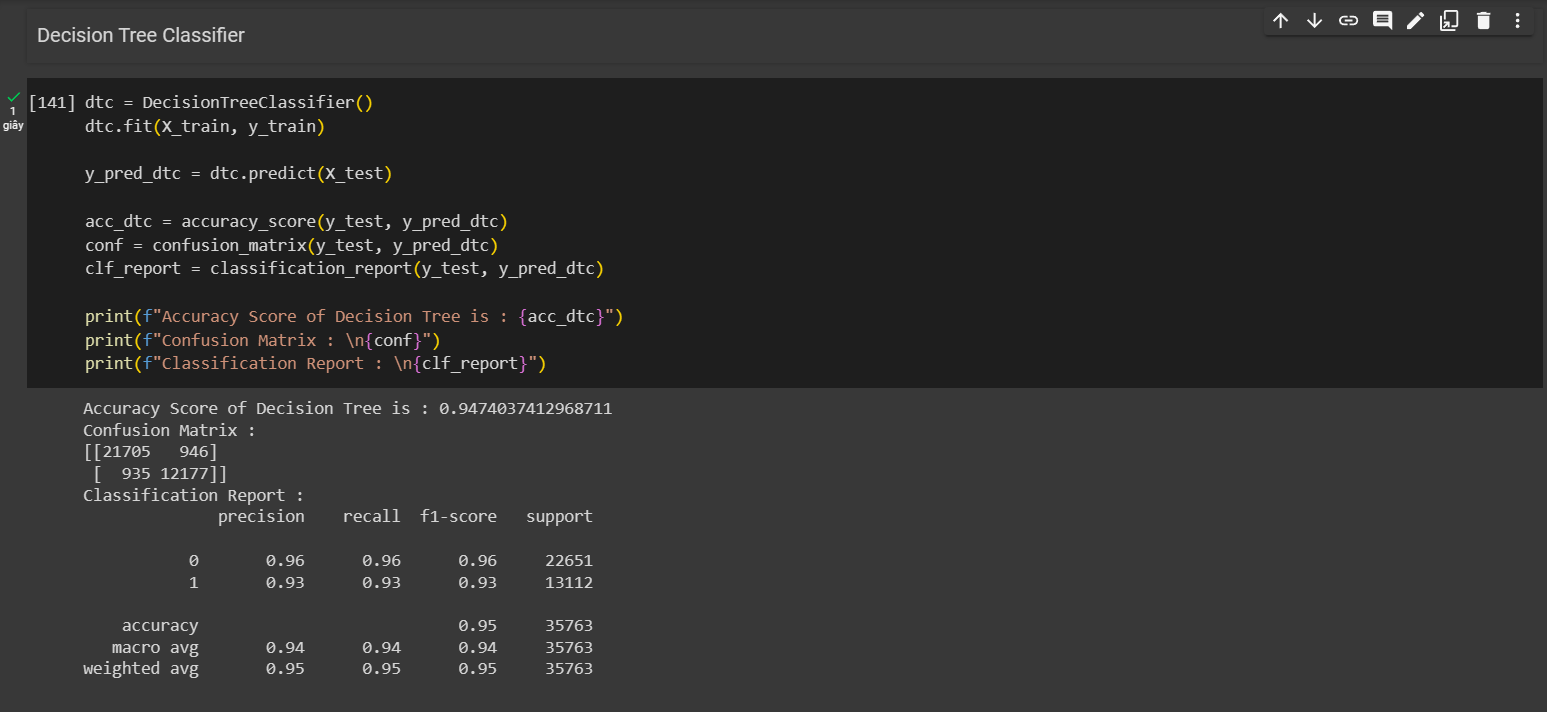
Recall:

