Tyler Goodwyn

UCLA extension : Machine learning with Python

Final project

League of Legends game analysis

Project overview

League of Legends is a video game with more than 80 million players worldwide . The format of the game consists of matches that pit 2 teams of five players against one another. A match unfolds over the course of about 45 minutes to an hour, and culminates with one team winning by destroying the base of the other team.

Figure 1 image of base being destroyed

Although a game can take as long as an hour and a half , the first 10 minutes are critical for determining how the remainder of the game will play out. Available on kaggle is a data set consisting of data taken from the first 10 minutes of 10,000 games between highly ranked players .

Broadly speaking , the goal of this project will be to create a report from the data data that has the potential to point to strategies for making decisions at the start of the game based off relationships between sets of features and their influence on the outcome of the game. These relationships will be shown by testing various linear regression models on the data.

They will principally be two things being measured for success . The first will be a comparison between various linear regression models in their efficacy on being able to predict successfully what the outcome was of a game from the features given. The purpose of this in large part is just for me to learn about the differences between various linear regression models . The second measurement has a slightly wider appeal , which is that it will be concerned with measuring the efficacy of different combinations of features . If a certain combination of features seems to produce in the model a higher score of successful predictions then that could mean those features deserve to be prioritized in a strategy for how to play the game at the beginning . We will see .

Data Set

The data comes from https://www.kaggle.com/bobbyscience/league-of-legends-diamond-ranked-games-10-min . This in turn comes from an API published by Riot Games (the creator of League of Legends). If I have time and it seems like it could be useful then I will explore this API more deeply ; however , I believe that there are plenty of insights to be gleaned from the data set as it exists currently on kaggle .

Looking at notebooks that have already begun to play around with the data , it appears that there could definitely be some promising correlations amongst the various features :

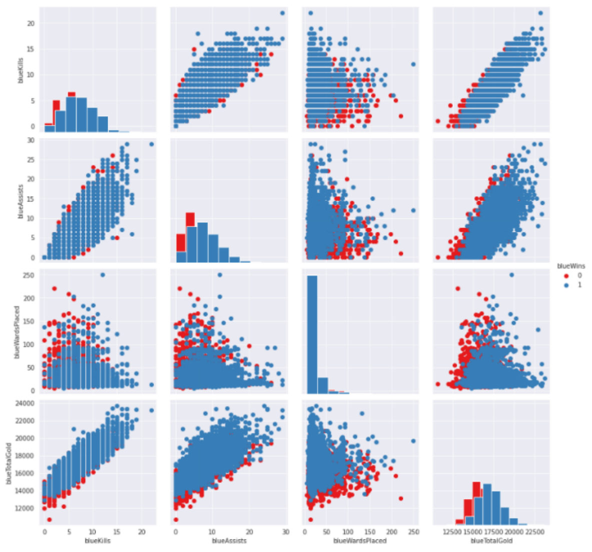


Figure 2 https://www.kaggle.com/xiyuewang/lol-how-to-win

I feel like this is also a good data set for a beginner project since it has a healthy mix of discrete and continuous variables

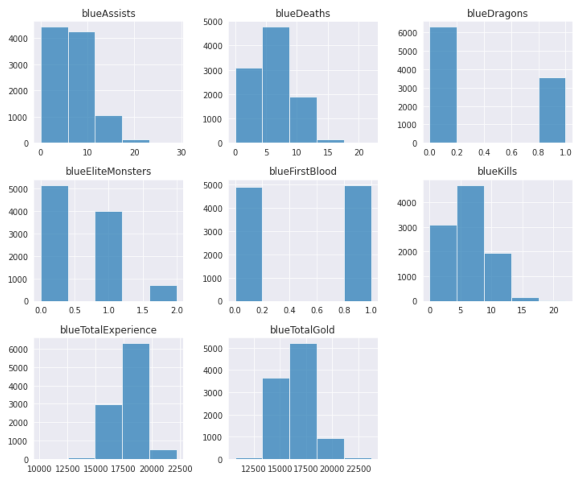


Figure 3 https://www.kaggle.com/xiyuewang/lol-how-to-win

Implementation

(Code for generating graphics below is in companion notebook)

