UỶ BAN NHÂN DÂN

THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

BÁO CÁO TỔNG KẾT MÔN
PHÂN TÍCH DỮ LIỆU

Lớp: DCT121C2

Thành viên tham gia:

Hà Quốc Bảo – 3121411021

Trương Tấn Tài - 3121411189

Trương Quang Hùng - 3121411081

Giảng viên hướng dẫn: Trịnh Tấn Đạt

Nguyễn Thị Tuyết Nam

Thành phố Hồ Chí Minh, Tháng 5 Năm 2024

LỜI CẢM ƠN

Trong quá trình học tập và làm việc nhóm, chúng em luôn nhận được sự quan tâm, hướng dẫn và giúp đỡ tận tình của giảng viên bộ môn.

Lời nói đầu tiên, nhóm em xin gửi lời cảm ơn sâu sắc, chân thành đến với các giảng viên bộ môn Phân tích dữ liệu - Thầy Trịnh Tấn Đạt và Cô Nguyễn Thị Tuyết Nam, người đã nhiệt tình, tận tâm trong việc giảng dạy & hướng dẫn chúng em trong suốt quá trình học tập hỗ trợ cả nhóm hoàn thiện báo cáo đồ án môn học này.

Tuy nhiên trong quá trình học tập, nghiên cứu và hoàn thiện đồ án, do trình độ tiếp thu kiến thức, lý luận và kinh nghiệm thực tiễn còn hạn chế nên còn tồn tại những thiếu sót khó tránh khỏi, vì vậy chúng em rất mong nhận được những ý kiến đóng góp chân thành từ Thầy và Cô để có thể trao dồi thêm được nhiều kinh nghiệm cho những bài báo cáo sau này.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn Thầy và Cô!

MỤC LỤC

Phần 1: Mở đầu

1.1) Giới thiệu

Thể thao là một phần không thể thiếu trong xã hội loài người. Thể thao không chỉ mang mục đích duy trì, cải thiện kĩ năng và sức khỏe, trau dồi kinh nghiệm cho các vận động viên mà còn tạo ra tinh thần giải trí, đoàn kết cho người xem. Trong đó NCAA Division I Men's Division I Basketball tournament (hay March Maddnes) là một trong những giải đấu bóng bầu dục dành cho các học sinh đại học, cao đẳng hàng đầu Hoa Kỳ được diễn ra hằng năm để tìm ra nhà vô địch. Với sự hấp dẫn của giải đấu, việc dự đoán cái tên nhà vô địch là vấn đề cực kì nan giải của người xem, các tạp chí và đồng thời trong đó có một số nhà khoa học.

Trong nghiên cứu này, chúng em sẽ tập trung vào việc sử dụng KNN, một thuật toán học máy phổ biến trong lĩnh vực phân loại và dự đoán,để dự đoán các đội bóng có khả năng vào vòng 16 đội của giải đấu March Madness. Sử dụng dữ liệu về các trận đấu từ năm 2023 trở về trước, thông tin về đội bóng, và các yếu tố khác, chúng em sẽ thực hiện một phân tích khám phá (EDA) để tìm hiểu các mối quan hệ và mẫu trong dữ liệu, từ đó xây dựng một mô hình dự đoán chính xác.

1.2) Bài toán dự đoán nhà vô địch giải đấu March Madness.

Hiện nay, với việc hiện đại hóa và các thông số của từng trận đấu trong một mùa giải được thông số hóa. Nhu cầu phân tích các thông số trên dựa trên các thuật toán để xác định tiêu điểm và các điểm nóng diễn ra trong các thời điểm của từng trận đấu là một mục đích rất quan trọng cho các ban tổ chức giải đấu nhằm định hướng người xem đến với những điểm nóng trên sân đấu nhanh nhất có thể là một điều hợp lý để thu hút người xem và tạo thêm sức hút , sự hấp dẫn trong từng cặp trận đấu và làm cho trận đấu thêm sôi động.

Các chỉ số trên không chỉ giúp các ban tổ chức trong việc tổ chức các cuộc khảo sát người xem xác định thành tích, các cập trận tâm điểm của mùa giải của các đội bóng mà còn giúp các đội bóng cải thiện chiến thuật, thông số của các cá nhân trong đội để tăng hiệu quả trong việc tập luyện và áp dụng vào những cặp đấu.

Cùng với sự đa dạng của các đội bóng hằng năm, với năm 2023 có sự tham gia của 64 đội bóng đến từ các trường đại học, cao đẳng trên khắp Hoa Kỳ. Các đội được chia thành các bảng như hình sau:

Hình 1.1 Hình mô tả các cặp đấu diễn ra trong giải đấu March Maddness 2023

Để dự đoán các đội bóng có khả năng vào vòng 16 (tức 16 đội). Các chỉ số dưới đây được tổng hợp lại từ những năm trước từ các đội bóng được thông số hóa để dễ dàng thực hiện xử lý dữ liệu.

1.3) Lý do chọn đề tài.

Thể thao nói chung, Bóng rổ nói riêng không chỉ là một môn thể thao, mà còn là một sân chơi của sự cạnh tranh và sự hứng thú từ hàng triệu người hâm mộ trên khắp thế giới. Việc dự đoán kết quả của các trận đấu không chỉ là một nhu cầu giải trí mà còn mang lại nhiều giá trị phân tích và thực tiễn trong lĩnh vực thể thao. Thế nên chúng em đã lựa chọn đề tài này với những lý do sau đây:

Đầu tiên, đam mê thể thao là nguồn cảm hứng chính cho việc chúng em chọn đề tài này. Sự đam mê với bóng rổ không chỉ là một sở thích cá nhân mà còn là nguồn động viên mạnh mẽ để khám phá và nghiên cứu sâu hơn về môn thể thao này. Bằng cách kết hợp niềm đam mê này với mục tiêu nghiên cứu, chúng em tin rằng sẽ có sự cam kết cao và nhiệt huyết hơn trong quá trình thực hiện dự án.

Thứ hai, giải đấu bóng rổ là một hình thức giải trí phổ biến và luôn thu hút sự quan tâm lớn từ cộng đồng. Tính cạnh tranh cao và tính dự đoán của mỗi trận đấu tạo ra một môi trường lý tưởng để áp dụng các phương pháp máy học. Các giải đấu như NBA, EuroLeague và CBA mang lại không chỉ những trận đấu kịch tính mà còn là nguồn dữ liệu phong phú để nghiên cứu và phát triển mô hình dự đoán.

Thứ ba, tính thường xuyên và tổ chức của các giải đấu bóng rổ cung cấp điều kiện thuận lợi cho việc thu thập dữ liệu và xây dựng mô hình. Lịch trình cố định và thông tin chi tiết về các trận đấu trước đó, đội hình, điểm số và kết quả đều dễ dàng truy cập và phân tích. Điều này tạo ra một cơ hội đáng giá để áp dụng các phương pháp máy học và thử nghiệm các mô hình dự đoán.

Cuối cùng, truyền thông phong phú và đa dạng về bóng rổ cung cấp nguồn thông tin đáng tin cậy và hữu ích cho quá trình nghiên cứu. Phân tích trước và sau mỗi trận đấu, dữ liệu thống kê và nhận định từ các chuyên gia thể thao cung cấp một cơ sở lý tưởng để xây dựng và cải thiện các mô hình dự đoán. Chúng em hy vọng rằng nghiên cứu này sẽ đóng góp vào việc hiểu sâu hơn về thể thao và phát triển các ứng dụng thực tiễn trong lĩnh vực máy học và dự đoán.

Thế nên, chúng em đã chọn đề tài "Áp dụng KNN và Neural Network trong Dự Đoán Kết Quả Giải Bóng Rổ" với những lý do như trên. Hy vọng rằng nghiên cứu này sẽ đóng góp vào việc hiểu sâu hơn về thể thao và phát triển các ứng dụng thực tiễn trong lĩnh vực máy học và dự đoán.

1.4) Ứng dụng dự đoán kết quả EDA.

Dự đoán nhà vô địch của giải đấu bóng bầu dục March Madness có thể mang lại một số ứng dụng như sau:

- Dự đoán kết quả trận đấu: Các thuật toán (KNN,...) có thể được sử dụng để dự đoán kết quả của các trận đấu cụ thể trong giải đấu. Bằng cách sử dụng dữ liệu lịch sử về các đội tham gia và kết quả của họ trong các trận đấu trước đó, thuật toán có thể đưa ra dự đoán về đội nào sẽ có khả năng chiến thắng cao hơn trong một trận đấu cụ thể.
- Phân tích dữ liệu, đưa ra các thông số khác: Trước khi sử dụng , việc tiền xử lý dữ liệu và phân tích dữ liệu là rất quan trọng. Việc này có thể giúp phát hiện ra các mẫu và xu hướng trong dữ liệu, giúp tăng cường hiệu suất của mô hình.
- Tối ưu hóa đội hình, chiến thuật cụ thể: Dựa trên dự đoán của thuật toán, các nhà quản lý đội bóng có thể điều chỉnh đội hình của họ, lựa chọn các cầu thủ phù hợp và chiến thuật tốt nhất và điểm lợi của đội tuyển mình và khuyết điểm của đối phương để tăng cơ hội chiến thắng trong giải đấu.
- Đánh giá đội bóng, cá nhân trong mùa giải: Thuật toán có thể được sử dụng để đánh giá sức mạnh của các đội bóng trong giải đấu, từ đó giúp người hâm mộ và nhà phân tích, nhà báo thực hiện các đánh giá tỷ lệ chiến thắng của mỗi đội và đưa ra dự đoán về cơ hội của họ trong việc vô địch.
- Xây dựng hệ thống thông minh: Dựa trên dữ liệu lịch sử và dự đoán của thuật toán, có thể xây dựng các hệ thống thông minh để cung cấp thông tin và dự đoán về giải đấu cho người hâm mộ và các nhà cái, từ đó tăng cường trải nghiệm và hiệu quả của họ.
- Sử dụng các chỉ số để nắm bắt các điểm nóng trong trận đấu: việc nắm bắt các thời điểm then chốt hay điểm nóng trong trận đấu một cách tức thời giúp cho trận đấu trở nên sôi động và hấp dẫn đối với người xem và người hâm mộ giải đấu. Việc đưa ra các chỉ số giữa các khoảng nghĩ giữa thời gian trận đấu giúp các nhà phân tích thực hiện phân tích các tình huống có thể xảy ra, góp phần tăng tính sôi động cũng như tăng tính cạnh tranh của các đội.

Phần 2: Cơ sở lý thuyết

2.1) Thuật toán K-Nearest-Neighbor (KNN)

Định nghĩa

Là một thuật toán phân lớp được sử dụng trong việc nhận dạng mẫu (Pattern Recognition) trong Machine Learning. KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

Một số tên gọi khác:

- Memory-Based Reasoning
- Example-Based Reasoning
- Instance-Base Learning

Lazy Learning

Ý tưởng

Thuật toán KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm củng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp. Dưới đây là các công thức cơ bản để tính khoảng cách giữa 2 điểm x, y với thuộc tính k:

Euclidean:

$$D = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \qquad 2.1.1$$

Manhattan:

$$\mathrm{D} = \sum_{i=1}^{k} |x_i - y_i| \qquad 2.1.2$$

Minkowski:

$$\mathrm{D} = \left(\sum_{i=1}^{k} (|x_i - y_i|)^q\right)^{\frac{1}{q}} \qquad 2.1.3$$

Các bước thực hiện thuật toán:

- Bước 1: Chuẩn bị dữ liệu, dữ liệu này nên được chia thành hai tập: tập dữ liệu huấn luyện và tập dữ liệu kiểm tra. Tập dữ liệu huấn luyện được sử dụng để huấn luyện mô hình, trong khi tập dữ liệu kiểm tra được sử dụng để đánh giá hiệu suất của mô hình.

- Bước 2. Chọn tham số K:

Tham số K = 'số láng giềng gần nhất' sẽ được sử dụng để dự đoán nhãn cho dữ liệu mới. Giá trị K thường được chọn bằng cách thử nghiệm các giá trị khác nhau và chọn giá trị cho độ chính xác cao nhất.

- Bước 3. Tính toán khoảng cách:

Tính toán khoảng cách giữa điểm dữ liệu mới và mỗi điểm dữ liệu trong tập dữ liệu huấn luyện. Có nhiều cách khác nhau để tính khoảng cách, chẳng hạn như khoảng cách Euclidean, khoảng cách Manhattan và khoảng cách Minkowski.

- Bước 4. Xác định K láng giềng gần nhất:

Xác định K láng giềng gần nhất với điểm dữ liệu mới.

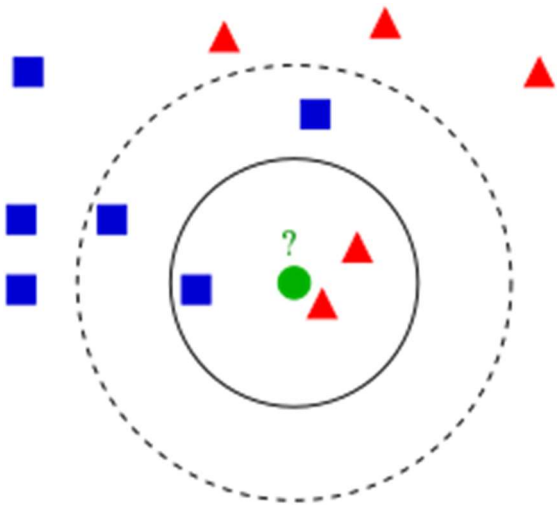
- Bước 5. Dự đoán nhãn:

Dự đoán nhãn cho điểm dữ liệu mới dựa trên nhãn của K láng giềng gần nhất. Có nhiều phương pháp khác nhau để dự đoán nhãn, chẳng hạn như phương pháp bỏ phiếu đa số, phương pháp trung bình trọng số và phương pháp k-nearest neighbors weighted.

- Bước 6. Đánh giá hiệu suất:

Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra. Có nhiều chỉ số khác nhau để đánh giá hiệu suất, chẳng hạn như độ chính xác, độ chính xác, độ thu hồi và điểm F1.

- Ví dụ về thuật toán KNN:



Hình 2.2 Cái nhìn chung về KNN

Trường hợp: sinh nhật của người bạn mới quen sắp đến, tôi muốn tặng quyển sách nhân dịp sinh nhật, nhưng không biết bạn có phải là người thích đọc sách không. Vì thể toi quyết định tìm hiểu những người bạn thân xung quanh của bản ấy để phân tích qua thuật toán KNN để tìm được sở thích của bạn.

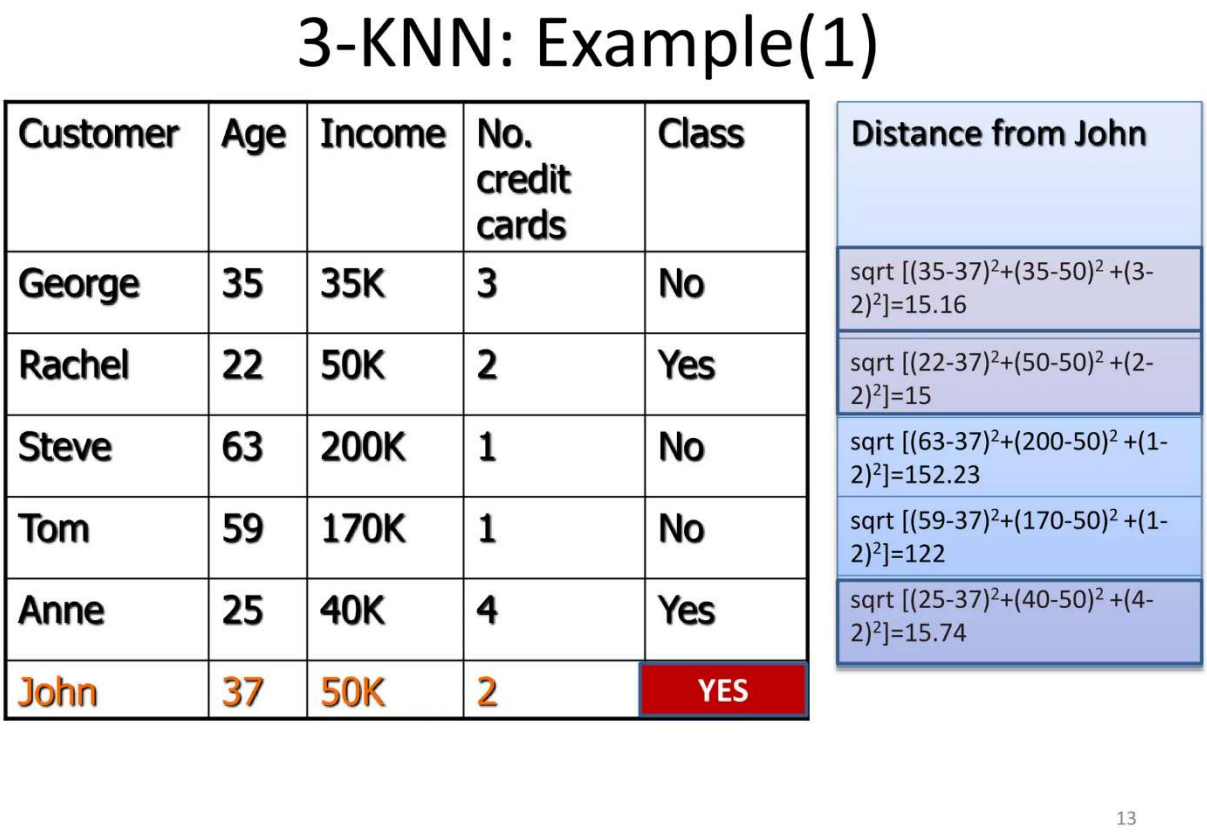
- Ô vuông màu xanh: những người thích đọc sách.
- Tam giác màu đỏ: những người không thích đọc sách.
- Chấm màu xanh: sở thích của người bạn mà tôi muốn phân loại.
- Vòng tròn bên ngoài: đại diện cho mức độ thân thiết của những người bạn thân.