* 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

2.2.3 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

2.2.5 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

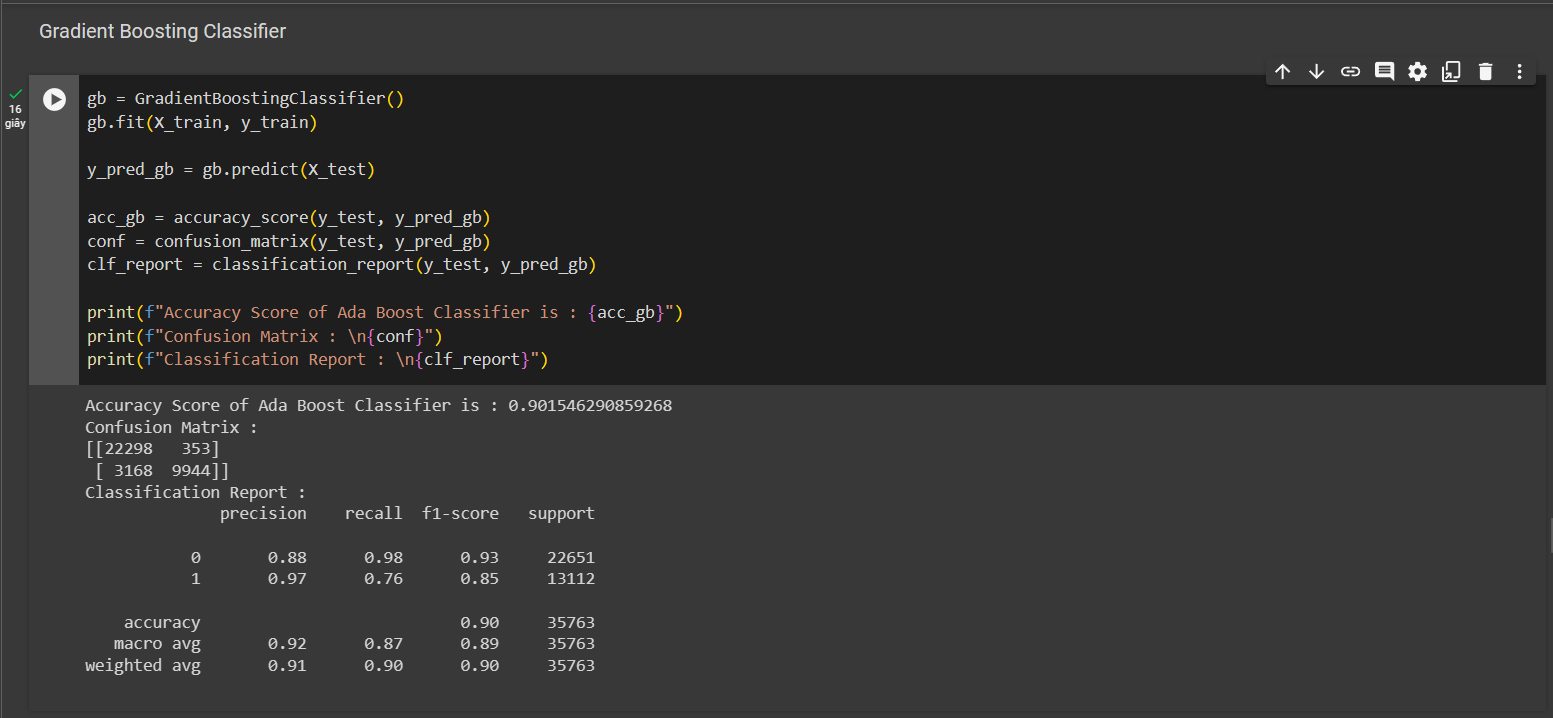
Recall:

* 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

2.2.6 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

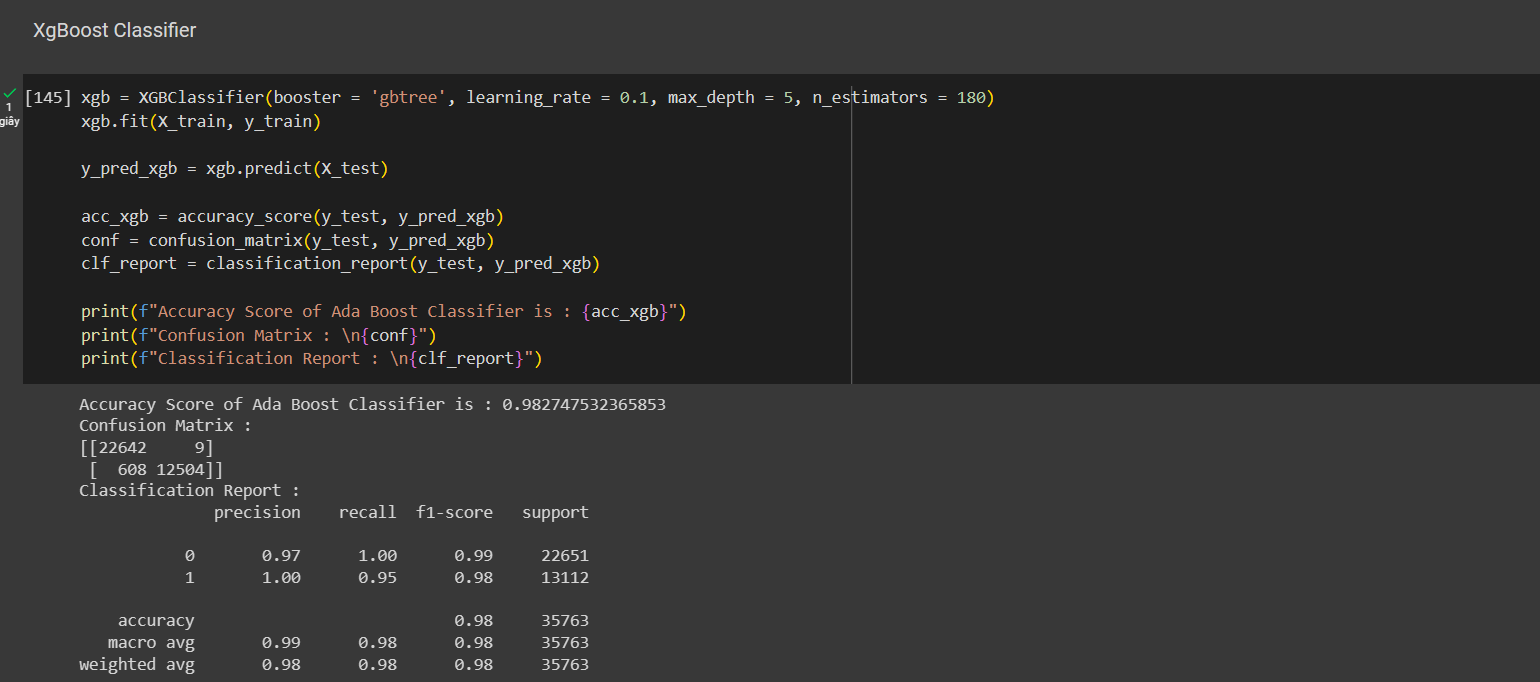
Recal:

* 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

2.2.7 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

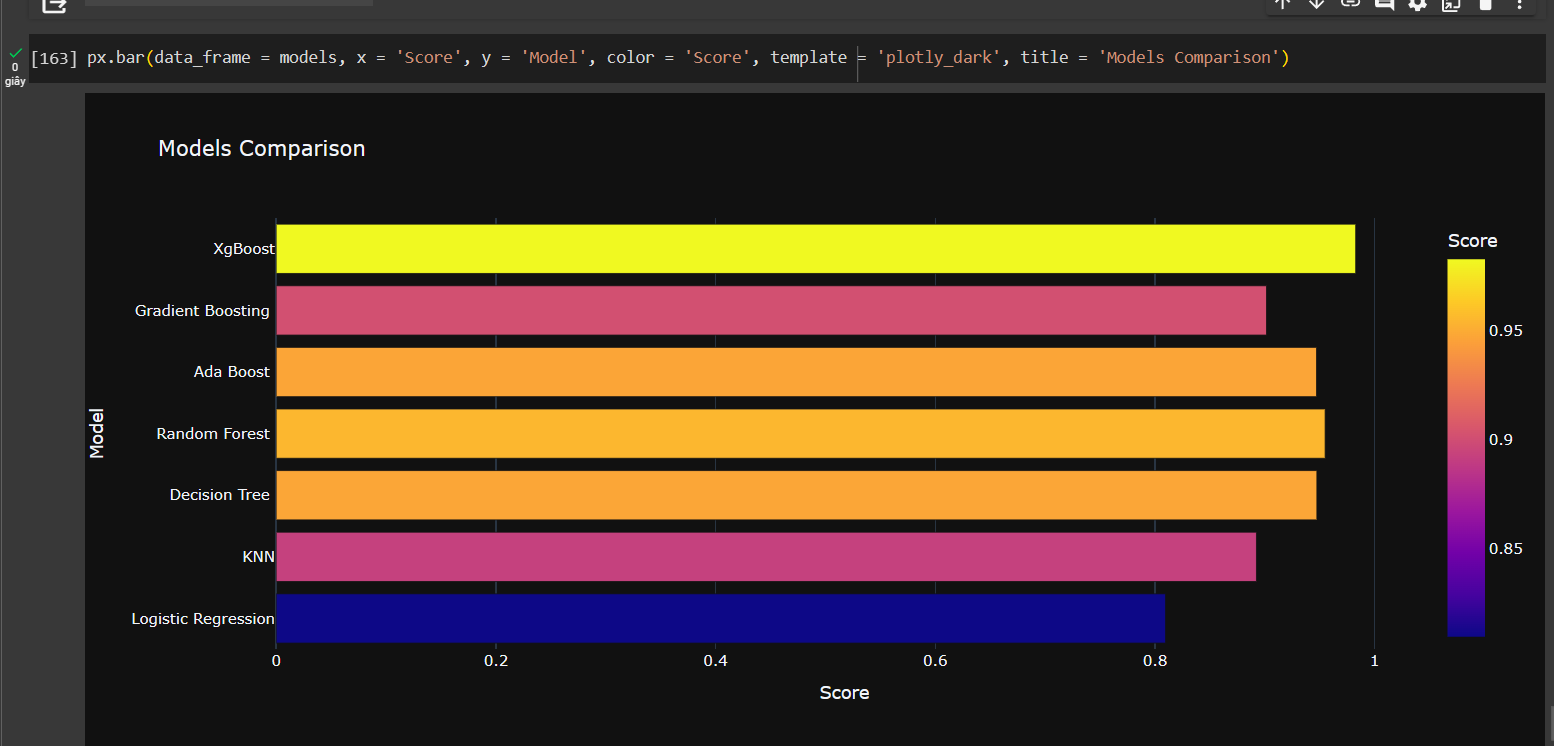
Recall:

* 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

2.2.8 Tổng quan



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.

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

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

2.5 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

2.5.1. 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.

2.5.2. 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.

2.5.3. 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.

2.5.4. 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.

2.5.5. 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