A Team Based Player Versus Player Recommender Systems Framework For Player Improvement

BLINDED FOR REVIEW

Abstract

Modern Massively Multi-player Online Games (MMOGs) have grown to become extremely complex in terms of the usable resources in the games, resulting in an increase in the amount of data collected by tracking the in-game activities of players. This has opened the door for researchers to come up with novel methods to utilize this data to improve and personalize the user experience. In this paper, a novel but flexible framework towards building a team based recommender system for player-versus-player (PvP) content in such MMOGs is presented, and applied to a case study in the context of the major commercial title *Destiny 2*. The framework combines behavioral profiling via cluster analysis with recommendation systems to look at teams of players as a unit, as well as the individual players, to make recommendations to the players, with the purpose of providing information to them towards improving their performance.

1 Introduction

Massive Multi-player Online Games (MMOGs or MMOs) are played by large number of players. Particularly popular today are Multi-player Online Battle Arena games, also known as MOBAs, which are a subsegment of the MMO market. Global revenues in the area of 30 billion USD yearly by end of 2017, growing from 24.4 bio. USD in 2014, with the revenue split almost equally across role-playing focused MMOGs, First-Person Shooter-focused titles such as *Destiny* and real-time strategy titles such as Multi-Player Online Battle Arena games (MOBAs) such as *League of Legends* and *Dota 2* (Statista 2017).

MMOGs and MOBAs have become increasingly complex in nature, which in turn drives players to learn more about them to improve at these games. Players are faced with numerous in-game choices that determine their success and mastering an MMOG has become a complex and time-consuming process. While this complexity might interest some players, it can also hinder performance improvement, which can lead to frustration and abandonement of the game. Similarly, complexity can deter new players from taking up the games. These topics are currently debated in e.g.

esports (games played competitively), but as yet there is limited evidence presented on skill learning in games (Stafford et al. 2017; Block et al. 2018; Aung et al. 2018). Along with the interest in adaptive games and the emergence of esports, the ability to employ the massive amounts of behavioral telemetry captured from commercial game titles towards helping the players improve their experience, performance and decision-making has emerged as a topic in game development (El-Nasr, Drachen, and Canossa 2013). One of the ideas discussed in the games industry is the application of **recommender systems** (Weber 2015; Drachen et al. 2016). Such systems can e.g. be applied to recommend areas to train or strategies. MMOGs like *Destiny 2* are complex and require so much decision making that players might not be aware how to improve their performance - across skill trees, training with weapons, which weapons to equip, which raids to participate in, which classes to play etc. While playing more will possibly help, the goal of *player-focused recommender systems* (as opposed to systems for monetization or adaptive systems (Weber 2015; Yannakakis 2012), is to assist players through the use of data. What makes recommender systems challenging to integrate in digital games is that players will have different preferences on how they play. A system therefore has to take a basis in the playstyle of the individual players (Sifa et al. 2018). In this paper, a multi-profile team based recommender system framework is proposed. The goal of the framework is to inform teams of players about the choices made by "better" teams - where better can be different features e.g. win ratio. Robust recommendation systems for competitive multi-player games have not been thoroughly explored (Sifa et al. 2018), and not at the team level. However, since MMOGs and MOBAs are based on team PvP (Player vs. Player) or feature such elements, it is of interest to look at the team as well as individual players. We apply the framework to the case of *Destiny 2*. Recommendations are intended to help teams improve by suggesting performance metrics/playstyles to improve and weapons to use, based on what higher performing teams who are more or less similar to the active team are utilizing.

2 Related Work

Game Analytics, a domain of business analytics focusing on games, has grown as the core topic in game research and

development today (El-Nasr, Drachen, and Canossa 2013; Sifa, A., and Bauckhauge 2018; Drachen et al. 2016; Normoyle and Jensen 2015). Games like *Destiny 2*, tend to collect a large amount of data with thousands of features for each player. This introduces opportunities for scientists to explore methodologies to improve the player experience (Stafford and Haasnoot 2017; Goldstone and Lupyan 2016).

