Thành viên nhóm:

52000049 - Pham Trí Hùng

52000086 - Huỳnh Thi Thảo Ngân

52000043 - Lê Thị Thúy Hằng

Bài 1: (5 điểm)

Giải quyết một bài toán phân loại (classification) trong học máy với các yêu cầu sau:

- Tự đặt ra hoặc tự tìm Bài toán với dữ liệu có sẵn hoặc dữ liệu tự xây dựng. Dữ liệu phải là dạng có cấu trúc, tức là dạng bảng biểu với cột là thuộc tính (attribute) và mỗi dòng là một đối tượng (instance). Dữ liệu phong phú với nhiều thuộc tính, các thuộc tính thuộc nhiều kiểu data khác nhau (numerical, categorical). Số lượng các phần tử (các dòng) càng nhiều càng tốt.
- Thực hiện các bước đọc dữ liệu, chuẩn hoá dữ liệu trước khi đưa vào mô hình để học.
- Sử dụng ít nhất 3 mô hình phân loại khác nhau. So sánh các mô hình này với các độ đo: accuracy, precision, recall, f1-score của từng class và weighted average of f1-score của toàn bộ dữ liệu. So sánh về thời gian training và thời gian testing của các models này.

Bài làm

Giải quyết bài tập bằng 3 mô hình là: GaussianNB, KNeighborsClassifier, DecisionTreeClassifier với dữ liệu weatherAUS.csv về thời tiết.

1.1 Mô hình GaussianNB

- Mô hình này dùng để tính giá trị trung bình và độ lệch chuẩn của từng biến đầu vào (x) cho từng giá trị lớp.

  trung bình (x) = 1 / n * tổng (x)

  n: là số lượng phiên bản

  x là các giá trị cho một biến đầu vào trong dữ liệu đào tạo của bạn

- Chúng ta có thể tính độ lệch chuẩn bằng phương trình sau:

độ lệch chuẩn (x) = sqrt (1 / n * sum (xi-mean (x) ^ 2))

- Đây là căn bậc hai của hiệu số bình phương trung bình của mỗi giá trị của x từ giá trị trung bình của x, trong đó n là số lượng phiên bản, sqrt () là hàm căn bậc hai, sum () là hàm tổng, xi là một giá trị cụ thể của biến x cho trường hợp thứ i và giá trị trung bình (x) được mô tả ở trên và ^ 2 là hình vuông.
- Đưa ra dự đoán với mô hình Gaussian Naive Bayes Xác suất của các giá trị x mới được tính bằng Hàm mật độ xác suất Gaussian (PDF).
- Khi đưa ra dự đoán, các tham số này có thể được cắm vào Gaussian PDF với đầu vào mới cho biến và đổi lại, Gaussian PDF sẽ cung cấp ước tính xác suất của giá trị đầu vào mới đó cho lớp

pdf (x, trung bình, sd) = (1 / (sqrt (2 * PI) * sd)) * exp (- ((x-mean ^ 2) / (2 * sd ^ 2)))

- Trong đó pdf (x) là Gaussian PDF, sqrt () là căn bậc hai, giá trị trung bình và sd là giá trị trung bình và độ lệch chuẩn được tính ở trên, PI là hằng số, exp () là hằng số e hoặc Euler được nâng lên thành công suất và x là giá trị đầu vào cho biến đầu vào.
- Sau đó chúng ta có thể cắm các xác suất vào phương trình trên để đưa ra dự đoán với các đầu vào có giá trị thực.

1.2 Mô hình KNeighborsClassifier

- KNN (K-Nearest Neighbors) là một thuật toán đơn giản nhất trong nhóm thuật toán Học có giám sát. Ý tưởng của thuật toán này đó là tìm output của một dữ liệu mới dựa trên output của K điểm gần nhất xung quanh nó. KNN được ứng dụng nhiều trong khai phá dữ liệu và học máy. Trong thực tế, việc đo khoảng cách giữa các điểm dữ liệu, chúng ta có thể sử dụng rất nhiều độ đo, tiêu biểu như là Manhattan, Ơ-clit, cosine,…
- Thuật toán:

+Xác định tham số K số làng giềng gần nhất

+Tính khoảng cách của đối tượng cần phân lớp tới tất cả các đối tượng có trong tập train

