1. For the dataset it's quite easy since you access all the info from RIOT gathers from League of legends servers through these three APIs(https://developer.riotgames.com/):

- /lol/match/v5/matches/by-puuid/{puuid}/ids
- /lol/match/v5/matches/{matchId}
- /lol/summoner/v4/summoners/by-account/{encryptedAccountId}

    a. The first one contains all the matches played in LOL, the second contains all the matches ids of a specific player's puuid and the last one contains all the information of an account.

    b. The reason I chose this Dataset is simply because I doubt I can find a better one since Riot themselves made it. On the other hand, I read somewhere that I can only make 100 requests every 2 minutes and I am not sure if that will be problematic for me in the future.

2. a- The APIs respond with a JSON serialized HTTPS response, to help me change the date to something better to understand I will be using the library Riot-Watcher. I will still need to preprocess the data further by taking the subset I need off it: Champions played and their win rates and mastery points, user's win rate in different modes.

   b- The model I decided to use is KNN and here is how it will work.
   First of all, I will take n number of players and for each player I will define n(i) number of most played champions (I say n(i) because this number will differ from player to player),then I will be creating a matrix of size nxn where each entry will be the number of common champions played by the player corresponding to that row and the player corresponding to that column. Then I will use NetworkX to graph my data based on that matrix. For the distance, I will use the Dijkstra's Algorithm. At the end, I 'll need to find the best k for the best results. In those k players, I will rank all the champions of their most played champions based on the total mastery points of a champion of all the players (think of mastery points (MP) as the time spent playing a champion), then remove the champions that the user already plays and VOILA. To better understand here is an example:
   Let's imagine after the KNN model, I got left with k=3 players:

Champions	Player 1	Player 2	Player 3	Total MP
Champion A	80k MP	2k MP	30k MP	112k MP
Champion B	30k MP	0	40k MP	70k MP
Champion C	0	140k MP	500k MP	640k MP

   In this case, champion C has the most total MP which means he will be recommended (of course in this simple example we chose to only recommend one champion it could be more).

c- To test my model I will be taking a list of players and find the most played champions for those players, and from these I will remove one of them and then test my model, for each player the sub-accuracy will be the fraction of the number of recommendations that are actually played by the player on the player's total number of recommendations. The total accuracy will be the mean of the sub-accuracies.
I will also use a confusion matrix of this form:

	Most played Champions that got removed from the test set players	Champions that are not that are not even mostly played by the players
Champions recommended	Number of champions recommended that got removed	Number of champions recommended that are not even mostly played by the players
Champions recommended	Number of champions recommended that are not even mostly played by the players	Number of champions not recommended that are not even mostly played by the players

For my expectations, I have no idea what to expect but one thing I am worrying about is the size of the dataset which might be very large and when If I try to make it smaller the accuracy will be affected greatly.

3. Application:
    - Input: the input will be generally the user's account name since I can access all the information I need using the APIs. I say generally because I might add some features in the future so that the output is more adapted to the user. For example, I could add recommendations for wanted role and the reason is in League of Legends, there are five roles, and most players play the same role, so if a player wants to change and uses this system the way I described it, he will most likely get recommended champions in the same role he is already playing which is not what he wants.
    - Output: a list of champions and their pictures.