Data preprocessing

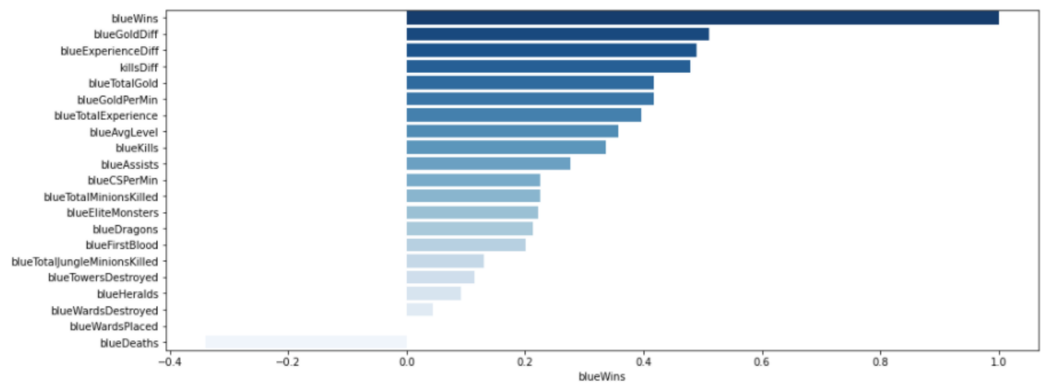
There are a couple things that we can do without even having to begin to visualize the data in order to prepare it for everything that comes after . For example , we can begin by separating the target column - which is the column with the boolean values for whether blue won or not - from the rest of the columns - which will comprise the features . Additionally , there is one column that we can be certain will have no influence on our machine learning models , and that is the value of game ID . This was an arbitrary value assigned to each instance during the data harvesting phase and has nothing to do with what occurred during each actual game , which is all that we are concerned with . Furthermore , data contains pairs of features for identical statistics from each game taken for both the red and blue sides . As many of these are mirrors or inverses of one another (for example blue gold difference and red gold difference are complements of one another), and others are corollaries of one another (for example blue deaths is the same as red kills), it is safe to assume that these duplicates will not improve the performance of the model . therefore , we go ahead and remove all of the red features .

Lastly , I noticed one odd thing about this data set is that it includes the differences between the two sides for several important features , EG gold and experience , but it does not include the difference in kills . From having played the game, it seems obvious to me that this is an important feature so I went ahead and added it , by calculating the blue total kills minus the blue total deaths , to the data set .

Exploratory data analysis

Ranking features

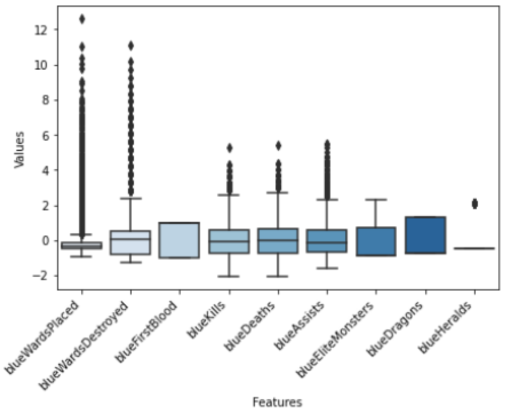
First thing I wanted to look at was simply how correlated are the features with the blue team winning . This is really a crude way of figuring out what elements of the game one should focus on during the 1st 10 minutes . After finishing the project though I realize this is really the best information that you can get from this data in terms of coming up with a strategy , and the results would not exactly be shocking to anyone who has played the game . The biggest thing I noticed was that the gold difference actually seemed to be a bigger determinant than the kills difference , which I was somewhat surprised by . So if there's one take away : <u>it's that having more gold by the end of the first 10 minutes is more correlated with winning than any of the other features</u> , although I'm not sure being about 54% correlated would be considered highly correlated .

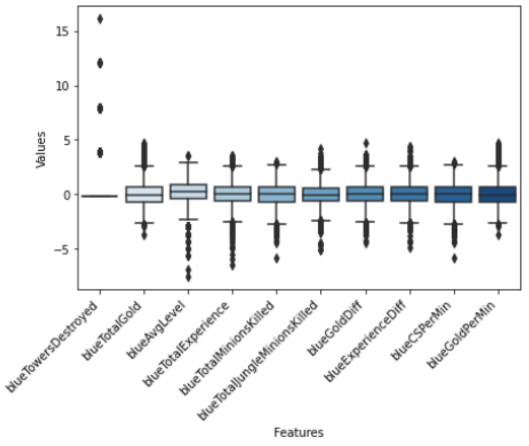


Examining feature distribution

Next I wanted to look at the distributions of the features themselves . This is not directly related to winning , but it is interesting . I can see some things that make sense to me : for example , the average level has a lot of outliers beneath the minimum . This makes sense because anyone who has watched highly ranked players play the game know that each team member assumes a certain role , and there is one team member whose role is just to support the other team members (actually called support), so this team member would regularly have a much lower level than the other 4 team members .

