Data Science in Practice

MGT - 415

The Sports Betting market: an Investment approach

Authors:
Allan Bellahsene
Antoine Marchal
Valentin Meylan

Supervisor:
Prof. Bruffaerts Christopher

Teaching Assistant:
Omar Ballester Gonzalez

11th May 2020

Contents

List of Figures

Introduction

Sport is a particularly lucrative leisure activity, and one industry is happily taking its share of the pie: sports betting. In France, a 2016 study indicated that 2,081 billion euros worth of bets were placed on the Internet that year, generating a turnover of nearly 350 million euros.[1] These staggering figures are indisputable: bettors lose a lot of money every year. This claim is further illustrated by the two figures below: for a given set of 7,202 English Premier League football games, we backtested two simple strategies consisting in i) randomly betting one of three possible football game outcome[2] for each game (Fig. 1) and ii) always betting on a victory of the team playing at home[3] (Fig. 2). In each of this strategy, a constant bet of 1% of the total bankroll is placed for each game. The conclusion is quite clear: the *Random* strategy went broke after 3,000 bets, while it took approximately 6,000 bets for the *Home* Strategy to lose 100% of the initial capital.

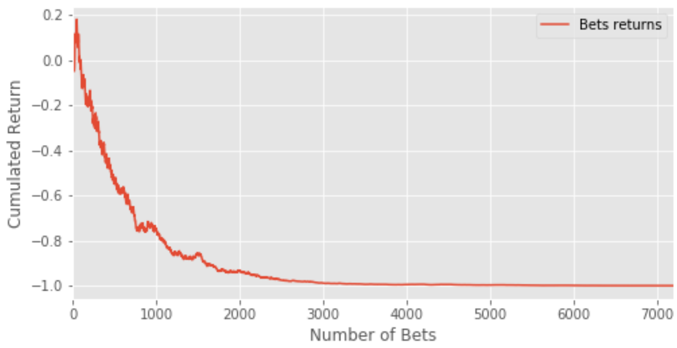


Figure 1: Cumulated Return on Invested Capital for *Random* strategy

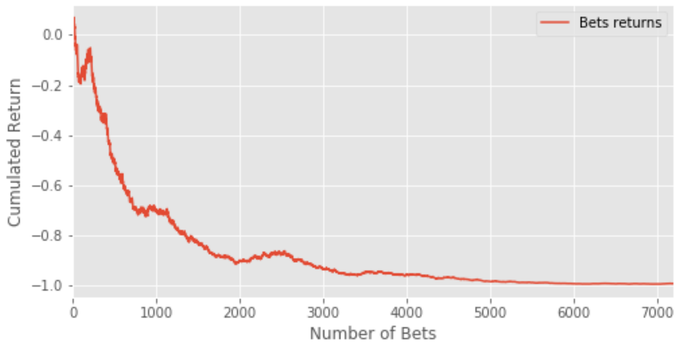


Figure 2: Cumulated Return on Invested Capital for *Home* strategy

[1] ARJEL, 2018.

[2] A classical Football game has three possible outcomes: a Home victory, a Draw or an Away victory.

[3] A team playing at Home has historically more chances to win than when playing Away. The odds used to backtest this strategy are the ones proposed by the bookmaker Interwetten.

Hence, one may wonder about the very aim of this practice: are we necessarily doomed to failure as bettors? A sport event naturally has a random component, so even the most astute experts are very often wrong about their predictions. Is it not possible, however, to adopt a quantitative approach in order to maximise the benefits while minimising the risks? Indeed, this is the approach used by financial specialists working in hedge funds.

The analogy between financial markets and the sports betting market seems quite natural: just as the results of matches are very difficult to predict on a regular basis (see Fig.1, Fig. 2), it is just as difficult to find financial instruments that will guarantee you both a stable and high return. Risk and return seem to be two inseparable concepts in both areas. In the financial sphere, most renowned investors and speculators are brilliant mathematicians who try to predict the market using complex mathematical models. However, there are extremely few who manage to beat the market on a regular basis, so the theory of efficient markets, which states that it is impossible to "beat the market," still finds many supporters nearly half a century after it was first coined by Eugene Fama [2]. So what is the market, then, in the case of sports betting? Bookmakers are the closest thing to it. Indeed, bookmakers quote the matches, via the odds. The odds reflect the estimated probabilities associated with each possible outcome of a given sporting event, to which bookmakers add a margin, ensuring a certain profit, regardless of the outcome. Therefore, what does it mean to "beat the market", in the case of sports betting, if every possible outcome is hedged by a margin? This can take many forms, but in this paper we choose to examine two of them.