Player retention in games is of key concern in digital games irrespective of the business model employed, and notably in online and/or persistent games such as *Destiny 2*. Towards this, player churn has been investigated across industry and academia. For example using survival analysis via Mixed Effects Cox Regression (Demediuk et al. 2018). Other works have looked at the problem as a marketing model of customer retention (Debeauvais et al. 2011), by using metrics such as play time, stop rate and number of years players have been playing the game. Hidden Markov models have also been explored to predict player churn (Tamassia et al. 2016). The first step towards building player-facing recommender systems is to create behavioral profiles for players. Several papers about profiling in games originating in Game AI (Yannakakis 2012) and business intelligence requirements (Drachen et al. 2012b) have been published. Cluster analysis have been used to reduce the dimensionality in behavioral telemetry (Normoyle and Jensen 2015). Previous work has looked at creating behavioral profiles for the players of *Destiny* to identify the best cluster models in high dimensional space (Drachen et al. 2016). Other previous work has presented behavioral profiling in commercial game titles (Drachen, Canossa, and Yannakakis 2009), dealing with self-organizing networks, and this was followed up by inspecting how profiles changed with progress in the game (Sifa et al. 2013). While in-game recommender systems remain relatively unexplored, there have been many proposals to recommend games similar to other commercial products like movies based on past preferences (Cutler and Breiman 1994; Cremonesi, Koren, and Turrin 2010; Sifa, Bauckhage, and Drachen 2014; Weber 2015) and using collaborative filtering methods (Anwar et al. 2017). A recommender system for individual players in *Destiny* was explored by Sifa et al. (Sifa et al. 2018). This analysis expands on this by operating at the level of teams. Another aspect considered in this analysis is the *learning rate* or the rate of skill acquisition of players and/or teams. Some teams learn and improve faster than others and achieve higher eventual levels of skill and performance. Recommending the strategies of teams that learn faster could be beneficial to beginning players. Variability in elements of play, like practice spacing and social play, affects subsequent skill development (Stafford et al. 2017). It has been suggested that greater initial variability in practice may drive higher subsequent performance (Stafford et al. 2012; Stafford and Dewar 2014), agreeing with computational accounts of how learning must balance exploration and exploitation of options (Sutton and Barto 1998; Humphries, Khamassi, and Gurney 2012). Variability is as much an engine of learning as consistency (Schmidt 1975; Van Rossum 1990; Newell and McDonald 1992; Ranganathan and Newell 2010). This raises the question of exactly which kinds of variability, and in what quantities, support optimal skill acquisition. On a final note, the aim here is to inform players rather than developing the most precise cluster solutions.

3 A PvP Recommender System Framework

Most esports games have teams of players competing against each other. The same is the case for MMOGs where team-based PvP forms at least part of the gameplay, e.g. *World of Warcraft, Destiny 2, Shen Zhou Online*. Players are typically provided with a variety of customizations for the character they choose, such as weapons, armor, abilities, etc. They are also free to manipulate their playing style, which is naturally dependent on their customization choices (and vice-versa) as well as innate tendencies they may have built up. However, it is still possible for players to consciously alter the way they play to win more matches.

A strategy that only optimizes individual player statistics provides no guarantee of winning games as a team. A successful team will not only make the optimal customizations and strategic decisions for each player but also find the right balance across the whole team. An experienced team may be willing to make smaller changes to their in-game selections and play styles than a less experienced one. The right suggestions for an experienced team would be different from those for a team still learning the ropes. A less experienced team may want to speed up their learning, while a veteran team would likely want fine-tuning suggestions to improve their win-loss ratio. In essence, there is an opportunity to provide players with relevant information to help them make these choices.

The objective of the team based recommender system is to assist teams in improving their performance by exposing them to the customization and play style choices of other similar but better teams. The system proposed here builds on Sifa et al. (Sifa et al. 2018) by proposing multiple player profiles, and aggregate them on a team level to create a profile for a team.

The general procedure to provide recommendations (also shown as a flowchart in Fig 1) is described below. Further information on these steps is provided in the following subsections, covering player profiling, clustering, recommendation features and team mapping.

1. Choose Profiles: Select the profiles that will be used to define players and teams.

2. Create Features and Cluster: Define features to construct these profiles

3. Choose a method to cluster players within these profiles.

4. Create Player Profile Vector: A one-hot vector with as many elements as there are total clusters in all the profiles, '1' indicating the membership of that player in a cluster.

