Machine Learning Application in *League of Legends*

Guanyu Zhang, Jiabao Liu, Kaiquan Liu,
Limeng Li, Ruoyi Chen

Part 1: Background

1.1 MOBA Games

Multiplayer Online Battle Arena (MOBA) games are an offshoot of Real-Time Strategy games.

In MOBA's players take charge of a single character as part of a team. Teams of players are taxed with the destruction of the opposing teams structures, with the critical structure found in the center of their base on one side of the map. Players must work as a team to destroy these structures. Typified by two spiritual successors of its origins, namely Valve's *DotA2* and Riot's *League of Legends*, MOBA games have generated the most power and energy in the 120 billion dollars video gaming industry. Also MOBA gave the birth of the largest eSports in the world in terms of the prize money awarded (a prize pool of over half a billion in 2018)

1.2 ML Application in MOBA Games

Since MOBA games had generated over billions of revenue in recent years. All major players are concerned about how to keep games alive and more profitable. And the core of their strategy is machine learning. There are a variety of ways in which machine learning can potentially impact areas of MOBA development, consumption and engagement.

Content Patches and Balance: As MOBA games are RTS battle games, the process of balancing each character for fair competition is crucial and long-lasting. Using performance data and algorithms to predict how major changes of a given patches would do to the overall fairness is an ongoing and also potential tool for game designers.

Community: Sadly, MOBAs have a poor reputation for their community and toxic players' behaviors. Cheating, passive in-game behaviors are really a major problems for companies. One great attempt is to use machine learning technique to smartly monitor in-game behavior and filter out the toxic players and give them warnings or fair punishes.

Items Pricing: MOBA games have a strong IP, meaning its characters in game can be very well-known and liked by players. Thus in game purchases like Character Skins, Decorations are considerably profitable given its low cost. In this case, an effective pricing strategy is important and also another field where machine learning techniques can be handy.

AI BOT: Artificial Intelligence BOT is gradually become a popular trend for MOBA game companies to attract players and display technical powers. Deep learning method applied in those AI BOT will continuously make them stronger and more competitive versus human player in the future..

ESports Analysis: Like traditional sports analysis, pre, in-game and post-game data can all be used to perform eSports analysis. Identifying players' patterns, focusing on winning details are becoming a main part for every eSports team.

1.3 Analysis Based on League of Legends of Riot's Games

In this report, our team will mainly focus on the game League of Legends and its provider Riot's Games. It is the major player and it is very representative regarding the business problems of the whole MOBA games industry.

Based on company's announcement, players' feedbacks and personal experience, our team identify the top three issues that would greatly affect the company's operations, profits and competitive strength in the market. The three issues are Balancing, Skin Pricing and Misbehavior Detection. A proper balancing will guarantee fairness within the games, a proper Skin pricing in League of Legends will affect the greatest portion of the company's revenue and misbehavior detection will help build greater community and increase player experience. Our general idea is to use predictive models to figure out the difficulty within each issue and deliver supports from prediction for companies to make business decisions.

1.4 Data Source and Overview

Fortunately, Riot's Games stored massive amount of data of all type including general stats of each player, detailed game data of every game, and complex in-game data stored in battle log. Moreover, the data is publicly released by Riot's Games in an online API. So we can easily access valid raw data for our modeling.

In this report, we will illustrate how we would establish predictive models to address the problems we want to solve. There are three models using different observations and modeling method to respectively address three problems. In the end, we would also deliver a decision support system interface to assist with the Misbehavior Detection problems.

Part 2: Predictive Problems

2.1 Skin pricing

2.1.1 Description

Skins sales is one of the main revenue resources for the league of legends game. To maximize revenue, the game company needs to predict how many players will buy a skin at a price. To sale skins targeting at certain groups of players, the game company needs to predict whether players with some features are willing to buy the skin. Our skin pricing model is used to solve these problems. By analyzing the lease features and ID features of a player, our model will classify players as 'will buy' or 'not likely to buy' and therefore help managers to make decision.