Ta dễ dàng thấy được hình tam giác màu đỏ gần chấm màu xanh nhất, từ đó có thể kết luật người mình tặng quà không thích sách. Tuy nhiên xung quanh có nhiều ô vuông xanh khác nhau nên xét điểm gần nhất chưa phải là tốt nhất nên ta sẽ

xét K điểm gần nhất. Đầu tiền ta xét K=3, ta dễ dàng thấy khi K=3 thì có 2 tam giác đỏ và một vuông xanh, do đó chấm xanh được xếp vào tam giác đỏ.

Tiếp theo ta xét K=5 thì ta được 3 hình vuông xanh và 2 tam giác đó, vì thế chấm xanh được xếp vào lớp ô vuông xanh. Một trường hợp khác, khi ta xét K=4 thì ta có 2 ô vuông xanh và 2 tam giác đỏ, khi đó ta sẽ có điểm bằng nhau, trong trường hợp này thì thuật toán KNN sẽ xử lý bằng cách so sánh tổng khoảng cách của các hình gần nhất so với điểm đang xét. Do xuất hiện trường hợp điểm bằng nhau nên người ta thường chọn K là số lẻ.

- Ví dụ về tính khoảng cách.

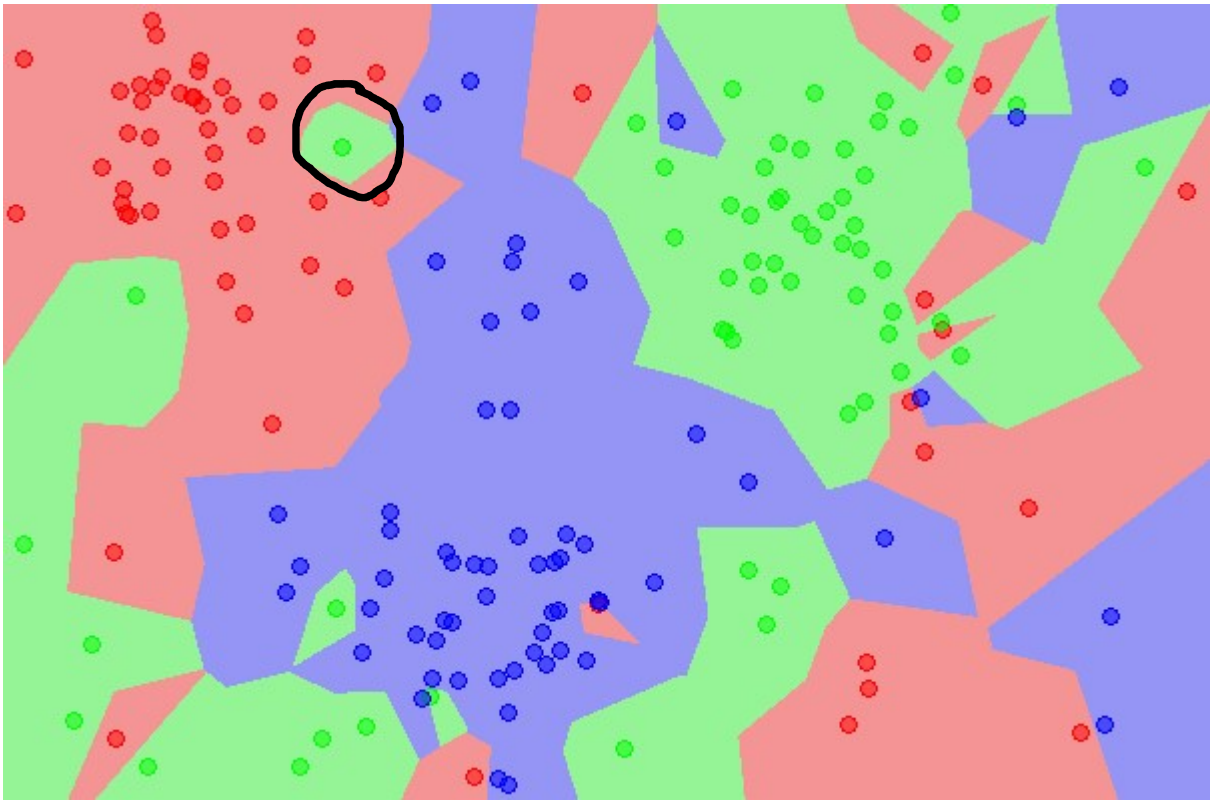


Hình 2.3 Ví dụ về KNN

Ở đây ta sẽ tính khoảng cách của từng Customer đã phân Class để xác định Class của John bằng việc chọn Customer nào có khoảng cách ngắn nhất. Sau khi

tính toán thì Customer Rachel có khoảng cách thấp nhất là 15 vậy ta sẽ phân John vào Class:Yes giống với Rachel.

KNN nhiễu



Hình 2.4 KNN nhiễu

Ví dụ trên đây là bài toán phân loại với 3 loại: Đỏ, Lam, Lục. Mỗi điểm dữ liệu mới sẽ được gán nhãn theo màu của điểm mà nó thuộc về. Trong hình này, có một vài vùng nhỏ xem lẫn vào các vùng lớn hơn khác màu. Ví dụ có một điểm màu Lục đã khoành tròn nằm giữa hai vùng lớn với nhiều dữ liệu màu Đỏ và Lam. Điểm này rất có thể là nhiễu, dẫn đến nếu dữ liệu test rơi vào vùng này sẽ có nhiều khả năng cho kết quả không chính xác.

Ưu điểm và nhược điểm.

Ưu điểm:

- Dễ sử dụng, cài đặt.
- Có thể áp dụng cho nhiều loại dữ liệu.
- Dự đoán kết quả mới dễ dàng nếu mẫu test đủ lớn.

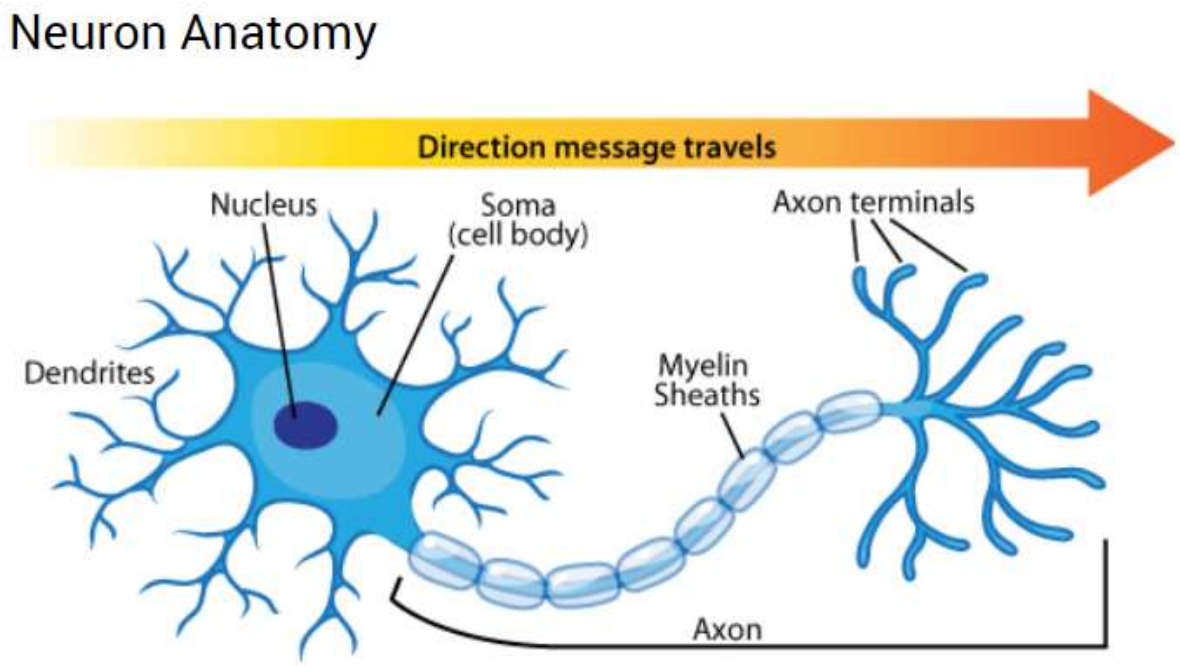
Nhược điểm:

- Tốn nhiều thời gian để phân loại mẫu mới.
- Cần tính toán khoảng cách từ mẫu mới với tất cả các mẫu khác.
- Việc chọn K cần phải kỹ lưỡng(kết quả không chính xác khi K quá nhỏ).
- Cần lượng mẫu lớn để có độ chính xác.

2.2) Neural Network Classification

Định nghĩa

Neural là tính từ của neuron (nơ-ron), network chỉ cấu trúc đồ thị nên neural network (NN) là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh, là một chuỗi những thuật toán được đưa ra để tìm kiếm các mối quan hệ cơ bản trong tập hợp các dữ liệu. Thông qua việc bắt bước cách thức hoạt động từ não bộ con người. Nói cách khác, mạng nơ ron nhân tạo được xem là hệ thống của các tế bào thần kinh nhân tạo.



Hình 2.5 Mạng Nơ Ron

Tuy nhiên NN chỉ là lấy cảm hứng từ não bộ và cách nó hoạt động, chứ không phải bắt chước toàn bộ các chức năng của nó.

Đặc điểm

Neural Network có chứa những lớp bao hàm các nút được liên kết lại với nhau. Mỗi nút lại là một tri giác có cấu tạo tương tự với hàm hồi quy đa tuyến tính. Bên trong một lớp tri giác đa lớp, chúng sẽ được sắp xếp dựa theo các lớp liên kết với nhau. Lớp đầu vào sẽ thu thập các mẫu đầu vào và lớp đầu ra sẽ thu nhận các phân loại hoặc tín hiệu đầu ra mà các mẫu đầu vào có thể phản ánh lại.

Kiến trúc của NN

Mạng Neural Network là sự kết hợp của những tầng perceptron hay còn gọi là perceptron đa tầng. Và mỗi một mạng Neural Network thường bao gồm 3 kiểu tầng là:

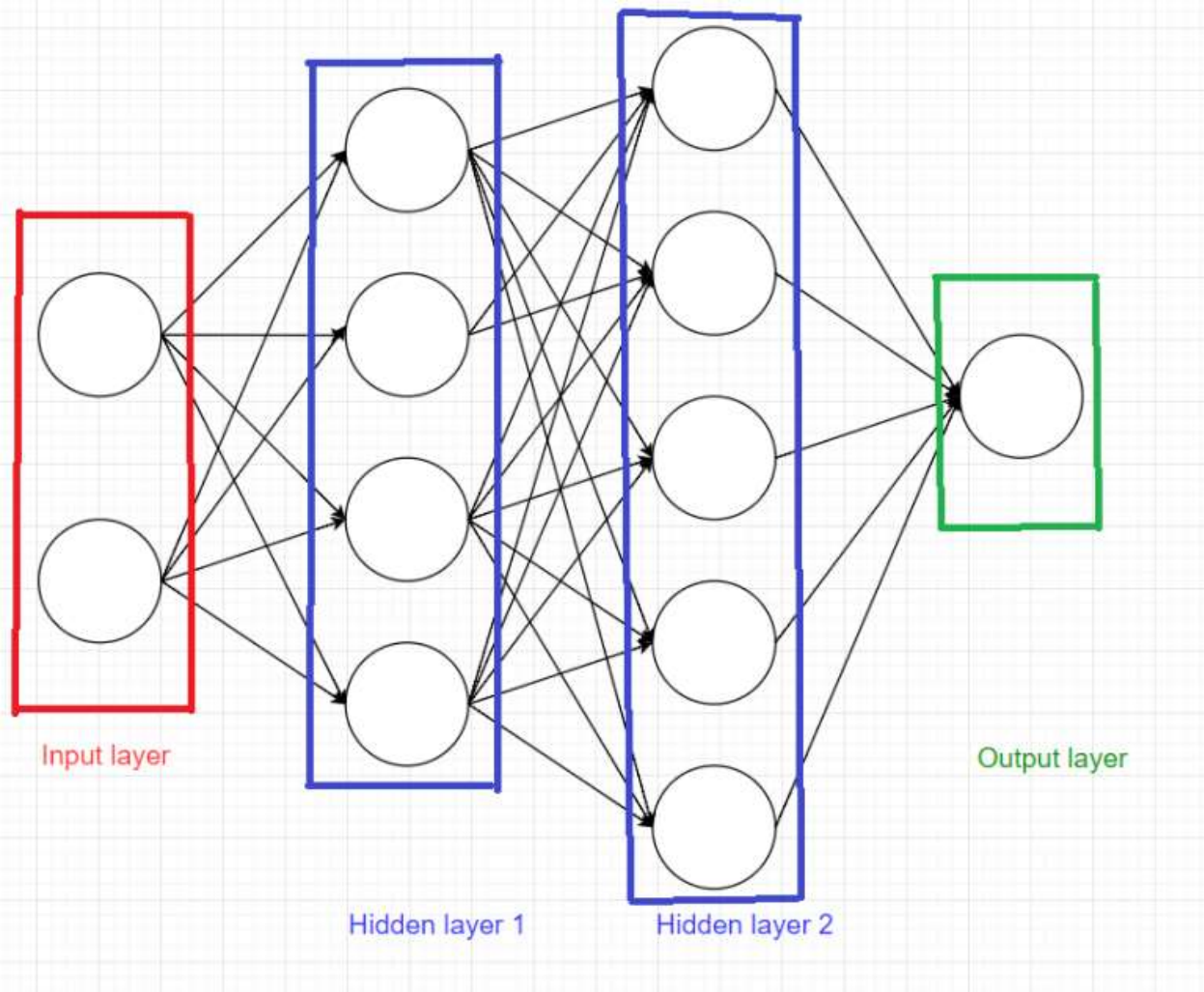
Tầng input layer (tầng vào): Tầng này nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng.

Tầng output layer (tầng ra): Là tầng bên phải cùng và nó thể hiện cho những đầu ra của mạng.

Tầng hidden layer (tầng ẩn): Tầng này nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng.

Các hình tròn được gọi là node.

Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không các hidden layer. Tổng số layer trong mô hình được quy ước là số layer – 1 (Không tính input layer).



Hình 2.6 Cấu trúc của Neural Network

Ứng dụng của Neural Network:

Mạng nơ ron nhân tạo được ứng dụng cho rất nhiều lĩnh vực như: tài chính, giao dịch, phân tích kinh doanh, lập kế hoạch cho doanh nghiệp và bảo trì sản phẩm. Neural Network còn được sử dụng khá rộng rãi cho những hoạt động kinh doanh khác như: dự báo thời tiết, và tìm kiếm các giải pháp nhằm nghiên cứu tiếp thị, đánh giá rủi ro và phát hiện gian lận.

Dự báo thời tiết: Mạng nơ-ron có thể được sử dụng để dự đoán thời tiết trong tương lai dựa trên dữ liệu quan sát từ các cảm biến và mô hình dự đoán thời tiết. Các mạng nơ-ron có thể học các mẫu phức tạp trong dữ liệu như sự biến đổi của áp suất không khí, nhiệt độ, và độ ẩm để đưa ra dự đoán chính xác về thời tiết.

Đánh giá rủi ro: Trong lĩnh vực bảo hiểm và tài chính, mạng nơ-ron có thể được sử dụng để đánh giá rủi ro cho các hợp đồng bảo hiểm hoặc các khoản vay. Bằng cách phân tích các yếu tố như lịch sử thanh toán, thông tin cá nhân và điều kiện thị trường, mạng nơ-ron có thể dự đoán rủi ro và đưa ra các quyết định định giá hoặc chính sách bảo hiểm.

Phát hiện gian lận: Trong các giao dịch tài chính và thanh toán trực tuyến, mạng nơ-ron có thể được sử dụng để phát hiện gian lận. Bằng cách phân tích các mẫu không bình thường trong các giao dịch, mạng nơ-ron có thể cảnh báo về các giao dịch có khả năng là gian lận và giúp ngăn chặn các hoạt động gian lận trước khi xảy ra.