Our first approach is analogous to a very famous investing strategy first theorized by arguably the most gifted investor that ever lived, Benjamin Graham [3]. This theory is called *value investing.* One of its main arguments is that the intrinsic value of an asset is not necessarily reflected by its price at time t, since it is largely impacted by the agents expectations, which are far from being rational at all times. As a result, the theory suggests that there are under and overvalued assets in the market at any given time. Hence, being able to identify them will enable the intelligent investor to make profits in the long run. This theory is naturally in contradiction with the one of efficient markets discussed above, but it has also had many proponents for many decades, including the iconic Warren Buffet. In the case of the sports betting market, there is a theory that is very similar to that of value investing: *value betting.* Just as asset prices can deviate from their true value at certain times, the "price" of some sports bets can deviate from their true value. In sports betting, the price of a bet is simply its odds, since it rewards the winning bettor proportionally to the odds of the outcome she has correctly predicted. In the case of classic odds, the higher the odds, the higher the potential payout associated with them. However, this potential payoff is obviously not without cost, since the higher the odds associated with a certain outcome, the more bookmakers believe that this result is unlikely to materialize. In other words, to use the famous expression popularized by Milton Friedman and others, *there is no free lunch.* As stated above, the odd a bookmaker attributes to a certain outcome is *mostly* the inverse of the probability estimated by the bookmaker for that outcome, to which a margin is added. Therefore, the concept of value betting is based on a simple objective: predicting *better* than bookmakers. This approach is meant to be extremely ambitious, and may even border on arrogance when we know that bookmakers hire teams of experts to estimate the most accurate probabilities, while at the same time having resources and data that are not publicly available. Nevertheless, there are frictions that can disrupt the sports betting market, as it is also governed by supply and demand, so that odds can be impacted by irrational betting behaviour, thus forcing bookmakers to adjust their odds... just as speculators can create bubbles that completely disconnect asset prices from their true value.

Our analysis will focus on one sport in particular: Football (or soccer). In the first part of this project, our goal will be to implement a data driven approach to predict Football games. In this section, we will mainly focus on Feature Engineering, as we believe this part is crucial given the initial data that we have and the dynamics inherent to Football. The features we created will then constitute inputs to different Machine Learning models. Then, we will try to implement a value betting strategy, using two approaches: a *passive* approach, where we will always bet on the outcomes predicted by the bookmakers (the minimal odds), and an *active* strategy, where we will place the bets of our value betting strategy based on the predictions of our Machine Learning models. Finally, a third part of this project will be dedicated to implement another strategy, which is very present in the hedge fund finance community, namely that of arbitrage opportunities. The aim will be to find loopholes in the way bookmakers price the odds. Although less attractive from an intellectual point of view, this method is often more profitable. However, our results will be qualified by the quality and relevance of the data we have at our disposal.

I. Football Data-Driven predictions

1. The Problem

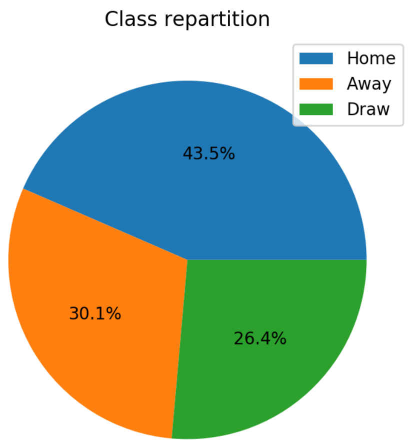


Figure 3: Distribution of the three target classes in the dataset

In Football, there are three possible results for a traditional game: a **H**ome team victory, a **D**raw or an **A**way team victory. In this section, we aim at predicting the final result of a game using data. We are therefore facing a classification problem. The data we use will be presented in the next section.

