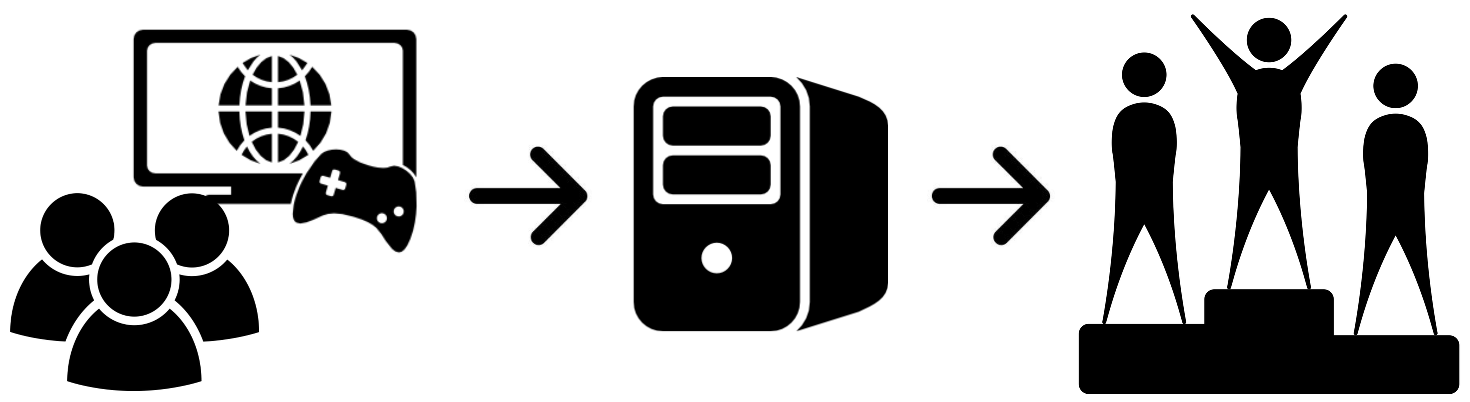
# Player Behavior and Optimal Team Compositions in Online Games

Hao Yi Ong, Sunil Deolalikar, and Mark Peng

*CS 229: Machine Learning Class Project*

## Introduction

- In online role-playing games, players work in teams to accomplish a common objective (e.g., defeating an opposing team)



- In a game, given the teams' player compositions and their player statistics, we want to predict the players' play style and forecast the game outcome

- **Player behavior**
  - In-game play style; e.g., prefers more offense-oriented strategies
  - Also encompasses skill level
  - Predict from player statistics
- **Team composition**
  - Types of players on a team, each classified by their play styles
  - Predict from game server database's match histories

## Problem Description

- Given
  - **Match histories** containing participant IDs and match statistics
  - **Player statistics** containing player histories and overall game statistics
- Output
  - **Play style classifier** that groups players by their in-game tendencies given their game histories
  - **Outcome predictor** that guesses which team will win given the various team compositions
- In order to
  - **Gain insight** on player behaviors and game strategies
  - **Maximize accuracy** on predicting game outcomes

## Target Game

- **League of Legends**
  - Multiplayer battle arena game with 27 million plays per day
  - Free online API to retrieve de-identified game data (120,000 training and 12,000 test samples)
  - Official guide provides clustering information for players based on in-game character choices (e.g., character with good defense)

## Baseline Outcome Predictor

- **Features** are team compositions for each team in a game based on official game guide's clustering information (each character is mapped to a play style)
- **Logistic regression** with 10% hold-out cross validation
- **Poor accuracy** of 55.1% on training samples, 54.4% on test samples

## Behavioral Clustering

- **Features** are normalized player statistics (damage dealt, money earned,...)
- Clustering algorithms
  - **k-means** with 10-fold cross validation over parameter k gave 12 clusters
  - **DP-means** is an expectation-maximization algorithm derived using a Dirichlet process mixture model (Kulis and Jordan, 2012)
    Intuitively, a new cluster is formed whenever a point is sufficiently far away from all existing centroids, as determined by some threshold distance $\lambda$
    We ran it with 10% hold-out cross validation with $\lambda = 3.3$, giving 8 clusters
  - **Implementation** in MATLAB, ran on 2.7 GHz Intel Core i7, 8 GB RAM