Các bước sử dụng Neural Network:

Chuẩn bị dữ liệu: Thu thập và tiền xử lý dữ liệu cho phù hợp với mạng nơ-ron, bao gồm chuẩn hóa dữ liệu, mã hóa các biến phân loại, và phân chia dữ liệu thành tập huấn luyện, tập kiểm tra và tập validation.

Xây dựng mô hình: Chọn kiến trúc mạng nơ-ron phù hợp cho vấn đề cụ thể của bạn, bao gồm số lượng lớp, số lượng nơ-ron trong mỗi lớp, và các hàm kích hoạt. Sau đó, khởi tạo và huấn luyện mô hình trên tập dữ liệu huấn luyện.

Đánh giá và tinh chỉnh: Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình và điều chỉnh các siêu tham số như tỷ lệ học, số lượng lớp và số lượng nơ-ron để cải thiện hiệu suất.

Kiểm tra và triển khai: Kiểm tra mô hình trên tập dữ liệu kiểm tra độc lập và đảm bảo rằng nó hoạt động đúng đắn. Sau đó, triển khai mô hình vào môi trường thực tế và duy trì nó theo thời gian.

Lợi ích trong việc sử dụng Neural Network:

Khả năng học tập phức tạp: Neural Network có khả năng học được các mẫu phức tạp và không tuyến tính trong dữ liệu, điều này làm cho chúng phù hợp cho các bài toán phân loại có cấu trúc phức tạp.

Tính linh hoạt: Neural Network có thể xử lý các loại dữ liệu khác nhau như dữ liệu số, dữ liệu hình ảnh, dữ liệu văn bản, v.v., và có thể được áp dụng vào nhiều lĩnh vực khác nhau.

Tự động hóa: Một khi được huấn luyện đúng cách, Neural Network có thể tự động phân loại dữ liệu mới mà không cần sự can thiệp của con người.

Hiệu suất cao: Trong một số trường hợp, Neural Network có thể cung cấp hiệu suất phân loại tốt hơn so với các phương pháp truyền thống khác, đặc biệt là đối với dữ liệu phức tạp và không gian chiều cao.

Bất lợi trong việc sử dụng Neural Network:

Yêu cầu dữ liệu lớn: Neural Network thường yêu cầu một lượng lớn dữ liệu huấn luyện để đạt được hiệu suất tốt và tránh overfitting.

Khó hiểu và khó diễn giải: Cấu trúc phức tạp của Neural Network có thể làm cho chúng khó hiểu và khó diễn giải, đặc biệt là đối với các mô hình sâu.

Yêu cầu tính toán cao: Huấn luyện Neural Network đòi hỏi tính toán lớn và có thể mất nhiều thời gian, đặc biệt là đối với các mạng sâu và dữ liệu lớn.

Nguy cơ overfitting: Neural Network có nguy cơ overfitting đặc biệt là khi dữ liệu huấn luyện không đủ hoặc khi mô hình quá phức tạp.

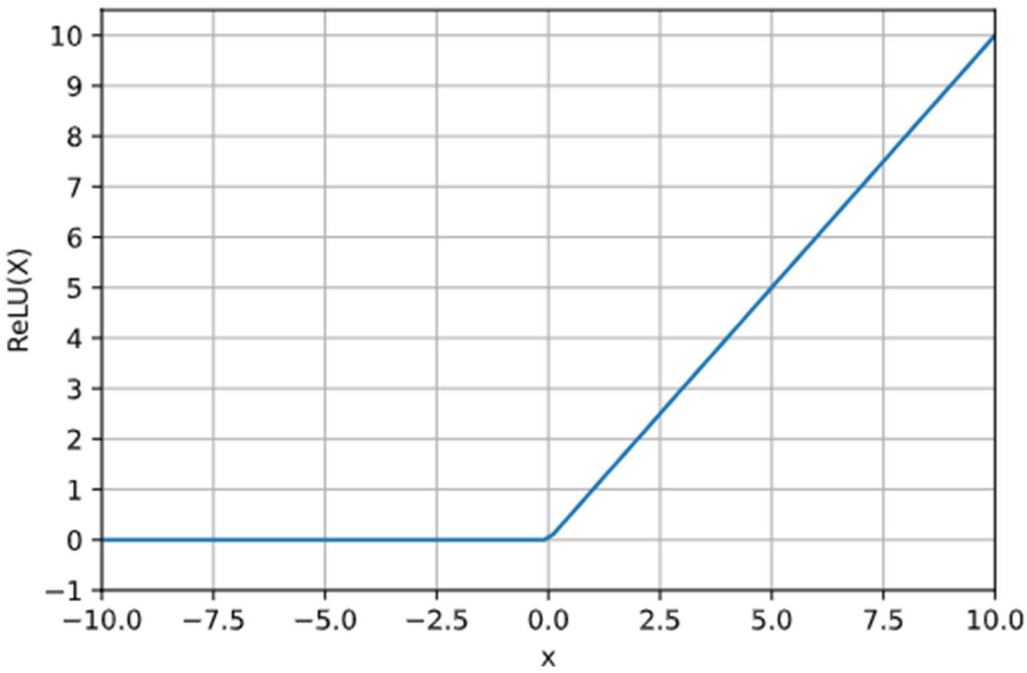
Tóm lại, Neural Network Classification là một công cụ mạnh mẽ cho việc phân loại dữ liệu trong nhiều lĩnh vực khác nhau. Tuy nhiên, việc sử dụng nó đòi hỏi kiến thức chuyên sâu và hiểu biết về các vấn đề cụ thể trong quá trình triển khai. Đồng thời, việc quản lý các bất lợi như yêu cầu dữ liệu lớn và nguy cơ overfitting là quan trọng để đảm bảo hiệu suất và độ tin cậy của mô hình.

Hàm Relu.

Là một trong những hàm kích hoạt (activation function) được sử dụng để huấn luyện trong neural network. Hàm được sử dụng để giải quyết vấn đề về độ dốc biến mất trong các mô hình deep learning. Hàm Relu có công thức sau đây:

$$\mathrm{f(x)} = \max(0, x) \qquad 2.2.1$$

Hàm Relu được sử dụng để lọc các giá trị <0, có thể thấy ở đồ thị sau:



Hình 2.7 đồ thị hàm relu

Hàm Relu có một số ưu điểm sau đây:

- Không bị mất độ dốc: Trong quá trình lan truyền ngược, hàm ReLU không gặp vấn đề mất độ dốc như hàm sigmoid hay hàm tanh.
- Tính toán nhanh, tốc độ hội tụ nhanh: Với hàm ReLU, tính toán đơn giản hơn so với các hàm khác như sigmoid hay tanh. So sánh với hàm sigmoid và tanh, việc không dùng đến các công thức phức tạp như exp của tanh, hay công thức phức tạp của hàm sigmoid. Cùng với đó hàm Relu có tốc độ hội tụ nhanh gấp 6 lần so với hàm tanh đến từ việc không bị bão hòa giá trị hai đầu như hàm tanh và hàm sigmoid.
- Không bị giới hạn đầu ra: Hàm ReLU không bị giới hạn đầu ra trong khoảng [0, 1] như hàm sigmoid hay [-1, 1] như hàm tanh.

Thuật toán tối ưu Adam (Adam - Adaptive Moment Estimation)

Định nghĩa thuật toán tối ưu Adam:

Với mục tiêu cải tiến các phương pháp học trong machine learning, thuật toán Adam đã được xuất hiện vào năm 2014 với mục tiêu cải tiến các phương pháp học đã có từ trước đó. Ví dụ như AdaGrad, SGD, ...

Thuật toán Adam là thuật toán tối ưu được kết hợp từ hai kỹ thuật là RMSP (RMS prop) và momentum. Thuật toán adam sử dụng hai internal states momentum (m) và squared momentum (v) của gradient cho các tham số. Sau mỗi batch huấn luyện, giá trị của m và v được cập nhật lại sử dụng exponential weighted averaging.

Với m và v trên ta có mã giải sau:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)_{g_t} \qquad 2.2.2$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)_{g_t^2} \qquad 2.2.3$$

Mt là thời điểm ước tính đầu tiên (giá trị trung bình).

Vt là ước tính mô men thứ hai (phương sai không tâm) của các gradient tương ứng.

Để chống lại các sai lệch bằng cách tính toán các ước tính mô men thứ nhất và thứ hai được hiệu chỉnh sai lệch bằng công thức dưới đây:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \qquad 2.2.4$$

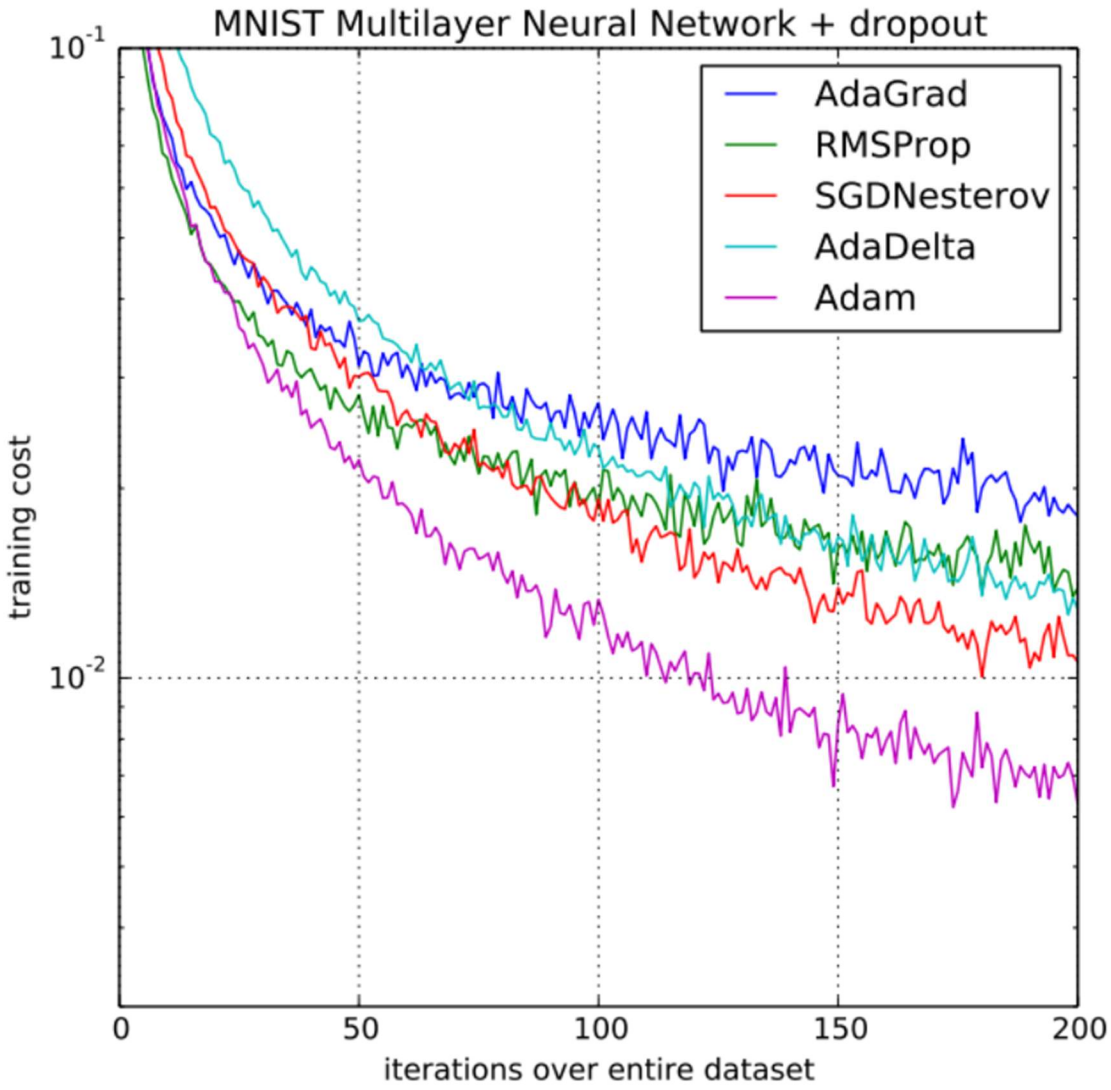
$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \qquad 2.2.5$$

Với beta là một siêu tham số có giá trị được đặt trước khi quá trình học bắt đầu, thì khi cập nhật các giá trị mt và vt thì theta được tính như sau:

$$\theta_t = \theta_t - 1 - \alpha \frac{m_t}{\sqrt{v_t}+\epsilon} \qquad 2.2.6$$

Trong đó, alpha là learning rate, epsilon là giá trị được chèn vào để không bị chia cho số 0.

Do sự có sự kết hợp của nhiều thuật toán, hiệu suất của thuật toánAdam với các thuật toán trước đó được mô tả dưới dạng biểu đồ sau :



Hình 2.8 Biểu đồ hiệu suất của thuật toán Adam so với các thuật toán trước đó

Dựa vào biểu đồ trên ta thấy được training cosst của thuật toán Adam được giảm đáng kể sau mỗi lần thực hiện một minibatch so với các thuật toán khác, vì Adam được kế thừa từ hai kỹ thuật momentum và RMS Prop được giải thích như sau:

+ Tính exponential moving average đạo hàm của giá trị được vào biến m và dùng nó là phần tử số của việc cập nhật hướng.Có ý nghĩa là nếu m có giá trị lớn, thì việc descent đang đi đúng hướng và cần có bước nhảy lớn hơn để đi nhanh hơn. Tương tự, nếu giá trị m nhỏ, phần descent có thể không đi về hướng tối tiểu và thực hiện đi 1 bước nhỏ để thăm dò. Đây là phần momentum của thuật toán.

+ Tính exponential moving average của bình phương đạo hàm của giả trị được lưu vào biến v và sử dụng nó là phần mẫu số của việc cập nhật hướng. Việc này có ý nghĩa như sau: Giả sử gradient mang các giá trị dương, âm lẫn lộn, thì khi cộng các giá trị lại theo công thức tính m ta sẽ được giá trị m gần số 0. Do âm dương lẫn lộn nên nó bị triệt tiêu lẫn nhau. Nhưng trong trường hợp này thì v sẽ mang giá trị lớn. Do đó, trong trường hợp này, chúng ta sẽ không hướng tới cực tiểu, chúng ta sẽ không muốn đi theo hướng đạo hàm trong trường hợp này. Chúng ta để v ở phần mẫu vì khi chia cho một giá trị cao, giá trị của các phần cập nhật sẽ nhỏ, và khi v có giá trị thấp, phần cập nhật sẽ lớn. Đây chính là phần tối ưu RMSProp của thuật toán.

2.3) Exploratory Data Analysis (EDA)

1.Định nghĩa

Là phương pháp khám phá dữ liệu, tìm ra các xu hướng, mẫu thử hoặc kiểm tra các giả định trong dữ liệu nhằm mục đích hiểu rõ về cấu trúc và tính chất của dữ liệu. Khi áp dụng các thuật toán học máy hoặc xây dựng các mô hình dự đoán, EDA góp phần quan trọng trong quá trình xử lý dữ liệu, giúp giải quyết các điều kiện ngoại lệ, giá trị thiếu và những vấn đề ảnh hưởng đến kết quả cuối cùng.

2. Mục đích sử dụng

Mục tiêu của trình bày biểu đồ là để giao tiếp thông tin rõ ràng, toàn vẹn, và hiệu quả hơn. Một biểu đồ được trình bày tốt sẽ khuyến khích sự tham gia của nhiều thành viên trong nhóm, cũng như giúp mọi người tập trung vào bài báo cáo hơn. Với tập dữ liệu đồ sộ, ta cần một cách hiệu quả để có thể hiểu được tính chất của tập dữ liệu đó. Hệ thống thị giác của con người là kênh đón nhận thông tin nhanh chóng và hiệu quả nhất nên việc nắm bắt các nguyên tắc khi trình bày là một kiến thức hữu ích.

Dưới đây là một số mục đích của EDA ta có thể áp dụng vào viêc phân tích dữ liệu:

- Tìm hiểu về cấu trúc dữ liệu: giúp xác định cấu trúc dữ liệu bao gồm số lượng, kiểu dữ liệu, trường dữ liệu, sự liên kết giữa các trường dữ liệu,... Khi xác định được cấu trúc dữ liệu thì có thể hiểu được mối quan hệ giữa các dữ liệu.

- Điều chỉnh và thay đổi: giải quyết các trường hợp thiếu giá trị, dữ liệu lỗi, các ngoại lệ trong dữ liệu.

- Xác định mối tương quan giữa các biến: Các biến đều chứa các giá trị riêng, EDA phát hiện các liên hệ tiềm ẩn và sự ảnh hưởng giữa các biến với nhau, tạo sự liên kết giữa các thông tin dữ liệu nhằm xây dựng một quy trình phân tích tổng thể, rõ ràng.

