



Khoa Công nghệ thông tin

BÁO CÁO

DỰ ĐOÁN NGƯỜI CHƠI CHIẾN THẮNG TRONG GAME PUBG

Môn: Khoa học dữ liệu

Giảng viên: Trần Trung Kiên

Nhóm 1: 1512416 – Nguyễn Tất Nam Phương
1512473 - Trương Ngọc Tài

Tp. Hồ Chí Minh, 8/1/2018.

Nội dung

I)	Đặt vấn đề	2
a)	Câu hỏi.	2
b)	Trả lời.....	2
c)	Dữ liệu.....	2
d)	Input và Output (chỉnh sửa sau vấn đáp).....	2
II)	Giải quyết vấn đề	3
a)	Thu thập dữ liệu.	3
b)	Tiền xử lí dữ liệu.....	3
c)	Xây dựng mô hình để mô hình hóa dữ liệu. (chỉnh sửa sau vấn đáp)	4
d)	Huấn luyện mô hình.	4
e)	Kiểm thử và chọn mô hình phù hợp. (chỉnh sửa sau vấn đáp)	4
f)	Đánh giá mô hình được chọn với dữ liệu thực.....	5
III)	Tham khảo.....	5

I) Đặt vấn đề

a) Câu hỏi.

“Ai sẽ là người chiến thắng?”



Đặt vấn đề (thêm vào sau vấn đáp): PUBG là một game sinh tồn rất nổi gần đây. Đối với người/đội tham gia chơi game thì việc dự đoán trước được tỉ lệ chiến thắng (thứ hạng của đối thủ trong danh sách người chơi) rất là có ích, để có kế hoạch đối phó. Từ đó nhóm mới suy nghĩ đến việc sẽ dự đoán xem liệu ai sẽ là người có tỉ lệ chiến thắng cao nhất trong danh sách người tham gia trận đấu đó.

b) Trả lời.

Nếu trả lời được câu hỏi này:

- Dự đoán trước được người chiến thắng trong game.
- Trường hợp là người chơi: ta có thể biết trước được khả năng thắng/thua của đối thủ để có cách đối phó.
- Đối với 1 tổ chức như Công ty Game hay các đội nhóm chơi game: có thể dựa vào dữ liệu dự đoán này để tiến hành sắp xếp/chọn lọc/cải thiện sức mạnh của đội.

c) Dữ liệu.

Dữ liệu được thu thập thông qua API của PUBG cung cấp.

Hoàn toàn hợp pháp.

d) Input và Output (chỉnh sửa sau vấn đáp)

Input: mảng kích thước $M \times 38$

- M là số lượng người chơi
- 38 là số thuộc tính cơ bản của 1 người chơi

Output: Thứ hạng của từng người chơi. Nằm trong khoảng $[0:1]$, và người chơi có hạng cao nhất, gần 1 nhất sẽ là người có tỉ lệ thắng cao nhất.

II) Giải quyết vấn đề

a) Thu thập dữ liệu.

Lấy dữ liệu thông qua API của PUBG.

Yêu cầu: đã có tài khoản PUBG

Quy trình:

- Khởi tạo một API với token PUBG cung cấp cho mỗi tài khoản
- Gửi yêu cầu lên server PUBG để lấy danh sách các giải đấu trên thế giới.
- Ứng với mỗi giải đấu: tiến hành lấy tất cả các trận đấu trong giải đấu đó.
- Ứng với mỗi trận đấu: lấy tất cả các người tham gia trong trận đấu đó.
- Ứng với mỗi người tham gia: lấy tất cả các thuộc tính của người chơi đó.

b) Tiền xử lí dữ liệu.

- Chia bộ dữ liệu thành 3 phần tỉ lệ 60:20:20 tương ứng train:validation:test.
- Phân tách cột: deathType ['alive', 'byplayer', 'logout', 'suicide'] thành các cột ứng với giá trị trong cột.
 - o Tương tự như bag-of-word: chuyển thành one-hot vector rồi nối vào vector feature.
- Loại bỏ các cột không cần thiết: playerId, groupId, matchId, name, deathType(loại sau khi đã phân tách)
 - o Vậy ban đầu ta có 38 thuộc tính

```
['DBNOs', 'assists', 'boosts', 'damageDealt', 'deathType', 'headshotKills',  
'heals', 'killPlace', 'killPoints', 'killPointsDelta', 'killStreaks', 'kills',  
'lastKillPoints', 'lastWinPoints', 'longestKill', 'mostDamage', 'name',  
'playerId', 'rankPoints', 'revives', 'rideDistance', 'roadKills', 'swimDistance',  
'teamKills', 'timeSurvived', 'vehicleDestroys', 'walkDistance', 'weaponsAcquired',  
'winPlace', 'winPoints', 'winPointsDelta', 'matchDuration', 'matchID', 'matchType',  
'groupId', 'numGroups', 'maxPlace', 'winPlacePerc']
```