As shown in Fig.3 these three classes are slightly unbalanced in our data. In order to take into account this unbalance and improve our classifier models, we will use two strategies[4]:

- **Stratification**: This ensures that the class ratio is the same in the train and test set. This is done when splitting the dataset in train and test set by using the 'stratify' option in the 'train_test_split' function of sklearn. As for the cross-validation, during which the train set is further split in train-test set multiple times, stratification is ensured by default in the 'GridSearchCV' function.
- **Class weighted learning**: This adjusts weights inversely proportional to class frequencies in the objective function, i.e. it gives higher weight to minority class and lower weight to majority class. The idea is similar to re-sampling (with an under sampling of the majority class and an over sampling of the minority class). This is simply achieved with the option 'class_weight' in the sklearn classifiers.

[4] The unbalance in the classes being not so pronounced, we limited ourselves to these two strategies. However note that there exist other way to deal with unbalanced data, notably, using re-sampling which is not implemented in the present work.

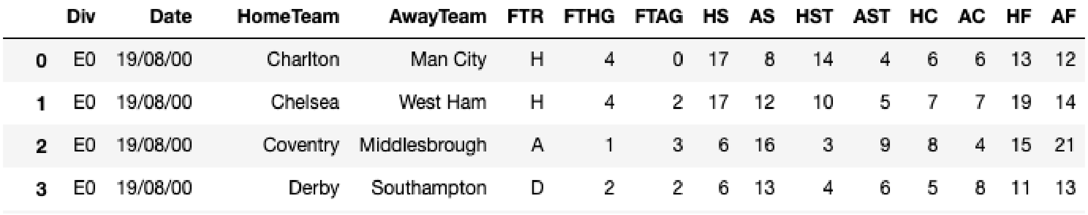
2. Data

Our dataset consists of observations of Football games in the first, second and third English divisions, for the seasons 2012-2013 to 2018-2019. The data is distributed as follows:

- Premier League (First Division): 2,284 observations
- EFL Championship (Second Division): 3,312 observations
- EFL League One (Third Division) : 3,312 observations

Hence, we initially have a total of $N = 8{,}908$ observations, imported from the specialized website *football-data.co.uk*, with a total of $D = 74$ features.

Each row of the dataset corresponds to statistics, measures and categories for a specific game including two teams, at a specific date.



	Div	Date	HomeTeam	AwayTeam	FTR	FTHG	FTAG	HS	AS	HST	AST	HC	AC	HF	AF
0	E0	19/08/00	Charlton	Man City	H	4	0	17	8	14	4	6	6	13	12
1	E0	19/08/00	Chelsea	West Ham	H	4	2	17	12	10	5	7	7	19	14
2	E0	19/08/00	Coventry	Middlesbrough	A	1	3	6	16	3	9	8	4	15	21
3	E0	19/08/00	Derby	Southampton	D	2	2	6	13	4	6	5	8	11	13

Figure 4: Table of a sample of the original dataset showing the main *game statistics* features

The main features of our dataset can be split as follows:

Categorical features

- Home Team (H) - name of the team that plays 'home', i.e. on its pitch
- Away Team (A) - name of the team that plays 'away', i.e. on the pitch of the opponent team
- Division (Div) - League Division of the game opposing Home Team and Away Team
- Full Time Result (FTR) - the final result of the game, which can only take three values: Home Team win, Away Team win, or Draw. This is the label we aim at predicting when using a classification approach