- Xây dựng cơ sở dữ liệu quan hệ: Các đối tượng dữ liệu quan trọng được phát triển mối quan hệ nhằm cấu trúc hóa dữ liệu theo sơ đồ, tiết kiệm thời gian xử lý những thông tin thừa, hạn chế sự sai sót trong phân tích.

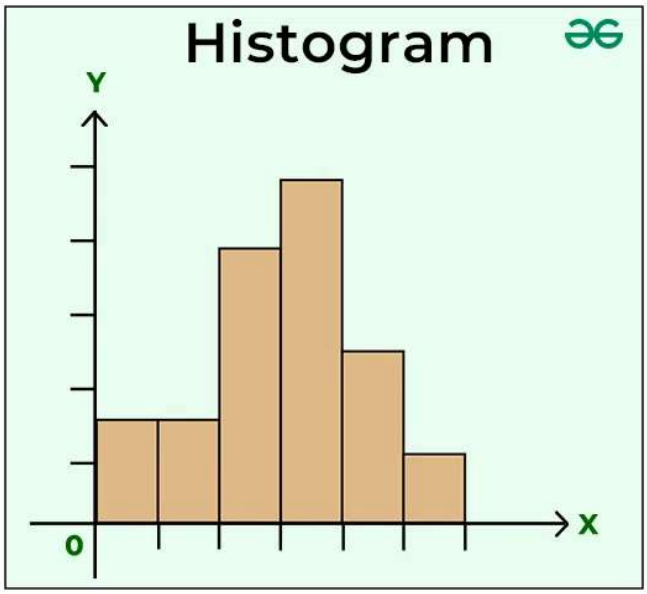
- Chuẩn bị cho bước phân tích tiếp theo: giúp loại bỏ các dữ liệu không cần thiết, dữ liệu thiếu giá trị và chuẩn hóa dữ liệu.

3. Các kỹ thuật được dùng trong EDA

Phần tích đơn biến.

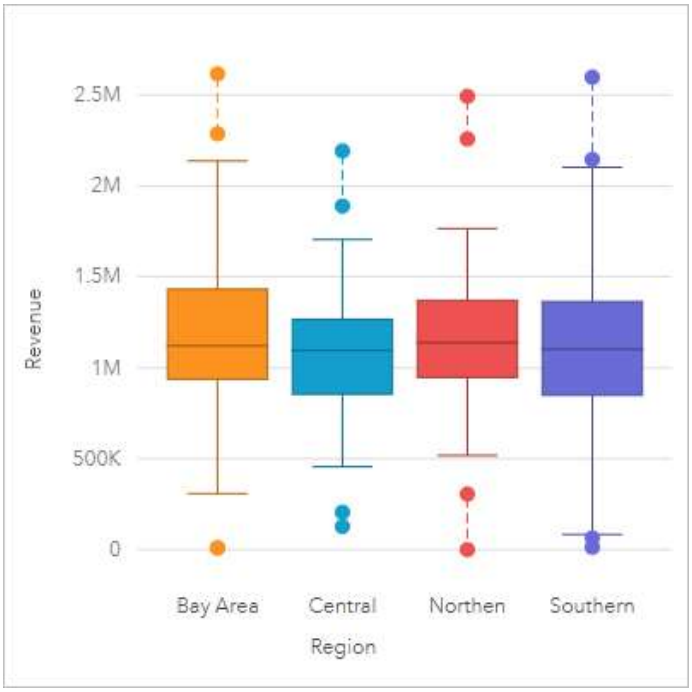
Phân tích đơn biến có mục đích là cho ta hiểu được sự phân bổ của các giá trị cho một biến duy nhất. Dữ liệu đơn biến không theo loại dữ liệu cụ thể mà được phân theo mục đích sử dụng hoặc bản chất riêng. Để phân tích một tập dữ liệu, các loại kỹ thuật phân tích đơn biến sẽ được sử dụng tùy thuộc vào các loại biến đề cập. Một số dạng biểu đồ được sử dụng nhiều trong phân tích đơn biến:

Histograms (Biểu đồ phân phối): Histogram hiển thị tần suất của từng giá trị hoặc nhóm giá trị trong dữ liệu số, xác định đỉnh, đuôi và các thông số thống kê liên quan.



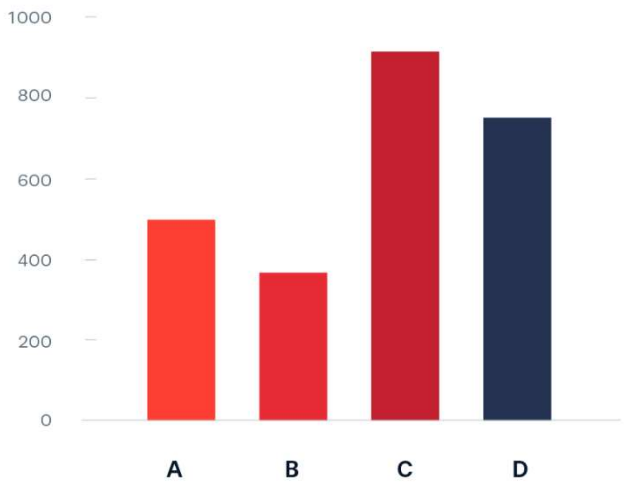
Hình 2.9 Biểu đồ Histogram

Boxplot (Biểu đồ hộp): Một Boxplot sẽ cung cấp một số thông tin quan trọng như phần tối thiểu, giá trị tối đa, giá trị trung vị,... Boxplot còn được sử dụng để xác định các dữ liệu ngoại lệ.



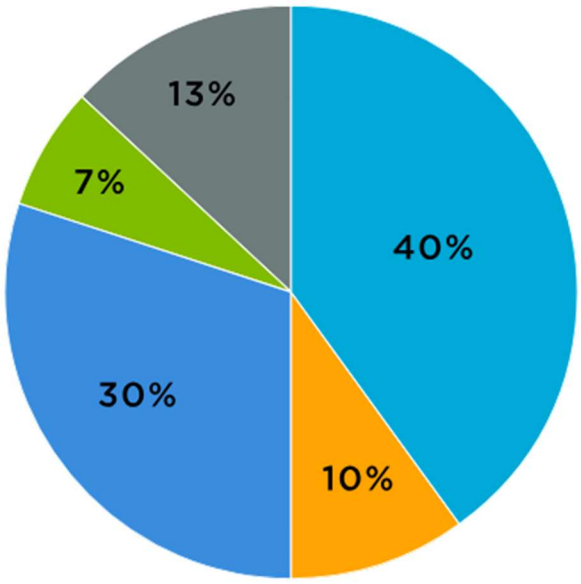
Hình 2.10 Biểu đồ Boxplot

Bar Chart (Biểu đồ cột): Chủ yếu là biểu đồ thanh tần số, được sử dụng để so sánh giá trị của các biến rời rạc và tìm tần suất của các phân loại dữ liệu khác nhau.



Hình 2.11 Biểu đồ Bar Chart

Pie Chart (Biểu đồ tròn): Biểu đồ tròn truyền tải thông tin như biểu đồ cột, khác biệt nằm ở cách thể hiện, với mỗi phần trong hình tròn là biểu thị tỷ lệ của từng danh mục trong dữ liệu.

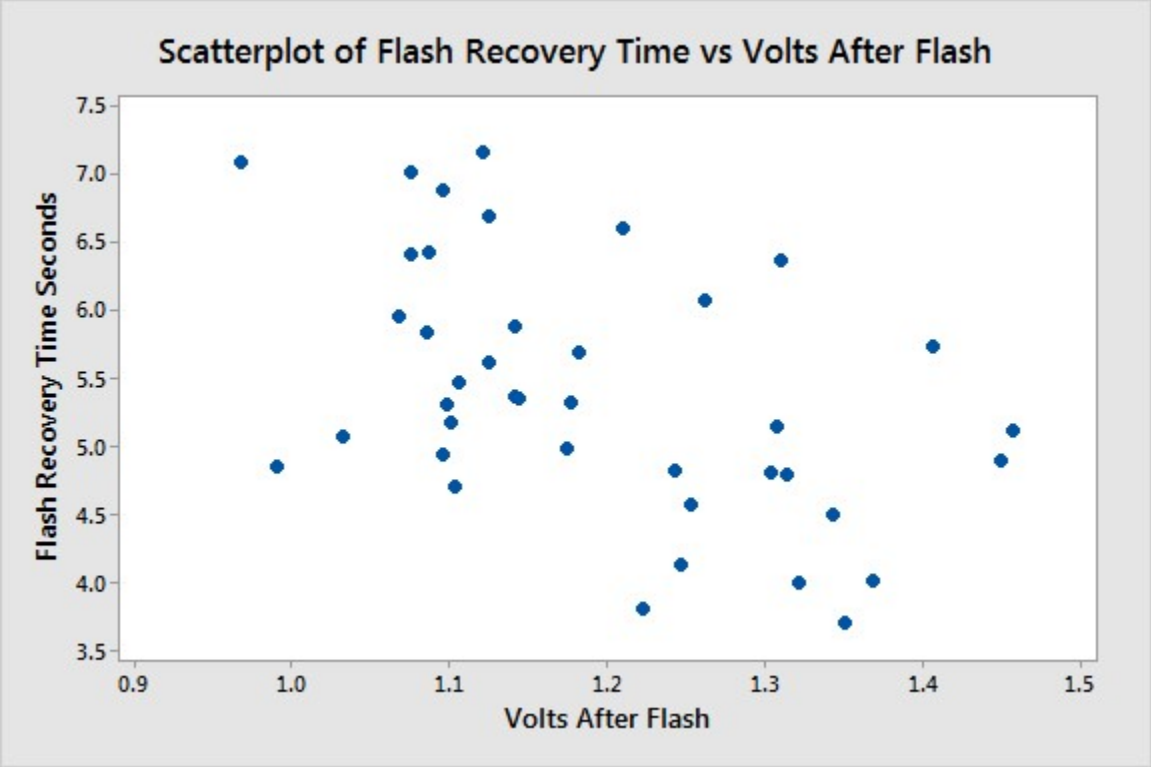


Hình 2.12 Biểu đồ PieChart

Phân tích hai biến

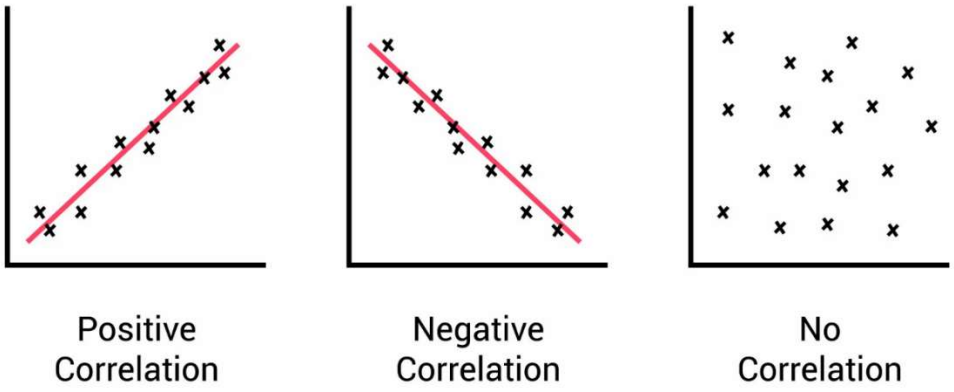
Là phương pháp kiểm tra sự liên quan giữa hai dữ liệu khác nhau, cách thức để xác định xem có mối liên hệ nào giữa hai biến hay không, nếu có thì mối liên hệ đó mạnh đến mức nào và thể hiện theo hướng nào. Đây là kỹ thuật phân tích giúp xác định cách kết nối giữa hai biến và tìm ra xu hướng trong dữ liệu. Các dạng biểu đồ phổ biến được sử dụng cho phân tích hai biến là:

Scatterplots (Biểu đồ phân tán): Biểu đồ phân tán cho biết hai biến có liên quan như thế nào. Thể hiện các giá trị của một biến trên trục X và các giá trị khác của biến trên trục Y.



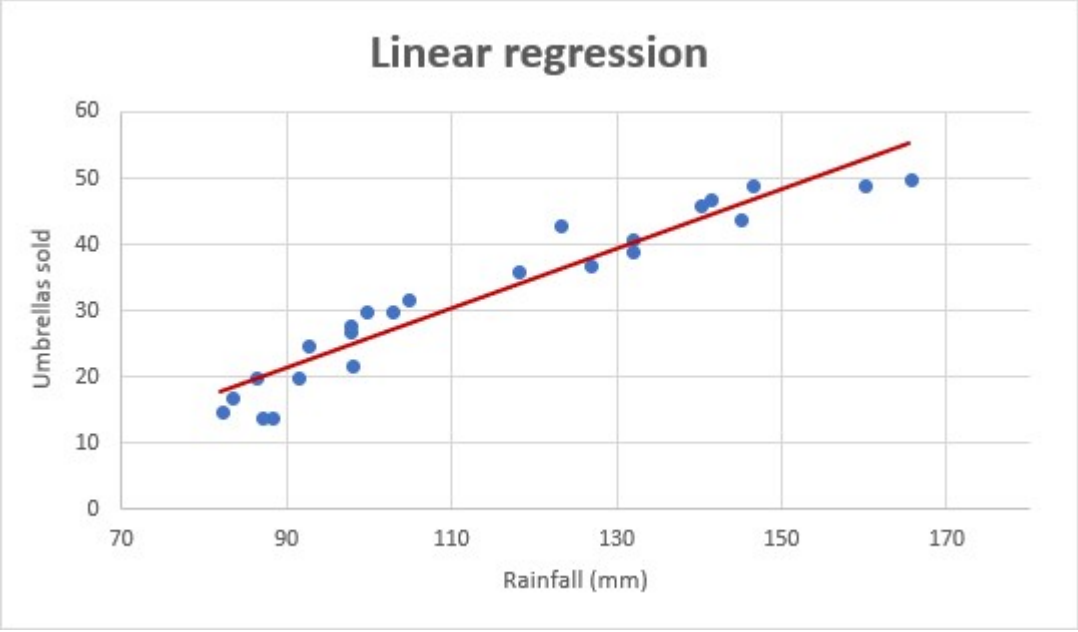
Hình 2.13 Biểu đồ Scatterplot

Correlation (Biểu đồ tương quan): Hệ số tương quan là phép đo thể hiện mức độ mạnh và định hướng của hai biến được liên kết. Mối tương quan tích cực là khi một biến tăng lên, biến còn lại cũng tăng theo. Mối tương quan tiêu cực là khi một biến tăng lên, biến còn lại sẽ giảm.



Hình 2.14 Biểu đồ Correlation

Regression (Biểu đồ phân tích hồi quy): Trong biểu đồ hồi quy, trục X đại diện cho biến độc lập và trục Y đại diện cho biến phụ thuộc. Khi các điểm được thể hiện trên biểu đồ, một đường hồi quy sẽ được vẽ để ước lượng mối quan hệ tuyến tính giữa hai biến.

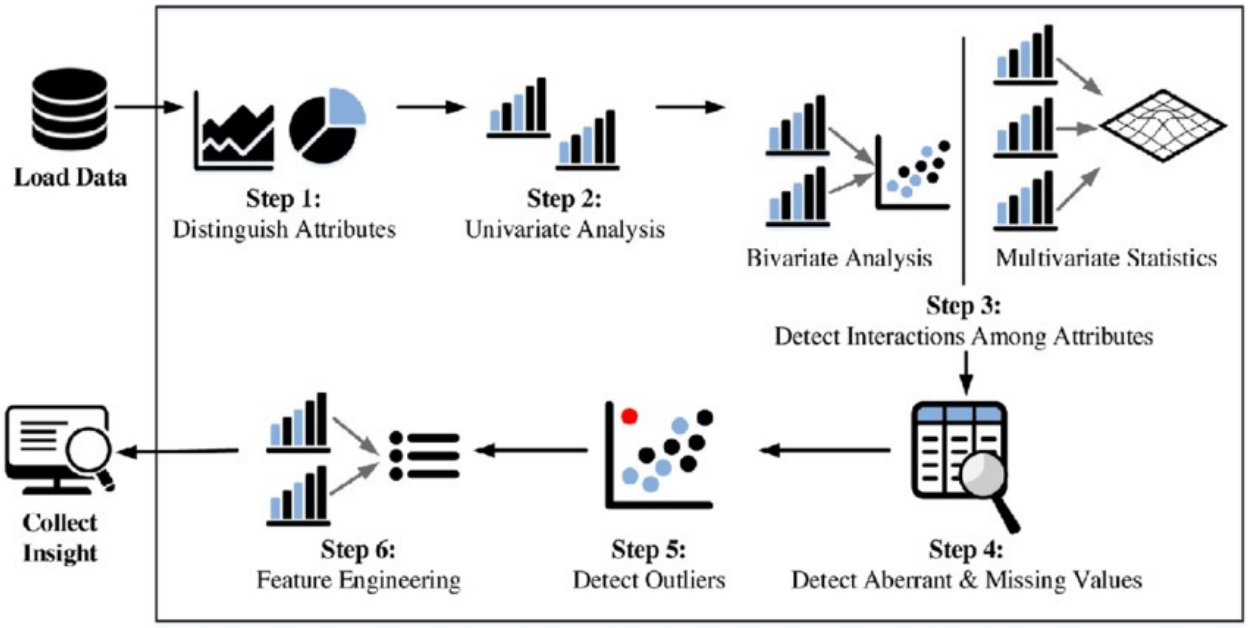


Hình 2.15 Biểu đồ Regression

Phần tích đa biến.

