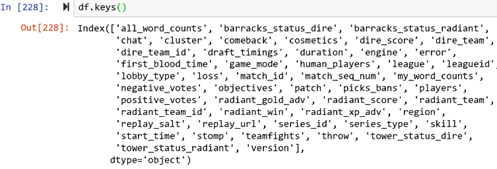
* 1. Introduction

DOTA 2 is a multiplayer online battle arena(MOBA) game developed by Valve. In this type of games, players form two random teams of 5 and play against the other team aiming to destroy the main turret of the opposing team. Each player can select a champion from a champion pool of over 100 champions, where each champion has unique skills and initial capabilities. In game, gold can be acquired by killing minions or killing champions on the opposing team. Gold acquired can be used to purchase items that improves the champions' capabilities substantially. Different champions are suited for different items in game, so the choosing of items purchased also has a significant impact on the result of the game.

In DOTA 2, the two teams—known as the Radiant and Dire—occupy fortified bases in opposite corners of the map, which is divided in half by a crossable river and connected by three paths, which are referred to as "lanes". The lanes are guarded by defensive towers that attack any opposing unit who gets within its firing range. Gold is only awarded to the player when the enemy target is killed by damage from the player, not the turrets or minions. This is referred to as ‘Last hit’ in games.

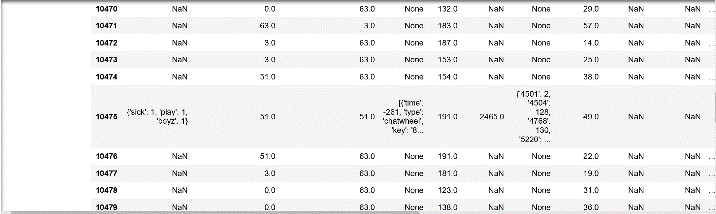
* 1. Data collection

The dataset we are using are recent DOTA 2 match statistics. We used public apis at opendota.com to pull down the data of roughly 54000 recent DOTA 2 matches. The dataset included the in-game statistics of all ten players, the duration of the game, the meta of the game and so on. A list of all the keys are as follows:

An example entry looks like this:

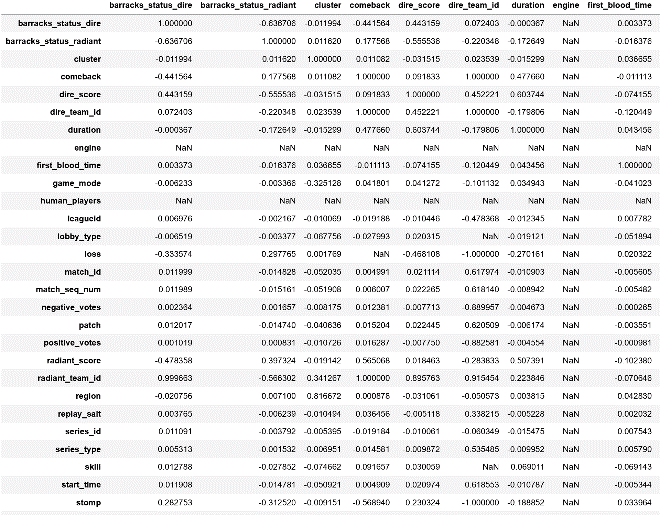


After converting this into a python dataframe, there are some null values. In order to smooth the dataset for classification, null values are filled with the mean of all values in that column. This is intended to avoid destroying the effectiveness of linear models.



