



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
II)	Giải quyết vấn đề .....	2
a)	Thu thập dữ liệu.....	2
b)	Tiền xử lí dữ liệu.....	3
c)	Xây dựng mô hình để mô hình hóa dữ liệu.....	3
d)	Huấn luyện mô hình.....	4
e)	Kiểm thử và chọn mô hình phù hợp.....	4
f)	Đánh giá mô hình được chọn với dữ liệu thực.....	4
III)	Tham khảo. ....	4

## I) Đặt vấn đề

### a) Câu hỏi.

“Ai sẽ là người chiến thắng?”



### 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.

## 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.
- Loại bỏ các cột không cần thiết: playerId, groupId, matchId, name, deathType(loại sau khi đã phân tách)
- Phân tách cột: deathType thành các cột ứng với giá trị trong cột.
- 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.
- Gắn nhãn cho dữ liệu: winPlacePerc  $\geq$  0.8 thì label=1 còn lại là 0.

Kích thước bộ dữ liệu:

```
#### Kích thước bộ dữ liệu ###
Train_X: (1839, 34)
Validation_X: (613, 34)
Test_X: (614, 34)
#####
```

Train chứa 1839 entity.

Validation chứa 613 entity.

Test chứa 614 entity.

Số lượng feature là 34.

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

##### Mô hình 1: Perceptron

- Tham số:
  - o Alpha=0.001
  - o Max\_iter=1
  - o randomState=0

##### Mô hình 2: Multi Layer Perceptron

- Tham số:
  - o 2 lớp ẩn: lớp thứ nhất có 50 node, lớp thứ 2 có 25 node ẩn.
  - o Activation='tanh'.
  - o Solver='lbfgs'
  - o Max\_iter=1000
  - o 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

#### d) Huấn luyện mô hình.

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

#### e) 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:

```
Model Perceptron:
0.054377379010331704
0.06851549755301795
-----
Model MLP:
0.0
0.008156606851549755
-----
Model SGD:
0.0
0.0065252854812398045
-----
Model Logistic Regression:
0.000543773790103317
0.011419249592169658
```

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

Mô hình tốt nhất là SGD.

Kết quả khi kiểm tra trên tập test:

Đánh giá model được chọn - SGD!

Độ lỗi trên tập test  
0.0

### III) Tham khảo.

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