Phân tích đa biến kỹ thuật phân tích ở cấp độ phức tạp hơn, được sử dụng khi có nhiều hơn hai biến trong tập dữ liệu. Phân tích đa biến giúp giảm thiểu và đơn giản hóa dữ liệu mà không làm mất bất kỳ chi tiết quan trọng nào trong tập dữ liệu. Điều quan trọng nhất trong phương pháp này là phải hiểu mối quan hệ giữa các biến dự đoán hành vi của các biến dựa trên quan sát.

Quy trình thực hiện EDA.



Hình 2.19 Quy trình thực hiện EDA

Bước 1 - Thu thập dữ liệu: Thu thập dữ liệu từ các nguồn, sau đó lưu trữ và tổ chức một cách chính xác để các bước tiếp theo được thực hiện một cách nhanh chóng.

Bước 2 - Kiểm tra dữ liệu: Kiểm tra sơ bộ về tệp dữ liệu, xem số lượng, kiểu dữ liệu, thuộc tính dữ liệu và các đặc điểm khác. Quá trình này sẽ giúp các nhà phân tích dữ liệu định hình được các phương án xử lý dữ liệu tiếp theo.

Bước 3 - Xử lý dữ liệu: Ở bước này, các nhà phân tích dữ liệu sẽ thực hiện các phần việc như bổ sung các giá trị thiếu, xóa các giá trị trùng lặp, xử lý các dữ liệu ngoại lệ và chuyển đổi định dạng dữ liệu.

Bước 4 - Trực quan dữ liệu: Sử dụng các kỹ thuật phân tích kết hợp với các biểu đồ để hiểu về các mẫu, xu hướng và mối tương quan giữa các dữ liệu. Tùy vào

mối quan hệ giữa các biến để ứng dụng các kỹ thuật phân tích để khai thác điểm đặc trưng của tệp dữ liệu.

Bước 5 - Đúc kết: Dựa trên các bước đã thực hiện, phân tích và đưa ra kết luận về các dữ liệu đã xử lý. Ghi nhận các mẫu quan trọng đã tìm thấy, trình bày các xu hướng và khía cạnh khác của dữ liệu.

Bước 6 - Báo cáo kết quả: Sử dụng các biểu đồ phân tích, hình ảnh và các mô tả liên quan để báo cáo kết quả dữ liệu một cách chi tiết và rõ ràng.

Phần 3 : Phương Pháp đề xuất

3.1) Các bước tiền xử lý dữ liệu.

Trước khi bắt đầu việc xử lý dữ liệu, việc tiền xử lý dữ liệu là một điều vô cùng quan trọng để chuẩn bị cho việc xử lý diễn ra hiệu quả và đạt hiệu suất cao. Các bước xử lý dữ liệu bao gồm các việc như sau:

- Lọc và xử lý dữ liệu thiếu.

Một trong những vấn đề phổ biến nhất khi làm việc với dữ liệu là sự xuất hiện của dữ liệu thiếu hoặc rỗng. Có nhiều phương pháp để xử lý dữ liệu thiếu, bao gồm thay thế giá trị thiếu bằng giá trị trung bình, giá trị trung vị hoặc xóa các hàng hoặc cột chứa dữ liệu thiếu.

- Chuẩn hóa dữ liệu.

Chuẩn hóa dữ liệu là quá trình biến đổi dữ liệu để đảm bảo rằng tất cả các biến có cùng phạm vi hoặc phân phối. Các phương pháp chuẩn hóa phổ biến bao gồm chuẩn hóa Min-Max và chuẩn hóa Z-score. Tuy nhiên, do dữ liệu đã được chuẩn hóa trừ trước nên chúng em không cần phải chuẩn hóa dữ liệu từ đầu.

- Xử lý và mã hóa dữ liệu hạng mục.

Dữ liệu hạng mục là dạng dữ liệu mà các giá trị của chúng không phải là số, chẳng hạn như loại sản phẩm, giới tính, hoặc màu sắc. Trong quá trình tiền xử lý, dữ liệu hạng mục thường được chuyển đổi thành dạng số để có thể sử dụng trong các mô hình học máy. Trong trường hợp này, chúng em sử dụng biến là sử dụng Local Outlier Factor (LOF). LOF đo lường mức độ "khác biệt" của mỗi điểm dữ liệu so với các điểm xung quanh nó. Các điểm có LOF cao hơn một ngưỡng được xác định có thể được coi là outliers và loại bỏ khỏi tập dữ liệu. Phương pháp này giúp cải thiện tính đồng nhất và hiệu suất của các mô hình dự đoán sau này bằng cách loại bỏ các điểm dữ liệu ảnh hưởng.

- Loại bỏ outliers.

Outliers là các điểm dữ liệu mà có giá trị rất khác biệt so với phần còn lại của dữ liệu. Khi không xử lý outliers, việc phân tích có thể bị sai lệch một khoảng lớn. Cho nên, việc loại bỏ outliers giúp cải thiện tính đồng nhất của dữ liệu và kết quả của phân tích.

3.2) Input, Output của mô hình dự đoán.

Với việc dự đoán các đội có khả năng vào vòng 16 đội cao nhất, việc cần thiết để thu thập dữ liệu được xác minh từ những năm trước đó. Trong bài toán này, chúng em sử dụng dữ liệu được thu thập từ năm 2012 đến năm 2023. Các chỉ số trên được thông số hóa và làm input của mô hình. Dữ liệu trên được

Dữ liệu trên được cung cấp bởi Google, là một trong nữa nhà cung cấp hệ thống điện toán đám mây cho NCAA và được thu thập mỗi năm để tổ chức cuộc thi. ham gia củng cố kiến thức về bóng rổ, thống kê, lập mô hình dữ liệu và công nghệ đám mây. Là một phần trong hành trình chuyển sang đám mây, NCAA đã di chuyển hơn 80 năm dữ liệu lịch sử và từng trận đấu, từ 90 giải vô địch và 24 môn thể thao, sang Google Cloud Platform (GCP). NCAA đã khai thác dữ liệu lịch sử bóng rổ trong nhiều thập kỷ bằng cách sử dụng BigQuery, Cloud Spanner, Datalab, Cloud Machine Learning và Cloud Dataflow để hỗ trợ phân tích hiệu suất của đội và cầu thủ.

Output của mô hình là các đội có khả năng vào vòng 16 đội được đánh số theo dạng 1 và 0. Số 1 là vào vòng 16 đội và số 0 là ngược lại.

3.3) Các thông tin Input của mô hình, tập dữ liệu kiểm tra và tập dữ liệu huấn luyện.

Input của mô hình là tổng hợp các chỉ số hiệu suất của từng đội bóng theo từng năm. Input của dữ liệu gồm 32 dòng, 707 cột.

Các chỉ số được xắp sếp như hình sau:

A	B	C	D	E	F	G	H	I	J
Year	Team	G	W	L	W-L%	SRS	SOS	Conf. W	Conf. L
2,023	Alabama	37	31	6	0.84	23.19	9.65	16	2
2,023	Arizona	35	28	7	0.80	19.08	8.34	14	6

Hình 3.1 Tên các cột input bộ dữ liệu (phần 1)

	K	L	M	N	O	P	Q	R
1	Home W	Home L	Away W	Away L	Team Points	Opp Points	FG%	3P%
2	15	0	9	3	3,027	2,526	0.44	0.34
3	15	2	6	4	2,866	2,490	0.49	0.38

Hình 3.2 Tên các cột input bộ dữ liệu (phần 2)

	S	T	U	V	W	X	Y	Z	AA
1	FT%	Home win rate	Away win rate	Conference win rate	Point diff %	AdjEM	AdjO	AdjD	AdjT
2	0.73	1	0.75	0.89	0.20	27.28	115.50	88.20	72.60
3	0.71	0.88	0.60	0.70	0.15	21.90	118.20	96.30	72

Hình 3.3 Tên các cột input bộ dữ liệu (phần 3)

AB	AC	AD	AE	AF	AG	AH
Luck	SOS AdjEM	OppO	OppD	NCSOS AdjEM	Seed	Made Round of 16
0.06	11.07	110.20	99.20	10.46	1	1
0.03	8.32	107.50	99.20	3.12	2	0

Hình 3.4 Tên các cột input bộ dữ liệu (phần 4)

Sau đây là 100 dòng đầu của một số cột của input:

Year	Team	G	W	L	W-L%	SRS	SOS	Team Points	Opp Points	Home win rate	Away win rate	Seed
2023	Alabama	37	31	6	0.838	23.19	9.65	3027	2526	1	0.75	1
2023	Arizona	35	28	7	0.8	19.08	8.34	2866	2490	0.88	0.6	2
2023	Arizona State	36	23	13	0.639	11.29	8.18	2559	2447	0.67	0.54	11
2023	Arkansas	36	22	14	0.611	15.99	9.87	2666	2446	0.82	0.2	8
2023	Auburn	34	21	13	0.618	14.35	9.29	2474	2302	0.88	0.33	9
2023	Baylor	34	23	11	0.676	17.3	10.54	2619	2389	0.82	0.5	3
2023	Boise State	34	24	10	0.706	12.78	6.21	2451	2198	0.93	0.5	10
2023	Colgate	35	26	9	0.743	2.2	-5.86	2733	2426	0.88	0.73	15
2023	Connecticut	39	31	8	0.795	22.95	8.51	3064	2501	0.88	0.55	4
2023	Creighton	37	24	13	0.649	17.83	9.8	2828	2531	0.87	0.45	6
2023	Drake	35	27	8	0.771	7.84	-1.14	2617	2236	0.93	0.5	12
2023	Duke	36	27	9	0.75	15.83	7.44	2592	2290	1	0.4	5
2023	FDU	37	21	16	0.568	-8.98	-9.83	2863	2744	0.67	0.47	16
2023	Florida Atlantic	39	35	4	0.897	13.92	2.3	3035	2547	1	0.79	9
2023	Furman	36	28	8	0.778	4.89	-2.9	2911	2563	0.88	0.73	13
2023	Gonzaga	37	31	6	0.838	18.99	8.04	3187	2715	0.94	0.78	3

2023	Grand Canyon	36	24	12	0.667	4.43	1.27	2706	2421	0.78	0.5	14
2023	Houston	37	33	4	0.892	22.2	4.79	2770	2126	0.89	1	1
2023	Howard	35	22	13	0.629	-5.04	-5.14	2634	2542	0.86	0.43	16
2023	Illinois	33	20	13	0.606	14.81	7.65	2452	2216	0.88	0.3	9
2023	Indiana	35	23	12	0.657	14.94	8.88	2616	2404	0.88	0.42	4
2023	Iona	35	27	8	0.771	7.61	-2.76	2660	2297	0.92	0.67	13
2023	Iowa	33	19	14	0.576	14.08	8.69	2642	2464	0.82	0.36	8
2023	Iowa State	33	19	14	0.576	15.36	10.39	2231	2067	0.81	0.27	6
2023	Kansas	36	28	8	0.778	19.2	11.84	2715	2450	0.94	0.64	1
2023	Kansas State	36	26	10	0.722	15.79	9.48	2742	2515	0.94	0.36	3
2023	Kennesaw State	35	26	9	0.743	0.6	-2.06	2626	2415	0.94	0.63	14
2023	Kent State	35	28	7	0.8	7.53	-1.22	2665	2305	1	0.6	13
2023	Kentucky	34	22	12	0.647	14.76	8.08	2532	2305	0.78	0.6	6
2023	Louisiana	34	26	8	0.765	4.27	-0.51	2630	2362	1	0.5	13
2023	Marquette	36	29	7	0.806	17.07	8.01	2856	2530	0.94	0.69	2
2023	Maryland	35	22	13	0.629	14.58	8.35	2440	2222	0.94	0.18	8
2023	Memphis	35	26	9	0.743	14.5	6.9	2778	2512	0.87	0.58	8
2023	Miami (FL)	37	29	8	0.784	13.98	6.84	2925	2661	0.94	0.64	5
2023	Michigan State	34	21	13	0.618	14.25	11.25	2411	2309	0.86	0.42	7
2023	Mississippi State	34	21	13	0.618	11.43	6.69	2234	2073	0.75	0.4	11
2023	Missouri	35	25	10	0.714	11.41	6.89	2762	2604	0.84	0.5	7
2023	Montana State	35	25	10	0.714	2.18	-1.43	2588	2342	0.92	0.67	14
2023	NC State	34	23	11	0.676	12.5	5.58	2643	2408	0.88	0.4	11
2023	Nevada	33	22	11	0.667	10.01	6.41	2397	2238	0.93	0.46	11
2023	Northern Kentucky	35	22	13	0.629	-0.83	-2.89	2372	2224	0.82	0.5	16
2023	Northwestern	34	22	12	0.647	13.68	8.71	2303	2134	0.72	0.64	7

2023	Oral Roberts	35	30	5	0.857	8.55	-2.84	2915	2458	1	0.73	12
2023	Penn State	37	23	14	0.622	13.04	9.02	2673	2524	0.76	0.36	10
2023	Pittsburgh	36	24	12	0.667	10.59	5.2	2704	2510	0.82	0.58	11
2023	Princeton	32	23	9	0.719	4.76	-0.97	2416	2192	0.8	0.64	15
2023	Providence	33	21	12	0.636	12.89	6.55	2552	2343	0.88	0.5	11
2023	Purdue	35	29	6	0.829	18.23	8.31	2543	2196	0.88	0.73	1
2023	Saint Mary's (CA)	35	27	8	0.771	17.41	7.18	2473	2105	0.89	0.75	5
2023	San Diego State	39	32	7	0.821	15.81	8.92	2775	2475	0.94	0.8	5
2023	Southeast Missouri State	36	19	17	0.528	-7.4	-6.81	2789	2734	0.69	0.31	16
2023	Southern California	33	22	11	0.667	13.06	8	2391	2224	0.88	0.5	10
2023	TCU	35	22	13	0.629	15.61	8.78	2634	2395	0.76	0.3	6
2023	Tennessee	36	25	11	0.694	20.84	7.98	2547	2084	0.88	0.4	4
2023	Texas	38	29	9	0.763	20.56	10.43	2963	2578	0.94	0.4	2
2023	Texas A&M	35	25	10	0.714	13.76	7.42	2549	2327	0.94	0.64	7
2023	Texas A&M-Corpus Christi	35	24	11	0.686	-2.57	-7.02	2803	2561	0.87	0.53	16
2023	Texas Southern	35	14	21	0.4	-10.56	-5.05	2422	2518	0.54	0.19	16
2023	UC Santa Barbara	35	27	8	0.771	3.3	-2.03	2510	2305	0.8	0.71	14
2023	UCLA	37	31	6	0.838	22.11	8.66	2743	2245	1	0.82	2

2023	UNC Asheville	35	27	8	0.771	-1.74	-4.35	2604	2418	1	0.63	15
2023	Utah State	35	26	9	0.743	14.23	6.81	2736	2446	0.88	0.6	10
2023	Vermont	34	23	11	0.676	1.63	-3.13	2471	2276	0.93	0.6	15
2023	Virginia	33	25	8	0.758	13.28	5.98	2237	1996	0.94	0.55	4
2023	Virginia Common wealth	35	27	8	0.771	9.89	1.95	2480	2202	0.83	0.73	12
2023	West Virginia	34	19	15	0.559	15.95	10.89	2583	2411	0.76	0.27	9
2023	Xavier	37	27	10	0.73	16.03	9.16	2995	2741	0.88	0.64	3
2022	Akron	34	24	10	0.706	0.91	-3.93	2402	2130	0.8	0.67	13
2022	Alabama	33	19	14	0.576	14.62	11.59	2623	2523	0.81	0.3	6
2022	Arizona	37	33	4	0.892	22.75	6.84	3107	2518	1	0.75	1
2022	Arkansas	37	28	9	0.757	16.27	8.62	2810	2527	0.94	0.56	4
2022	Auburn	34	28	6	0.824	19.2	8.17	2660	2285	1	0.73	2
2022	Baylor	34	27	7	0.794	21.73	8.76	2619	2178	0.88	0.73	1
2022	Boise State	35	27	8	0.771	11.93	5.22	2389	2130	0.81	0.75	8
2022	Bryant	32	22	10	0.688	-4.8	-7.41	2497	2348	0.93	0.5	16
2022	Cal State Fullerton	32	21	11	0.656	-0.04	-2.28	2239	2129	0.85	0.5	15
2022	Chattanooga	35	27	8	0.771	6.12	-0.94	2596	2253	0.8	0.75	13
2022	Colgate	35	23	12	0.657	0.98	-6.37	2646	2349	0.94	0.44	14
2022	Colorado State	31	25	6	0.806	11.51	4.77	2275	2047	0.93	0.67	6
2022	Connecticut	33	23	10	0.697	16.4	6.88	2469	2155	0.88	0.5	5
2022	Creighton	35	23	12	0.657	11.34	8.54	2422	2324	0.73	0.55	9
2022	Davidson	34	27	7	0.794	10.43	1.92	2567	2244	0.93	0.75	10
2022	Delaware	35	22	13	0.629	0.32	-1.82	2568	2456	0.64	0.62	15
2022	Duke	39	32	7	0.821	19.55	7.26	3122	2643	0.83	0.82	2
2022	Georgia State	29	18	11	0.621	-1.11	-1.34	2049	1906	0.67	0.55	16

2022	Gonzaga	32	28	4	0.875	25.46	4.5	2790	2119	1	0.83	1
2022	Houston	38	32	6	0.842	22.55	6.47	2844	2233	0.94	0.7	5
2022	Illinois	33	23	10	0.697	16.69	9.48	2456	2218	0.81	0.64	4
2022	Indiana	35	21	14	0.6	12.71	8.08	2478	2316	0.78	0.27	12
2022	Iowa	36	26	10	0.722	18.9	6.87	2995	2562	0.83	0.5	5
2022	Iowa State	35	22	13	0.629	12.38	9.33	2296	2189	0.74	0.4	11
2022	Jacksonville State	32	21	11	0.656	-0.19	-3.79	2353	2151	0.73	0.64	15
2022	Kansas	40	34	6	0.85	22.28	11.3	3129	2690	0.94	0.6	1
2022	Kentucky	34	26	8	0.765	20.9	8.02	2701	2263	1	0.55	2
2022	Longwood	33	26	7	0.788	-0.35	-6.11	2499	2170	0.94	0.58	14
2022	Louisiana State	34	22	12	0.647	17.45	8.3	2467	2156	0.88	0.27	6

3.4) Đề xuất mô hình sử dụng.