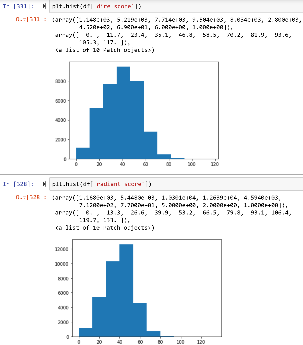
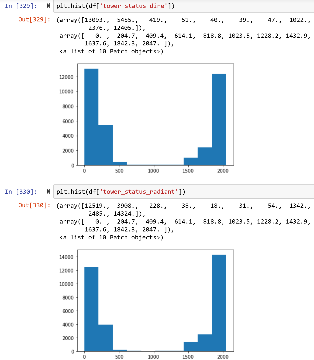
* 1. Exploratory analysis

We first calculated the correlation matrix of the columns in the dataframe. The results are shown below:

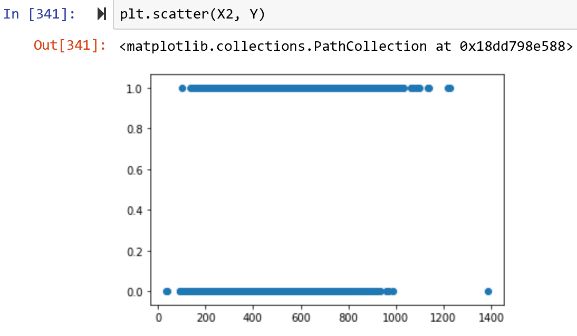


Since most correlation values are close to 0, we can safely assume that these features are relatively independent from one another, thus we shouldn’t worry too much about the problem of double counting while constructing feature vectors or doing naïve bayes.

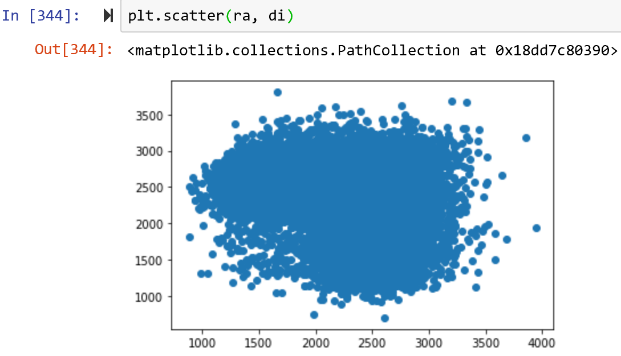
We then plotted histograms of “radiant\_score”, “dire\_score”, “tower\_status\_radiant” and “tower\_status\_dire”. It seems that the distribution of radiant and dire team statistics is very similar. This means that the game is well balanced for different factions in the game at this point. Whether the player is on the dire team or the radiant team shouldn’t be a good feature for the model. Figures are shown below.



We plotted the gold acquired by each player with respect to the result of the game. It seems that there is a clear difference in the average of gold obtained per minute between games won and games lost. This means that gold per minute can be a good feature to be incorporated into the design of our model.



We also plotted a scatterplot of total gold of the radiant team and the dire team. The winning team tends to have higher total gold, so the datapoints right of the diagonal is more likely to be a radiant victory, the datapoints left of the diagonal is likely to be a dire victory. A lot of datapoints are located near the diagonal, making the classification task difficult.



2.1 Predictive Task

We intend to use the data from these matches to train a real-time predictor of the probability of winning for both sides in game. An ongoing match should have data to all the features in our dataframe accessible in real time. Therefore, we can build a model that predicts the result of the game with it. We also want to predict the probability of winning with these real time data.

2.2 Baseline

Since this is a binary classification problem, we set the baseline accuracy to be 0.5.

2.3 modeling methodologies

Since our exploratory analysis show that the features in the dataframe are relatively independent, we intend to do a naïve bayes on the data ???

In the scatterplot we made on total gold, classification is hard around the diagonal. Support vector machines are useful for handling situations like this, as SVM focuses on the datapoints close to the line that separates the two categories.

In our exploratory data analysis, we identified gold per minute as a good feature to use. Therefore, we are interested in building a feature vector that consists of the gold per minute data of all the 10 players in the match. The first 5 players are on the radiant team and the latter 5 are on the dire team. We can then run a logistic regression with what we have.

Other features we considered are one-hot encodings of champion selection and items purchased. As shown in the introductory part, champions and items strongly influences the result of the match, so including these as features should be beneficial for our model.

3.1