Twitter Sentiment Analysis on NBA Game Predictions

Abhishek Dendukuri, Arya Tayebi, Farris Atif, Nasser Al-Rayes, Neeraj Joshi, Zixiao Chen

I. Introduction

For our final project, we decided to add advanced python techniques we learned in homeworks to an NBA game prediction model. With the influx of sports betting in the US, there has been a rise of analytics in predicting outcomes of games for financial reward. However, one of the main issues is the turnaround time from data aggregation and model prediction to putting down a bet on a team. In order to take full advantage of the odds of a team winning at a certain moment in time, data aggregation and model prediction needs to be not only accurate but quick.

In this project, we took NBA odds data along with twitter sentiment scores prior to a game to predict an outcome of the event. From an advanced python perspective, we used cython in the data aggregation phase to speed up the overall processes of receiving an outcome.

II. Datasets

We had two data sources. The first data source was from SportData.io, a data company that contains odds, results, statistics, and more on multiple sports. Specifically for NBA data, we needed to use two endpoints of their api, one for odds data and one for results data. Due to their api documentation, the easiest and most efficient way to pull data was through node.js. We called the API every gameday from early March and grabbed over 200 games. The results and odds data of each game were joined in a pandas dataframe via the id of each game. The id was specified as “home team + away team + game date”.

The second data source was from the twitter developer API. We pulled NBA game related tweets from 5 hours prior to each game, to the end of the game. We specify this time frame because we are mainly interested in what people think right before and during the game, since that’s when the odds are determined. This data is collected for further usage in sentiment analysis, which will be elaborated in the next section.

From the two aforementioned data sources, we collected odds data, tweets for sentiment analysis, and game results data. We merged three datasets together and our final dataset is around 200 rows, with each row representing one NBA game.

III. Methodology

A. Feature Extraction

We used both Twitter sentiment data and odds data as input features to feed into our model.

We first conducted sentiment analysis on the tweets we collected. We cleaned collected tweets by removing links and punctuation. We also removed retweets to ensure that the same tweet does not have duplications to outweigh other tweets.

After data cleaning, we adopted SentimentIntensityAnalyzer to generate sentiment scores for each team. Sentiments are roughly divided into three buckets: positive, negative, and neutral. We generated three different sentiment scores for both home team and away team (Figure 1.). For example, in the first row of Figure 1 we can see the sentiment scores of the game between Boston Celtics and Milwaukee Bucks. Homepositive = 38.56 represents that out of all the tweets related to the home team (Boston Celtics), 38.56% tweets have a positive sentiment toward the home team winning. Whereas awayPositive = 37.19 means that out of all the tweets related to the away team (Milwaukee Bucks), 37.19% tweets have positive attitudes of the away team winning. We structured the sentiment score this way because we want to ensure the score is not biased because of an imbalance in the number of tweets per team. All six scores (homePositive, homeNegative, homeNeutral, awayPositive, awayNegative and awayNeutral) are included in our final model as input features.

	gameID	homeTeam	awayTeam	homePositive	homeNegative	homeNeutral	awayPositive	awayNegative	awayNeutral	total_tweets_analyzed
0	BOSMIL512022	Boston Celtics	Milwaukee Bucks	38.565022	6.278027	55.156951	37.198068	2.898551	59.903382	430
0	BOSMIL532022	Boston Celtics	Milwaukee Bucks	54.166667	5.833333	40.000000	55.421687	3.212851	41.365462	489
0	MEMGS512022	Memphis Grizzlies	Golden State Warriors	48.214286	3.571429	48.214286	53.080569	1.895735	45.023697	435
0	MEMGS532022	Memphis Grizzlies	Golden State Warriors	37.142857	10.952381	51.904762	46.031746	8.730159	45.238095	462
0	MIAPHI522022	Miami Heat	Philadelphia 76ers	54.393305	7.949791	37.656904	50.490196	6.862745	42.647059	443
0	MIAPHI542022	Miami Heat	Philadelphia 76ers	39.700375	13.483146	46.816479	31.981982	14.864865	53.153153	489
0	PHODAL522022	Phoenix Suns	Dallas Mavericks	42.748092	11.068702	46.183206	56.569343	4.744526	38.686131	536
0	PHODAL542022	Phoenix Suns	Dallas Mavericks	49.765258	9.859155	40.375587	47.027027	7.567568	45.405405	398

Figure 1. Sentiment analysis results

Besides sentiment scores, we also extracted features from the odds dataset. We included odd, spread, and spreadOdds for both home team and away team separately. Spread and SpreadOdds refer to how many points a team will win/lose by and the odds of that happening, respectively. Odds related features are included in the final model because it is a good proxy of people's sentiments towards a game since it's financial rewards related.

