ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

----------OoO----------

BÁO CÁO ĐỒ ÁN

DỰ ĐOÁN KẾT QUẢ BÓNG ĐÁ BẰNG THUẬT TOÁN MÁY HỌC

Môn học: Máy học

Giáo viên hướng dẫn:

PGS.TS Lê Đình Duy

THS. Phạm Nguyễn Trường An

Sinh viên:

Huỳnh Minh Trí – MSSV: 18520176

Thành phố Hồ Chí Minh, tháng 07 năm 2020

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

----------OoO----------

BÁO CÁO ĐỒ ÁN

DỰ ĐOÁN KẾT QUẢ BÓNG ĐÁ BẰNG THUẬT TOÁN MÁY HỌC

Môn học: Máy học

Giáo viên hướng dẫn:

PGS.TS Lê Đình Duy

THS. Phạm Nguyễn Trường An

Sinh viên:

Huỳnh Minh Trí – MSSV: 18520176

Thành phố Hồ Chí Minh, tháng 07 năm 2020

Mục Lục

I. Giới thiệu

1. Tổng quan bài toán

Trong bóng đá và các môn thể thao khác, kết quả của một trận đấu luôn bị ảnh hưởng bởi rất nhiều yếu tố khác nhau như phong độ, tình hình sức khỏe của các cầu thủ, chiến thuật, huấn luận viên, thời tiết, sân thi đấu,… và có cả yếu tố may mắn. Vì thế việc dự đoán kết quả một trân đấu thể thao đã luôn đặt nhiều thách thức và khó khăn cho cả những chuyên gia thể thao cũng như cho cả máy tính. Trong đồ án này, việc dự đoán kết quả một trân đấu bóng đá sẽ sử dụng các yếu tố: chỉ số của cầu thủ từ FIFA và những thông số về lịch sử đấu của các đội bóng. Các thuật toán máy học phổ biến như K-Nearest-Neighbor, Logistic Regresion, Support Vector Machine và Neural Network được áp dụng để đưa ra dự đoán kết quả cho đội chủ nhà là thắng, hòa hoặc thua.

2. Ứng dụng và mục tiêu

Cá cược thể thao đặc biệt là bóng đá đã được hợp pháp ở nhiều quốc gia như Anh, Mỹ, Úc, Hà Lan, Thụy Sĩ, …Thị trường cá cược ở Mỹ được dự kiến sẽ đạt 8 tỷ USD vào năm 2025[1]. Ở Việt Nam, theo nghị đinh 06/2017/NĐ-CP, Chính phủ đã cho phép thí điểm đặt cược bóng đá quốc tế. Nhưng trong đó vẫn còn một số khuất mắc nên chưa có doanh nghiệp nào vào cuộc. Tuy nhiên có thể thấy việc hợp pháp đặt cược bóng đã sẽ được tiến hành và phát triển ở Việt Nam trong một tương lai không xa.

Mục tiêu của đồ án là đưa ra dự đoán kết quả của một trận đấu bóng đá ở các giải vô địch quốc gia hàng đầu Châu Âu. Dự đoán này mong muốn sẽ giúp ích cho các doanh nghiệp khi phát triển thị trường cá cược bóng đá và giúp ích cho cả người chơi.

Đồ án cũng giúp tìm ra thuật toán nào là phù hợp cho bộ dữ liệu có được. Tìm hiểu xem những chỉ số đánh giá cầu thủ từ FIFA có ảnh hưởng đến kết quả trận đấu như thế nào.

[1] Theo trang báo MarketWatch, link bài báo: https://www.marketwatch.com/story/firms-say-sports-betting-market-to-reach-8-billion-by-2025-2019-11-04

II. Dữ liệu

Bài toán sử dụng dữ liệu trong 10 mùa giải (2010-2020) (đối với mùa 2020 tính đến 7/2020) của 4 giải vô địch quốc gia hàng đầu Châu Âu bao gồm: Premier League (Anh), Laliga (Tây Ban Nha), SerieA (Ý) và Bundesliga (Đức). Dữ liệu được thu thập chia làm 2 nhóm: dữ liệu của các cầu thủ và dữ liệu của các trận đấu:

- Dữ liệu của các cầu thủ mà thành phần chủ yếu là các chỉ số đánh giá từ trò chơi *FIFA*. Đây là một trò chơi của hãng *Electronic Arts*, nó có tính phổ biến và uy tín trên toàn thế giới nên có thể chọn những chỉ số đánh giá cầu thủ[2] của trò chơi này để làm đầu vào cho bài toán. Trò chơi này bao gồm dữ liệu của các cầu thủ ở các giải đấu quốc gia hàng đầu Châu Âu và được cập nhật qua hàng năm, đáp ứng được sự thay đổi phong độ và thay đổi câu lạc bộ của các cầu thủ. Bên cạnh các chỉ số đánh giá, chúng ta còn thu thập thêm dữ liệu về tên cầu thủ, tên đội đang thi đấu, vị trí thi đấu, số áo thi đấu, tuổi và mùa giải thi đấu tương ứng. Toàn bộ dữ liệu này được lấy về từ trang web https://www.fifaindex.com/.

- Dữ liệu trận đấu bao gồm tên 2 đội bóng, thời điểm thi đấu, đội hình ra sân và số bàn thắng mà 2 đội ghi được. Những dữ liệu này được lấy về từ trang báo https://www.skysports.com/. Đây là trang báo về thể thao lớn ở nước Anh nên có tính chính xác và tin cậy cao.

Để đáp ứng về việc chuyển nhượng cầu thủ, phong độ của đội bóng và cầu thủ ở các mùa giải khác nhau thì dù cùng là một đội bóng, một cầu thủ nhưng ở các mùa giải khác nhau sẽ xem như là khác nhau.

Sử dụng 2 công cụ Web scrapper và Scrapy để lấy dữ liệu từ các trang web. Lưu 2 nhóm dữ liệu dưới định dạng file csv.

Kích thước bộ dữ liệu bao gồm 15.045 trận đấu và 26.227 cầu thủ trong 10 mùa giải.

[2] Tham khảo cách tính chỉ số cầu thủ tại link: https://www.fifauteam.com/player-ratings-guide-fifa-19/#:~:text=A%20player's%20overall%20rating%20is,other%20refers%20to%20International%20Reputation.&text=As%20you%20can%20see%20above,attributes%20with%20the%20international%20reputation.

III. Tiền xử lí dữ liệu

1. Chuẩn hóa dữ liệu

a. Xử lí chênh lệch và đồng bộ ở 2 nhóm dữ liệu

Có sự chênh lệch về số đội bóng ở 2 nhóm dữ liệu: trong nhóm dữ liệu các cầu thủ có số đội bóng là 195 đội trong khi ở nhóm dữ liệu các trận đấu chỉ có 133 đội.

- Chênh lệch này do một số đội bóng đổi tên qua các mùa giải. Ví dụ như trong nhóm dữ liệu cầu thủ, đội *Barcelona* của Laliga được lưu dưới 2 cái tên: *fc-barcelona* và *f-c-barcelona*.

- Bên cạnh đó còn có sự khác biệt tên của cùng một đội bóng ở 2 nhóm dữ liệu do 2 trang web quản lí tên đội khác nhau. Ví dụ như ở nhóm dữ liệu các cầu thủ tên đội chỉ là là *inter* trong khi ở nhóm dữ liệu các trận đấu tên của đội này là *inter-milan*.