+Lấy top K cho giá trị nhỏ nhất (hoặc lớn nhất)

+Trong top K giá trị vừa lấy, ta thống kê số lượng của mỗi lớp, chọn phân lớp cho số lượng lớn nhất

- Một câu hỏi đặt ra đó là có phải cứ chọn K càng lớn thì càng tốt, thì câu trả lời đó là còn tùy thuộc vào dữ liệu đó như thế nào. Không phải lúc nào K càng lớn thì cho kết quả tốt và ngược lại. Việc lựa chọn tham số K của mô hình sẽ tiến hành thông qua thực nghiệm nhiều lần để chọn ra kết quả tốt nhất.

- Với các bước như trên, chúng ta nhận thấy rằng thuật toán của KNN rất đơn giản, dễ thực hiện, dễ cài đặt. Việc dữ đoán kết quả thật là dễ dàng, độ phức tạp của thuật toán nhỏ. Bên cạnh đó sẽ tồn tại nhiều nhược điểm như:

  +Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.Nếu tập train của chúng ta có kích thước rất lớn, thì việc duyệt qua tất cả các điểm dữ liệu để tính toán là rất mất thời gian, đặt biệt là trong thời kỳ hiện nay thì dữ liệu thu thập được rất lớn

  +KNN rất nhạy cảm với dữ liệu nhiễu, đặc biệt là khi ta chọn K nhỏ. Việc này sẽ dẫn đến kết quả không tốt.

1.3 DecisionTreeClassifier

- Mô hình *cây quyết định* là một mô hình được sử dụng khá phổ biến và hiệu quả trong cả hai lớp bài toán phân loại và dự báo của học có giám sát. Khác với những thuật toán khác trong học có giám sát, mô hình *cây quyết định* không tồn tại phương trình dự báo. Mọi việc chúng ta cần thực hiện đó là tìm ra một cây quyết định dự báo tốt trên tập huấn luyện và sử dụng cây quyết định này dự báo trên tập kiểm tra.
- Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.
- Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết**.**

1.4 Accuray, Precision, Recall và F1-score:

- Accuracy là một chỉ số dùng để đánh giá mô hình trong học máy được định nghĩa là tỉ lệ giữa số điểm được phân loại đúng và tổng số điểm.
- Precision là một chỉ số dùng để đánh giá các mô hình trong học máy được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive (TP + FP) .
- Recall là một chỉ số dùng để đánh giá các mô hình trong học máy được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN).
- F1-score là một chỉ số dùng để đánh giá mô hình trong học máy là harmonic mean của precision và recall.

Bài 2: (3 điểm)

Feature Selection là bài toán lựa chọn các đặc trưng (feature or attribute) quan trọng và loại bỏ các đặc trưng không quan trọng hoặc dư thừa. Từ đó ta có thể xây dựng mô hình học máy hiệu quả hơn (nhanh hơn, ít tham số hơn và độ chính xác cũng tương tự hoặc thậm chí tốt hơn khi dùng toàn bộ tập feature).

Hãy tìm hiểu vấn đề Feature Selection với các yêu cầu sau:

a) Phương pháp dựa trên Correlation. Hiển thị các đồ thị minh hoạ.

b) Thử nghiệm các tập feature được lựa chọn khác nhau từ phương pháp ở câu (a) cho một bài toán Regression và sử dụng thuật toán Linear Regression. Dữ liệu và bài toán tự chọn. So sánh thông qua độ đo Mean Absolute Error (MAE).

Bài làm

- Khi xây dựng mô hình học máy cho tập dữ liệu ngoài đời thực, chúng ta gặp rất nhiều đặc trưng trong tập dữ liệu và không phải lúc nào tất cả các đặc trưng này đều quan trọng. Thêm các tính năng không cần thiết trong khi huấn luyện mô hình dẫn đến chúng ta giảm độ chính xác tổng thể của mô hình, tăng độ phức tạp của mô hình và giảm khả năng tổng quát hóa của mô hình và làm cho mô hình bị sai lệch. Do đó, lựa chọn đặc trưng là một trong những bước quan trọng trong khi xây dựng mô hình học máy.
- Mục tiêu của nó là tìm ra bộ tính năng tốt nhất có thể để xây dựng mô hình học máy hiệu quả hơn (nhanh hơn, ít tham số hơn và độ chính xác cũng tương tự hoặc thậm chí tốt hơn khi dùng toàn bộ tập feature).
- Lựa chọn đặc trưng có bốn cách tiếp cận khác nhau như cách tiếp cận bộ lọc (filter approach) , cách tiếp cận trình bao bọc (wrapper approach), cách tiếp cận nhúng (embedded approach) và cách tiếp cận kết hợp (hybrid approach).