Game statistics

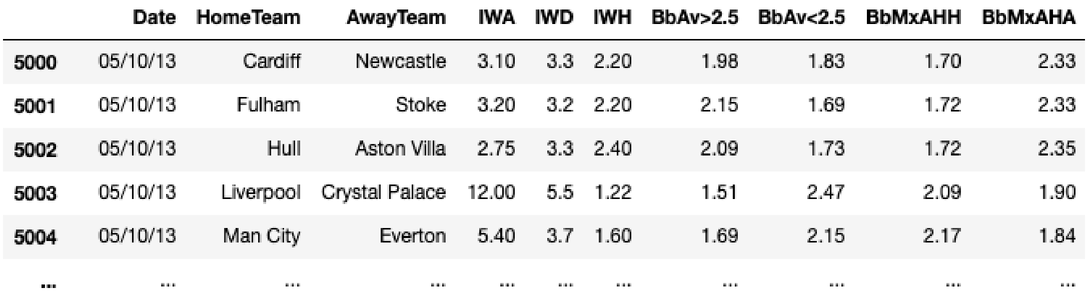
Each game statistics (goals scored, shots, fouls committed, etc.) is reported, for each row, for each team (see Fig. 4). To differentiate between the statistics of the two teams, we separate the statistics of the home team and of the away team. A few features are presented below:

- Full Time Home Goal (FTHG) - total number of goals scored by the Home Team,

- Full Time Away Goal (FTAG) - total number of goals scored by the Away Team
- Home Shots (HS) - total number of shots done by Home Team
- Away Shots (AS) - total number of shots done by Away Team
- Home Target Shots (HTS) - total number of on target shots by Home Team
- Away Target Shots (ATS) - total number of on target shots by Away Team
- etc.

There is a total of 18 game-statistics related features.

Odds



	Date	HomeTeam	AwayTeam	IWA	IWD	IWH	BbAv>2.5	BbAv<2.5	BbMxAHH	BbMxAHA
5000	05/10/13	Cardiff	Newcastle	3.10	3.3	2.20	1.98	1.83	1.70	2.33
5001	05/10/13	Fulham	Stoke	3.20	3.2	2.20	2.15	1.69	1.72	2.33
5002	05/10/13	Hull	Aston Villa	2.75	3.3	2.40	2.09	1.73	1.72	2.35
5003	05/10/13	Liverpool	Crystal Palace	12.00	5.5	1.22	1.51	2.47	2.09	1.90
5004	05/10/13	Man City	Everton	5.40	3.7	1.60	1.69	2.15	2.17	1.84
...	...	...	...	...	...	...	...	...	...	...

Figure 5: Table of a sample of the original dataset showing a few *bookmakers odds* features

The last main type of features in our dataset are odds from different bookmakers. For each game, and for each bookmaker, the odds cover the following outcomes:

- Odds regarding the Full Time Result - Home Win odd, Away Win odd, Draw odd. For example, the columns *IWA, IWD* and *IWH* in Fig. 5 are the full time results odds of the bookmaker *Interwetten* (IW).
- Odds regarding the number of goals at a given game. For instance, the columns *BbAv≥2.5* and *BbAv≤2.5* represent the odds proposed by the bookmaker *BetBrain* corresponding to the following event: the number of total goals for a given game will be larger than 2.5 or smaller than 2.5.

Our dataset includes these kinds of odds for 8 different bookmakers. A completely detailed description of all features is available on the ANNEXE which completes this report.

3. Features Engineering

The concept of *momentum*

To predict the outcome of a game occurring at time t, we cannot use the game statistics features available at time t. Indeed, we can only use the game statistics available until time $t-1$, because the statistics displayed at time t in our dataset are in reality only available once the game is finished. Hence, we can formulate the problem as follows:

$$\hat{P}(FTR_t = i) = \phi_i(\mathbf{F}_{t-1}, \mathbf{F}_{t-2}, ..., \mathbf{F}_{t-k}), i = \{0, 1, 2\} \tag{1}$$

meaning, the estimated probability $\hat{P}$ that a game at time t takes one of the three possible outcomes i is a function of the D *game-statistics* features f from $t-k$ up to $t-1$, where

$$\mathbf{F}_{t-j} = \begin{pmatrix} f_{1,t-j} & f_{2,t-2} & \cdots & f_{D,t-j} \end{pmatrix}^T$$

are $D \times 1$ vectors, for $j = \{1, ..., k\}$.