Để đồng bộ cùng một đội bóng với nhau, tiến hành đánh số ID cho các đội, lấy tên các đội bóng trong nhóm các trận đấu làm chuẩn vì trong nhóm dữ liệu này, tên đội không bị thay đổi qua các mùa giải. Các đội sẽ có ID từ 0-132.

b. Tổ chức quản lý các đối tượng

Quản lý các cầu thủ, các đội bóng, các trận đấu dưới dạng các dictionary trong python.

- Một cầu thủ có các thông tin sau:

```
Player {
    'name': player's name,
    'pos': player's position,
    'ovr': player's overal rating,
    'age': player's age,
}
```

- Một đội bóng có các thông tin sau:

Team {

Number player 1: {player 1},

Number player 2: {player 2},

Number player 3: {player 3},

…

}

- Một trận đấu có các thông tin sau:

Match {

'date': date,

'home_id': home team's ID,

'away_id': away team's ID,

'home_goals': goals of home team,

'away_goals': goals of away team,

}

Trong một mùa giải, các trận đấu của một đội bóng sẽ được gom lại với nhau và sắp xếp theo thứ tự tăng dần của ngày thi đấu để quản lý và dễ dàng tính toán các đặc trưng về lịch sử đấu.

2. Xây dựng đặc trưng

Chỉ số của các cầu thủ được dùng làm đặc trưng cho các trận đấu. Tuy nhiên, vì trong bóng đá, yếu tố đội hình cũng như vị trí thi đấu của các cầu thủ trên sân thể hiện tính chiến thuật của mỗi đội và có ảnh hưởng lớn đến kết quả trận đấu. Bóng đá có độ đa dạng về đội hình thi đấu như: 4-3-3, 4-4-2, 3-5-2, 4-1-2-1-2, … Có thể thấy ở các đội hình khác nhau thì số cầu thủ ở hàng hậu vệ, tiền vệ, tiền đạo là khác nhau. Nên thay vì chỉ dùng 11 vị trí cho 11 chỉ số đánh giá của cầu thủ ở mỗi đội, chúng ta sẽ dùng 18 vị trí để thể hiện rõ hơn cách

bố trí cầu thủ của các đội hình khác nhau. Trong đó vị trí đầu tiên dành cho thủ môn, 6 vị trí tiếp theo cho các chỉ số đánh giá của các cầu thủ hàng hậu vệ, 7 vị trí kế tiếp cho hàng tiền vệ và cuối cùng dành cho hàng tiền đạo. Nếu vị trí nào không có cầu thủ thi đấu thì được cho bằng 0.

Ví dụ cho mảng chỉ số của đội hình xuất phát là 5-2-3:

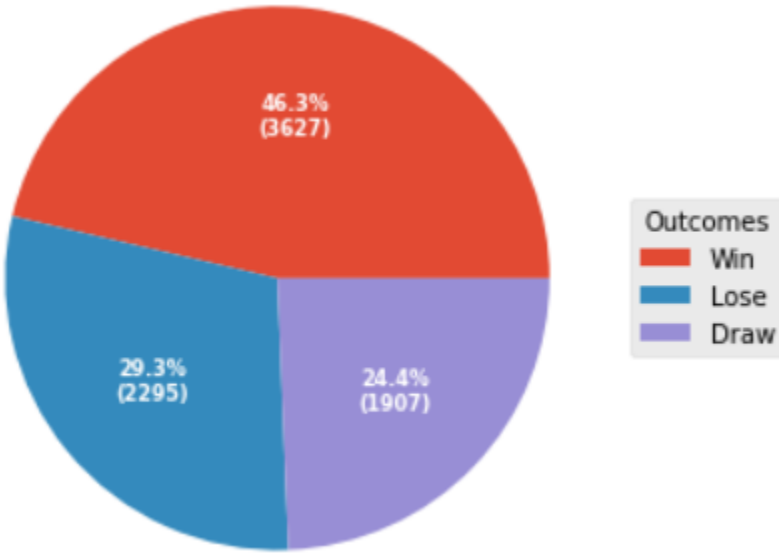
89	81	82	82	82	86	0	85	88	0	0	0	0	0	91	84	84	0

Kết quả của một trận đấu bóng đá cũng chịu ảnh hưởng không nhỏ đến từ phong độ của 2 đội bóng. Một đội bóng đang có phong độ tốt thì luôn có thể giành kết quả có lợi trước một đội đang có phong độ thấp hơn. Vì vậy các thông số từ lịch sử đấu được thêm vào bên cạnh những chỉ số đánh giá để thể hiện được phong độ của 2 đội. Các thông số này bao gồm số trận thắng, số trận hòa, số bàn thắng và số bàn thua trong 5 trận gần nhất. Những thông số này thể hiện được rằng khi có tỉ lệ trận thắng cao hay số bàn thắng ghi được nhiều thì hàng công của đội bóng đang thi đấu tốt, ngược lại thì hàng phòng ngự đang thi đấu tệ.

3. Phân chia dữ liệu

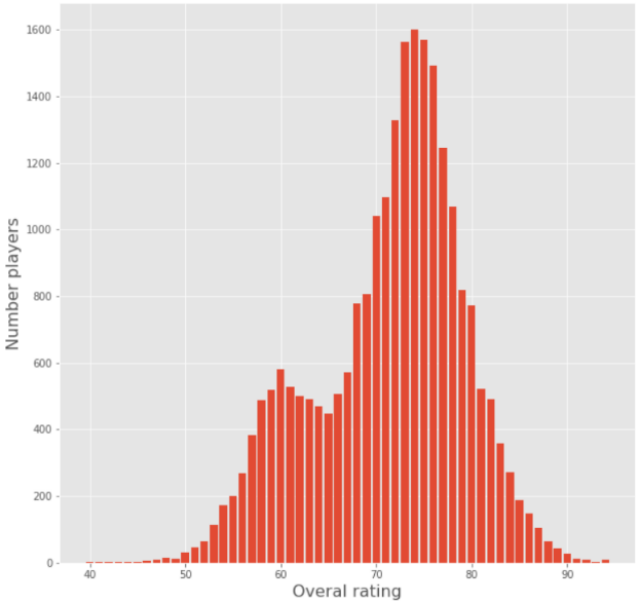
Dùng dữ liệu các mùa giải từ 2009/2010 - 2018/2019, bao gồm 13.679 trận đấu và 23.869 cầu thủ (cầu thủ), để xây dựng các model. Trong đó 80% dùng để train các thuật toán và 20% để thẩm định kết quả.

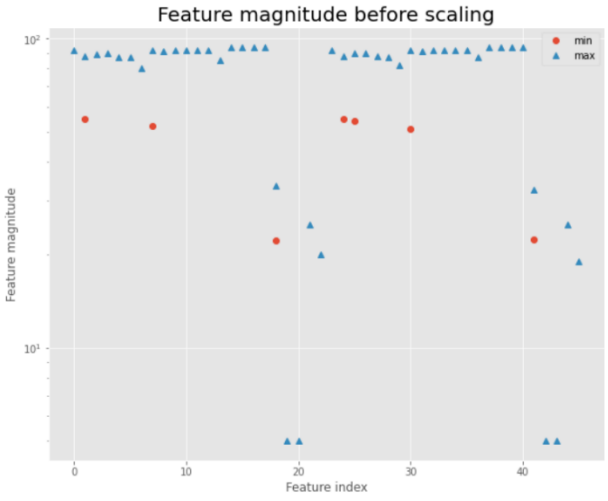
Dữ liệu dùng để xây dựng model có sự cân bằng của 3 nhãn



Dữ liệu mùa giải 2019/2020, bao gồm 1366 trận đấu và 2358 cầu thủ, là bộ dữ liệu các trận đấu mới nhất, để kiểm tra hiệu quả của thuật toán tốt nhất chọn được.