5. Create Team Profile Vector: The summation of the player profile vectors, representing the number of players belonging to every profile considered. The elements of this vector, corresponding to different profiles, may be multiplied by different factors, giving them appropriate weight depending on their significance.
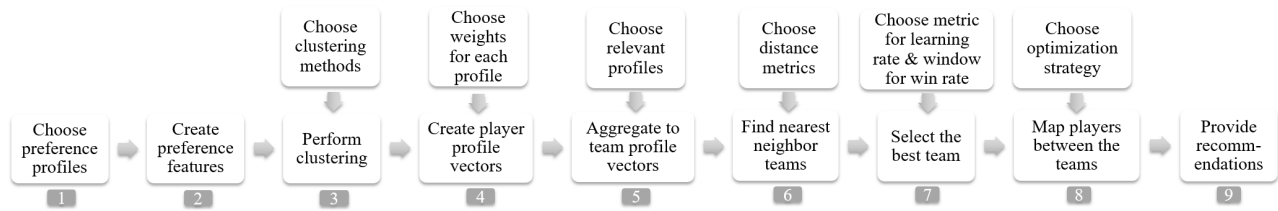
Figure 1: Flowchart describing the process and the multiple choices to be made for the recommendation system framework.

6. Find Similar Teams: Find a pool of teams closest to the target team using the nearest neighbors algorithm with the appropriate distance metric. Fix a minimum of T teams in this pool. T would be dependent on the amount of data.

7. Select "Best" Team: The team with the highest win rate or learning rate (based on the maturity of the target team) is selected from this pool, after choosing the metric to define learning rate and the matches M for win rate.

8. Map players between teams: Map the players from the selected best team to those of the target team. This is done by considering distances between players themselves, represented by the player profile vectors. The mapping process is discussed in detail in the next section.

9. Provide Recommendations: For each of the players in the target team, provide recommendations based on the mapping between the target players and the best team, i.e based on the features selected show the difference between the matched players in the target and best team.

Player Profiling with Clustering

Clustering has been used extensively to create profiles of users in multiple recommender systems (Drachen et al. 2016; Normoyle and Jensen 2015; Sifa, A., and Bauckhauge 2018). The idea is to group the players based on some key features that define player behavior and reflect on future behavior, such as weapon or armor selections, attribute allocations such as agility, health, special abilities, etc. Different clustering methods have been used in player profiling based on the game requirements like centroid seeking strategies such as k-means and convex hull methods such as archetype analysis (Sifa et al. 2018; Drachen et al. 2012a).

A profile category contains defining features which are used to create clusters for that profile. Given the multitude of choices that players can make in esports games, creating a single player profile for every player would be suboptimal due to the loss of granular information, and the difficulty in making this profile very interpretable or actionable. Therefore, a collection of profiles should be used depending on the objectives of the game. These profiles can be largely grouped into three major categories based on choices made before and during matches.

1. Character Preference Profiles: These profiles consider the character that a player selects, such as a *medic* or a *recon* in a shooter like *Battlefield*, a specific character in a MOBA such as *DoTA*, or any other fundamental character class in a game. This choice can often be definitive in

terms of the availability of equipment, base statistics, abilities, and optimal gameplay strategy to contribute most effectively. This profile might not be applicable to some games such as *Fortnite* or *Rocket League*.

2. Equipment Preference Profiles: These profiles consider choices made outside the match itself, such as equipment (weapons, armor, etc.) that a player tends to prefer (or is more adept with), allocation of different skills or powers, etc.

3. Play Style Preference Profiles: These profiles look at the in-game behavior of players. For example, they could capture tendencies to use certain abilities, group up and aid teammates through assists or revives, make objective based contributions such as flag capture equivalents, etc.

With more data and complexity, these profile categories can be subdivided into several profile types, such as armor, weapon and stat-allocation profiles within the equipment profiles. Similarly, measurable contributions to various game objectives may be separated from other play style attributes such as player aggression, teamwork or preference for a type of play. All the features should be normalized due to different units and sparsity. After clustering players to assign them profiles, the profiles are aggregated to the team level as discussed later.