  Các thông số để lựa chọn tính năng:

  Các tham số được phân loại dựa trên hai yếu tố:

  Sự giống nhau của thông tin được đóng góp bởi các tính năng:

CORRELATION

- Các đặc trưng được phân loại là có liên quan hoặc tương tự chủ yếu dựa trên yếu tố tương quan của chúng. Trong tập dữ liệu, chúng ta có nhiều tính năng tương quan với nhau. Các đặc trưng tương quan là: nếu f1 và f2 là hai đặc trưng tương quan của một tập dữ liệu, thì mô hình phân loại hoặc hồi quy bao gồm cả f1 và f2 sẽ cho kết quả giống như mô hình dự đoán so với kịch bản trong đó f1 hoặc f2 đã được đưa vào tập dữ liệu. Điều này là do cả f1 và f2 đều có tương quan và do đó chúng đóng góp cùng một thông tin về mô hình trong tập dữ liệu. Có nhiều phương pháp khác nhau để tính toán hệ số tương quan, tuy nhiên, hệ số tương quan của Pearson được sử dụng rộng rãi nhất.
- Công thức cho hệ số tương quan của Pearson (p) là:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

Trong đó

+cov (X, Y) - hiệp phương sai

+sigma (X) - độ lệch chuẩn của X

+sigma (Y) - độ lệch chuẩn của Y

- Do đó, các đặc trưng tương quan không liên quan, vì tất cả chúng đều đóng góp thông tin tương tự .Chỉ một đại diện của toàn bộ các đặc điểm tương quan hoặc được kết hợp sẽ cho cùng một kết quả phân loại hoặc hồi quy. Các đặc trưng này là dư thừa và bị loại trừ cho mục đích giảm kích thước sau khi chọn một đại diện cụ thể từ mỗi nhóm tính năng được liên kết hoặc tương quan bằng cách sử dụng các thuật toán khác nhau.

TƯƠNG QUAN VỚI NHÃN LỚP ĐÍCH (CORRELATION WITH THE TARGET CLASS LABEL):

- Mối tương quan giữa nhãn lớp mục tiêu và các tính năng quyết định mức độ tương quan của từng tính năng đối với nhãn lớp đích. Có nhiều kỹ thuật tương quan khác nhau như Pearson, Spearman, Kendall, v.v. để tìm mối tương quan giữa hai đối tượng địa lý.
- df.corr() trả về với hệ số tương quan con người giữa các đối tượng địa lý. Từ bản đồ nhiệt tương quan ở trên cho dữ liệu titanic , các tính năng như ‘sex’, ‘Pclass’có ‘fare’tương quan cao với nhãn lớp mục tiêu ‘Survived Và do đó đóng vai trò là các tính năng quan trọng. Trong khi các tính năng như ‘PassengerId’, ‘SibSp’không tương quan với nhãn lớp đích và có thể không phục vụ các tính năng quan trọng cho việc lập mô hình. Do đó, các tính năng này có thể bị loại bỏ.

TƯƠNG QUAN GIỮA CÁC TÍNH NĂNG (CORRELATION BETWEEN THE FEATURES):

- Sự tương quan giữa các tính năng dẫn đến tính cộng đồng, có thể ảnh hưởng đến hiệu suất của mô hình. Một đối tượng địa lý được cho là có tương quan với các đối

tượng địa lý khác nếu chúng có hệ số tương quan cao, do đó sự thay đổi trong một đối tượng địa lý cũng dẫn đến thay đổi đối tượng địa lý tương quan khác.