Tuy nhiên trong bộ dữ liệu, các đặc trưng về chỉ số đánh giá cầu thủ có giá trị dao động từ 50 đến 90 trong khi các đặc trưng khác như tỉ lệ trận thắng trong 5 trận gần nhất có giá trị từ 0 -1. Điều này sẽ làm cho việc đánh giá các đặc trưng bị mất cân bằng và làm cho việc hội tụ của Gradient descent trong một số thuật toán như Neural network, Linear classification,… chậm đi.





Từng đặc trưng trong được chuyển giá trị về dao động từ 0-1:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

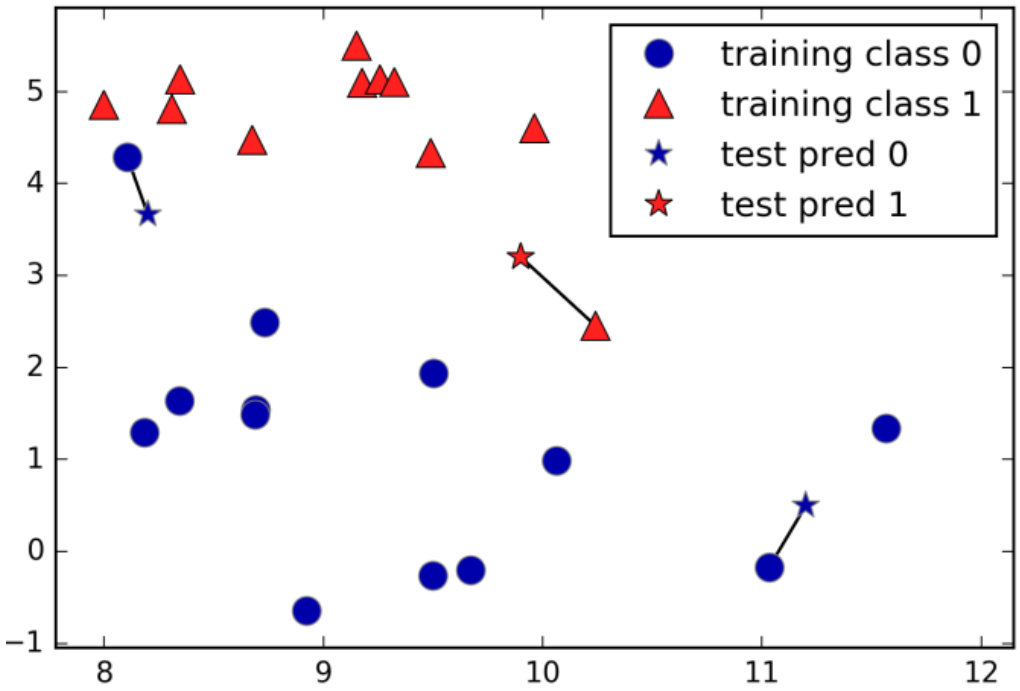
Khi áp dụng các thuật toán máy học, sẽ sử dụng cả 2 bộ dữ liệu chưa được scaled và đã được scaled. Điều này giúp tìm xem dữ liệu nào sẽ phù hợp cho thuật toán nào và giúp tìm ra cách xử lý dữ liệu phù hợp nhất để đạt kết quả cao nhất cho bài toán.

IV. Các thuật toán máy học

Các thuật toán máy học được sử dụng từ thư viện **scikit-learn**.

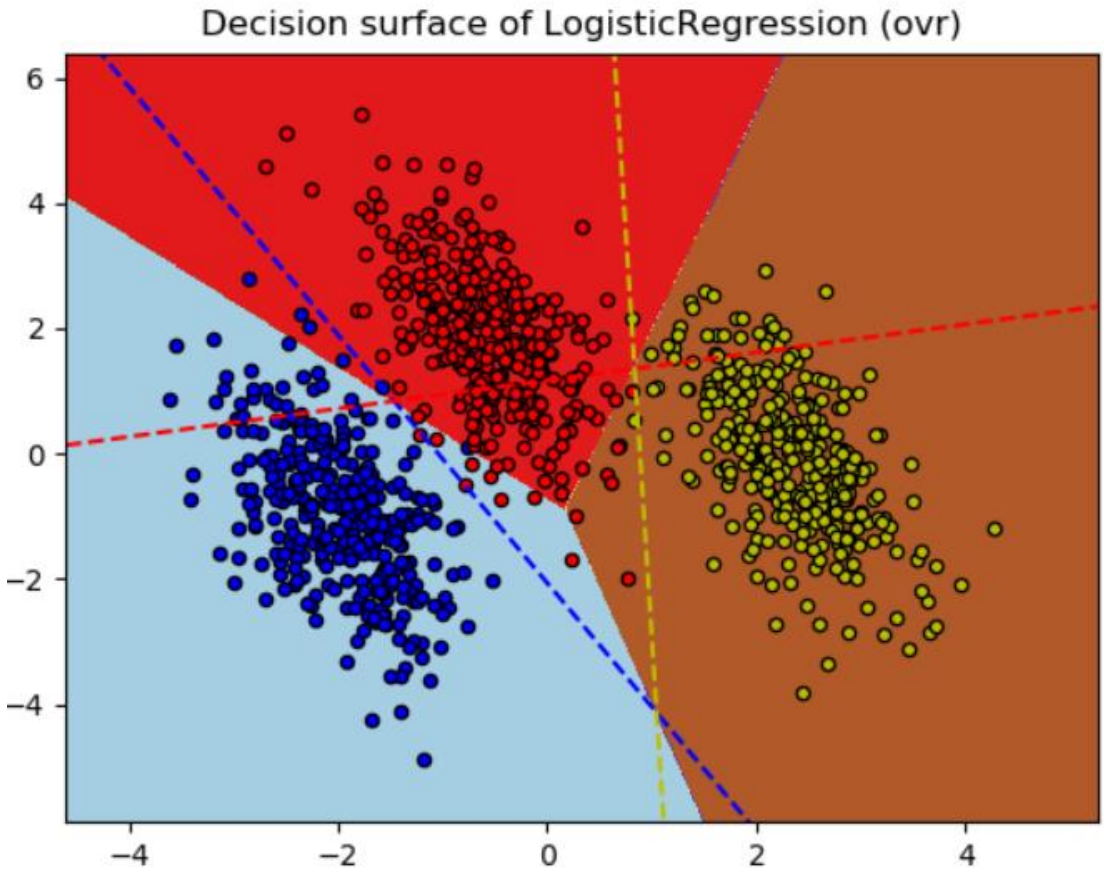
1. K-Nearest-Neighbors (k-NN):

k-NN là một thuật toán máy học đơn giản và dễ tiếp cận, nó được xây dựng bằng cách lưu trữ bộ dữ liệu train và thực hiện dự đoán một dữ liệu mới dựa theo **k** điểm dữ liệu trên tập train gần nó nhất.