We make the following assumption: the more recent the data, the more weight should be given to it to predict the outcome of a game. We base this assumption on the concept of *momentum*, meaning that a team which had recently performed good is most likely to perform good on the next game, while a team that has performed bad lately will most likely perform badly in the future. To capture this concept in our model, we use the Exponential Weighted Moving Average (EWMA) function. For a given feature f, the EWMA function is defined as follows:

$$EWMA(f_t) = \frac{f_{t-1} + (1-\gamma)f_{t-2} + ... + (1-\gamma)^{k-1} f_{t-k}}{1 + (1-\gamma) + ... + (1-\gamma)^{k-1}} \tag{2}$$

where the parameter $\gamma \in (0, 1)$ is a coefficient that gives more weight to the more recent observations. In other words, the weights given to each observation decrease exponentially as the observations become far in time, and the higher γ, the higher this exponential decay. k is the *span* parameter, which indicates how far in time one should 'go'. Both hyperparameters are optimized when selecting the best machine learning model.

Hence, for each prediction at time t, the features used for estimating the labels are in fact transformed features. Therefore, we have:

$$\hat{P}(FTR_t = i) = \Phi_i(\tilde{F}_{1,t}, \tilde{F}_{2,t}, ..., \tilde{F}_{D,t}), \tag{3}$$

where

$$\tilde{F}_{d,t} = EWMA(f_{d,t}),$$

$d = \{1, ..., D\}$

Hence, equation (3) summarizes that the estimated probabilities for the three possible outcomes of a game occurring at time t are computed as a function of the game statistics features available until time $t-1$. $\tilde{F}$ is what we called the *Momentum Transformed Feature*, which is done by applying the EWMA function defined in equation (2) to all relevant game statistics features.

The function Φ_i thus refers to the model applied to these features, which we discuss in the next section.

To be perfectly rigorous, equation (3) only takes into account game statistics features to estimate the labels, while in reality we also use another type of features to which the momentum transformation is not applied: the odds of the different bookmakers. Therefore, for a total number of D features used for our prediction, let D_1 denote the game statistics related features, and D_2 denote the bookmakers-related features, such that $D = D_1 + D_2$. The D_2 bookmakers related features are available at date t, hence no transformation is needed. Therefore, we model the estimated probability of a game at time t finishing with a Home Victory, a Draw or an Away Victory by the following equation:

$$\hat{P}(FTR_t = i) = \Phi_i(\tilde{F}_{1,t}, \tilde{F}_{2,t}, ..., \tilde{F}_{D_1,t}; F_{1,t}, F_{2,t}, ..., F_{D_2,t}), \tag{4}$$

$i = \{0, 1, 2\}$

Features relevance

After implementing this feature transformation, two steps are left before implementing any machine learning model. The first step consists in normalizing the features for them to contribute to the algorithms in the same scale. Then, the second step consists in implementing Principal Component Analysis (PCA). Indeed, PCA will help us reduce the dimensionality, which we know is not very desirable for some algorithms (curse of dimensionality), and can help us to drastically diminish the computing time of the different algorithms. After performing PCA on our features, we managed to reduce the dimensionality from D = 74 to D = 30. As one can see from Fig. 6, the explained variance ratio stops increasing after D = 30.

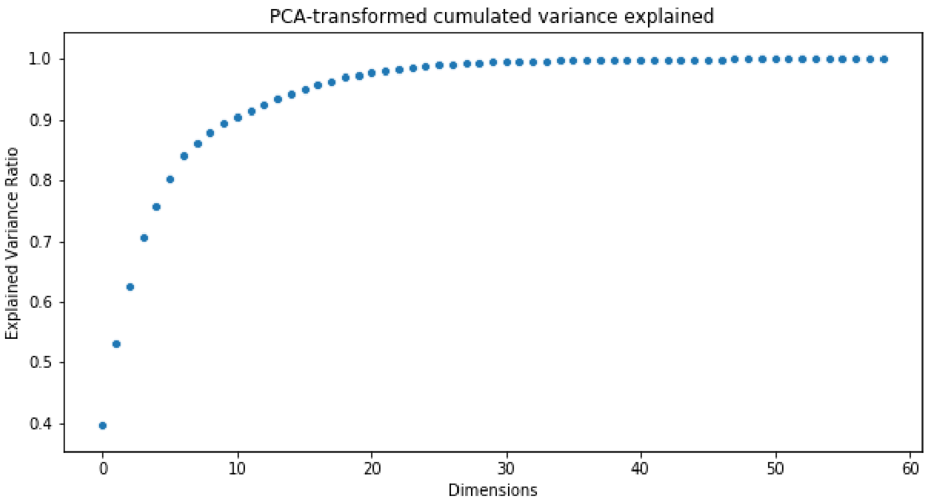


Figure 6: Principal Component Analysis cumulative explained variance as a function of number of features