- o Sau khi loại bỏ và phân tách còn: $38-1+4-7-1=33$ thuộc tính.

```
['DBNOs', 'assists', 'boosts', 'damageDealt', 'headshotKills',  
'heals', 'killPlace', 'killPoints', 'killPointsDelta',  
'killStreaks', 'kills', 'lastKillPoints', 'lastWinPoints',  
'longestKill', 'mostDamage', 'rankPoints', 'revives',  
'rideDistance', 'roadKills', 'swimDistance', 'teamKills',  
'timeSurvived', 'vehicleDestroys', 'walkDistance', 'weaponsAcquired',  
'winPoints', 'winPointsDelta', 'matchDuration', 'winPlacePerc',  
'deathType_alive', 'deathType_byplayer', 'deathType_logout', 'deathType_suicide']
```

- Điền thêm giá trị thiếu: dùng giá trị trung bình trong tập train để điền cho giá trị thiếu.

```
#####
##### TRUNG BÌNH các field trong tập TRAIN #####
[8.725234411502307e-16, 1.489741399808593e-15, -2.3959774158105776e-16, 8.035115599586816e-16, -9.992419025418925e-17, 4.173
456420545436e-15, -2.2463896881788684e-17, 0.0, 0.0, 3.878162216026488e-16, 7.148656460720448e-17, 0.0, 0.0, 1.6825273452753
091e-16, 0.0, 0.0, -1.0867707986413277e-15, -2.006616929985991e-15, -6.922756342787978e-16, 9.877577242780817e-16, -3.916974
7234711915e-16, -5.989480260259862e-16, 5.757840626969191e-15, 1.279062579557024e-16, -8.688030512622122e-17, 0.0, 0.0, -1.0
086104388216486e-15, -3.62078484570352e-17, 5.58951582677797e-16, 7.612913761309018e-16, 4.3810261221635035e-16]
#####
```

- Scale dữ liệu bằng standardScaler của sklearn: chuyển đổi dữ liệu có trung bình bằng 0, độ lệch chuẩn là 1.

Kích thước bộ dữ liệu:

```
##### Kích thước bộ dữ liệu #####
Train_X: (5392, 32)
Validation_X: (1798, 32)
Test_X: (1798, 32)
#####
```

Train chứ 5392 dòng.

Validation chứ 1798 dòng.

Test chứ 112 dòng.

Số lượng đặc trưng là 32.

c) Xây dựng mô hình để mô hình hóa dữ liệu. (chỉnh sửa sau vấn đáp)

Mô hình 1: Linear regression

Mô hình 2: MLPRegressor

- Tham số:
 - o Hidden_layer_size.
 - o Activation.
 - o Solver

d) Huấn luyện mô hình.

e) Kiểm thử và chọn mô hình phù hợp. (chỉnh sửa sau vấn đáp)

- Đối với mô hình Linear Regression:
 - o Độ lỗi MSE trên tập validation: 0.04151382715813545
 - o Độ lỗi MSE trên tập train: 0.0333471427306854
- Đối với mô hình MLPRegressor:



Tham số	Độ lỗi
hiddenLayer = (50, 10) active = 'logistic' solv = 'lbfgs'	Độ lỗi MSE trên tập validation: 0.024611556856295968 Độ lỗi MSE trên tập train: 0.00886674544141948
hiddenLayer = (100, 10) active = 'relu' solv = 'adam'	Độ lỗi MSE trên tập validation: 0.028584275717725392 Độ lỗi MSE trên tập train: 0.012929614252051129

f) **Đánh giá mô hình được chọn với dữ liệu thực.**

Mô hình tốt nhất là MLPRegressor.

Kết quả khi kiểm tra trên tập test:

➤ Độ lỗi MSE trên tập test: 0.04102559914512375

III) Tham khảo.

- Kaggle: <https://www.kaggle.com>
- PUBG API: <https://documentation.playbattlegrounds.com/en/getting-started.html>