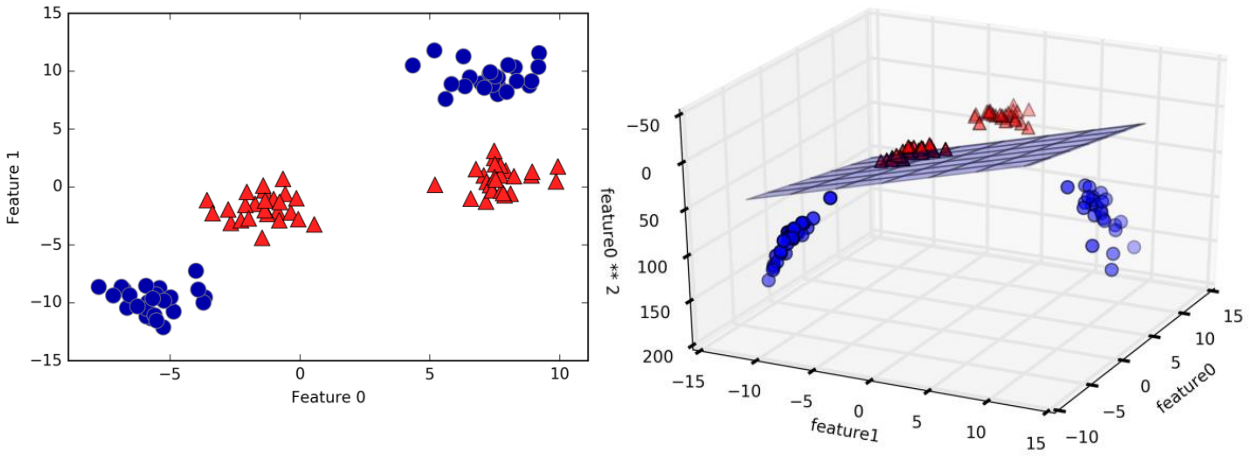
2. Logistic Regresion

Logistic Regresion là một model tuyến tính trong machine learning. Các thuật toán tuyến tính thường được sử dụng cho bài toán phân loại 2 lớp. Tuy nhiên để mở rộng cho những bài toán phân loại nhiều lớp hơn, các model tuyến tính sẽ áp dụng cách tiếp cận one-vs.-rest hoặc one-vs.one. Trong đồ án này sử dụng kĩ thuật one-vs.rest.



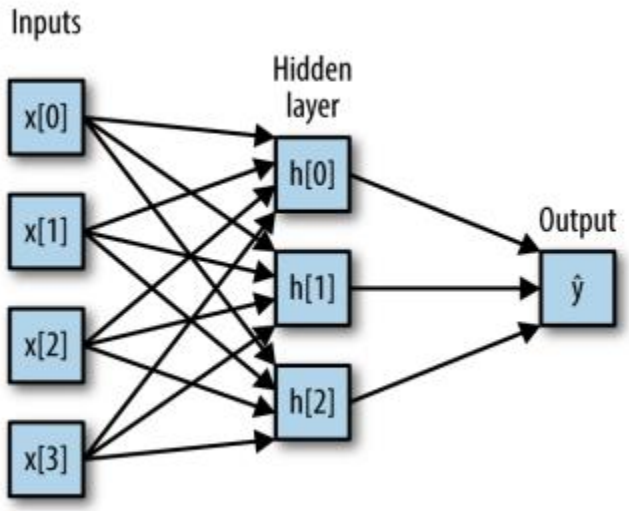
3. Support vector machines (SVMs)

SVMs là một model mở rộng từ những model tuyến tính như Logistic Regression hay LinearSVC. Nó phù hợp với những bộ dữ liệu phức tạp, phi tuyến tính. SVMs thêm vào những đặc trưng mới đồng thời cũng làm tăng không gian của bộ dữ liệu, từ đó đưa ra những ranh giới giữa các điểm dữ liệu gọi là các vector. Những vector này là cơ sở để đưa ra dự đoán những dữ liệu mới.



4. Neural Network (MLPs)

MLPs model có thể được xem gần giống như một model tuyến tính vì nó sử dụng kết quả từ nhiều lớp liên tiếp nhau để đưa ra dự đoán. Mỗi lớp này chứa nhiều node và trọng số của mỗi lớp sẽ được đưa vào những hàm phi tuyến tính như hàm *relu*, *tanh*. Kết quả của những hàm này sẽ được đưa vào các node ở những lớp sau.



V. Training và hiệu quả của các thuật toán

Áp dụng các thuật toán trên với tham số mặc định từ thư viện **scikit-learn**.

1. K-Nearest-Neighbors (k-NN):

- f1-score trên dữ liệu không scale cao hơn so với dữ liệu đã được scale : 0.44 so với 0.41.

- Nhãn Draw có hiệu quả dự đoán thấp hơn so với 2 nhãn còn lại.

- Có overfitting trên cả 2 bộ dữ liệu.

```
              REPORT ON UNCALED DATA
Accuracy on training set: 0.6135057471264368
Accuracy on test set: 0.4425287356321839
              precision    recall  f1-score   support

        Lose       0.40      0.49      0.44       462
        Draw       0.26      0.21      0.23       399
         Win       0.56      0.55      0.55       705

    accuracy                           0.44      1566
   macro avg       0.41      0.41      0.41      1566
weighted avg       0.44      0.44      0.44      1566

              REPORT ON SCALED DATA
Accuracy on training set: 0.6143039591315453
Accuracy on test set: 0.4099616858237548
              precision    recall  f1-score   support

        Lose       0.36      0.42      0.39       462
        Draw       0.28      0.22      0.25       399
         Win       0.50      0.51      0.51       705

    accuracy                           0.41      1566
   macro avg       0.38      0.38      0.38      1566
weighted avg       0.41      0.41      0.41      1566
```

2. Logistic regression:

- f1-score trên 2 bộ dữ liệu bằng nhau.

- Hiệu quả khi dự đoán nhãn Draw thấp hơn so với 2 nhãn còn lại.

```
              REPORT ON UNSCALED DATA
Accuracy on training set: 0.5432630906768838
Accuracy on test set: 0.5178799489144317
              precision    recall  f1-score   support

        Lose       0.49      0.51      0.50       462
        Draw       0.11      0.00      0.00       399
         Win       0.53      0.82      0.65       705

    accuracy                           0.52      1566
   macro avg       0.38      0.44      0.38      1566
weighted avg       0.41      0.52      0.44      1566

              REPORT ON SCALED DATA
Accuracy on training set: 0.5450191570881227
Accuracy on test set: 0.5210727969348659
              precision    recall  f1-score   support

        Lose       0.50      0.50      0.50       462
        Draw       0.12      0.00      0.00       399
         Win       0.53      0.83      0.65       705

    accuracy                           0.52      1566
   macro avg       0.39      0.44      0.38      1566
weighted avg       0.42      0.52      0.44      1566
```

3. Support vector machine (SVMs)

- f1-score trên 2 bộ dữ liệu bằng nhau.

- Trong trường hợp này, SVMs không thể dự đoán đúng bất kì nhãn Draw nào.

```
               REPORT ON UNSCALED DATA
Accuracy on training set: 0.54757343550447
Accuracy on test set: 0.5191570881226054
              precision    recall  f1-score   support

        Lose       0.54      0.40      0.46       462
        Draw       0.00      0.00      0.00       399
         Win       0.51      0.89      0.65       705

    accuracy                           0.52      1566
   macro avg       0.35      0.43      0.37      1566
weighted avg       0.39      0.52      0.43      1566

               REPORT ON SCALED DATA
Accuracy on training set: 0.5555555555555556
Accuracy on test set: 0.5178799489144317
              precision    recall  f1-score   support

        Lose       0.52      0.41      0.46       462
        Draw       0.00      0.00      0.00       399
         Win       0.52      0.88      0.65       705

    accuracy                           0.52      1566
   macro avg       0.35      0.43      0.37      1566
weighted avg       0.39      0.52      0.43      1566
```

4. Neural Network (MLPs)

- f1-score trên dữ liệu scale cao hơn so với dữ liệu không được scale : 0.51 so với 0.47.

