Using Twitter and Elo to predict football matches and bet on them.

Mignard Balthazar
Hoxha Maximilien
Lafon Mathis
Duval Denis

https://github.com/FLAGGED43497714/MALISPROJECT1

Abstract: The goal of this project was to be able to predict the outcome of football matches thanks to data collected on twitter, specifically user tweets related to a given game. Football games from the top 5 European leagues serve as the basis for this study although other ligues were included as well. Numerous obstacles related to language processing of the tweets, tweet collection by keywords, and unbiasedness were overcome in order to classify tweets by polarity. An average sentiment was calculated for a given team's odds of winning a match which were paired with its Elo rating (accessible via online tables) to obtain meaningful predictions. These were the input features of our machine learning algorithms. We tested a wide array of methods to see which would best fit our data before settling on what we called an assembly of SGDs. The term assembly refers to the voting process that was undertaken to counteract random predictions and outliers that were bound to arise in a small dataset. The final part of the study consisted in maximizing betting gains through the use of our trained algorithms. Utility theory played an important role in the rather promising results that our methods yielded.

Part I : Gathering Tweets.

1.1) Methodology

Firstly, all of the tweets used to predict the outcome of games were collected prior to the start of the match so as to avoid biased tweets dating from during or after the game. For variety and universality, the idea was to take an interest in leagues from different countries; the top leagues such as the BPL (British Premier League), the Bundesliga, La Liga, Serie A, and Ligue 1 were our primary case studies. However, smaller leagues and national cups were included as well. The idea was to collect roughly 2000 tweets for every game; 1000 tweets would be allocated to the home team and 1000 to the away team. These files with heaps of tweets acquired thanks to the Twitter API had taken the team's name or nickname in order to compute the queries. As one can imagine, the sheer number of games during the weekend made it prohibitive to manually launch the tweet gathering process for every game. There was thus an optimizing process that was undertaken to automate this step. Namely, a python script verifying the date and time of football matches allowed us to dedicate more time to other tasks.

1.2) Problems faced and proposed solutions.

The twitter API does not give free tier users access to tweets that are more than a week old. This meant that in order to maximize our data collection, we needed to branch out to as many leagues as possible for every week. As mentioned above we collected data for teams all over Europe and in various divisions, from the most prestigious to the more modest ones, and by taking cup competitions into account. Another simple problem was assessing tweet relevancy; oftentimes famous teams' names would be used only to gain visibility, not because of its footballistic content. Through trial and error, a list of "banned words" that were popping up repeatedly in irrelevant tweets was put in place so as to eliminate them in the future. We also cracked down on spam and duplicate tweets. There were also a couple of issues concerning the actual names of the teams themselves; some were too long and complicated to be spelled out by the average user ; some teams' names were simply the city in which the team was based which led to some tweets dealing with daily life in the city. To circumvent the problem we could use recognized acronyms such as ASSE for Saint-Étienne (which otherwise is a city in France) or the team's twitter name (ex : @kleeblattfuerth for the SpVgg Greuther Fürth). This was not without its problems, however, since journalists often write the full name of the team correctly, while "random and ordinary" people may not be as conscientious. This could lead to discrepancies in the representation of the population's sentiment towards their team.

Part II : Pre-processing

2.1) Methodology

As is always the case for scientific experiments, the data needed to be representative of the overall sentiment regarding the games. Links and # had to be taken out of the tweets so as not to interfere with the sentiment analysis. Then came the translating of the tweets into English for all the tweets originally in foreign languages. The next step consisted in analyzing the sentiment associated with each tweet. We used TextBlob, a Python library able to process textual data. It gave us a polarity score within the range [-1.0, 1.0] where -1.0 is a very negative tweet and 1.0 is very positive. An average for each team's tweets would then be calculated and would be considered representative of twitter's beliefs in each team. The teams' Elo ratings were not calculated by our group: online registered values can be found on the website clubelo.com. Moreover, the betting odds for every game were taken from official online bookmakers.

2.2) The Elo rating system.