Features to Recommend Against

When attempting to provide recommendations that can help teams improve their PvP performance, it is important to carefully choose the features to recommend against. These features should be able to identify "better" teams. The recommendation would be made using two different perspectives of "better" teams as defined below.

1. Win Rate: Measures the overall success of a team. A simple way of defining a better team is a team that wins more often. Recent matches tend to indicate more about the success of a team than older matches. Therefore, only the most recent M matches played by a team should be considered. Win rate is then defined as the proportion of the last M matches won by a team. M can be changed depending on the population a developer is tailoring their system for and the richness of the data, for example, including more matches if only considering mature teams.

2. Learning Rate: Teams that are just starting out are interested in knowing how they can improve and learn faster as opposed to statistically increasing their current win rate.

It has been suggested that higher variability in initial practice drives higher eventual performance (Stafford et al. 2012; Stafford and Dewar 2014). Players have a tendency to under-explore the space of possible actions, thus, learning sub-optimally (Stafford et al. 2017). To introduce this variability, players are provided recommendations from teams that are learning much *faster*. The two important questions that pop up are: learning *what* and *faster* how?

The *what* is a fundamental *metric* to the game, such as the Kills-Deaths (KD) ratio in combat games, an intuitive metric such as Scoring Chances Created vs Scoring Chances Conceded in a low scoring sports game, or Average Number of Crashes in a team racing game. To define *faster*, consider the slope of the linear regression line fit to the team's average *metric* against match number (Stafford et al. 2017). The higher the slope, the faster the team is improving. While improving such a metric may not immediately guarantee more wins, building up these skills would leave a team better placed to start making finer strategic decisions in order to win more frequently at a mature level of competition.

Player to Player Mapping Between Teams

Once the best team is determined, the players from the target team are mapped to the ones from the best team using the player profile vectors. This is necessary to provide information regarding what changes players should make. If teams are comprised of n players, $n!$ player mappings are possible. While an intuitive approach would be to select the mapping which minimizes the total distance across all $n!$ pairs, it might not be the most actionable mapping. This is because it is possible, especially with higher dimensional data and fewer teams in the population, that all the pairs of players are slightly different from each other when total distance is minimized. Players may instead prefer a sequential optimization for mappings of the first p players, followed by the remaining players. Minimizing the total distance for p out of n players would result in at least these p players being mapped to genuinely similar counterparts in the best team. p is another tunable parameter depending on data availability. $p = n$ would minimize the total distance across all pairs. When mapping players, the character preference features may be given higher weight by multiplying the corresponding elements of the player profile vector by a number C greater than one, if the character preference has a high impact on the weapon preferences and play style of a player. The choice of p and C is referred to as the optimization strategy in Figure 1.

4 Evaluation for *Destiny 2*: A Case Study

Destiny 2, the sequel to 2014's *Destiny*, is a hybrid online first-person shooter game blended with MMOG elements. Activities in *Destiny 2* are divided among Player versus Environment (PvE) and Player versus Player (PvP) game types. This analysis focuses on PvP, where players participate in four-versus-four team combat.

Destiny 2 provides a variety of customizations on both character initialization and weapon loadouts (the list of weapons carried by a player), which have a large impact on the playing style and success of the players. For example, individual weapons vary significantly through a variety of lore-based advantages, as well as a differing balance of major attributes such as magazine size, range, stability, reload speed, rate-of-fire and damage dealt. Using the correct weapon for their playing style, given the situation, can greatly improve a players effectiveness and scores.

Dataset Description

The datasets were pulled from a random sample of 100,000 players who played at least one PvP match, and had at least two hours of gameplay. All 1,077,413 PvP matches for those players from September 2017 to March 2018 were extracted as JSON objects through the Bungie API along with 60 metrics for each player per match. A team was defined as a set of four players which plays a match against another team of four players. Since the goal was to recommend better teams, a threshold was set for the minimum number of matches played as a team. Given the size of the dataset and the cumulative distribution of matches played by all teams (Figure 2), only teams that played more than 8 matches were considered. After applying these filters, the dataset was reduced to 9200 matches, 935 teams, and 3579 players.

