

## **CS446 Machine Learning Semester Project Proposal**

### **Mobile game (Clash Royale) match result prediction using multiple machine learning algorithms**

- **Group Members**

- Zheng Li (zhengli6)
- Fangwen Wu (fwu11)
- Yiming Wang (wang532)

- **Background and motivation**

Clash Royale is a worldwide popular real-time strategy mobile card game. It won the best mobile game of the year 2016 and has more than 70 million downloads on App Store. The game consists of building decks by choosing 8 out of 70 battle cards and real-time battling with opponents. Cards are characterized by its type (troop, building, spell), rarity (common, rare, epic, legendary) as well as level. Given the different combination of cards and play skills, it brings complex dynamics and outcome of the game.

Given the popularity and complexity of the game, a research question can be asked as is there a deck that always performs better than another one ? Does the cards combination itself provides enough information for us to tell if it will win against its opponent? We acknowledged that human plays a very important role in this game and the same deck with the players from the different levels will yield significant contrasting outcomes. Consequently, selecting worldwide top 1000 players' match records as samples will sufficiently represent the strengths and weakness of deck since we can assume that all those players would have excellent skills and the skill difference is relatively subtle. In this game, the players are only competing with the players with similar ranks. In addition, top 1000 players always have maximum level cards which eliminates any bias caused by cards level differences. Thus, we are able to exclude unrelated factors (human skills, card levels, etc) and focus on the combination of cards that determines the match outcome: win, loss and draw.

- **Methods and Implementation**

The data set of this project comes from the real world match results. We use the approximately 200,000 recent match records (win, draw, loss) from the top 1000 players from *Stats Royale*[1] which is a website contains the information of real-time top 1000 players game tags as well as their recent match records. We will write crawler program to pull the data from the website and use packages like Pandas, SkLearn to clean and organize data. It is noted that the website contains real-time match records, we can always extend our training data if it requires more samples.

The learning problem can be identified as a multi-class classification problem since the outcome (win, loss, draw) of a match is not binary. The feature space can be defined as  $\{0,1\}^{140}$  since currently there are 70 cards in the game and two players in one match.  $x_1$  to  $x_{70}$  represents deck used player 1 and  $x_{71}$  to  $x_{140}$  represent player 2's deck. Since for each battle, at most 16 cards will be used, so 16 out of 140 features may not be expressive enough. So we will adapt features like cards combination, type of cards, rarity of cards in the deck, etc to make feature space more tight. The proposed machine learning training algorithms for this problem include multi-class SVM [2], Random Forest [3], Naive Bayes and Neural Network. We will implement several algorithms and compare the results to select the one with the best performance.

- **Project widget:**

Time	Content	Check Point
3.13-3.26	<ul style="list-style-type: none"> <li>● Get clear view of the whole project</li> <li>● Know better about data available</li> <li>● Write scrapy program to collect data from website</li> </ul>	<ul style="list-style-type: none"> <li>● Feature list for this project</li> <li>● Program to scrape data</li> <li>● data scraped from website</li> </ul>
3.27-4.2	<ul style="list-style-type: none"> <li>● Clean and organize data</li> </ul>	<ul style="list-style-type: none"> <li>● Well organized data for further learning</li> </ul>
4.3-4.30	<ul style="list-style-type: none"> <li>● Implement machine learning algorithm to train the model</li> <li>● Compare performance of different algorithm and tune parameter</li> </ul>	<ul style="list-style-type: none"> <li>● Programs for machine learning algorithms</li> <li>● Table for performance comparison and parameter tuning</li> </ul>
5.1-5.7	<ul style="list-style-type: none"> <li>● Make further discovery of this problem</li> <li>● Summary and write report</li> </ul>	<ul style="list-style-type: none"> <li>● Final report</li> <li>● Organized programs for this project</li> </ul>

- **References:**

- [1] Stats Royale website: <https://statsroyale.com/top/players>
- [2] Mohamed Aly. "Survey on Multiclass Classification Methods" *Technical Report, Caltech* (2005)
- [3] Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.