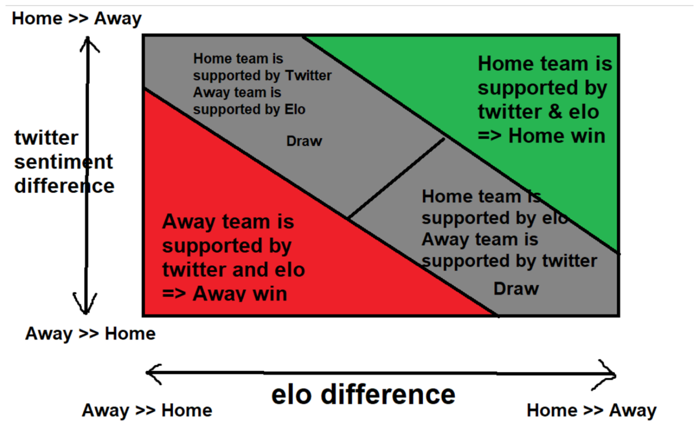
The Elo rating was originally invented as an improved chess-rating system. We will not get into details about the Elo system but you can find a lot of information online. The two key things to understand for the rest of this paper are :
- You cannot compute it yourself if you don't keep track of the team's last 200 ~ 300 games.
- Defining the score of a player by 1 if he wins, 0.5 if he draws and 0 if he loses, the elo difference between the two players can produce an expected score of each player between 0 and 1. This expected score gives you the probability of one player performing better than another one but not a probability of winning / drawing / losing. For example an expected score of 0.75 for player A could represent 75% chance of him winning, 0% chance of him drawing and 25% chances of him losing or it could represent a 50% chance of him winning, 50% chances of him drawing and 0% chances of him losing.

Part III : Models

As we only had around 100 matches to train and test our model on, we decided to use a SGD classifier.

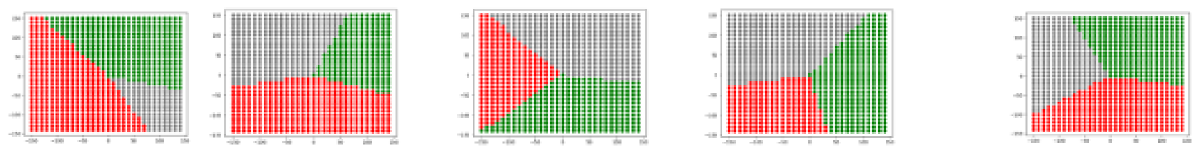
3.1) What is there to expect ?

The following schema shows the intuition we had about what the decision boundary of our SGD should've looked like by feeding the SGD with the difference of Elo (home team's Elo - away team's Elo) and the difference of sentiment (twitter sentiment about home team - sentiment about away team) :


Home >> Away
twitter sentiment difference
Away >> Home
Home team is supported by Twitter Away team is supported by Elo
Draw
Home team is supported by twitter & elo => Home win
Home team is supported by elo Away team is supported by twitter
Draw
Away team is supported by twitter and elo => Away win
elo difference
Away >> Home
Home >> Away

Green : Home team wins
Gray : Draw
Red : Away team wins

And the following plots are what the decision boundaries actually were for 5 different SGD classifiers we obtained :



The problem with SGD Classifiers is that different results are possible for a given dataset and as we see here they can vary a lot. But, these decision regions are not meaningless. In fact we can see the top right region being green most of the time, the red region being on the bottom left most of the time and the gray region being either on the top left or on the bottom right.

So the solution we came up with to stabilize the results is what we called an SGD Assembly : we trained 100 of them and then we had them voting on the outcome of the match. The outcome with the most votes is then chosen as the prediction.

This was our better agent but you can see in the following tables all the models we tested and their results.

	Twitter Only					
	SGD Assembly	Random Forest	KNN1	KNN3	NN (1 3 1)	NN (1 3 3 1)
Average	38,89 %	46,17 %	46,67 %	44,43 %	44,44 %	30,56 %
	Elo Only					
	SGD Assembly	Random Forest	KNN1	KNN3	NN (1 3 1)	NN (1 3 3 1)
Average	55,00 %	47,86 %	48,49 %	48,82 %	31,67 %	31,11 %
	Twitter & Elo					
	SGD Assembly	Random Forest	KNN1	KNN3	NN (2 3 1)	NN (2 3 3 1)
Average	70,56 %	60,00 %	52,78 %	54,45 %	43,33 %	40,00 %

Part IV : Betting

4.1) Betting 101.