Team Based Recommender System

The Gaussian mixture model (GMM) served as the clustering method of choice. GMM is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. GMM generalizes k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. GMMs, unlike k-means, do not assume equal variance of all the features in a cluster, nor the approximately equal size of all clusters, hence offering a superior alternative due to their flexibility. The steps outlined in Section 3 and Figure 1 are followed to build a recommender system for *Destiny 2*:

Steps 1, 2 & 3: In accordance with the three profile categories discussed in the framework section, the following features and profiles were created using GMMs.

Character Preference Profile: In *Destiny 2*, players can choose from three character classes, namely "Titan", "Warlock" and "Hunter", varying in skills, resiliency, agility, recovery, abilities, and super-powers. A player's choice of character class has an impact on their optimal gameplay strategy, as well as choices of class-exclusive weapons and armor. The character preference profile contained three clusters, one for each of the three character classes.

Weapon Preference Profile: Weapons were considered for equipment preference, given their impact on playing style and performance. Players can equip three weapons from among 14 different weapon classes such as Auto Rifles, Fusion Rifles, Shotguns, etc. Weapon preference features were created to quantify players' preferences for these 14 different weapon classes. The features captured the total kills earned with each weapon class as a proportion of total kills across all matches for each player. Figure 3 shows the distribution of total community kills with each weapon class, displaying the overall popularity of the weapon classes.
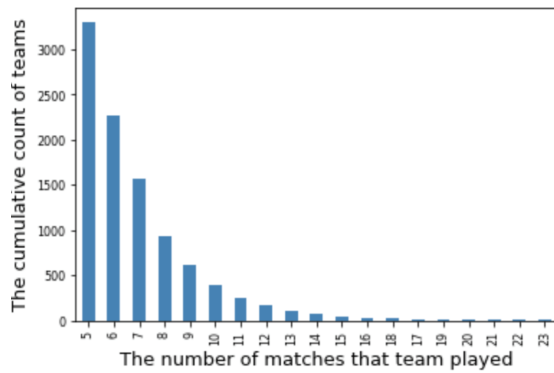
Figure 2: The count of teams by the number of matches played. The count is cumulative starting from the last bar (23 matches) which helps in picking the minimum threshold of matches played as 8 since there are close to 1000 teams. The bars start from 5 matches for smooth visualization.
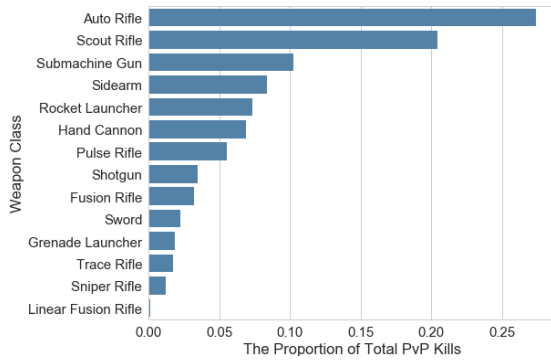

Figure 3: Distribution of total kills by weapon class.

Figure 4: 7 clusters describing players' weapon preferences. Observe how the Auto and Scout Rifles are most used, also seen in Figure 3. Some clusters focus on weapon range. Cluster B shows heavy usage of the mid-range weapons, here, Shotguns and Submachine guns. Other clusters such as D, show a heavy preference for a specific weapon, here, Pulse Rifles.
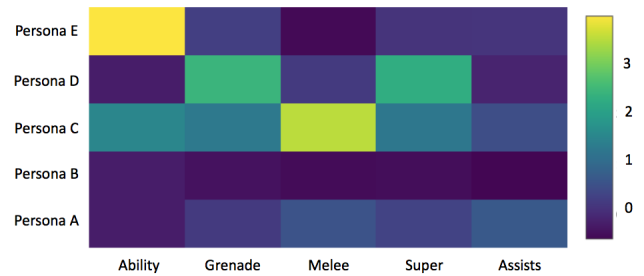
Figure 5: 5 clusters describing play styles of players based on non weapon based skills and assists.

Cluster	Description	Inference
A	High assists, low on other skills	Skills for assists
B	Low assists and other skills	Weapons for kills
C	Very high melee, high other skills	Frequent Close combat
D	More kills with grenades and super	Grenades judiciously
E	Very high on Ability	Character abilities for kills

Table 1: Play style cluster definitions

The weapon preference features were used to build the dimension-space for the weapon preference profile and Figure 4 shows a heatmap of the seven profiles thus created.