- Hiệu quả dự đoán nhãn Draw cao hơn so với các model trước.

```
               REPORT ON UNCALED DATA
Accuracy on training set: 0.5292145593869731
Accuracy on test set: 0.4661558109833972
              precision    recall  f1-score   support

        Lose       0.45      0.38      0.41       462
        Draw       0.30      0.36      0.32       399
         Win       0.59      0.59      0.59       705

    accuracy                           0.47      1566
   macro avg       0.45      0.44      0.44      1566
weighted avg       0.47      0.47      0.47      1566

               REPORT ON SCALED DATA
Accuracy on training set: 0.5775862068965517
Accuracy on test set: 0.5063856960408685
              precision    recall  f1-score   support

        Lose       0.54      0.34      0.41       462
        Draw       0.29      0.07      0.12       399
         Win       0.52      0.86      0.65       705

    accuracy                           0.51      1566
   macro avg       0.45      0.42      0.39      1566
weighted avg       0.46      0.51      0.44      1566
```

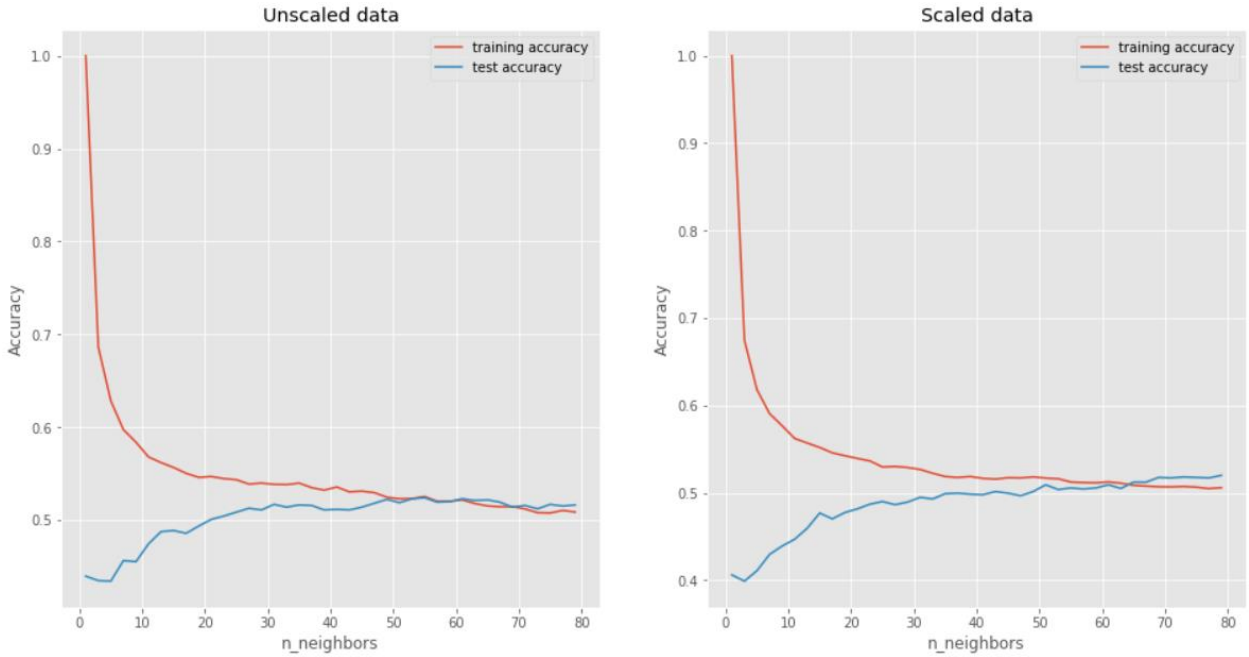
Nhìn chung, trên tất cả các model và trên cả 2 dạng dữ liệu scale và không scale, kết quả dự đoán gần giống nhau, trong đó Logistic Regression, SVMs có hiệu quả cao nhất trên tập test (f1-score = 0.52). Nhãn Draw có hiệu quả dự đoán thấp trong tất cả các model.

VI. Thử các tham số

1. K-Nearest-Neighbors (k-NN)

Các tham số quan trọng:

- k: số điểm neighbors dùng để dự đoán. Dựa theo đồ thị, tham số k sẽ cho kết quả tốt trong khoảng từ 20 – 80.



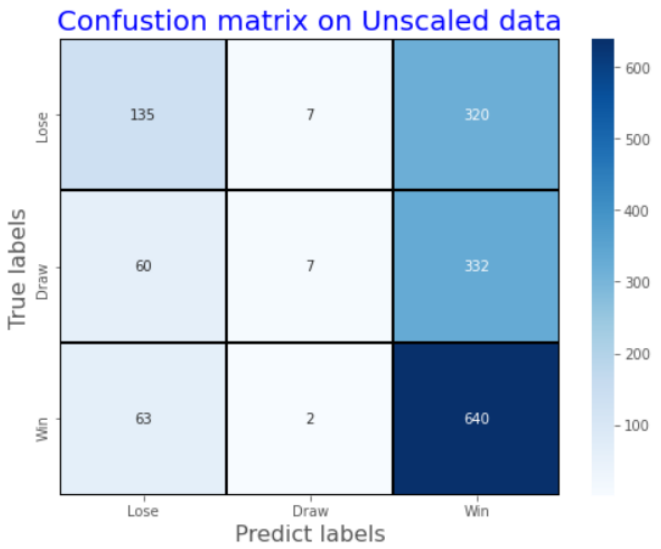
- metric: cách tính khoảng cách 2 điểm.

Các tham số tốt nhất sau khi tune:

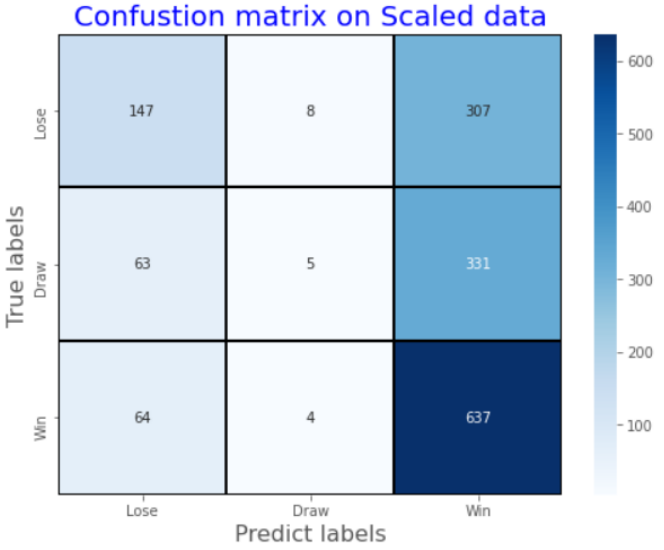
```
Best uncaled model:
Best performance: 0.521763 using {'metric': 'manhattan', 'n_neighbors': 96}
Best scaled model:
Best performance: 0.519741 using {'metric': 'manhattan', 'n_neighbors': 76}
```