Betting is actually going one step further than predicting the outcomes in terms of difficulty and there are several reasons for that. The first one is that even with 80% of correct predictions you might lose money. That's because the expected value of your bet is the following :

Expected value = (Probability of the outcome) x (Odd of the outcome) x Amount you betted – Amount you betted

The left term is what you are expecting to win and the right one is what you are paying to play.
So even if you have a great 80% good prediction, you can lose money if the games you bet on satisfy the condition :

$$P_{outcome} \times Odd_{outcome} < 1 \iff P_{outcome} < \frac{1}{Odd_{outcome}}$$

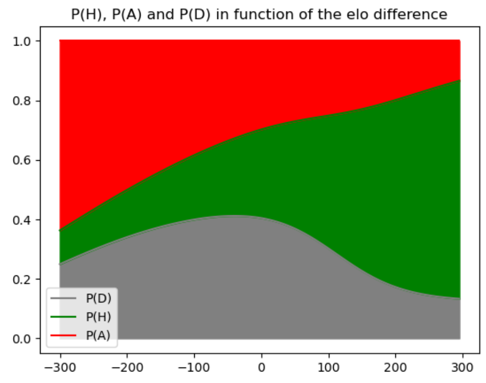
So it is not just about predicting the outcome, it is also about finding the probability of this outcome.

The other thing that makes betting ‘harder’ than predicting is that the betting companies are taking a margin on these odds. So let's say you guessed that the probability of an outcome was 33% and the betting company made the same prediction, then they would fix odds of something around 2.9 and even if your calculation were as accurate as theirs, you would lose money.

The problem is that our SGD Assembly is not producing any probability for the outcomes and thus we still need to find a way to get these probabilities.

4.2) Evaluating the probabilities.

The first solution we tried is the following : As the elo system is based on a probability of one player performing better than its opponent, there must be a way to get probabilities out of Elo difference. Actually, the Elo system is based on the ‘expected score’ of a player which is a value between 0 (100% chance of losing) and 1 (100% chance of winning). And so with this we get a probability of one team performing better than another one. The remaining problem is to evaluate the probability of a draw between these two teams. And the Elo system doesn’t offer an answer for that. So the way we did it is that we trained a neural network on detecting draws from Elo difference (the training dataset was the French Ligue 1 of 2020/2021 so it did not include any game we would test our bets on). And then instead of asking this Neural Network what it thinks will be the outcome, we ask it the probability of the outcome being a draw. Then we just fill the remaining probabilities with the expected score from Elo rating of the home team and of the away team and we obtain this :
Naming the event H an home win, D a draw and A an away win.



.

The second solution we tried is much simpler. We just use the votes of the assembly to determine the probabilities, i.e for a certain match, if among the 100 different SGD, 80 voted for a home win, 12 voted for a draw and 8 voted for an away win, we say that P(H) = 80%, P(D) = 12% and P(A) = 8%. This second solution seems much more disconnected from the reality of the game of football; rather, it relies on how confident the Assembly is for its prediction.

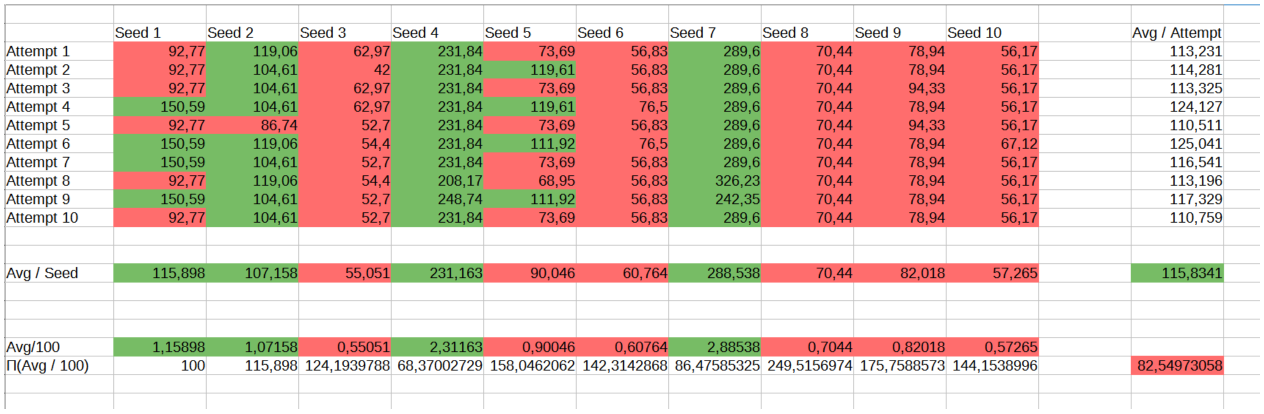
4.3) How much do we bet ?