Play Style Profile: Five features were created to indicate the behavior of players and reflect the role and performance of an individual players within a team.

1. Kills from Non-Weapon Skills: Kills can also be obtained in *Destiny 2* without using weapons. These features are created to capture such kills and, just like the kills from weapons, are proportions of the total number of kills.

   (a) Kills from Ability: The proportion of kills earned by special abilities such as solar, void, and arc abilities.
   (b) Kills from Grenade: The proportion of kills earned with hand grenades.
   (c) Kills from Melee: The proportion of kills earned using melee in close quarters combat.
   (d) Kills from Super: The proportion of kills earned using superpowers, which vary across the character classes.

2. Assists: An *assist* is when a player helps another player kill an enemy without scoring the killing shot themselves. Teams scoring proportionately more assists would likely
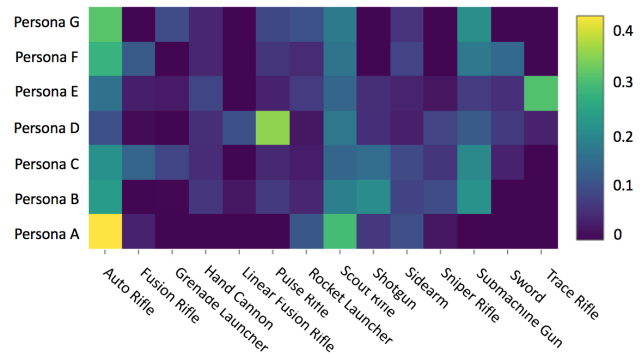
tend to stick together rather than playing as "lone wolves", and have multiple players target the same opponent.

This play style profile is built in the dimension-space created using these features. The five clusters created are described in Table 1 and a heat map is shown in Figure 5.

It should be noted that these features do not capture the specific use of abilities, grenades, melee or super abilities to either score assists or in the case of abilities, help the team in other ways such as creating a "healing circle". Unfortunately, these actions are not captured in the data but would help refine the play style profile further. Specific actions towards objectives for different game modes, such as zone-

captures in the "Control" game mode, are also not captured.

All the features were normalized to avoid the issue of different units and sparsity of weapon usage since some players might only make use of a few weapon classes. To decide the optimal number of clusters for the GMM clustering in this analysis, the optimal elbow point in the scree plot for the BIC (Bayesian Information Criterion) was looked at, keeping in mind the interpretability of the resulting clusters.

Steps 4 and 5: The player and team profile vectors were created using the three profiles discussed above. The elements of the team profile vector corresponding to the character class were multiplied by a factor of 1.5 in order to give more weight to differences in the character class composition, given the impact of the character class on the optimal play style and weapons available to players in the team.

Steps 6: The L^1 distance was used to calculate distance between team and player profile vectors since it is highly interpretable and less influenced by outliers than the L^2 (the Euclidean distance) in high dimensions because the points become further apart in higher dimensions. The pool of nearest neighbor teams to the target team was determined and fixed to have a minimum of $T = 5$ teams, with more teams allowed if the distance between the teams is four or less (i.e. at most two players have different profiles in a single dimension space, or one player has a different profile in two dimension spaces, because of how the L^1 distance operates).

Steps 7: To compute the win rate of a team, the number of recent matches considered were $M = 15$. The team's average kills-deaths-assists ratio, defined as ((Kills + Assists/2) / Deaths), was the metric chosen to compute the learning rate of a team, because it has been a traditional skill indicator for shooter games to evaluate player skills.

Steps 8: The parameters for the optimization strategy used to map the players between the two teams were $p = 2$ and $C = 2$. Therefore, the differences in character class were given twice the weight as given to differences in other profiles. This was to ensure that recommendations were relevant, given the strong impact that character class has on optimal play style and some of the weapons available to a player. Furthermore, users may also prefer seeing a mapping of their players to players with the same character class as them, in order to take the recommendations seriously.

Steps 9: Weapon loadout recommendations were made using all three profiles and play style adjustments were recommended using weapon preference and character class.

Example Recommendation Output

Recommendations were made for a randomly selected team in the sample (Tables 2 and 3). Given the approach for player-to-player mapping, the recommendations for the top $p = 2$ players in each table were the most pertinent.