Kết quả trên tập test với những tham số tốt nhất:

```
                REPORT ON UNCALED DATA
Accuracy on training set: 0.5292145593869731
Accuracy on test set: 0.49936143039591313
              precision    recall  f1-score   support

        Lose       0.52      0.29      0.38       462
        Draw       0.44      0.02      0.03       399
         Win       0.50      0.91      0.64       705

    accuracy                           0.50      1566
   macro avg       0.49      0.41      0.35      1566
weighted avg       0.49      0.50      0.41      1566
```




```
                REPORT ON SCALED DATA
Accuracy on training set: 0.5348020434227331
Accuracy on test set: 0.5038314176245211
              precision    recall  f1-score   support

        Lose       0.54      0.32      0.40       462
        Draw       0.29      0.01      0.02       399
         Win       0.50      0.90      0.64       705

    accuracy                           0.50      1566
   macro avg       0.44      0.41      0.36      1566
weighted avg       0.46      0.50      0.41      1566
```

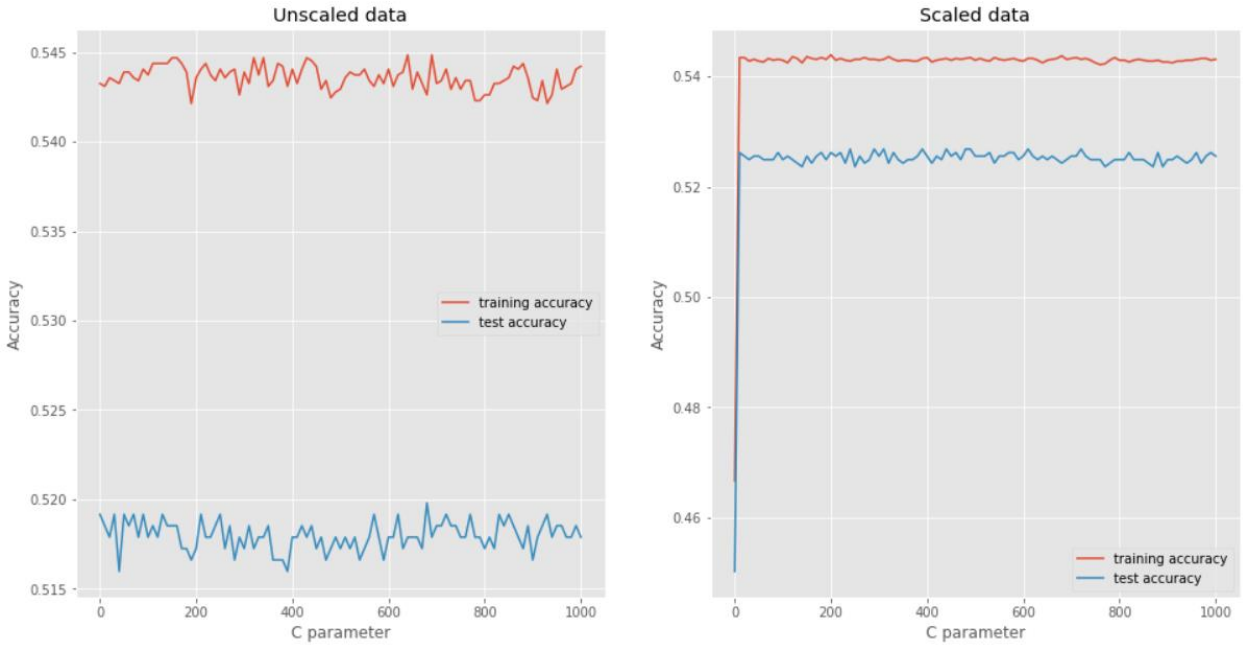


Hiệu quả dự đoán đã được tăng, model có f1-score bằng nhau trên cả 2 bộ dữ liệu. Tuy nhiên đối với nhãn Draw, dù precision có cao hơn nhưng recall và f1-score lại có kết quả thấp hơn so với model mặc định.

2. Logistic Regression

Các tham số quan trọng:

- C: tham số của regularization.



- penalty: lựa chọn regularization.

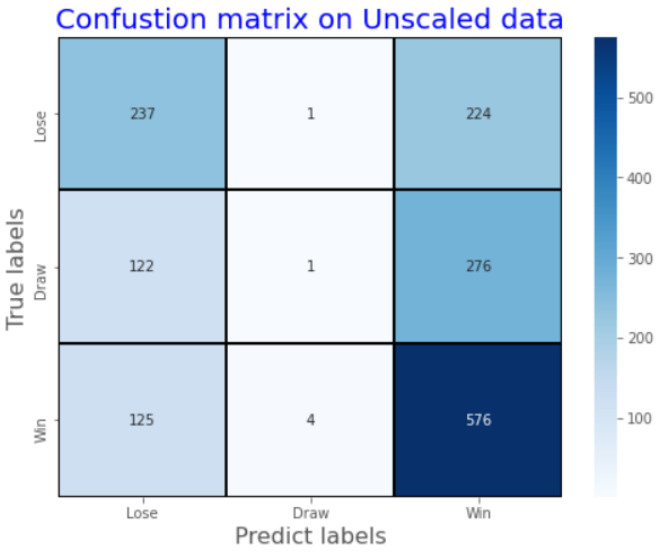
- solver: thuật toán dùng tối ưu hóa vấn đề.

Các tham số tốt nhất sau khi tune:

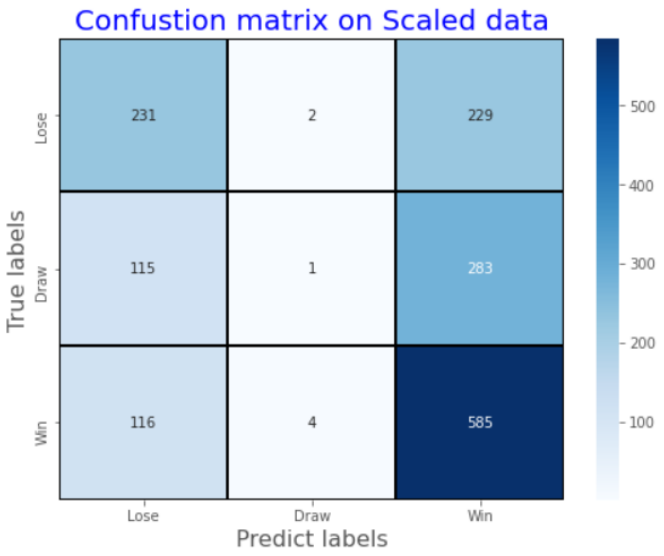
```
Best uncaled model:
Best performance: 0.536927 using {'C': 0.1, 'penalty': 'l1', 'solver': 'liblinear'}
Best scaled model:
Best performance: 0.537514 using {'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}
```

Kết quả trên tập test với những tham số tốt nhất:

```
              REPORT ON UNSCALED DATA
Accuracy on training set: 0.544220945083014
Accuracy on test set: 0.5197956577266922
              precision    recall  f1-score   support

        Lose       0.49      0.51      0.50       462
        Draw       0.17      0.00      0.00       399
         Win       0.54      0.82      0.65       705

    accuracy                           0.52      1566
   macro avg       0.40      0.44      0.38      1566
weighted avg       0.43      0.52      0.44      1566
```



```
              REPORT ON SCALED DATA
Accuracy on training set: 0.5451787994891443
Accuracy on test set: 0.5217113665389528
              precision    recall  f1-score   support

        Lose       0.50      0.50      0.50       462
        Draw       0.14      0.00      0.00       399
         Win       0.53      0.83      0.65       705

    accuracy                           0.52      1566
   macro avg       0.39      0.44      0.38      1566
weighted avg       0.42      0.52      0.44      1566
```