Để thực hiện việc dự đoán các đội có khả năng vào vòng 16. Vì đây là dạng EDA, việc xử lý các dữ liệu có tính chất đặc trưng và các mối quan hệ , cấu trúc của dữ liệu cần có cái nhìn tổng quan về chi tiết và tính chất của nó. Mặc dù EDA phần lớn thiên về tập trung vào việc sử dụng các phương pháp thông kê mô tả và trực quan hóa dữ liệu, việc áp dụng một số mô hình dự đoán cũng góp phần giải quyết hay cung cấp một số thông tin có giá trị nhất định trong việc xác định các mối quan hệ giữa các biến.

Với những việc cần áp ứng như trên,chúng em đề xuất hai mô hình dự đoán là KNN Classification, KNN và Neural network Classification.KNN Classification (binary classification) phân loại các đội có khả năng vào vòng 16, KNN Regressor (Logistic Regression) để đưa ra tỉ lệ các đội có khả năng vào vòng 16. Neural network Classification sử dụng để đưa ra tỉ lệ và thực hiện phân loại nhị phân sử dụng trình tối ưu hóa Adam để đưa ra các đội có khả năng vào vòng 16 theo độ chính xác trong một epoch. Neural network sẽ thực hiện huấn luyện dữ liệu đầu vào theo số epoch nhất định.

Phần 4: Thực nghiệm và đánh giá kết quả.

4.1) Bộ dữ liệu đội bóng.

Các đội bóng được chia thành các hạt giống (Seeds) được đánh dấu bên phải hình 1.1. Ví dụ trong trận chung kết, Uconn đã đánh bại San Diego St. để đăng quang ngôi vô địch. Thông thường, các seed càng nhỏ thường sẽ ít hơn các seed cao. Việc seed cao chạm tráng seed nhỏ một phần sẽ giúp các seed cao có lợi thế. Tuy nhiên sẽ không tránh được các bất ngờ. Ví dụ việc Priceton (seed 15) bất ngờ đánh bại Arizona (seed 2) ở vòng 64 đội. Các chỉ số được giải thích lần lượt như sau:

- Team (Đội): Tên của đội tuyển bóng đầu dục của các trường đại học, cao đẳng tham gia giải đấu.
- G (Số trận đấu): Tổng số trận đấu mà đội bóng đã tham gia.
- W (Số trận thắng): Tổng số trận đấu mà đội đã giành chiến thắng.
- L (Số trận thua): Tổng số trận đấu mà đội đã thua.
- W-L% (Tỉ lệ số trận thắng trên tổng số trận đấu): Phần trăm trận thắng của đội, được tính bằng cách chia số trận thắng cho tổng số trận đấu.
- SRS (Simple Rating System): Hệ thống đánh giá đơn giản sử dụng điểm số trung bình và sức mạnh của lịch thi đấu để đo lường chất lượng của một đội.
- SOS (Strength of Schedule): Mức độ khó khăn của lịch thi đấu, được tính bằng cách xem xét điểm số trung bình của đối thủ.
- Conf. W (Số trận thắng): Số trận thắng mà đội đã có .
- Conf. L (Số trận thua): Số trận thua mà đội đã gặp.
- Home W (Số trận thắng ở đội nhà): Số trận thắng mà đội đã có khi chơi ở sân nhà.
- Home L (Số trận thua ở đội nhà): Số trận thua mà đội đã gặp khi chơi ở sân nhà.
- Away W (Số trận thắng khi thi đấu với tư cách đội khách): Số trận thắng mà đội đã có khi thi với tư cách là đội khách.
- Away L (Số trận thua khi thi đấu với tư cách đội khách): Số trận thua mà đội đã gặp khi thi với tư cách là đội khách.
- Team Points (Điểm ghi được bởi đội): Tổng số điểm mà đội đã ghi được trong tất cả các trận đấu.
- Opp Points (Điểm ghi được bởi đối thủ): Tổng số điểm mà đối thủ của đội đã ghi được trong tất cả các trận đấu.
- FG% (Tỉ lệ ném trúng): Phần trăm bóng ném trúng của đội.

- 3P% (Tỉ lệ ném 3 điểm trúng): Phần trăm bóng ném 3 điểm trúng của đội.
- FT% (Tỉ lệ ném phạt trúng): Phần trăm bóng phạt trúng của đội.
- Home win rate (Tỉ lệ thắng khi thi đấu ở nhà): Phần trăm trận thắng của đội khi chơi ở sân nhà.
- Away win rate (Tỉ lệ thắng khi thi đấu ở đội khách): Phần trăm trận thắng của đội khi chơi ở đội khách.
- Conference win rate (Tỉ lệ thắng): Phần trăm trận thắng của đội.
- Point diff % (Phần trăm chênh lệch điểm số): Phần trăm chênh lệch giữa điểm ghi được và điểm ghi của đối thủ.
- AdjEM (Điểm hiệu chỉnh): Số điểm được hiệu chỉnh để đo lường hiệu suất của đội.
- AdjO (Tấn công hiệu chỉnh): Hiệu suất tấn công của đội sau khi được điều chỉnh.
- AdjD (Phòng thủ hiệu chỉnh): Hiệu suất phòng thủ của đội sau khi được điều chỉnh.
- AdjT (Số lần tấn công hiệu chỉnh): Số lần tấn công được điều chỉnh của đội.
- Luck (May mắn): Mức độ may mắn mà đội đã gặp phải, được đánh giá dựa trên kết quả so sánh với dự đoán.
- SOS AdjEM (Số hiệu chỉnh về sức mạnh của lịch thi đấu): Điểm số được điều chỉnh dựa trên sức mạnh của lịch thi đấu.
- OppO (Tấn công của đối thủ): Hiệu suất tấn công của đối thủ của đội.
- OppD (Phòng thủ của đối thủ): Hiệu suất phòng thủ của đối thủ của đội.
- NCSOS AdjEM (Số hiệu chỉnh về sức mạnh của lịch thi đấu): Điểm số được điều chỉnh dựa trên sức mạnh của lịch thi đấu.
- Seed (Hạng của đội trong giải đấu): Hạng của đội trong giải đấu, thường dựa trên xếp hạng của các đội bóng trong các trong mùa giải trước.
- Made Round of 16 (Có vượt qua vòng 16 hay không):Có đội đã vượt qua vòng 16 của giải đấu hay không.

4.2) Simple Rating System (SRS)

Chỉ số SRS hay Simple Rating System (có tên gọi khác là Sports Rating System) là một hệ thống xếp hạng thể thao là hệ thống phân tích kết quả của các cuộc thi thể thao để cung cấp xếp hạng cho mỗi đội hoặc người chơi. Các hệ thống phổ biến bao gồm các cuộc thăm dò của các nhà chuyên gia, thu thập ý kiến từ người không chuyên và các hệ thống máy tính. Xếp hạng, hoặc các chỉ số sức mạnh, là các biểu đồ số hóa về sức mạnh cạnh tranh, thường có thể so sánh trực tiếp để dự đoán

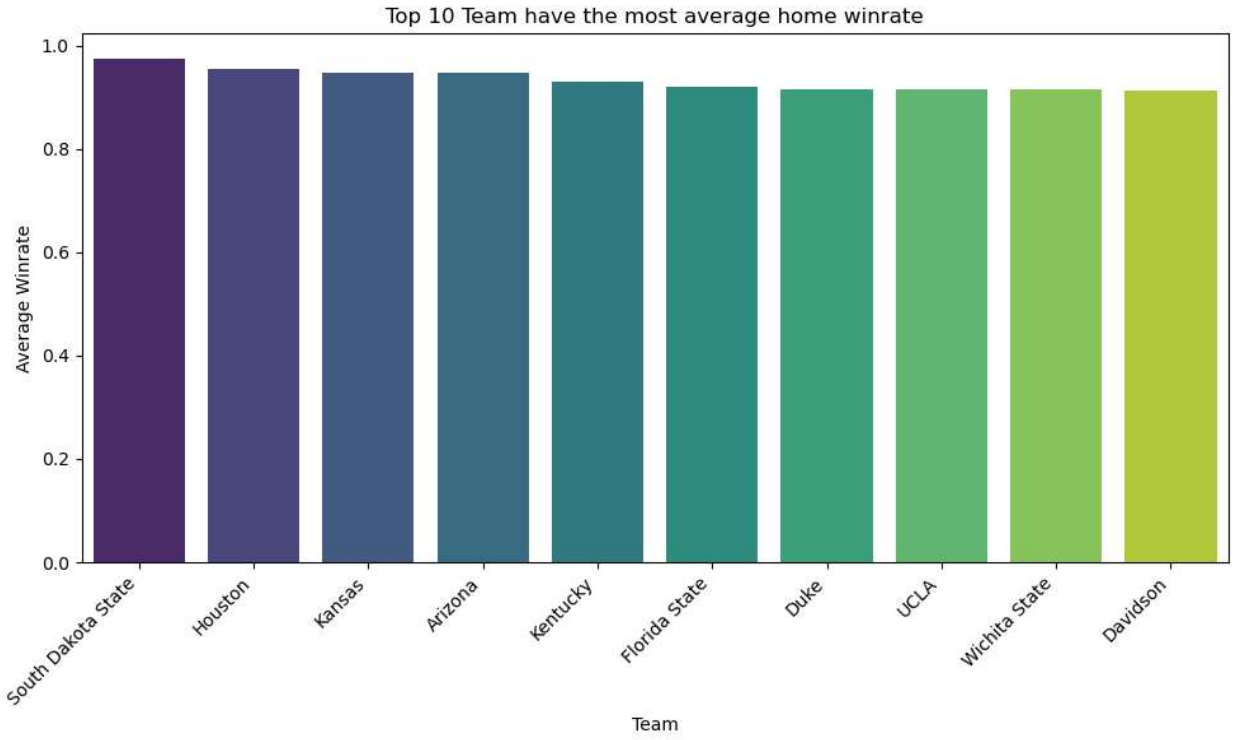
kết quả trận đấu giữa hai đội hoặc hai người chơi. Các hệ thống xếp hạng cung cấp một lựa chọn thay thế cho bảng xếp hạng thể thao truyền thống dựa trên tỷ lệ thắng-thua-hòa.

Hệ thống xếp hạng trên đã có từ 80 năm trước khi mà việc tính toán xếp hạng được thực hiện trên giấy chứ không phải bằng máy tính như hầu hết ngày nay. Một số hệ thống máy tính cũ vẫn đang được sử dụng hiện nay bao gồm: hệ thống của Jeff Sagarin, hệ thống của New York Times và Chỉ số Dunkel, được thành lập từ năm 1929. Trước sự ra đời của giải đấu bóng đá đại học, việc xác định các đội tham dự trận chung kết của Bowl Championship Series đã được xác định thông qua sự kết hợp giữa các cuộc thăm dò của các chuyên gia và các hệ thống máy tính.

SRS được xác định bởi:

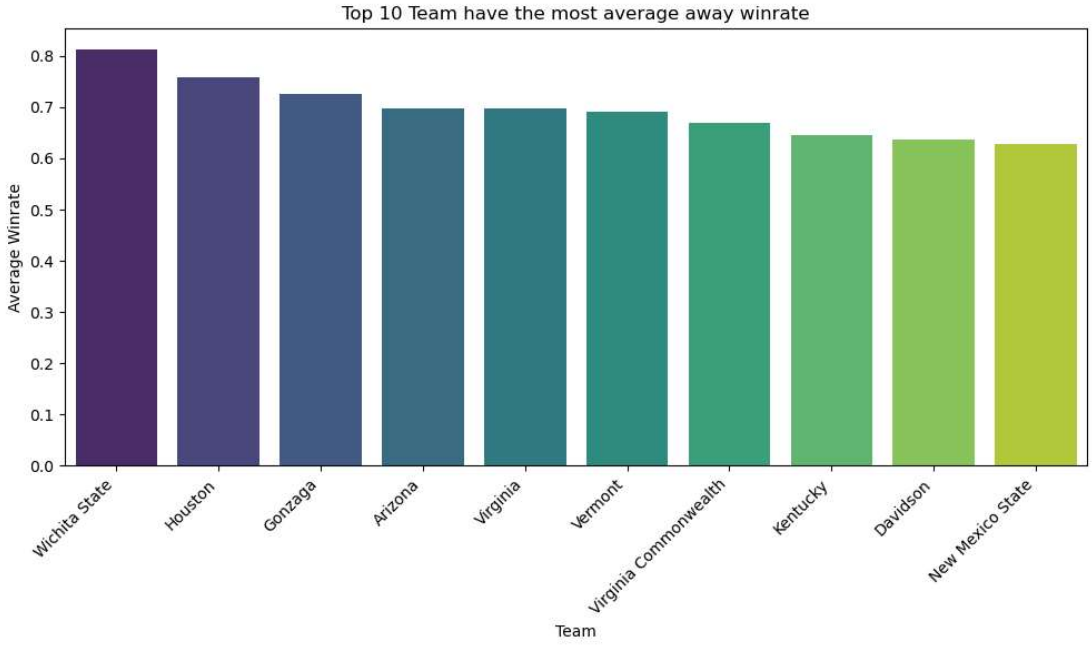
+ Lợi thế sân nhà:

Khi hai đội có chất lượng tương đương thi đấu, đội chơi tại nhà thường chiến thắng nhiều hơn. Việc này có thể thay đổi dựa trên thời kỳ thi đấu, loại trận đấu, độ dài mùa giải, môn thể thao, thậm chí số múi giờ được vượt qua. Nhưng trong mọi điều kiện, "việc chỉ đơn giản là thi đấu tại nhà đã tăng cơ hội chiến thắng". Việc giành chiến thắng khi đối đầu xa nhà được xem tích cực hơn so với việc giành chiến thắng tại nhà, vì nó khó khăn hơn. Lợi thế chơi tại nhà (mà đối với các môn thể thao được thi đấu trên sân thường được gọi là "lợi thế sân nhà") cũng phụ thuộc vào các đặc điểm của sân vận động và khán giả. Ví dụ, đây là hình biểu thị tỉ lệ chiến thắng khi ở sân nhà của các đội có số lần tham dự trên 5 lần:



Hình 4.1 Tỉ lệ chiến thắng của các đội ở sân nhà trung bình mỗi năm.