Table 2 shows the currently preferred weapon in slot 1 of 3 for each player and their corresponding recommended weapons against win rate, meaning that the team with highest win rate was chosen as the recommendation. Table 3 indicates positive and negative changes that should be made in each play style metric when recommending against learning rate, meaning that the fastest learning team was recommended. A single P (positive) or N (negative) sign indicates

Target	Current	Recommended
Player 1	The Wardcliff Coil: Rocket Launcher	Seven-Six-Five: Scout Rifle
Player 2	Last Hope: Sidearm	Jiangshi AR4: Auto Rifle
Player 3	Hoosegow: Rocket Launcher	Uriel's Gift: Auto Rifle
Player 4	Quickfang: Sword	A Sudden Death: Shotgun

Table 2: Weapon recommendations for weapon slot 1 using win rate

Target	Ability	Grenade	Melee	Super	Assists
Player 1	null	N	NN	null	N
Player 2	null	P	N	NN	NN
Player 3	P	NN	N	NN	NN
Player 4	null	NN	N	P	N

Table 3: Play style recommendations using learning rate

that the mapped player has a score greater or less than the target player's score in that dimension by an amount less than $1/4^{th}$ the standard deviation of that dimension, while two signs indicates a greater difference. "null" indicates that the score difference in that dimension was less than 1/20 of its standard deviation. Similar weapon and play style recommendations can be made for learning rate and win rate respectively. The best team based on Win Rate (the recommended column in Table 2) employed a more diverse range of specialized weapons like Scout Rifles and Shotguns which are considered long range and short range weapons respectively. Similarly, the play style recommendations for improving learning rate encouraged a more back-to-basics approach using fewer non-weapon skills, while the absolute best team tended to employ their Super abilities and other tools like grenades significantly more.

5 Conclusion and Future Work

In this paper, a multi-profile team based recommender system was proposed for team-based PvP games, and evaluated for *Destiny 2*. Players were profiled across three sets of behavioral metrics: character class, weapon preference, and play style preference. Player profiles were aggregated to create team profiles. Recommendations are made based on matching with teams with higher win ratios. The framework proposed is flexible and can be applied across multi-player competitive games, with adaptation to the specific title. For example, the specific number and types of profiles can be altered depending on the environment of the game. There is a substantial number of tunable parameters, as described in 4, which will change depending on the type and amount of data available, and the objective of the system. The framework proposed here is a step towards building player-facing systems in esports. Future work aims to implement an online version of the recommender system and open it up to the *Destiny 2* player community, for evaluation and continued development.

References

Anwar, S. M.; Shahzad, T.; Sattar, Z.; Khan, R.; and Majid, M. 2017. A game recommender system using collaborative filtering (gambit). In *Applied Sciences and Technology (IBCAST), 2017 14th International Bhurban Conference on*, 328–332. IEEE.

Aung, M.; Bonometti, V.; Drachen, A.; Cowling, P.; Kokkinakis, A. V.; Yoder, C.; and Wade, A. 2018. Predicting Skill Learning in a Large, Longitudinal MOBA Dataset. In *Proceedings of IEEE Computational Intelligence in Games*.

Block, F.; Hodge, V.; Hobson, S.; Sephton, N.; Devlin, S.; Ursu, M.; Drachen, A.; and Cowling, P. 2018. Narrative Bytes: Data-Driven Storytelling in Esports. In *ACM International Conference on Interactive Experiences for Television and Online Video*.

Cremonesi, P.; Koren, Y.; and Turrin, R. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, 39–46. ACM.

Cutler, A., and Breiman, L. 1994. Archetypal analysis. *Technometrics* 36(4):338–347.

Debeauvais, T.; Nardi, B.; Schiano, D. J.; Ducheneaut, N.; and Yee, N. 2011. If you build it they might stay: retention mechanisms in world of warcraft. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, 180–187. ACM.

Demediuk, S.; Murrin, A.; Bulger, D.; Hitchens, M.; Drachen, A.; Raffe, W. L.; and Tamassia, M. 2018. Player retention in league of legends: a study using survival analysis. In *Proceedings of the Australasian Computer Science Week Multiconference*, 43. ACM.