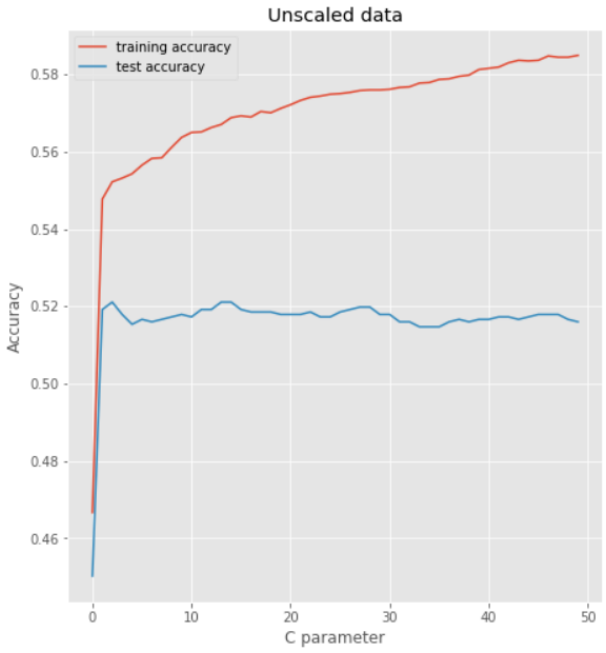


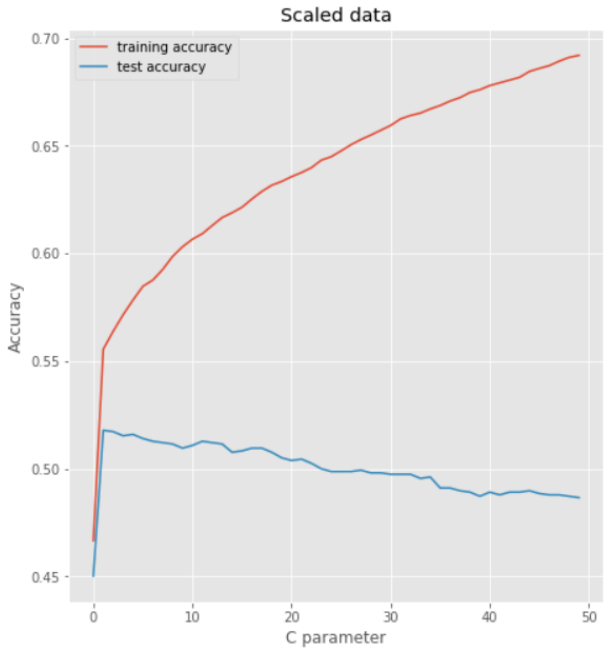
Kết quả không có nhiều thay đổi so với model mặc định.

3. Support Vector Machines (SVMs)

Các tham số quan trọng:

- C: tham số regularization.





- kernel: lựa chọn các kernel.

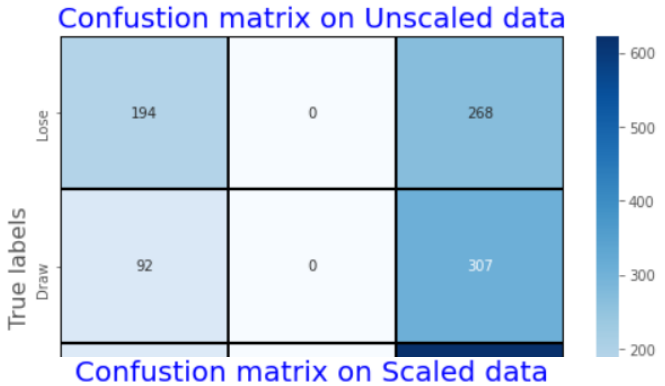
- gamma: điều khiển độ rộng của Gaussian kernel.

Các tham số tốt nhất sau khi tune:

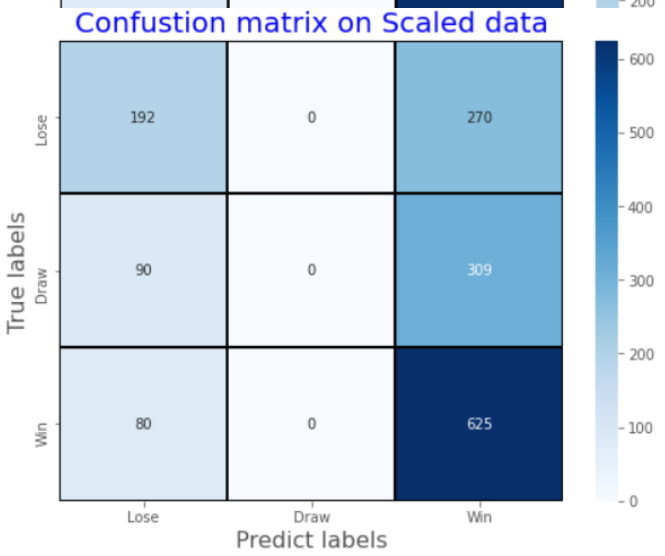
```
Best uncaled model:
Best performance: 0.539911 using {'C': 0.42500000000000004, 'gamma': 'scale', 'kernel': 'poly'}
Best scaled model:
Best performance: 0.535121 using {'C': 0.15, 'gamma': 'scale', 'kernel': 'poly'}
```

Kết quả trên tập test với những tham số tốt nhất:

```
REPORT ON UNSCALED DATA
Accuracy on training set: 0.5501277139208174
Accuracy on test set: 0.5217113665389528
              precision    recall  f1-score   support

        Lose       0.53      0.42      0.47       462
        Draw       0.00      0.00      0.00       399
         Win       0.52      0.88      0.65       705

    accuracy                           0.52      1566
   macro avg       0.35      0.43      0.37      1566
weighted avg       0.39      0.52      0.43      1566
```




```
REPORT ON SCALED DATA
Accuracy on training set: 0.5522030651340997
Accuracy on test set: 0.5217113665389528
              precision    recall  f1-score   support

        Lose       0.53      0.42      0.47       462
        Draw       0.00      0.00      0.00       399
         Win       0.52      0.89      0.65       705

    accuracy                           0.52      1566
   macro avg       0.35      0.43      0.37      1566
weighted avg       0.39      0.52      0.43      1566
```



Kết quả không có nhiều thay đổi so với model mặc định.

4. Neural Network (MLPs)

Các tham số quan trọng:

- hidden_layer_sizes: số lượng layer và số node mỗi layer.
- solvers: thuật toán dùng tối ưu hóa trọng số.

- activation: hàm activation, tính trọng số ở các node.

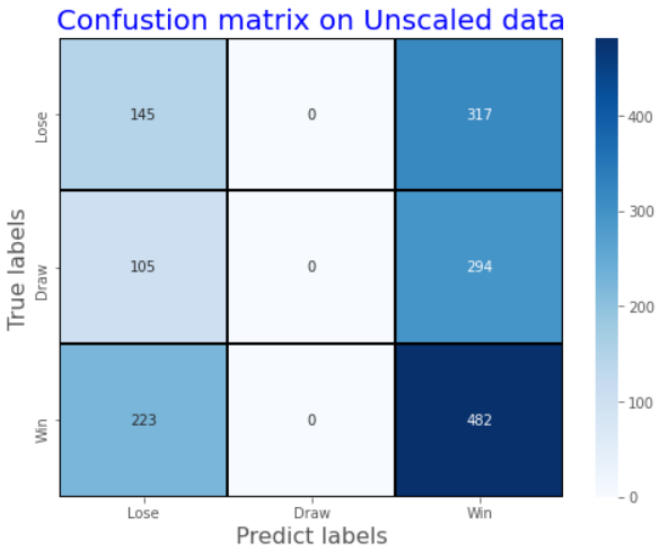
- alpha: tham số l2 penalty.

