|  |  |
| --- | --- |
| Trường Đại học Khoa học tự nhiên  Thành phố Hồ Chí Minh | 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](#_Toc534850458)

[a) Câu hỏi. 2](#_Toc534850459)

[b) Trả lời. 2](#_Toc534850460)

[c) Dữ liệu. 2](#_Toc534850461)

[II) Giải quyết vấn đề 3](#_Toc534850462)

[a) Thu thập dữ liệu. 3](#_Toc534850463)

[b) Tiền xử lí dữ liệu. 3](#_Toc534850464)

[c) Xây dựng mô hình để mô hình hóa dữ liệu. 5](#_Toc534850465)

[d) Huấn luyện mô hình. 5](#_Toc534850466)

[e) Kiểm thử và chọn mô hình phù hợp. 6](#_Toc534850467)

[f) Đánh giá mô hình được chọn với dữ liệu thực. 6](#_Toc534850468)

[III) Tham khảo. 6](#_Toc534850469)

# Đặt vấn đề

## Câu hỏi.

“Ai sẽ là người chiến thắng?”



## 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.

## Dữ liệu.

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

Hoàn toàn hợp pháp.

# Giải quyết vấn đề

## Thu thập dữ liệu.

Lấy dữ liệu thông qua API của PUBG.

Yêu cầu: đã có tài khoảng PUBG

Quy trình:

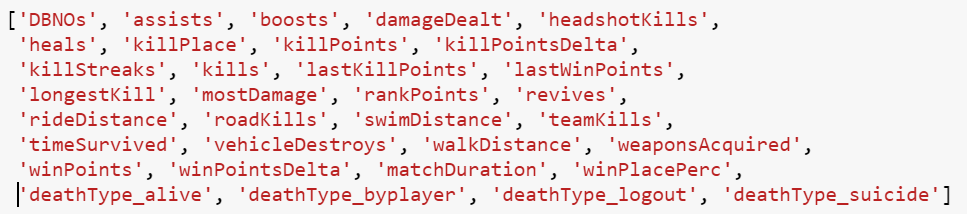
* Khởi tạo một API với token PUBG cung cấp cho mỗi tài khoảng
* Gửi yêu cầu lên server PUBG để lấy danh sách các giải đấu trên thế giới.
* Ứng với mỗi giải đấu: tiến hành lấy tất cả các trận đấu trong giải đấu đó.
* Ứng với mỗi trận đấu: lấy tất cả các người tham gia trong trận đấu đó.
* Ứng với mỗi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

## Tiền xử lí dữ liệu.

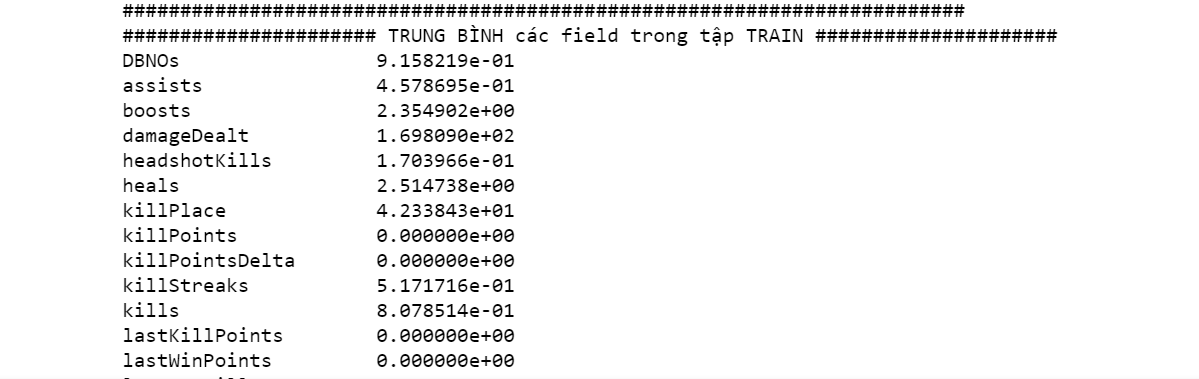
* Chia bộ dữ liệu thành 3 phần tỉ lệ 60:20:20 tương ứng train:validation:test.
* Phân tách cột: deathType [‘alive’, ‘byplayer’, ‘logout’, ‘suicide’] thành các cột ứng với giá trị trong cột.
  + Tương tự như bag-of-word: chuyển thành one-hot vector rồi nối vào vector feature.
* Groupby cột ‘playerId’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