Với hình trên, tỉ lệ chiến thắng của các đội sân nhà khá cao , đạt khoảng gần hơn 85%. Để việc so sánh tương quan hơn thì đây là biểu đồ của các đội có tỉ lệ chiến thắng ở sân khách cao nhất:

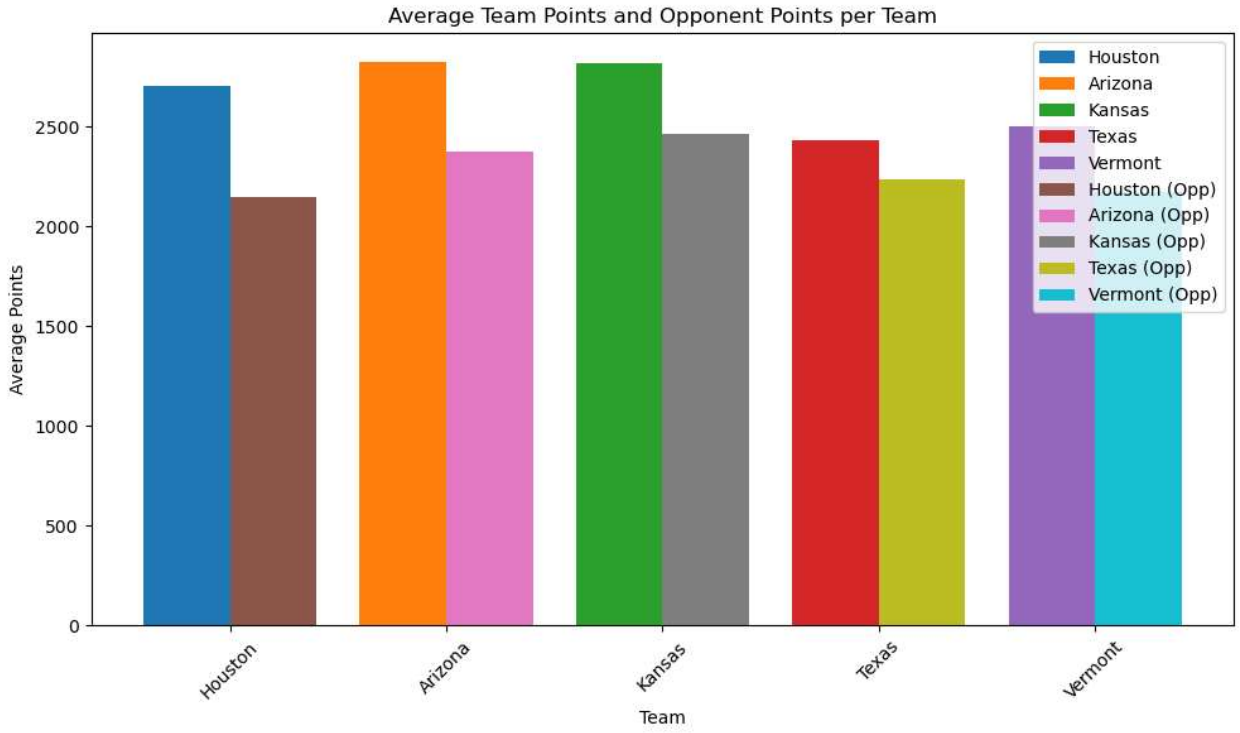


Hình 4.2. Tỉ lệ chiến thắng của các đội ở sân khách trung bình mỗi năm.

Dựa vào biểu đồ trên có thể thấy rằng việc có sự chênh lệch nhất định giữa tỉ lệ thắng sân khách và sân nhà của các đội. Việc này khẳng định yếu tố sân khách có ảnh hưởng đến chiến thắng của các đội bóng.

+ Strength of Schedule (SOS):

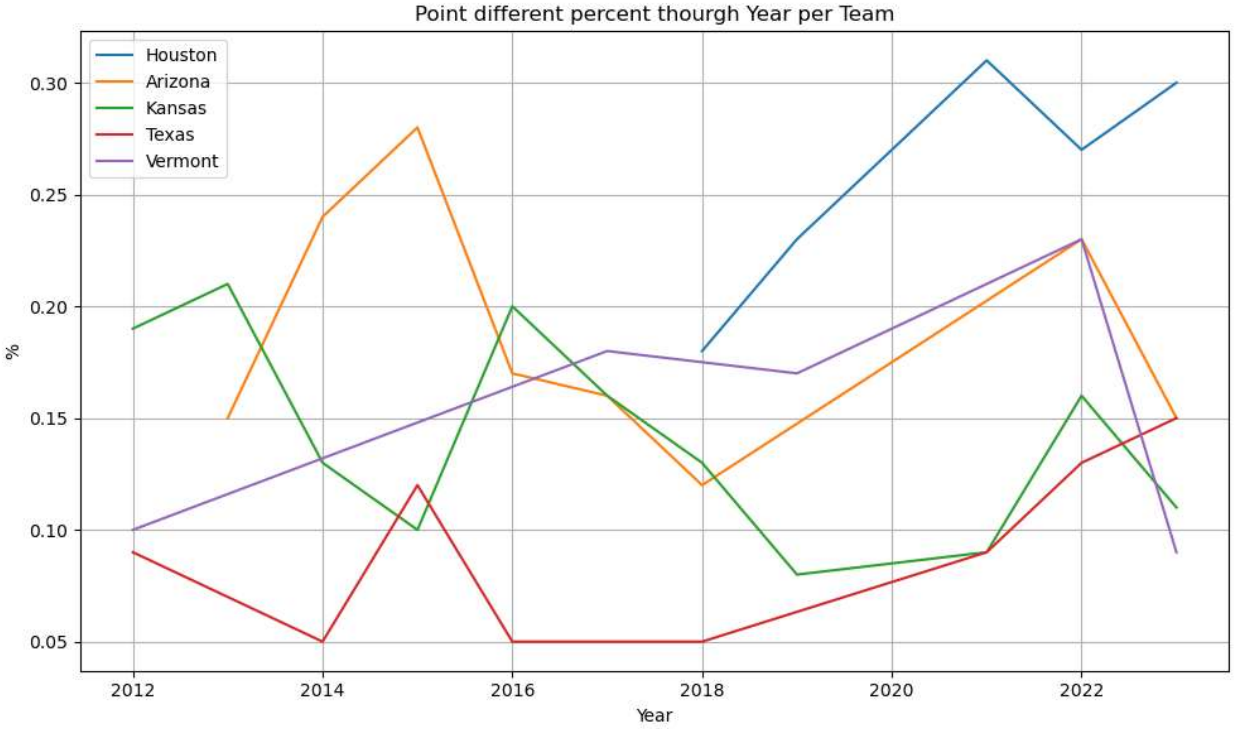
SOS đề cập đến chất lượng của các đối thủ của một đội. Một chiến thắng trước một đối thủ kém hơn thường được coi ít tích cực hơn so với một chiến thắng trước một đối thủ xuất sắc hơn. Thường thì các đội trong cùng một giải đấu, được so sánh với nhau để xem xét về chức vô địch hoặc vào vòng loại, không có cùng đối thủ. Do đó, việc đánh giá các kết quả thắng-thua tương đối của họ là phức tạp. Ví dụ , xét một số đội bóng ngẫu nhiên dựa theo điểm số chênh lệch trung bình mỗi năm, ta có sơ đồ sau:



Hình 4.3 Biểu đồ Team Point và Opponent Point của một số đội.

Đây là một số yếu giúp xác định chỉ số SOS, việc chiến thắng có một sự chênh lệch nhất định về chỉ số điểm ghi bàn giữa các trận giúp cho việc đánh giá sức mạnh

của đội bóng được trực quan hơn. Có thể thấy rằng chỉ số chênh lệch giữa Houston và Arizona có khoảng cách lớn hơn hẳn so với Kansas hoặc Texas. Để trực quan hơn, ta xét chỉ số chênh lệch điểm của các đội đó, vậy chúng ta sẽ có biểu đồ như sau:



Hình 4.4 Biểu đồ tỉ lệ điểm số chênh lệch qua các năm của các đội.

Dựa vào hình 4.4 và 4.3, ta có thể thấy các đội như Texas có sự chênh lệch điểm số giữa điểm số ghi được và điểm số đối thủ không có sự chênh lệch quá lớn qua các năm. Trong khi đó các đội có sự chênh lệch lớn có thể kể đến như Houston và Arizona. Tuy nhiên, đối với 2 đội trên thì tỉ lệ có sự thay đổi lớn qua từng năm. Đối với Kansas thì tỉ lệ có phần ổn định qua các năm hơn so với các đội trên và vẫn đảm bảo tỉ lệ chênh lệch điểm đủ lớn.

+ Điểm số khi chiến thắng:

Một sự phân chia chính trong các hệ thống xếp hạng thể thao nằm ở cách biểu diễn kết quả trận đấu. Một số hệ thống lưu trữ điểm số cuối cùng dưới dạng ba sự kiện rời rạc: thắng, hòa và thua. Các hệ thống khác ghi lại điểm số cuối cùng chính xác, sau đó đánh giá các đội dựa trên biên độ chiến thắng.

+ Thông tin tình trạng đội tuyển:

Bên cạnh điểm số hoặc số trận thắng, một số nhà thiết kế hệ thống chọn thêm nhiều thông tin chi tiết hơn về trận đấu. Các ví dụ bao gồm thời gian kiểm soát bóng,

số liệu cá nhân và sự thay nhân sự. Dữ liệu về thời tiết, chấn thương, hoặc các trận đấu "mang tính thủ tục" gần cuối mùa giải có thể ảnh hưởng đến kết quả trận đấu nhưng khó mô hình hóa. Trận đấu “mang tính thủ tục” là các trận đấu trong đó các đội đã giành được vị trí vào vòng loại và đã đảm bảo hạt giống của mình trước khi kết thúc mùa giải thường muốn cho các cầu thủ chính thức của mình nghỉ ngơi bằng cách giữ họ ra khỏi các trận đấu còn lại trong mùa giải thường. Điều này thường dẫn đến các kết quả không thể dự đoán và có thể làm lệch kết quả của hệ thống xếp hạng.

4.3) Seeds.

Seeds là một thuật ngữ dùng để xác định thứ hạng của các đội tham dự vào giải đấu playoff. Cụ thể, Seeds là việc xếp hạng các đội dựa trên thành tích của họ trong mùa giải chính thức trước khi vào giai đoạn loại trực tiếp của giải đấu. Càng cao seed của một đội, họ sẽ đối đầu với một đối thủ "yếu" hơn ở vòng đấu sớm hơn trong playoff. Seed thấp hơn thường ám chỉ việc đội đó phải đấu với các đối thủ mạnh hơn ở giai đoạnsớm hơn trong playoff. Điều này giúp tạo ra sự cân đối và hấp dẫn trong việc xác định nhà vô địch của giải.

Thông thường, Seeds được xác định từ những thành tích mà đội đã đạt được từ những năm trước đây. Ở trường hợp này, các đội được chia thành 16 seed. 16 seed biểu hiện cho sếp hạng của từng đội theo từng khu vực. Tuy nhiên, gần đây đã có sự tham gia của 68 đội , nên giải đấu đã ra quyết định sử dụng thêm 1 vòng đấu mới được gọi là “First Four”.

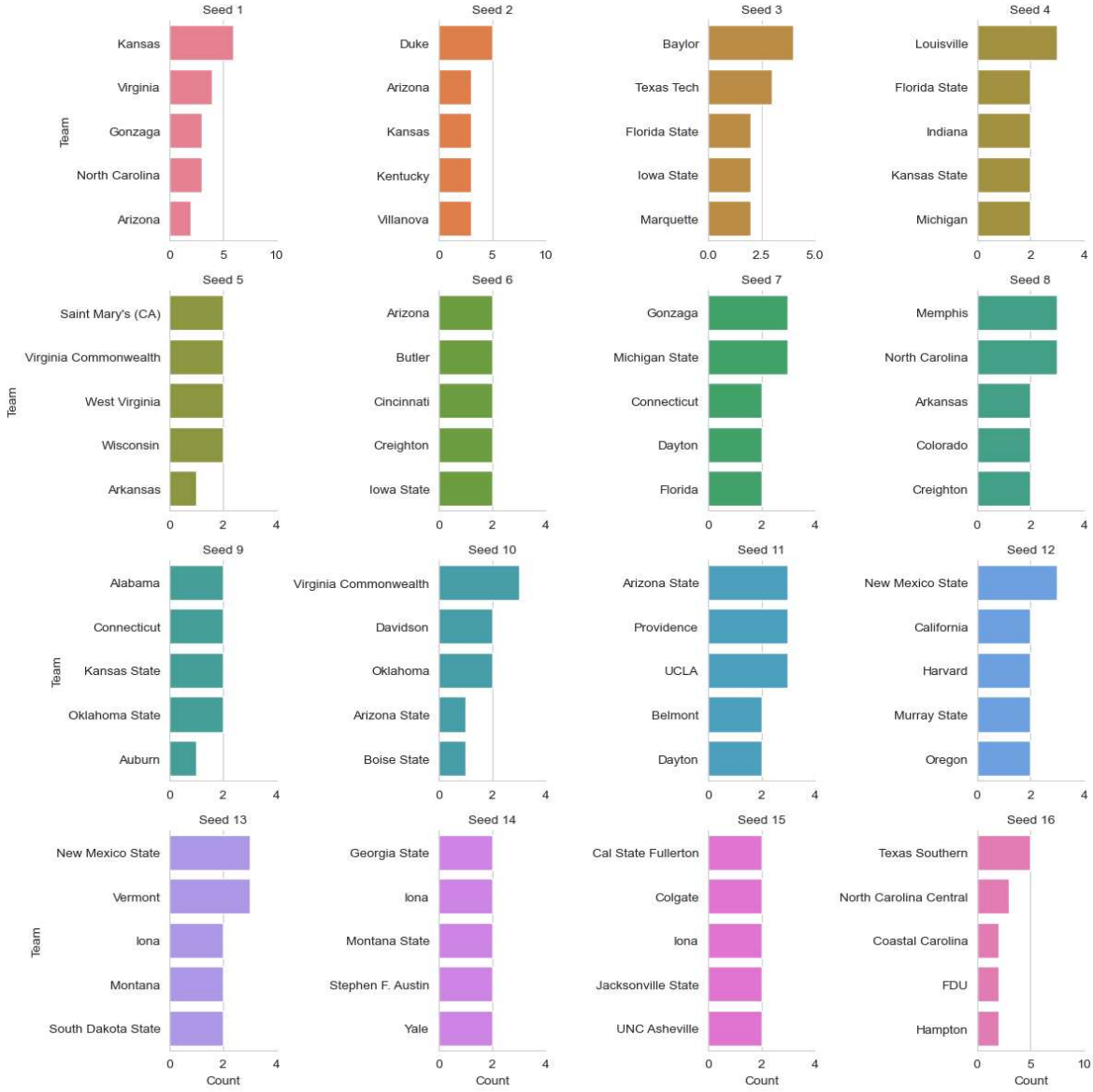
Để đưa về con số lý tưởng, dễ chia và để tạo sự phấn khích cho người hâm mộ, bốn vòng loại tự động được xếp hạt giống thấp nhất và bốn đội được xếp hạt giống thấp nhất nói chung. các đội (trong số các đội không thắng nhưng được mời thi đấu) thi đấu vòng “First Four” và những đội chiến thắng sẽ đi tiếp, để lại cho chúng ta 64 đội được chia đều. Điều này có thể thấy ở hình 1.1.

Quay trở lại, để xác định seed cho các đội là một điều phức tạp. Tuy nhiên, việc xác định các đội thuộc nhóm seed có thể dựa vào một số yếu tố sau đây:

- Thành tích trong mùa giải chính thức: bao gồm các yếu tố như điểm số, số trận thắng, số trận thua, và thành tích chung của mỗi đội trong suốt mùa giải sẽ quyết định seed của họ. Đội có thành tích tốt hơn thường được seed cao hơn.
- Thành tích trong các trận đấu quan trọng: Các trận đấu với đối thủ mạnh hơn, đánh giá mạnh hơn hoặc trong các giải đấu lớn có thể được coi trọng hơn và có thể ảnh hưởng đến việc xác định seed.

- Chỉ số thống kê cá nhân và đội hình: Những yếu tố như chỉ số điểm, tỷ lệ ném trúng, và hiệu suất phòng thủ có thể được xem xét khi xác định seed.