4. Models

The three categorical[5] classes H, D and A being known in advance, we face a supervised learning problem. In this context, the learning models that we will use in this are the multi-class classification model, seen or mentioned in class:

- K-Nearest Neighbors
- Logistic regression
- Support Vector Machine (SVM)
- Random forest
- Naïve Bayes

Note that, some models such as SVM and logistic regression models are not inherently appropriate for multi-class classification. However, all binary classifiers can be extended to accommodate multi-class classification. This is done through a strategy called '*One-VS-All*' that consists in separately fitting each class vs the others in a binary mode.

5. Metrics

The choice of metric to optimize in the cross-validation will play a crucial role into the selection of models and the tuning of the hyperparameters. The general metrics used for classification are listed below:

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

where TP and TN denotes respectively the True Positive and Negative, and FP, FN respectively denotes the False Positive and Negative. In general $F1$ score is preferred over the very common accuracy as it better takes into account false positive and false negative. Doing so it is better suited for unbalance classes, as it is slightly the case in this project (see Fig.3). However, despite the fact that all the metrics described above could be used in a general classification problem, every problem is different. The more we understand our problem, the more we can use its specificities to our advantage.

First of all note that our classification problem aims to classify each match in three category, it is a multi-class problem. Therefore the $F1$ score can not take into account the three classes simultaneously. It can only be computed for each class separately and then be averaged over

[5] Notice that, at first sight, these categorical variables seems nominal. However they present some kind of ordinality. Indeed a prediction of a draw is closer to an actual win or defeat than a win prediction is from an actual defeat (or the contrary). We will try to take this into account when it will come to choose the score metric for the model selection.

them.

Secondly, in spite of the fact that our categorical variables H, D and A may seem nominal at first sight they present in fact some kind of ordinality, in the sense that if the outcome of the match is A, a prediction of D is closer to the realized outcome than a prediction of H. The false predictions are not equally wrong as it would have been the case for purely nominal variables. For instance if our goal was to predict the nominal beverage preference of an individual between Coke, Fanta and Icetea[6]; predicting Fanta or Icetea when the true answer is Coke, is equally wrong. These two errors should be equally penalized. Here it is not the case.

In order to take into account these two aspects, we chose to use the **R**ank **P**robability **S**core as metric. This choice is motivated by the work of Constantinou & Fenton [4] that demonstrated that this metric is the more suitable to assess probabilities assigned to football match outcomes. It is defined as follows:

$$RPS = \frac{1}{r-1}\sum_{i=1}^{r-1}\left(\sum_{j=1}^{i}(p_j - e_j)\right)^2$$

where r is the number of potential outcomes and $p_j \in [0,1]$ and $e_j \in \{0,1\}$ are respectively the forecast probability and observed outcomes. In the case $r = 3$ and and the expression reduces to:

$$RPS = \frac{1}{2}\sum_{i=1}^{2}\left(\sum_{j=1}^{i}(p_j - e_j)\right)^2$$

where e_j takes the value 1 when the outcome correspond to the j^{th} category in $\{A, D, H\}$. This metric takes simultaneously into account the three categories and, in addition, the penalty attribution is sensitive to the ordering described above. This is a loss function for a single instance, the averaged sum of these losses for all matches (i.e. the objective function) should be minimized over the test sets under the cross validation.

6. Results of most effective models

[6] Here are two other brands to comply with our no product placement policy: Rivella and Ovomaltine Drink.