Our next problem is to figure out how much we want to bet if we know the expected value of our bet. This question is really not as trivial as it seems and those who wish to know more about it can research the Saint Petersburg paradox.

The way we did it was through the use of the Kelly Criterion. The Kelly Criterion is based on the formula
“Happiness = log(€)” and so instead of trying to maximize the expected value of your capital, you want to max the expected value of the log of your capital. Surprisingly, this theorem is widely used and seems to be true. Since the 2000s it has become mainstream investment theory and the claim has been made that well-known investors including Warren Buffet and Bill Gross use Kelly methods. Hence, we choose the amount we bet in order to maximize the expected value of

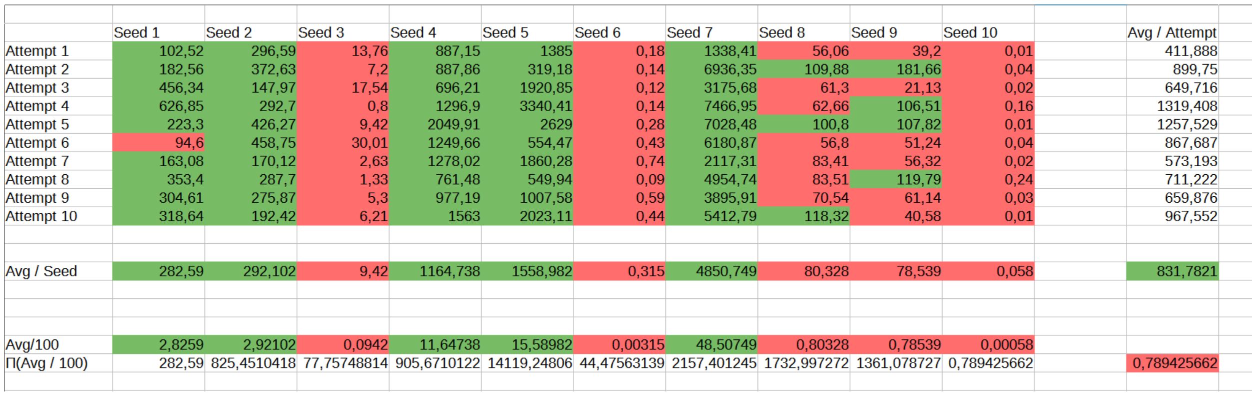
the log of our capital. Finally, in order to test our algorithm, we created 10 different 'seeds' that are permutations of the orders of our matches. That way we can train and test on multiple seeds and thus avoid having just one lucky strike or one bad streak. For each seed we will do 10 attempts and note the results (the training variability is reduced by the 'assembly' but different results can still occur). The following table represents for each seed and for each attempt the final capital in percentage of the initial capital (i.e how much euros we would have ended up with if we started with 100€).

For the solution 1 (Neural Network to predict draw probabilities) :



	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Seed 6	Seed 7	Seed 8	Seed 9	Seed 10		Avg / Attempt
Attempt 1	92,77	119,06	62,97	231,84	73,69	56,83	289,6	70,44	78,94	56,17		113,231
Attempt 2	92,77	104,61	42	231,84	119,61	56,83	289,6	70,44	78,94	56,17		114,281
Attempt 3	92,77	104,61	62,97	231,84	73,69	56,83	289,6	70,44	94,33	56,17		113,325
Attempt 4	150,59	104,61	62,97	231,84	119,61	76,5	289,6	70,44	78,94	56,17		124,127
Attempt 5	92,77	86,74	52,7	231,84	73,69	56,83	289,6	70,44	94,33	56,17		110,511
Attempt 6	150,59	119,06	54,4	231,84	111,92	76,5	289,6	70,44	78,94	67,12		125,041
Attempt 7	150,59	104,61	52,7	231,84	73,69	56,83	289,6	70,44	78,94	56,17		116,541
Attempt 8	92,77	119,06	54,4	208,17	68,95	56,83	326,23	70,44	78,94	56,17		113,196
Attempt 9	150,59	104,61	52,7	248,74	111,92	56,83	242,35	70,44	78,94	56,17		117,329
Attempt 10	92,77	104,61	52,7	231,84	73,69	56,83	289,6	70,44	78,94	56,17		110,759
Avg / Seed	115,898	107,158	55,051	231,163	90,046	60,764	288,538	70,44	82,018	57,265		115,8341
Avg/100	1,15898	1,07158	0,55051	2,31163	0,90046	0,60764	2,88538	0,7044	0,82018	0,57265		
Π(Avg / 100)	100	115,898	124,1939788	68,37002729	158,0462062	142,3142868	86,47585325	249,5156974	175,7588573	144,1538996		82,54973058