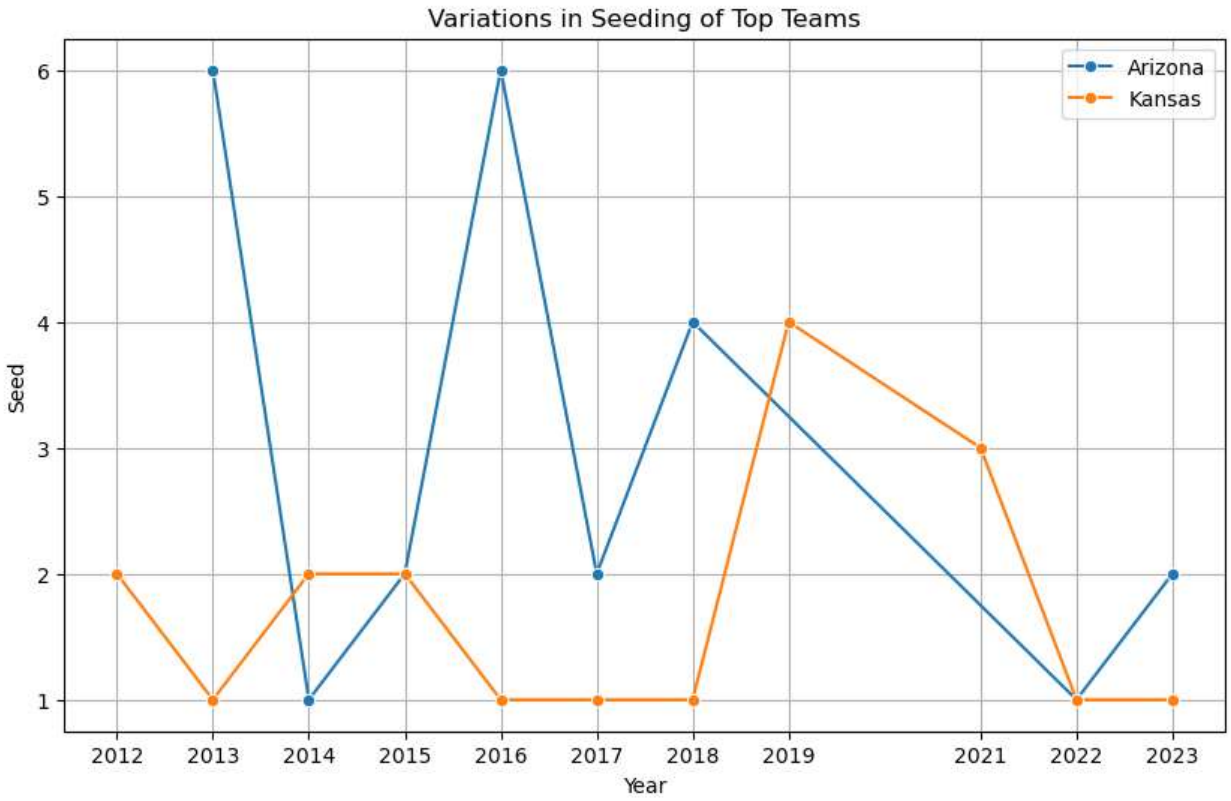
Để hình dung, đây là nhóm biểu đồ biểu hiện top 5 số lần các đội đạt được seed từ 1 đến 16 lớn nhất:



Hình 4.5 biểu đồ số lần các đội đạt được seed từ 1 đến 16

Với hình 4.5, có thể thấy rằng việc các đội được sếp thành seed cao khá thường xuyên, việc này biểu hiện ở tỉ lệ thắng của đội tuyển có được ở hình 4.1 và 4.2. Ví dụ, Kansas có số lần đạt seed 1 cao nhất và đứng thứ 2 ở seed 2. Dựa vào hình trên có thể thấy có một số đội có sự không ổn định, có thể đến từ việc thay đổi nhân sự hoặc phong độ thi đấu , các kết quả không có sự đồng đều (Arizona). Để dễ hình

dung hơn, sau đây là biểu đồ thay đổi hạt giống của 2 đội tuyển đã đề cập qua các năm:



Hình 4.6 Sự thay đổi Seed của 2 đội qua từng năm

Dựa vào hình 4.5 và 4.6 cho chúng ta thấy sự thay đổi các seed qua từng năm có sự không đồng đều, đều này diễn ra từ việc thay đổi nhân sự , chiến thuật cũng như phong độ, thành tích đội tuyển qua từng mùa giải.

4.4) Kết quả các bước tiền xử lý data

Để bắt đầu ta lấy 1 tập dữ liệu được lưu trong file ncaa.xlsx.

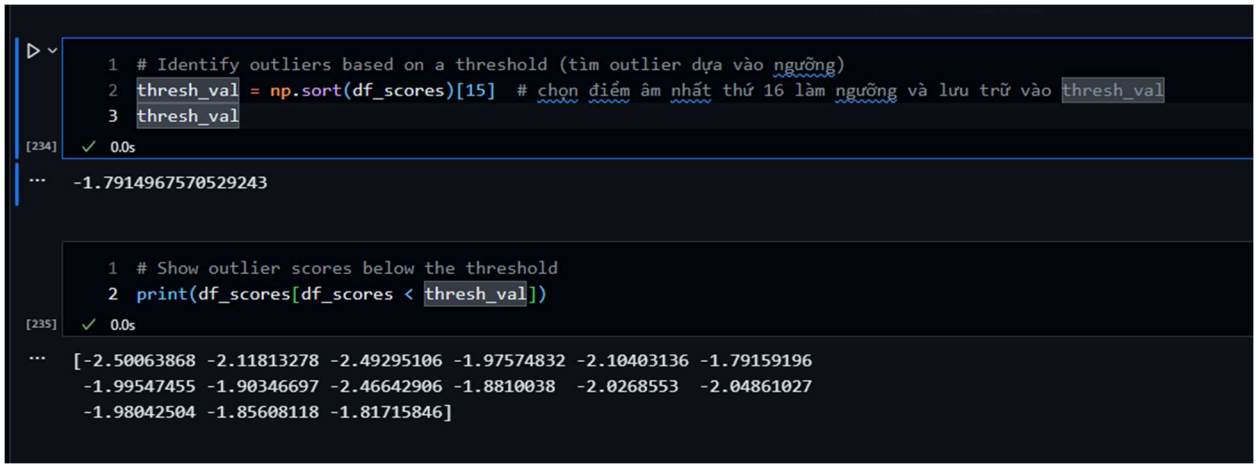
df = pd.read_excel('ncaa.xlsx')

Sau đó kiểm tra nếu tập dữ liệu đó có giá trị NaN hay NULL thì ta tiến hành xử lý, nếu không thì bỏ qua bước này

```
1  # Check for missing values in each variable
2  df.isnull().sum()
```

```
Year                  0
Team                  0
G                     0
W                     0
L                     0
W-L%                  0
SRS                   0
SOS                   0
Conf. W               0
Conf. L               0
Home W                0
Home L                0
Away W                0
Away L                0
Team Points           0
Opp Points            0
FG%                   0
3P%                   0
FT%                   0
Home win rate         0
Away win rate         0
Conference win rate   0
Point diff %          0
AdjEM                 0
AdjO                  0
...
OppD                  0
NCSOS AdjEM           0
Seed                  0
Made Round of 16      0
dtype: int64
```

Tiếp theo ta sử dụng LOF (Local outlier factor) sử dụng các mẫu lân cận để phát hiện ngoại lệ . Ở đây chúng em lấy các giá trị outlier và thay thế bằng giá trị tại ngưỡng.



```
# Identify outliers based on a threshold (tìm outlier dựa vào ngưỡng)
thresh_val = np.sort(df_scores)[15]  # chọn điểm âm nhất thứ 16 làm ngưỡng và lưu trữ vào thresh_val
thresh_val
```

[234] ✓ 0.0s

```
-1.7914967570529243
```

```
# Show outlier scores below the threshold
print(df_scores[df_scores < thresh_val])
```

[235] ✓ 0.0s

```
[-2.50063868 -2.11813278 -2.49295106 -1.97574832 -2.10403136 -1.79159196
 -1.99547455 -1.90346697 -2.46642906 -1.8810038  -2.0268553  -2.04861027
 -1.98042504 -1.85608118 -1.81715846]
```

```
# Pressure Imputation (Replacing outliers) (thay thế các điểm ngoại lệ)
pressure_val = df_LOF[df_scores == thresh_val]
print(pressure_val)
# Selects the columns containing categorical data (object type) from the original dataframe (chọn các cột chứa dữ liệu phân loại từ khung dữ liệu gốc)
df_objects = df.select_dtypes(include=['object'])
```

[308] ✓ 0.0s

```
      Year   G   W  L   W-L%    SRS   SOS  Conf. W  Conf. L  Home W  ...  \
141   2021  24  16  8  0.667  12.49  9.49       10        6      11  ...  \

       AdjO  AdjD  AdjT  Luck  SOS AdjEM   OppO  OppD  NCSOS AdjEM  Seed
141   106.4  90.9  64.0   0.1      14.39  109.7  95.3         11.6     7  \

     Made Round of 16
141                0

[1 rows x 33 columns]
```

```
# Converting DataFrame outliers to List of Dictionaries:
res = outliers.to_records(index = False)
```

[309] ✓ 0.0s

```
# Overwriting res with List from pressure_val (Potential Issue) (ghi đè res bằng danh sách từ pressure_val)
res[:] = pressure_val.to_records(index = False)
```

[310] ✓ 0.0s

```
# Updating Rows in df_LOF Based on Condition and res (Cập nhật hàng trong df_LOF dựa trên điều kiện và res)
df_LOF[df_scores < thresh_val] = pd.DataFrame(res,df_LOF[df_scores < thresh_val].index)

```

Việc thay thế Outlier Data bằng giá trị trung bình để đảm bảo sự toàn vẹn của dữ liệu từ đó cải thiện hiệu quả của các ML model.

4.5) Thông số cho mô hình

Đối với mô hình KNN ta dùng thư viện sklearn.neighbors để huấn luyện mô hình và sklearn.metrics để lấy các thông số đánh giá mô hình. Còn với mô hình Neural Network ta dùng thư viện tensorflow, tensorflow.keras.models và tensorflow.keras.layers để xây dựng một chuỗi các lớp một cách tuần tự. Trong mô hình neural network này sử dụng các thông số optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'] có nghĩa là Chọn thuật toán tối ưu hóa là 'adam', sử dụng hàm mất mát ‘'binary_crossentropy', đánh giá hiệu suất mô hình bằng chỉ số 'accuracy'.

4.6) Độ đo đánh giá

Về độ đo đánh giá chúng ta sử dụng các thông số là accuracy và F1-score để đánh giá cho mô hình KNN và Neural Network (Classification), Mean Absolute Error (MAE) và Mean Squared Error (MSE) cho mô hình KNN và Neural Network (Regression).

	Classification		Regression	
Mô hình	Accuracy	F1-score	MSE	MAE
k-nearest neighbors	0.9859	0.9677	0.0121	0.0323
Neural network	1.0000	1.0000	0.0366	0.0873

Phần 5: Kết luận

5.1) Dữ liệu và Phương pháp

- **Dữ liệu:** thông số các đội tham gia NCAA được thống kê từ năm 2012 đến năm 2023.
- **Phương pháp:**
  - Thuật toán KNN được sử dụng với giá trị k = 5.
  - Mạng nơ-ron được cấu trúc gồm 3 lớp:
    - Lớp ẩn đầu tiên với 128 nơ-ron và hàm kích hoạt ReLU.
    - Lớp ẩn thứ 2 với 64 nơ-ron và hàm kích hoạt ReLU.
    - Lớp đầu ra với 1 nơ-ron và hàm kích hoạt sigmoid.
  - Cả hai mô hình được huấn luyện và đánh giá trên cùng một tập dữ liệu chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ 80/20.

5.2) Kết quả

Trong quá trình thực hiện và kiểm thử hai thuật toán KNN và Neural network. Chúng em đã thu được một số kết quả đáng chú ý khi nghiên cứu các phương pháp, cơ sở lý thuyết của hai thuật toán KNN và Neural network.

Khi thực hiện bài toán dự đoán EDA, chúng em đã tiếp cận các phương pháp như KNN binary classification, KNN Regressor, các phương pháp thuật toán trong mạng Neural network như Adam Optimizer hay sử dụng hàm Relu để thực hiện xử lý số liệu, thông số đầu vào của dữ liệu và đầu ra của bài toán.

Sau khi thực hiện các phương pháp trên, chúng em đã thu được các kết quả và có thực hiện một số so sánh như sau:

Neural network có hiệu suất phân loại cao hơn KNN:

- F1-score của neural network đạt 1.0, cao hơn so với 0.9677 của KNN, cho thấy khả năng phân biệt lớp tốt hơn.
- Accuracy_score của neural network đạt 1.0, cao hơn so với 0.9859 của KNN, cho thấy tỷ lệ dự đoán chính xác cao hơn.
- Regression: KNN đạt được Sai số Trung bình Bình phương (MSE) thấp hơn (0.012112676056338027) so với Neural Network (0.036054734619609556).

Điều này cho thấy KNN có thể tốt hơn trong việc giảm thiểu sai số bình phương giữa các giá trị dự đoán và thực tế.

Tuy nhiên, có các lưu ý sau:

- Hiệu suất có thể thay đổi tùy theo tập dữ liệu và cách thức cài đặt mô hình. Trong trường hợp này, việc phân tích hay xử lý số liệu của các đội bóng có thể đúng một phần theo tính khách quan. Tuy nhiên trong việc xử lý thực tế, có thể sẽ xuất hiện một số trường hợp làm cho dữ liệu không đúng đắn. Ví dụ: giả sử có sự thay đổi một số thành viên trong đội bóng, tình hình thực tế hay phong độ, hiệu suất của thành viên trong đội bóng thay đổi,..
- KNN có ưu điểm đơn giản hơn và dễ giải thích hơn so với Neural network. KNN là thuật toán được sử dụng phổ biến, có thể nói là đơn giản hơn so với một số thuật toán phức tạp như Neural network.
- Neural network thường đòi hỏi nhiều tài nguyên tính toán hơn để huấn luyện. Vì thông thường, Neural network sử dụng nhiều lớp để phân loại, các lớp được kết nối với nhau qua các trọng số.

5.3) Kết Luận

Cả KNN và Neural Network đều là những thuật toán phân loại. Các thuật toán trên đều mang lại hiệu quả nhất định. Thế nên, Việc lựa chọn thuật toán sao cho phù hợp hoàn toàn phụ thuộc vào nhu cầu và điều kiện cụ thể của từng bài toán.

Trong trường hợp khi đối diện với các bài toán phức tạp đòi hỏi độ chính xác cao và có sẵn nguồn tài nguyên tính toán mạnh, việc sử dụng Neural Network có thể mang lại nhiều lợi ích đáng kể như:

- Có thể đạt độ chính xác cao cho các bài toán phức tạp. Neural network có khả năng xử lý các bài toán phức tạp như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên. Với việc có các cấu trúc phức tạp, Neural netowrk có thể tìm ra các quan hệ giữa các mẫu. Trong trường hợp trên, các yếu tố về chỉ số phòng thủ, tấn công, điểm số, tỉ lệ thắng,.... có thể tạo ra một số quan hệ phức tạp giữa các biến. Việc Neural network có khả năng tính toán và tìm ra các mỗi quan hệ giữa các biến giúp cho việc dự đoán thêm phần chính xác hơn.
- Có sẵn nguồn tài nguyên tính toán mạnh. Neural network là thuật toán phức tạp và sử dụng rất nhiều tài nguyên của máy tính. Đồng nghĩa với việc nếu máy tính có cấu hình càng mạnh, thuật toán có tốc độ thực hiện càng nhanh để thực hiện đọc các dữ liệu, thực hiện huấn luyện trên một tập dữ liệu lớn.

Trong trường hợp khi tài nguyên máy tính bị hạn chế, hoặc đối mặt với một số bài toán đơn giản, không có sự phức tạp của dữ liệu hay dữ liệu không có sự phức tạp quá lớn. Có thể sử dụng KNN khi:

- Cần một mô hình đơn giản và dễ giải thích. KNN hoạt động theo cơ chế: điểm dữ liệu mới sẽ được phân loại dựa trên đa số phiếu bầu từ các điểm gần nhất trong tập huấn luyện. Do đó, KNN tạo ra một mô hình dự đoán dễ hiểu và giải thích. Khi cần một giải pháp đơn giản và không cần quá nhiều sự phức tạp, việc sử dụng KNN làm cho quá trình hiểu và giải thích kết quả trở nên dễ dàng. Ví dụ, trong bài toán dự đoán kết quả giải bóng rổ, việc sử dụng KNN giúp dễ dàng giải thích dự đoán dựa trên các trận đấu gần nhất trong quá khứ, mà không cần phải hiểu sâu vào cấu trúc phức tạp của mô hình như trong Neural Network.
- Có hạn chế về tài nguyên tính toán. KNN không yêu cầu quá nhiều tài nguyên tính toán so với Neural Network. Với KNN, việc "huấn luyện" mô hình chỉ đơn giản là lưu trữ toàn bộ tập dữ liệu huấn luyện. Kết hợp với độ phức tạp thấp, việc sử dụng KNN giúp giảm bớt áp lực về tài nguyên tính toán trong việc dự đoán.

TÀI LIỆU THAM KHẢO

1. Nearest neighbors. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/neighbors.html

2. sklearn.neighbors.KNeighborsClassifier. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

3. Nttuan. (2021, March 23). *Bài 3: Neural network*. Deep Learning Cơ Bản. https://nttuan8.com/bai-3-neural-network-2/

4. Neural network models (supervised). (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/neural_networks_supervised.html5

5. Van Anh Tran T. (2023, June 8). EDA là gì? 4 Loại EDA phổ biến. *Gimasys*. https://gimasys.com/exploratory-data-analysis-eda-la-gi/

6. Ong H. (2018, January 22). *Exploratory Data Analysis: Các nguyên tắc trình bày biểu đồ*. Ông Xuân Hồng. https://ongxuanhong.wordpress.com/2015/09/16/exploratory-data-analysis-cac-nguyen-tac-trinh-bay-bieu-do/

7. EDA là gì? Mục đích của việc sử dụng Exploratory Data Analyst. (n.d.). https://mindx.edu.vn/blog/eda-la-gi