1. With F_1 score metric

	precision	recall	f1-score	support
A	0.426773	0.563444	0.485677	662.000000
D	0.304348	0.253448	0.276576	580.000000
H	0.563919	0.495798	0.527669	952.000000
accuracy	0.452142	0.452142	0.452142	0.452142
macro avg	0.431680	0.437564	0.429974	2194.000000
weighted avg	0.453918	0.452142	0.448620	2194.000000

Table 1: Classification report of logistic regression

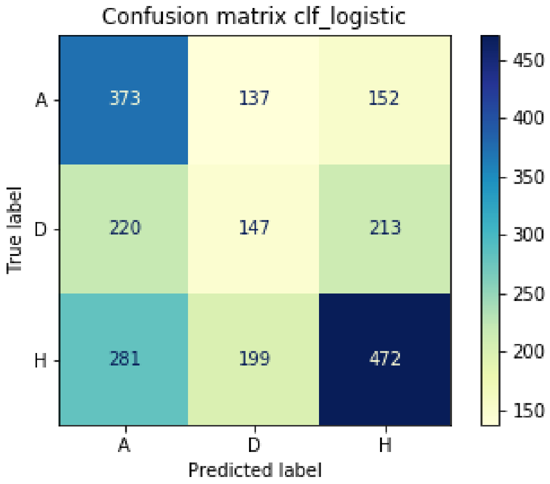


Figure 7: Confusion matrix for logistic

2. With custom score metric

	precision	recall	f1-score	support
A	0.450535	0.509063	0.478014	662.000000
D	0.302826	0.387931	0.340136	580.000000
H	0.600284	0.443277	0.509970	952.000000
accuracy	0.448496	0.448496	0.448496	0.448496
macro avg	0.451215	0.446757	0.442707	2194.000000
weighted avg	0.476465	0.448496	0.455431	2194.000000

Table 2: Classification report of Support Vector Machine

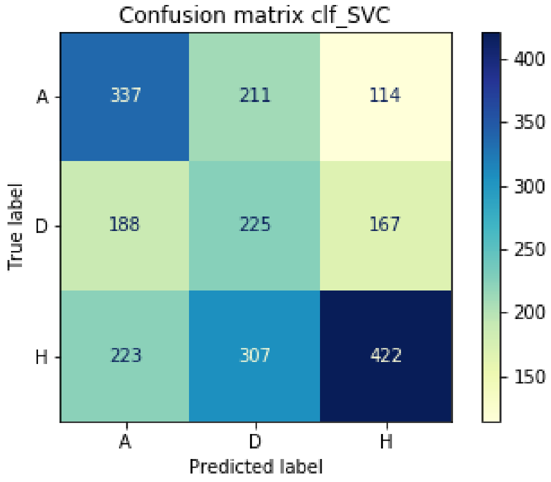


Figure 8: Confusion matrix for Support Vector Machine for custom metric

II. Value Betting Strategy

1. Value betting: definition

In value betting, the goal is to find events that are mispriced by bookmakers. To understand how an event can be mispriced, one first needs to understand what does an odd tell us.

In theory, an odd associated to a certain outcome reflects the implied estimated probability of this outcome. In fact, an odd is said to be *fair* if, for a given outcome A, we have:

$$Odd_A = \frac{1}{P_A}, \tag{5}$$

namely, the odd of an event A is fair if it is equal to the inverse of the probability of this event.

The expected gain relative to a bet of value w placed on event A is given by:

$$\begin{aligned} E[A] &= w \times (Odd_A - 1)P_A - w \times (1 - P_A) \\ &= w[(Odd_A) - 1)P_A - (1 - P_A)] \\ &= w(Odd_A P_A - 1) \end{aligned}$$

The concept of a fair odd is naturally tied to the one of expected gain. If the odd of a given event is fairly priced, then the expected return associated to it is equal to zero. Mathematically, if equation (1) is satisfied, we have:

$$E[A] = w(\frac{1}{P_A} P_A - 1) = 0 \tag{6}$$

Therefore, a fair market is characterized by a null expected gain. Is the sports betting market a fair market?

The two strategies illustrated in the introduction seem to indicate that it is not the case, and this is further reinforced by another 'naive' strategy we implemented. For the same set of 7,202 English Premier League football games that the first two strategies were implemented on, we implemented a third strategy that consisted in always betting in the minimal odd proposed by the bookmaker Interwetten for each game. Indeed, if the odds are fairly priced, then the minimal odd should represent the bookmaker's estimation of the highest probabilities. The cumulative return of this strategy is plotted on the figure below.