Drachen, A.; Sifa, R.; Bauckhage, C.; and Thurau, C. 2012a. Guns, Swords and Data: Clustering of Player Behavior in Computer Games in the Wild. In *Proc. of IEEE CIG*.

Drachen, A.; Sifa, R.; Bauckhage, C.; and Thurau, C. 2012b. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 163–170. IEEE.

Drachen, A.; Green, J.; Gray, C.; Harik, E.; Lu, P.; Sifa, R.; and Klabjan, D. 2016. Guns and guardians: Comparative cluster analysis and behavioral profiling in destiny. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, 1–8. IEEE.

Drachen, A.; Canossa, A.; and Yannakakis, G. N. 2009. Player modeling using self-organization in tomb raider: Underworld. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 1–8. IEEE.

El-Nasr, M. S.; Drachen, A.; and Canossa, A. 2013. *Game Analytics - Maximizing the Value of Player Data*. Springer.

Goldstone, R. L., and Lupyan, G. 2016. Discovering psychological principles by mining naturally occurring data sets. *Topics in cognitive science* 8(3):548–568.

Humphries, M. D.; Khamassi, M.; and Gurney, K. 2012. Dopaminergic control of the exploration-exploitation trade-off via the basal ganglia. *Frontiers in neuroscience* 6:9.

Newell, K., and McDonald, P. 1992. Searching for solutions to the coordination function: Learning as exploratory behavior.

Normoyle, A., and Jensen, S. T. 2015. Bayesian clustering of player styles for multiplayer games. *Proc. AAAI AIIDE*.

Ranganathan, R., and Newell, K. 2010. Motor learning through induced variability at the task goal and execution redundancy levels. *Journal of motor behavior* 42(5):307–316.

Schmidt, R. A. 1975. A schema theory of discrete motor skill learning. *Psychological review* 82(4):225.

Sifa, R.; A., D.; and Bauckhauge, C. 2018. Proling in Games: Understanding Behavior from Telemetry. In K., L.; Sukthankar, G.; and Wigand, R. T., eds., *Social Interactions in Virtual Worlds*. Cambridge University Press. 337–374.

Sifa, R.; Bauckhage, C.; and Drachen, A. 2014. Archetypal game recommender systems. In *LWA*, 45–56.

Sifa, R.; Drachen, A.; Bauckhage, C.; Thurau, C.; and Canossa, A. 2013. Behavior evolution in tomb raider underworld. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, 1–8. IEEE.

Sifa, R.; Pawlakos, E.; Zhai, K.; Haran, S.; Jha, R.; Klabjan, D.; and Drachen, A. 2018. Controlling the crucible: A novel pvp recommender systems framework for destiny.

Stafford, T., and Dewar, M. 2014. Tracing the trajectory of skill learning with a very large sample of online game players. *Psychological science* 25(2):511–518.

Stafford, T., and Haasnoot, E. 2017. Testing sleep consolidation in skill learning: A field study using an online game. *Topics in cognitive science* 9(2):485–496.

Stafford, T.; Thirkettle, M.; Walton, T.; Vautrelle, N.; Hetherington, L.; Port, M.; Gurney, K.; and Redgrave, P. 2012. A novel task for the investigation of action acquisition. *PloS one* 7(6):e37749.

Stafford, T.; Devlin, S.; Sifa, R.; and Drachen, A. 2017. Exploration and skill acquisition in a major online game. In *The 39th Annual Meeting of the Cognitive Science Society (CogSci)*. York.

Statista. 2017. Mmo gaming - statistics & facts. https://www.statista.com/topics/2290/mmo-gaming/.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tamassia, M.; Raffe, W.; Sifa, R.; Drachen, A.; Zambetta, F.; and Hitchens, M. 2016. Predicting player churn in destiny: A hidden markov models approach to predicting player departure in a major online game. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, 1–8. IEEE.

Van Rossum, J. H. 1990. Schmidt's schema theory: The empirical base of the variability of practice hypothesis: A critical analysis. *Human movement science* 9(3-5):387–435.

Weber, B. 2015. Building a recommender system for everquest landmark's marketplace.

Yannakakis, G. N. 2012. Game ai revisited. In *Proc. of the 9th conference on Computing Frontiers*, 285–292. ACM.