- Từ bản đồ nhiệt tương quan ở trên cho dữ liệu titanic , hệ số tương quan Pearson giữa 'Pclass' và 'Fare', do đó sự thay đổi trong một biến sẽ tác động tiêu cực đến biến kia.

Bài 3 (2 điểm):

Tìm hiểu các thuật toán tối ưu (thuật toán học, cập nhật tham số) trong học máy. Mô tả, giải

thích thuật toán và có code minh hoạ:

- Thuật toán Stochastic Gradient Descent
- Thuật toán Adam (Adam Optimization Algorithm)

Bài làm

3.1: Thuật toán Stochastic Gradient Descent

- Thuật toán SGD là thuật toán tối ưu hóa cơ bản theo họ gradient. Thuật toán này rất triển khai, có nền tảng lý thuyết vững chắc, cực kỳ ổn định trong quá trình huấn luyện, kết quả đạt được có thể so sánh với các thuật toán khác. Ý tưởng của thuật toán khá đơn giản, đó là "tính giá trị gradient của mỗi tham số, và đi một bước nhỏ theo chiều của gradient". Nếu chúng ta lặp đi lặp lại quá trình này, và ngẫu nhiên chọn (stochastic) một tập batch trong tập huấn luyện, mô hình chúng ta sẽ được cải tiến dần đến điểm hội tụ.
- Giả sử, bạn có một triệu mẫu trong tập dữ liệu của mình, vì vậy nếu bạn sử dụng kỹ thuật tối ưu hóa Gradient Descent điển hình, bạn sẽ phải sử dụng tất cả một triệu mẫu để hoàn thành một lần lặp trong khi thực hiện Gradient Descent, và nó phải được thực hiện cho mỗi lần lặp cho đến khi đạt đến cực tiểu. Do đó, nó trở nên rất

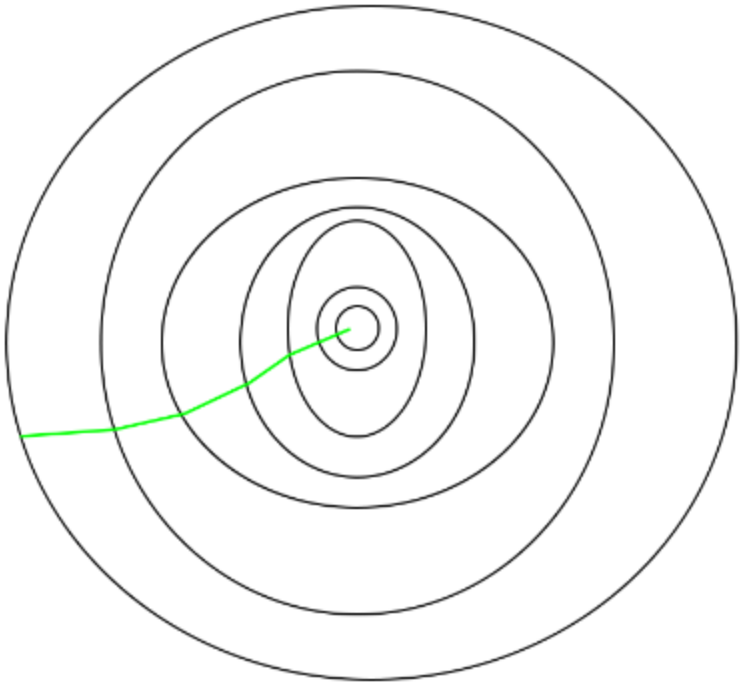
tốn kém về mặt tính toán để thực hiện.Vấn đề này được giải quyết bằng Stochastic Gradient Descent. Trong SGD, nó chỉ sử dụng một mẫu duy nhất, tức là kích thước lô của một, để thực hiện mỗi lần lặp. Mẫu được xáo trộn ngẫu nhiên và được chọn để thực hiện lặp lại.