2.1.2 Model

Input Variables	**Definition**
SkinPrice	The expected releasing price of the skin
Player ID	Each player has an unique account

Feature Variables	**Definition**
SkinOwnNumbers	The number of skins the player already has.
ChampionName	Which champion a skin is designed for?
HistoricalPurchaseRecord	The skin purchase record of a player in the past.
ChampionPreference	A player's preference for the champion of a skin.
Series	Whether a player has other skins that is in the same series as the new skin.
SkinTier	Each player has an unique account

The model treats individuals as units of analysis. Given a price, the model uses a series of variables to predict whether a player will buy a specific skin. This prediction process will be repeated under different prices. Then we can analyze the results to see which price can attract most buyers for the skin.

We choose six variables to predict whether a player will buy a skin. The first three variables describe features of the skin, and the last three variables describe a player's characteristics. SkinOwnNumbers and HistoricalPurchaseRecord reflect a player's purchase habit and income level. SkinOwnNumbers can be measured by the total skin purchased, or average number of skins purchased per month.

HistoricalPurchaseRecord can be measured by total amount spent on skin purchases or average price of skin purchased. ChampionName will influence players' decisions, since people only buy skin for the champions they have. SkinTier influences players because the rarer a skin is, the more likely players want to obtain it. ChampionPreference influences players because people tend to buy skins for their favorite champions. The preference can be measured by 'proficiency'. Series is another important variable. People usually have psychological trend to have a whole set of skins. If a player has one of a group of skins, he or she is more likely to buy other skins in the same series.

Training data of this model is the historical data for skins sold in the past. All these data can be directly obtained from players' records. After analyzing the above six variables, the model will classify the player as 'yes', which means the player are likely to buy the skin, or 'no', which means the player will not buy the skin.

When analyzing some special skins, we can add some constraints to the model. Take volume constraint for example, the number of some rare skins are limited, so when the predicted number of sales far exceeds the total number of skins, we can set a higher price for the skin. Since the game company doesn't provide so much skin to meet the high demand.

2.1.3 Validation and Measuring Criteria

We will use 'Accuracy' to measure whether the model successfully solved the problem. That is, the number of players whose purchase behavior are correctly predicted divided by total number of players.

2.1.4 Challenge.

1) The model is based on individuals, so huge amount of calculation is unavoidable. The game company must provide powerful data processor, which will bring higher operating costs.

2) An important factor that is not properly taken into account is heterogeneity. It's hard to use a variable to represent a player's aesthetic preference on a skin's visual presence.

2.2 Game balance

2.2.1 Description

The game balance is the most frequently discussed topic in almost every Multiplayer Online Battle Arena game. An unbalanced game will influence players' game experience and is harmful for the long-run development of a game. As a result, the strengths of a character should be offset by a proportional drawback in a certain area to prevent the domination of this character. In a similar way, the weak points of it should also be compensated by adding some advantages so that this character would not be so vulnerable. But, as the mechanism is complex, there is no perfectly balanced game. Thus, the game designers of League of Legends continuously adjust attributes of the champions. Sometimes, however, it is difficult to find out the champions who destroyed the game balance since the game designers are unable to monitor every game. This predictive model is designed to help them figure out the suspicious champions destroying the game balance and the changes should be made about this champion.

2.2.2 Model

To fulfill this purpose, five features need to be used to measure whether a champion is in the dominant/weak position in this game and whether it needs to be strengthened or be weakened. Our model will take Champion ID as input and using related feature variables to output a variable to label whether the champion needs to be strengthened or be weakened.

The following are those five features.

Feature Variables	Definition
Win rate	The percent of this champion win in a period in the whole game.
Pick rate	The percent of this champion picked by all players in a period in this game.
Ban rate	The percent of this champion banned by all players in a period in this game.
Pro pool	Whether this champion is frequently selected by professional players.
Avg Mastery score	The avg degree of proficiency that a player use this champion.

The reason why these five features are chosen is that if the overall win rate of a champion is far more than others, then it is possible that this champion is in dominant position. Because a high win rate means most of skillful and unskillful players could win using this champion. However, only use win rate may cause some error in prediction. Thus, the other four features are added to improve the prediction accuracy. If a champion is more likely to be picked or banned, it may indicate that this champion is in the leading position, because picking this champion made players more like to win and it also because of that this champion is banned to prevent their rival to win. What's more, if a champion is popular with professional players, it may also means using this champion made them more likely to win. Master score as a final feature also played an important role in determining whether the game is balanced. There must some problems with this champion when players with lower master score in using this champion who could also own a higher win rate.