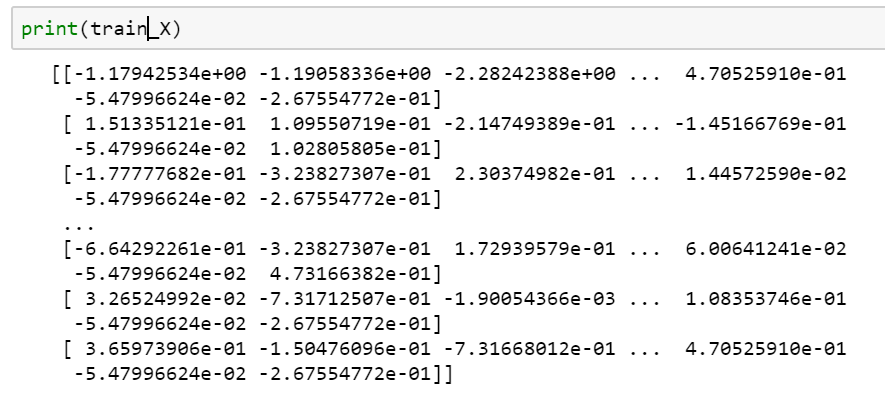
* + Sau khi loại bỏ và phân tách còn: 38-1+4-7-1=33 thuộc tính.



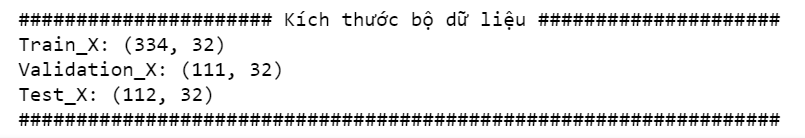
* Điền thêm giá trị thiếu: dùng giá trị trung bình trong tập train để điền cho giá trị thiếu.



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



* Gắn nhãn cho dữ liệu: winPlacePerc >= 0.5 thì label=1 còn lại là 0.

Kích thước bộ dữ liệu: 

Train chứ 334 entity.

Validation chứ 111 entity.

Test chứ 112 entity.

Số lượng feature là 32.

## Xây dựng mô hình để mô hình hóa dữ liệu.

**Mô hình 1: Perceptron**

* Tham số:
  + Alpha=0.001
  + Max\_iter=1
  + randomState=0

**Mô hình 2: Multi Layer Perceptron**

* Tham số:
  + 2 lớp ẩn: lớp thứ nhất có 50 node, lớp thứ 2 có 25 node ẩn.
  + Activation=‘tanh’.
  + Solver=‘lbfgs’
  + Max\_iter=1000
  + randomState=0

**Mô hình 3: Stochastic Gradient Descent**

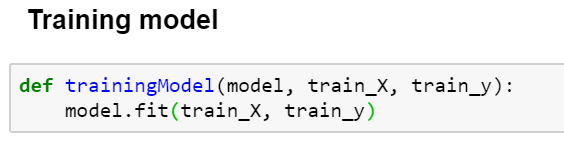
* Tham số:
  + Alpha=0.001
  + Max\_iter=100

**Mô hình 4: Logistic Regression**

* Tham số:
  + C=1
  + Max\_iter=100

## Huấn luyện mô hình.

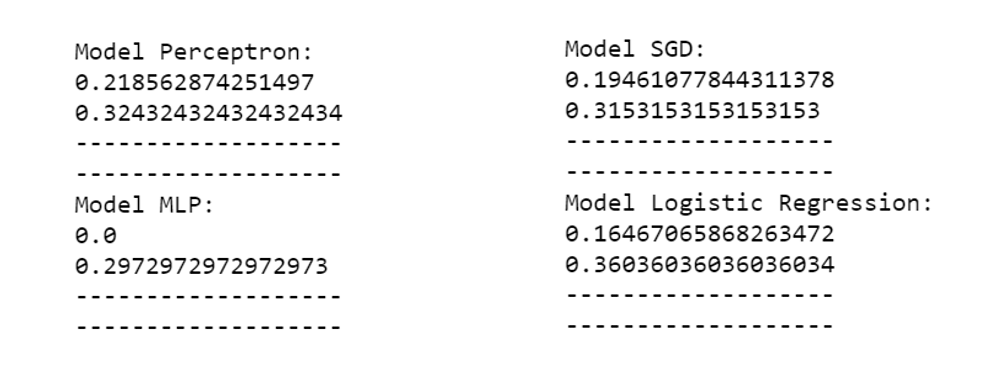
Huấn luyện mô hình với dữ liệu đã được tiền xử lí trên và với nhã đã được gắn tương ứng.



## Kiểm thử và chọn mô hình phù hợp.

Tiến hành kiểm thử từng mô hình dữ liệu đã xây dựng trên tập dữ liệu validation.

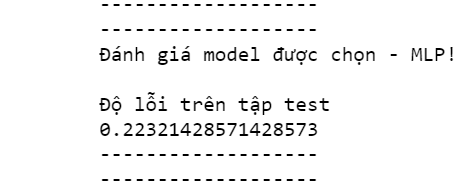
Kết quả kiểm thử:



## Đánh giá mô hình được chọn với dữ liệu thực.

Mô hình tốt nhất là MLP.

Kết quả khi kiểm tra trên tập test:



# Tham khảo.

* Kaggel: https://www.kaggle.com
* PUBG API: https://documentation.playbattlegrounds.com/en/getting-started.html