For the solution 2 (asking the assembly how confident it is) :



	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Seed 6	Seed 7	Seed 8	Seed 9	Seed 10		Avg / Attempt
Attempt 1	102,52	296,59	13,76	887,15	1385	0,18	1338,41	56,06	39,2	0,01		411,888
Attempt 2	182,56	372,63	7,2	887,86	319,18	0,14	6936,35	109,88	181,66	0,04		899,75
Attempt 3	456,34	147,97	17,54	696,21	1920,85	0,12	3175,68	61,3	21,13	0,02		649,716
Attempt 4	626,85	292,7	0,8	1296,9	3340,41	0,14	7466,95	62,66	106,51	0,16		1319,408
Attempt 5	223,3	426,27	9,42	2049,91	2629	0,28	7028,48	100,8	107,82	0,01		1257,529
Attempt 6	94,6	458,75	30,01	1249,66	554,47	0,43	6180,87	56,8	51,24	0,04		867,687
Attempt 7	163,08	170,12	2,63	1278,02	1860,28	0,74	2117,31	83,41	56,32	0,02		573,193
Attempt 8	353,4	287,7	1,33	761,48	549,94	0,09	4954,74	83,51	119,79	0,24		711,222
Attempt 9	304,61	275,87	5,3	977,19	1007,58	0,59	3895,91	70,54	61,14	0,03		659,876
Attempt 10	318,64	192,42	6,21	1563	2023,11	0,44	5412,79	118,32	40,58	0,01		967,552
Avg / Seed	282,59	292,102	9,42	1164,738	1558,982	0,315	4850,749	80,328	78,539	0,058		831,7821
Avg/100	2,8259	2,92102	0,0942	11,64738	15,58982	0,00315	48,50749	0,80328	0,78539	0,00058		
Π(Avg / 100)	282,59	825,4510418	77,75748814	905,6710122	14119,24806	44,47563139	2157,401245	1732,997272	1361,078727	0,789425662		0,789425662

As we can see both these solutions make profits on average (+15.83% and +731.78% on 18 matches) but if all the bets were taken consecutively (starting seed 2 with the balance you get at the end of seed 1, starting seed 3 with the balance you get at the end of seed 2 etc …) both solution would have ended up losing money (82.55 - 100 = - 17.45% and 0.79 - 100 = - 99.21%). So some further work can be done on optimizing the time to cash out.

One last thing to say is that when the algorithm only bases the amount he bets on how confident is the assembly (i.e solution 2) he is much more confident than in solution 1 and thus leading to big wins (up to 7367% increase) or overconfidence and big loses (melting up to 99.9% of its capital) which is a good highlight on why betting is much more complicated than predicting the outcomes.

Bibliography :

Kampakis S. & Adamides A (2014). Using Twitter to predict football outcomes.
https://arxiv.org/ftp/arxiv/papers/1411/1411.1243.pdf

Paul Deheuvels : La probabilité, le hasard et la certitude (2008). Editions Presses Universitaires de France.
https://www.cairn.info/la-probabilite-le-hasard-et-la-certitude--9782130571254.htm

How Science is Taking the Luck out of Gambling - with Adam Kucharski. The Royal Institution.
https://www.youtube.com/watch?v=658xlubwnDc

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter : What 140 Characters Reveal about Political Sentiment.
https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1441/1852

Lock, D., & Nettleton, D. (2014). Using random forests to estimate win probability before each play of an NFL game. Journal of Quantitative Analysis in Sports, 10(2), 9.
https://dr.lib.iastate.edu/entities/publication/434d718d-85ee-4ac9-96a0-b809dbfc868d

Ruiz E., Hristidis V., Carlos Castillo., Gionis A., Jaimes A (2012). Correlating Financial Time Series with Micro-Blogging Activity.
https://www.cs.ucr.edu/~vagelis/publications/wsdm2012-microblog-financial.pdf