B. Modelling

We first started with a baseline model that only includes odds related features as inputs. We chose logistic regression since we have a relatively small dataset. After building the baseline model, we improved the model by adding sentiment scores as input features, along with odds features. We believe that both sentiment scores and odds should effectively reflect how people think about a certain game, which would help the prediction. The model's accuracy improved by 5% after adding sentiment scores.

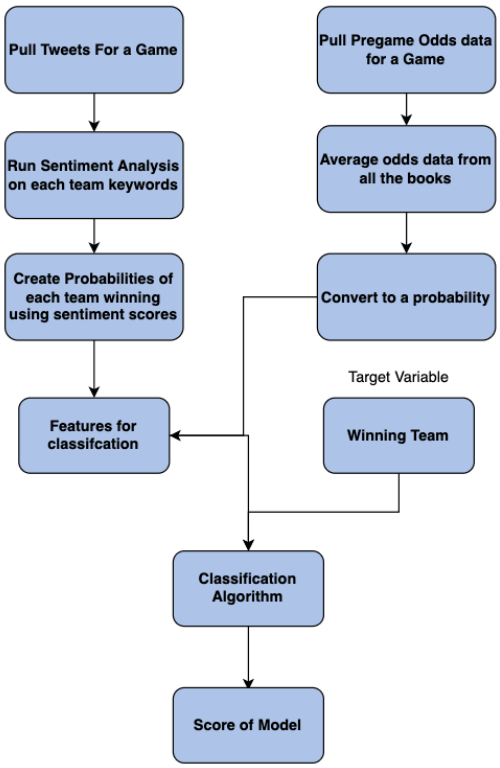

Figure 2. Modeling

IV. Results

Our baseline model obtained a decent accuracy of 71%. The precisions are 0.89 and 0.58 for class 1 and class 0. The recall are 0.62 and 0.88 for class 1 and class 0 (Figure 3).

After adding sentiment featuresour mode accuracy increased by 5%. The precision and recall also increased in the improved model (Figure 4).

	Precision	Recall
Class 1	0.89	0.62
Class 0	0.58	0.88

Figure 3. Precision and recall of baseline model

	Precision	Recall
Class 1	0.90	0.69
Class 0	0.64	0.88

Figure 4. Precision and recall of improved model

V. Optimization Technique

We adopted Cython as an optimization technique for our code. Cython is a superset of the Python language that additionally supports calling C functions and declaring C types on variables and class attributes. It is an easy way to incorporate compiled C/C++ code in Python programs, which provides OO functional and dynamic programming. We chose Cython for the following three reasons: a.) it speeds up critical parts of Python code using static typing, b.) it accelerates data exchange between array-like structures, c.) it integrates Python with existing C/C++ applications and liberties. Whether there are significant improvement by Cython is largely dependent on the program itself. Usually if it's a numerical program, the improvement is small. Whereas for programs with loops, the improvement is more noticeable. In our case, we did not have many loops in our program, hence the improvement is rather small.

We also considered other optimization techniques we've learned in the class such as concurrency and parallelism. However, since our dataset is relatively small and the nature of our problem setting is rather straightforward, these optimization techniques are not necessary and may not provide significant improvement to our program.

VI. Conclusion

In this project, we utilized twitter developer API and sportsdata.io API to extract NBA game odds data, sentiment data and results data. We also conducted sentiment analysis based on the raw twitter data we collected. We leveraged logistic regression to predict the game result for

each team, using both tweets sentiment scores and odds data. Lastly, we adopted Cython to improve our program.

Note: Github code at https://github.com/aryatayebi/AdvPythonProject

Please see README file to run notebook