Prediction System
for League of Legends

Gavin Saldanha

Motivation

- Millions of people play every day - affects many
- 5 random people put on team, given a few minutes to pick champions
- (124 choose 10) > 163 trillion possible matchups
- Hard to know the best champions to pick for a particular situation

Overview

- Riot Games keep records of every game ever
- Provide API to access this info
- Can apply Machine Learning techniques to discover “best” compositions
- Can use models to inform players which champions to pick

Dataset

- No easy way to get large number of historical games
- API calls for getting 10 recent games for specific player
- Start with myself in a queue
- For each player in queue:
  - Lookup recent games, add to dataset
  - Add other players from each game to queue
- My dataset has about 45000 games, ~ 1.5 GB

Training

- Used Spark to process data, do all training
- Trained various classifiers
  - SVM with gradient descent
  - Logistic regression with gradient descent, limited memory BGFS
  - Decision trees / random forests
- Compared to baseline method - average champion win rate
- Evaluated on held-out test set, chose best classifier

Results

- Baseline method gets about 53.5% accuracy
- SVM/Regression get 55-56.5%
  - Up to 5% improvement
- Decision trees and random forests perform worse (50-53%)
- As game is skill based, team comp can only be so impactful - unreasonable to expect high accuracy

Website

- Simple Django webserver
- Runs training script on startup, saves model
- Enter both team compositions, predicts winner with confidence measure
- Enter partial team compositions, predicts best champion to pick next