Also, of note is just how high the standard deviation is for ward placements . This was perhaps more interesting to me than any other thing I found while doing this project . I have not played a lot of the game , but I would have assumed that ward placement was highly correlated with winning , and the more wards place the better . <u>But the data does not substantiate that at all</u> . Ward placement is not something that I really understand but I know that it is a key factor for higher level play , and I now believe that doing similar project but that focused exclusively on ward data could yield a lot of interesting insights into team strategy - however , that data is not readily available and collecting it would require a deeper understanding of the game and of Riot's API.

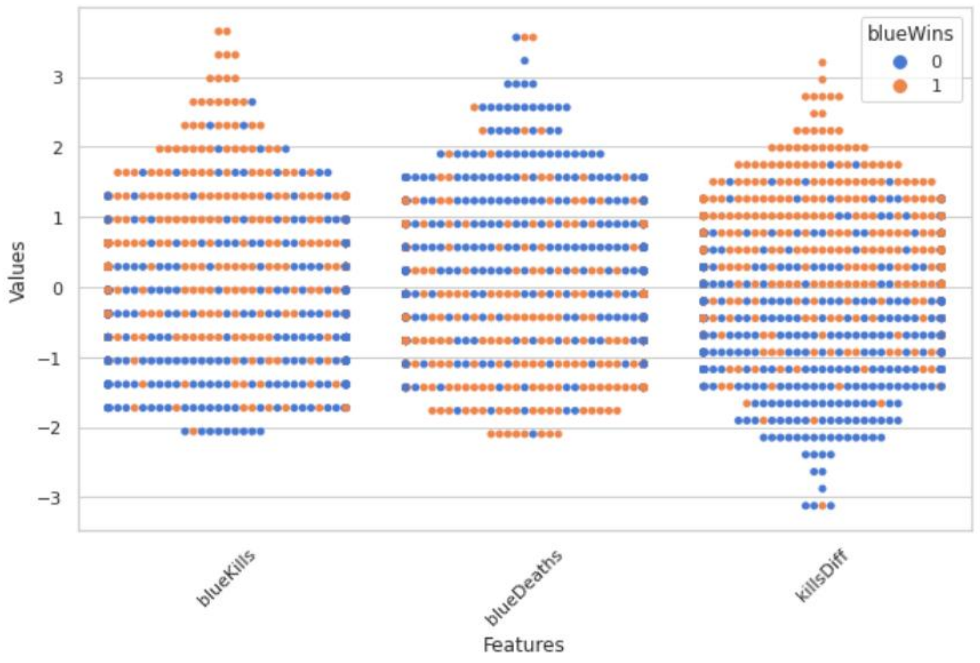




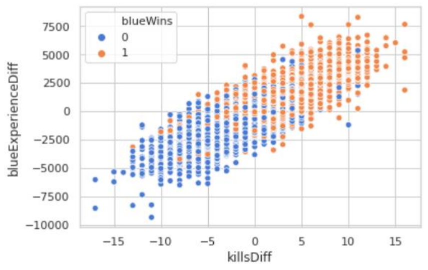
Choosing 2 features vs PCA

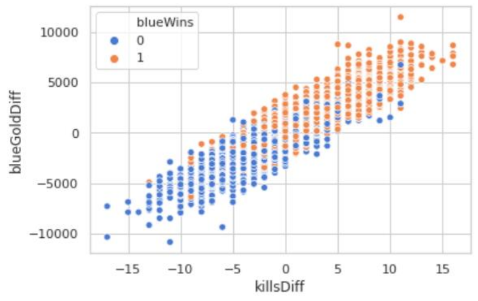
The last thing I wanted to do was just to perform a sort of eyeball test on what I thought the two most important features were , check the correlation of these 2 features combined with winning and then compare that result to a PCA .