2.2.3 Validation and Measuring Criteria

Since League of Legends was published in 2009, there must be so many champion adjustments to satisfy the game balance. Therefore, plenty of data from past version concerning these adjustments could be gathered and they are necessary in training the model. It is worth mentioning that assume all changes in the past are reasonable, which means designers will not strength a champion if it already in the dominant position.

The historical data should be first divided into training set and test set. In the training set, for each of the champion, before they are adjusted, select their win rate, pick rate, ban rate, pro pool and master score as variables and use the adjustment result as the output that is be strengthened, be weakened or not change. In the test set, also select these five variables and comparing the results generated by the model with the actual results. If most of results generated by the model are matched with the actual result, then the model could be considered successfully predicted which champion should be adjusted in current version.

2.2.4 Challenge

The biggest challenge in this model is that game designers only know they should weak a champion if it in the dominant position, but they do not know how to do it. For example, they are not sure whether they should weak its attack number or defense number and also, they do not know how much attack or defense should be decreased for this champion.

2.3 Misbehavior Detection

2.3.1 Description

Combating cheats is an ever-evolving arms race. The scope and complexity of cheat development grows every year along with the stakes in online gaming. There are certain behaviors that we discourage in *League of Legends*, the Scripting, Boosting, and Passive behavior. We classify these behaviors as cheating because they ruin the competitive experience that makes us love League. However, the pressure of us is to detect and prevent these bad cheating events, so we are going to use machine learning here to create a predictive model to help us. After getting the result from detection, we could apply different punishment on these users.

2.3.2 Model

When we input the all the input variables we want to focus into the system. The output of these models is whether they cheat or not, so the output should be "Y" and "N". Since the 3 different types of cheating are totally different, to specify each one, we are going to build 3 models to identify these 3 cheating behaviors.

In the first model, we build it to detect players who use Script, the feature variables here are as follow.

Feature Variables	**Definition**
Win rate	The percentage of victory game in this period of each player
KDA	The kill-death-assistance ratio
Reports	The amount of reports from others of each player
Hit rate	The hit rate in all the game of each player

To determine one player uses script or not, we have to analyze its behaviors in game. If the win rate or KDA of one player is too high, we consider that they are more likely to use the script, for example, they may have the wards of all map. However, some skillful players could create excellent win rate and KDA in game, so these two features are just part of the variables we need to consider. At the same time we will combine other features to help us determine. The hit percentage should also be taken into account. If one player always has extremely high hit rate, there is the chance that the script helps him to use abilities, as a result it will ruin the balance of LOL. The reports from other players are also valuable, if one player receive too many reports from different players, the probability of using a script is high.

In the second model, we build it to detect players who invite boosters. The feature variables here are as follow.

Feature Variables	**Definition**
Win rate	The percentage of victory game in this period of each player
KDA	The kill-death-assistance ratio
Deviation of the most proficient champions	The recent 10 champions the player uses most/ The 10 most proficient champions of the player
Reports	The amount of reports from others of each player
In-game behavior	The in-game text contains some sensitive words

The win rate and KDA are the most significant features here to detect boosters. Because the player who invites booster is to help them to a higher rank. If some players have the extremely high win rate and KDA than before, we will treat them as suspicious. Another important feature is the deviation of the most proficient champions, if someone who uses totally different champions recently to win a game, we will consider this player might invite boosters. The reports from other players are also valuable, if one player receive too many reports from different players about inviting boosters, the probability of boosting is high. The in-game behavior detecting is effective, if one player said some sensitive words like boosting in the game, we will consider they are prone to invite boosters.

In the third model, we build it to detect players who have some negative behaviors. The feature variables here are as follow.