Thuật toán SGD:

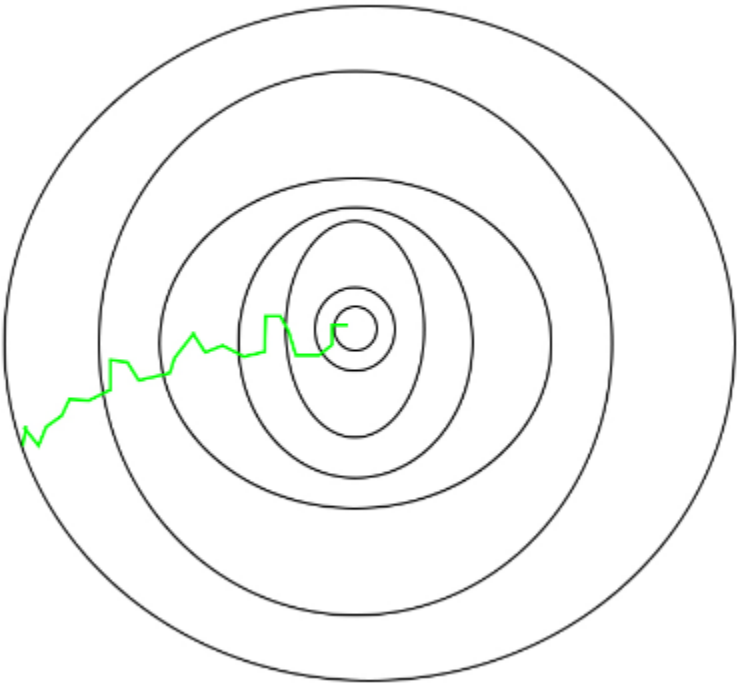
for i in range (m) :

$$\theta_j = \theta_j - \alpha \left(\widehat{y}^i - y^i\right) x_j^i$$

- Trong SGD, vì chỉ một mẫu từ tập dữ liệu được chọn ngẫu nhiên cho mỗi lần lặp, nên đường dẫn mà thuật toán thực hiện để đạt đến cực tiểu thường ồn ào hơn so với thuật toán Gradient Descent điển hình của bạn. Nhưng điều đó không quan trọng lắm vì đường đi của thuật toán không quan trọng, miễn là chúng ta đạt đến cực tiểu và với thời gian đào tạo ngắn hơn đáng kể.
- Đường dẫn được thực hiện bởi Batch Gradient Descent như hình dưới đây:



Một con đường đã được thực hiện bởi Stochastic Gradient Descent



- Ta thấy SGD có đường đi khá là zic zắc , không mượt như Gradient Descent điển hình, nên nó thường mất một số lần lặp cao hơn để đạt đến cực tiểu, vì tính ngẫu nhiên trong gốc của nó. Mặc dù nó yêu cầu số lần lặp lại cao hơn để đạt đến cực tiểu so với Gradient Descent điển hình, nhưng về mặt tính toán nó vẫn ít tốn kém hơn nhiều so với Gradient Descent điển hình. Do đó, trong hầu hết các tình huống, SGD được ưu tiên hơn Batch Gradient Descent để tối ưu hóa thuật toán.

*Ưu điểm :

- Thuật toán giải quyết được đối với cơ sở dữ liệu lớn mà GD không làm được.
- Thuật toán tối ưu này hiện nay vẫn hay được sử dụng.

*Nhược điểm :

- Thuật toán vẫn chưa giải quyết được 2 nhược điểm lớn của gradient descent (learning rate, điểm dữ liệu ban đầu). Vì vậy ta phải kết hợp SGD với 1 số thuật toán khác như: Momentum, AdaGrad,..

3.2 Thuật toán Adam (Adam Optimization Algorithm)

- Adam, (Adaptive moment Estimation), là một phương pháp ước lượng learning rate cho mỗi tham số. Adam được coi như là sự kết hợp của AdaDelta hay RMSProp và momentum. Trong khi momentum có thể xem như một quả bóng đang chạy xuống dốc, Adam lại giống như một quả bóng nặng với ma sát. Adam sử dụng bình phương gradient để chia tỷ lệ learning rate như RMSProp và tận dụng "đà" giống như momentum.
- Công thức cập nhật đầy đủ của Adam:

$$\begin{cases} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\ \theta_t &= \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\varepsilon} \end{cases}$$

- Để tốt hơn về chi phí tính toán, 3 dòng cuối cùng ta có thể viết lại như sau:

$$\eta_t = \eta \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t}$$

$$\theta_t = \theta_{t-1} - \eta_t \frac{m_t}{\sqrt{v_t} + \varepsilon}$$