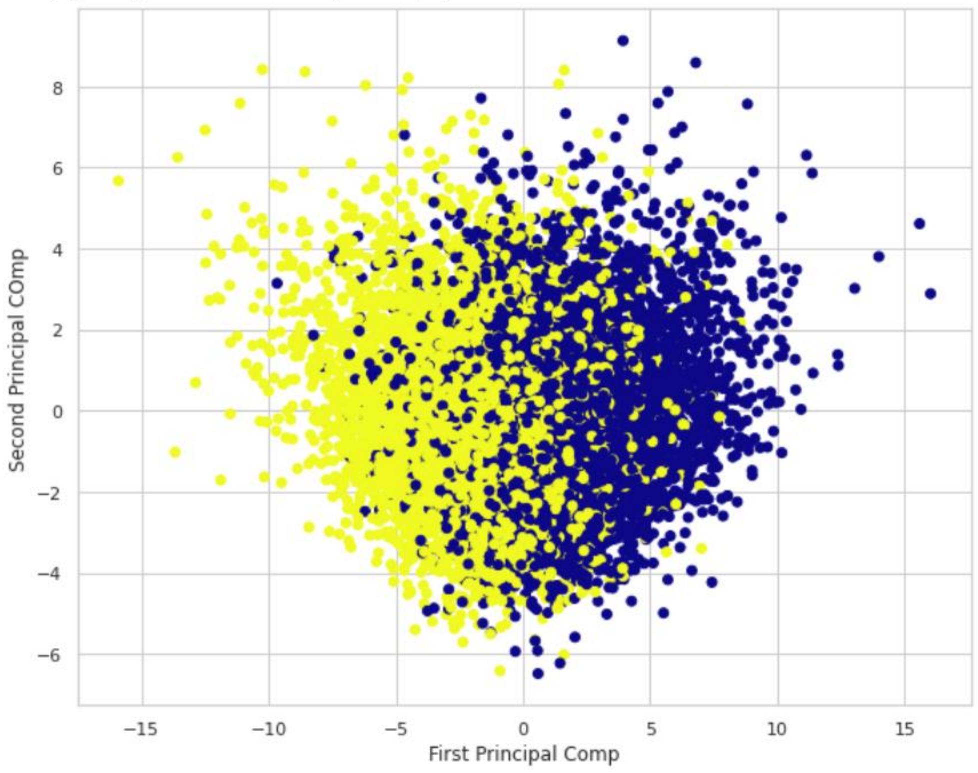
Here you can see a swarm plot comparing the correlation between the kill difference between the two teams and winning versus just the gross deaths or kills of the blue team . It is clear that the kill difference is much more strongly correlated with winning then the gross values alone .



Because it seems like the kill difference , the gold difference , and the experience difference are the three most important features correlated with winning , I went ahead and did a sort of rudimentary PCA combining some of those features to check out their combinations correlated with winning . as you can see , even just using two of those features you can establish a pretty strong correlation between the values of those features and whether or not the blue team won . Also included is a PCA that similarly divides the label into two sets based off a linear combination of weighted feature values.







Machine Learning

For the actual machine learning part of the project , I just wanted to do a logistic regression analysis using some of the models that we had learned in class . I chose to do a KNN classifier , a regular logistic regression solver , a decision tree , and a random forest classifier . The results pretty much lined up with what they had for our projects in class , with logistic regression in random forests having the strongest accuracy scores .

```
+-----------+----------+---------+-----------+---------+
| Algorithm | Accuracy |  Recall | Precision | F-Score |
+-----------+----------+---------+-----------+---------+
|    KNN    | 0.71862  | 0.74492 |  0.70617  | 0.72502 |
|  Decision | 0.69383  | 0.67886 |  0.69801  |  0.6883 |
|     LR    | 0.74494  | 0.74085 |   0.7454  | 0.74312 |
|  R Forest | 0.73229  | 0.73374 |  0.73003  | 0.73188 |
+-----------+----------+---------+-----------+---------+
```

Conclusion

Interestingly , the EDA for this project took way more time than the actual machine learning , and I realized shortly after I began that the EDA also would bear way more information on learning game strategy than the outcomes of the models .

The outcomes of the models though do tell you some information - essentially , the main question that they answer is : just how confident can you be at the end of 10 minutes do you know what the final outcome of the game will be ? I find this actually to be a really useful and interesting question even if not for purposes of developing strategy . the reason I find this interesting is two fold : I like to watch professional League of Legends matches , and a common theme , like with any competition , is the question of “just how likely is a comeback going to be given the current state of the game ? “ With the results of this model , I now have a sense that about 75% of the games are decided by what happens in the first 10 minutes . For 25% though , there is still a lot of game to be played after that .

Which leads me to the second reason why I find this useful : the few times that I have attempted to play online , I found that teammates give up very quickly based off a few things going wrong early in the game . Now when this happens , I can say “there's no need to quit , 25% of games are not decided in the first 10 minutes“.