Các tham số tốt nhất sau khi tune:

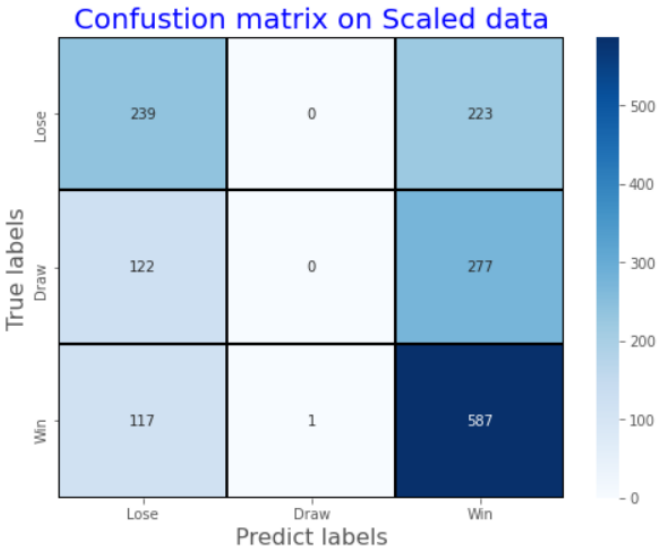
```
Best uncaled model:
Best performance: 0.539112 using {'activation': 'relu', 'alpha': 10, 'hidden_layer_sizes': (100, 50, 10), 'solver': 'lbfgs'}
Best scaled model:
Best performance: 0.538633 using {'activation': 'identity', 'alpha': 10, 'hidden_layer_sizes': (100, 50, 10), 'solver': 'lbfgs'}
```

Kết quả trên tập test với những tham số tốt nhất:

```
REPORT ON UNSCALED DATA
Accuracy on training set: 0.41522988505747127
Accuracy on test set: 0.4003831417624521
              precision    recall  f1-score   support

        Lose       0.31      0.31      0.31       462
        Draw       0.00      0.00      0.00       399
         Win       0.44      0.68      0.54       705

    accuracy                           0.40      1566
   macro avg       0.25      0.33      0.28      1566
weighted avg       0.29      0.40      0.33      1566
```




```
REPORT ON SCALED DATA
Accuracy on training set: 0.5429438058748404
Accuracy on test set: 0.5274584929757343
              precision    recall  f1-score   support

        Lose       0.50      0.52      0.51       462
        Draw       0.00      0.00      0.00       399
         Win       0.54      0.83      0.66       705

    accuracy                           0.53      1566
   macro avg       0.35      0.45      0.39      1566
weighted avg       0.39      0.53      0.44      1566
```



Hiệu quả trên bộ dữ liệu scale tăng, nhưng vẫn không có hiệu quả đối với nhãn Draw.

VII. Kết quả và đánh giá

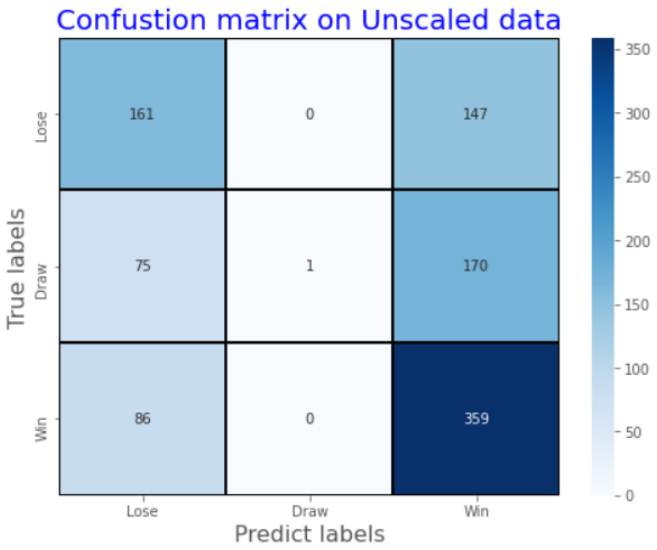
Sau khi xây dựng và kiểm thử với 4 thuật toán (K Nearest-Neighbors, Logistic Regresion, Support Vector Machines, Neural Network) thuật toán có hiệu quả tốt nhất là Neural Network. Mặc dù Logistic Regresion và Support Vecotr Machines cũng cho kết quả tương tự nhưng Logistic Regresion phù hợp hơn với bài toán phân loại 2 lớp còn Support Vector Machine có thời gian train và tune các tham số lớn.

Bộ dữ liệu cho đã được scale về trong khoảng 0 -1 sẽ giúp Neural Network hoạt động tốt hơn đối với bài toán này. f1-score trên tập test là 0.53.

Áp dụng model chọn được trên dữ liệu của các trận đấu mùa 2019-2020:

Accuracy: 0.5215215215215215

	precision	recall	f1-score	support
Lose	0.50	0.52	0.51	308
Draw	1.00	0.00	0.01	246
Win	0.53	0.81	0.64	445
accuracy			0.52	999
macro avg	0.68	0.44	0.39	999
weighted avg	0.64	0.52	0.44	999



- Đạt được f1-score = 0.52

- Model xây dựng được có hiệu quả trên bộ dữ liệu mới có thể coi như tương tự như khi xây dựng trên bộ train và test.

Neural Network xây dựng được có thể hoạt động tốt trên các bộ dữ liệu mới giống như trên dữ liệu cũ. Mặc dù không bị overfitting nhưng hiệu quả dự đoán vẫn còn thấp.

VIII. Kết luận

Bài toán đạt được kết quả dự đoán f1-score = 53% trên dữ liệu dùng để xây dựng model từ các mùa giải cũ (mùa giải từ 2010-2019) và f1-

score = 52% trên bộ dữ liệu của mùa giải mới (mùa 2020). Kết quả dự đoán còn thấp, đặc biệt nhãn Draw không có kết quả dự đoán tốt.

Sau quá trình thử qua các tham số, hiệu quả của các model vẫn không tăng quá đáng kể so với các tham số mặc định. Nguyên nhân có thể từ bộ dữ liệu. Trên thực tế, kết quả của một trận đấu bóng đá chịu ảnh hưởng bởi rất nhiều yếu tố. Số đặc trưng được đưa xem xét còn ít nên chưa đạt được hiệu quả cao.

Để nâng cao kết quả dự đoán, cần đưa vào thêm nhiều yếu tố khác như phong độ thi đấu, tình hình sức khỏe của từng cầu thủ, yếu tố về chiến thuật, huấn luận viên, sân vận động, thời tiết thi đấu hay cả những yếu tố bên ngoài chuyên môn có hưởng đến tâm lý cầu thủ, dẫn đến ảnh hưởng phong độ thi đấu. Cần có sự phân biệt giữa các giải đấu của các quốc gia khác nhau. Các tham số khi tune còn ít về giá trị, đặc biệt là tham số *hidden_layer_sizes* của Neural Network. Cần tăng thêm số layer cũng như số node mỗi layer để tăng hiệu quả của model.

Tài liệu tham khảo

Andreas C. Müller & Sarah Guido. Introduction to Machine Learning with Python. 2017.

Alexandre Bucquet & Vishnu Sarukkai. The Bank is Open: AI in Sports Gambling. 2018.

Bradley. Predicting Football Matches using EA Player Ratings and Tensorflow. 2018.

Scikit-learn. https://scikit-learn.org/stable/#