

Machine Learning – Clustering Team Roles

1. How do you frame your main question as a machine learning problem? Is it a supervised or unsupervised problem? If it is supervised, is it a regression or a classification?

A: This is an unsupervised problem that determines whether or not individual players (aka summoners) in professional League of Legends eSports leagues can be clustered into five different groups corresponding to their declared team roles in a match based on their performance data throughout a split season.

2. What are the main features (also called independent variables or predictors) that you'll use?

A: There are 17 features being used for this problem. They are: - **kills**: The number of times a player dealt a killing blow to individual enemy champions. - **assists**: The number of times a player has damaged or hindered individual enemy champions within 10 seconds of a teammate killing said enemies. - **magic damage dealt**: The cumulative amount of magic damage dealt to enemy champions, monsters, and structures (e.g., towers, inhibitors, the nexus). - **physical damage dealt**: The cumulative amount of physical damage dealt to enemy champions, monsters, and structures (e.g., towers, inhibitors, the nexus). - **magic damage dealt to champions**: The cumulative amount of magic damage dealt to enemy champions only. - **physical damage dealt to champions**: The cumulative amount of physical damage dealt to enemy champions only. - **total heal**: The total amount of health points the player has regained. - **total units healed**: The total number of entities (teammates, self, allied structures, allied minions) a player healed. - **damage self mitigated**: The total amount of health points that were not lost from sources of damage due to various sources such as shields, armor, and magic resistance. - **total damage taken**: The total amount of damage a player took from various sources, such as enemy champions/minions/towers, neutral monsters. - **neutral minions killed**: The number of neutral monsters killed by a player. Neutral monsters include those in the jungle, the dragons that spawn throughout a match, the Rift Herald, and the Baron Nashor. - **time crowd controlling others**: as explained from source: > timeCCingOthers is the new stat (not yet deployed) which offer a more precise "score" of CC, as each CC has a value, hard CC value is the duration the CC as been applied, soft CC only a 1/2 of the duration and slow for 1/6 of the duration. - **total time crowd control dealt**: as explained from source: > totalTimeCrowdControlDealt is just the sum of all CC applied. - **champ level**: The (experience) level of a player at the end of a match. - **vision wards bought in game**: The number of wards (i.e. surveillance items) a player purchased from the shop at the team's base. - **wards placed**: The total number of wards a player placed in the Summoner's Rift (i.e. the arena). - **wards killed**: The total number of enemy wards placed in the Summoner's Rift that a player destroyed.

3. Which machine learning technique will you use?

A: **k-means clustering** will be used for first testing its clustering performance of a dataset.

The k-means object generated will have a variable named "centers", which will be used as the input (training) model for when we use the **k-nearest-neighbor** technique (setting $k = 1$) for other datasets.

4. How will you evaluate the success of your machine learning technique? What metric will you use?

A: Since I manually added team roles to the datasets being used for this problem, I will output a table of said team roles against the clustering results. This table will represent a confusion matrix. From this matrix, we can calculate an overall accuracy rate for each team role ($[\text{true positives} + \text{true negatives}] / N$). This will be the metric used for evaluating the success of the machine learning technique.