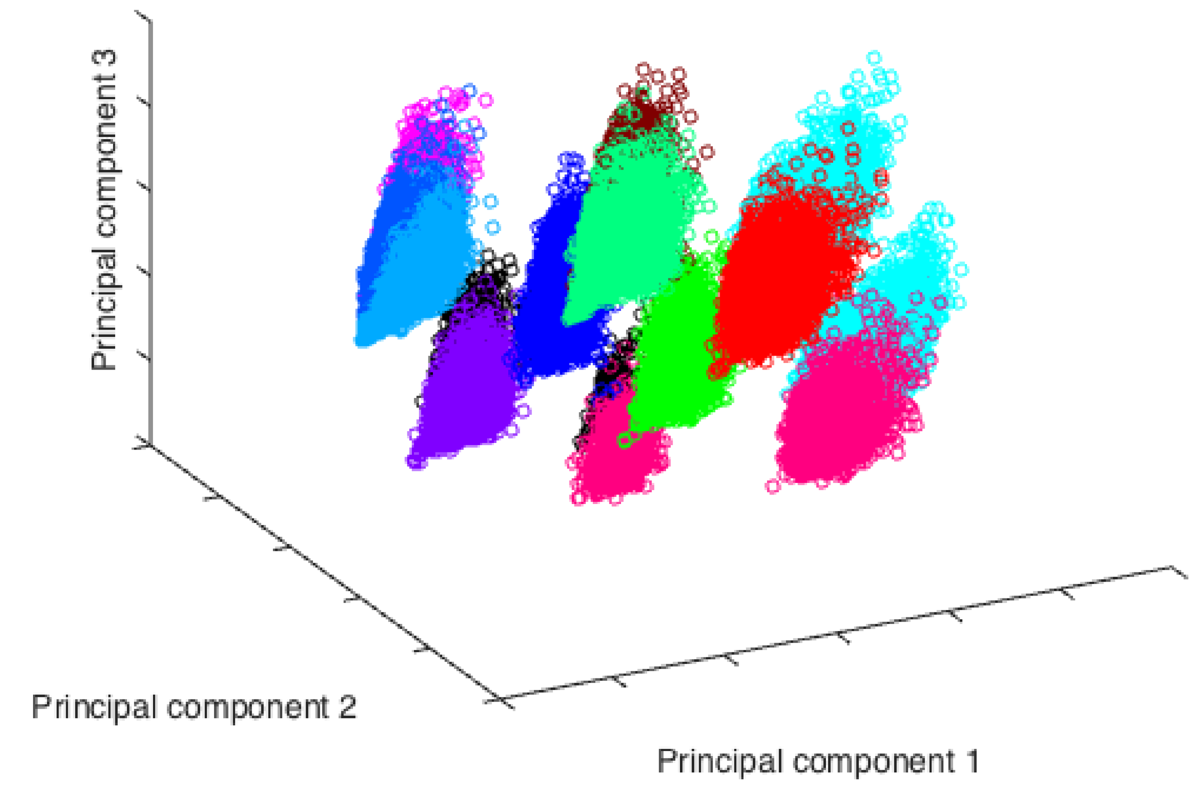
**given** training set of size $N$, threshold distance $\lambda$
**repeat until** clusters converge
  For $n = 1, \dots, N$
    1. Assign sample $n$ to the closest cluster if the contribution to objective from the squared distance is at most $\lambda^2$
    2. Otherwise, form a new cluster with just sample $n$

Summary: Play style clustering algorithm results (10 trials average)

| | cross validation method | no. of clusters | cpu time (s) |
|---|---|---|---|
| k-means | k-fold (k = 10) | 12 | 154.1 |
| DP-means ($\lambda = 3.3$) | 10% hold-out | 8 | 65.4 |

## Cluster Visualization

- **Principal component analysis** We used 3 principal components to visualize our full dataset in 3D; observe that the data is clearly clustered




## Game Outcome Prediction

- **Features** are team compositions for each team in a game based on behavioral groupings generated from k-means and DP-means clustering
- Classification algorithms
  - **Logistic regression** Lorem ipsum
  - **Gaussian discriminant analysis** Lorem ipsum
  - **Support vector machine** Lorem ipsum

Summary: Outcome prediction algorithm results (10 trials average)

| | k-means (%) | | DP-means (%) | | cpu time (s) | |
|---|---|---|---|---|---|---|
| | train acc. | test acc. | train acc. | test acc. | k-means | DP-means |
| LR | 72.25 | 68.75 | 69.67 | 67.11 | 7.4 | 7.1 |
| GDA | 74.79 | 70.14 | 70.88 | 68.39 | 7.7 | 7.1 |
| SVM | 74.75 | 70.39 | 71.71 | 69.21 | 91.2 | 41.6 |

## Conclusion and Extensions

- Lorem ipsum

## Acknowledgements

We thank Professor Andrew Ng, the instructor team, and fellow students for their help with and feedback on our work.