Feature Variables	Definition
Win rate	The percentage of victory game in this period of each player
KDA	The kill-death-assistance ratio
AFK	The time of away from the keyboard
Reports	The amount of reports from others of each player
In-game conversation	The in-game text contains some sensitive words
In-game behavior	Abnormal behavior in the games such as buy no items and equipment in a game or intentional feeding

The players behave negatively in the game because they may have some conflicts with their teammates, consequently they take some negative actions like die on purpose or AFK. The win rate and KDA will be low if they want to lose game easily, like ADC produce a 0-10 KDA. The AFK measures the time of away from the keyboard, if a player doesn't take actions for a long time in one game, it will affect other players' game experience. The reports from other players are also valuable, if one player receive too many reports from different players about negative behaviors, this probability will be high. We also could detect the in-game text to select the negatively behaved players, if one player curses other teammates with dirty words, the probability of behaving negatively will be high. Besides, we can also detect the abnormal behaviors which indicate the player is not play for the win and causes the negative effect on its teammates.

2.3.3 Validation and Measuring Criteria

Although the variables used in above three models are different, the rationale behind these models are similar. We could use the machine learning models to classify and label player, such as decision trees, randomforests and support vector machine etc. 70% historical data of cheaters and normal players can be used as training data, and 30% of it could be used to choose a fittest model. After we built the model with best performance, we put the specific data of each players into our model so that we can find out if they are suspicious. And if the final accuracy, precision, and recall have great performance, we could treat the model as success.

After getting the prediction we have predicted, we could trace these users' specific performance more thoroughly into each game, and then judge it cheat or not. Comparing this result with the earlier prediction, we could measure our success.

2.3.4 Challenge

The first challenge here is how to handle the text information in the game. The information text could act in different kinds of format, and sometimes it's vague to tell even for people. So detecting these vague text information is difficult for the machine learning. Another challenge is that we build this model on the previous data, which could be considered as training data, so we have to assume we have made correct detection before, only with this assumption, our prediction could be right.

Part 3: DSS Interface - Misbehavior Detection

We designed the Decision Support System to help the employees in Riots detect the suspicious account. As described in Part 2, the predictive model could help label the players who is more likely to have cheating behaviors. The anti-cheating system is linked to the player data thus when using the system, there is no need for the users to enter the details information of individual player.

In the end of 2018, League of Legends has an active player base of over 80 million monthly players, or over 27 million players every single day. There are millions of the players to be checked by our predictive model. In order to make the prediction more specific and efficient, the system allows users to filter certain players to check by selecting the periods, regions, ranks and maps. In addition, the users could choose to check a specific type of cheating. After that, the system would run the predictive model for each player who satisfied the pre-set criteria and label it as suspicious or not.

The Drop-down menu of each filed are listed below;

Field	Drop-down menu
Region	Brazil; EU Nordic and East; Japan; Latin America North; Latin America South; North America; Oceania; Russia; Turkey
Rank	Iron; Bronze; Silver; Gold; Platinum; Diamond; Master; Grand Master; Challenger
Map	Summoner's Rift; The Crystal Scar; Twisted Treeline; Howling Abyss; Butcher's Bridge; Cosmic Ruins; Valoran City Park; Substructure 43; Crash Site; Nexus Blitz
Match	Custom; Normal; Ranked

The "Overview" tab would display the total number of the suspicious accounts and the "Details" table then shows the general information of all the accounts labeled as suspicious. If the user clicks into each record in the "Details" tab, the data linked to that account will be stated in the "Account Information" tab, including match history, reported record and IP address etc. By doing so, the system helps users identify accounts for further investigation.

The filter is used to narrow down the checking area. If the requirements are not specified, more players would been involved in our analysis. For example, if the users selected all three types of cheating, the system would run the three models for each user separately. The "Overview" tab would show the numbers of suspicious accounts for each type of cheating. If the Region is selected as all regions, the players in all the regions would be put into the predictive model and labeled as suspicious account or not, the "Overview" tab could then show the distributions of suspicious account in different region.

The Anti-Cheating system should be maintained by the data analytics in the Riots. The company could generate more and more data over time, therefore the model should be updated periodically to increase its accuracy. Overall, this system should be helpful to assist the users find the suspicious cheating account for further investigation purpose.

Sample DSS Interface: