At first, we intended to design an algorithm that could give match result prediction before the game starts. That is, given the dual players' information and heroes information, we can give a prediction with satisfying accuracy. However, we failed to do so for multiple reasons.

One main reason is that some players refuse to give their information to public. Thus, in order to analyze the relevance between team winning rate and players winning rate, we have to select matches where all 10 players' information are open to public. This makes data collection really complex yet trivial. We then consider using professional matches data, as most professional players are willing to open their information to public, but the prediction accuracy based on these data is not so promising.

As a result, we decided to build a model that could give real time winning rate analysis.

In the jupyter-notebook file, we compared several algorithms for regression logistic regression, MLP regression, Decision Tree Regression and SVM Regression. (there are also some other ways of regression, yet they are not suitable for this task, the reason is similar to that why we not use some mentioned regression algorithms but only select one)

As can be seen in this file, three regression models have the same classification accuracy (in fact most models have the same accuracy), and SVM regression has a lower accuracy. If our goal is simply classification, we have many choices. However, our target is to give a winning rate prediction, thus, not only do we need to select the algorithm with highest classification accuracy, but we also need to analyze which one will give a reasonable regression result.

Now that we need to analyze the three models with same high accuracy logistic regression, MLP regression and decision tree. In decision tree model, the deeper the tree is, the more possible values we have in the result. However, as the depth increases, decision tree model becomes more likely to over fit the data. And when the tree depth is small (as given in the file 3), the model can give a fair regression of the data, but the result is not continuous. Thus, this model is not suitable for this task.

In MLP regression model, we do not want the model to be too complex, as we need to build multiple models for different time steps. And we tried many structures that are not too complex and deep. In this file, we illustrated a shallow neural network with 2 layers. The outcome is not exact what we desire. In our intuition, when one side has advantage very large, the winning rate of that side should reach 1. However, in this model, the winning rate of neither side will go to 1. It may have a better fit of the data if we expand the size of the network, but that would cost too much computation resources.

Compared with all three models above, logistic regression provides a much more reasonable result. We can directly use its result as the winning rate prediction. Plus the model size is very small, thus we choose this model as our algorithm.