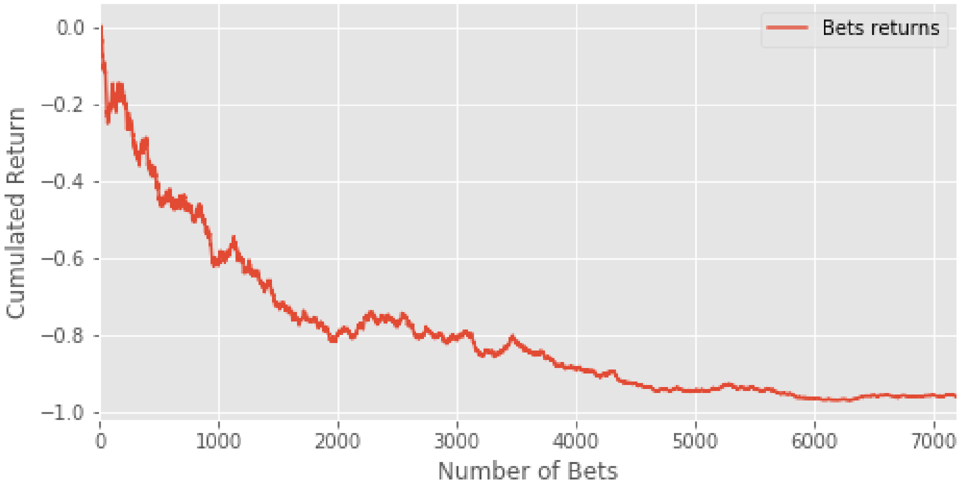


Figure 9: Cumulative Return on Invested Capital for *Minimum Odd* strategy

The cumulative return of this strategy is clearly negative, and exponentially decreasing with the number of bets, until reaching almost 95% of loss on invested capital after betting on all games.

This example indeed illustrates that, as the saying goes, *the house always wins.* Indeed, the odds proposed by any bookmaker deviate from the fair odds by a constant α, which represents their margin:

$$Odd(A)_{Bookmaker} = Odd(A)_{Fair} - \alpha, \tag{7}$$

where $\alpha \in (0, 1)$.
Inserting this equation in equation (2) yields:

$$E[A] = w(P_A(\frac{1}{P_A} - \alpha) - 1) = -\alpha w P_A < 0 \tag{8}$$

This implies that any bet is expected to yield a loss for the bettor !

However, this claim is true if and only if the probability estimated by the bookmakers for each event they quote correctly estimates the true probability of each event. Indeed, equations (2) and (4) imply that we can simply replace the odd of any event by the inverse of the probability, which means that the bookmakers *perfectly* estimate the probability associated to any event. This cannot be true in practice. In fact, if a bettor manages to estimate the probability associated to an event better than a bookmaker, then she can have a positive expected gain, even though the odd associated to this bet is unfair. Indeed, let P_B denote the true probability associated to event B and $\hat{P}_B$ denote the probability of event B estimated by the bookmaker. The expected gain of the *bookie* is naturally equal to minus the expected gain of the bettor:

$$\begin{aligned} E^{bookie}[B] &= -E^{bettor}[B] \\ &= -w(Odd_B^{bookie} P_B - 1) \\ &= w(1 - \frac{P_B}{\hat{P}_B + \alpha}) \end{aligned}$$

And therefore, we have:

$$E^{bookie}[B] \geq 0 \iff \alpha \geq |P_B - \hat{P}_B| \tag{9}$$

In other words, the expected gain of the bookmaker associated to bet B is positive only when the margin of the bookmaker α is large enough to cover the difference between the true probability and the probability estimated by the bookmaker. Therefore, a *value bet* for the bettor is any bet B such that this condition is not satisfied, i.e.

$$\alpha \leq |P_B - \hat{P}_B| = \delta \iff E^{bettor}[B] \geq 0 \tag{10}$$

Hence, the higher δ, the higher the expected gain for the bettor.

This is exactly the aim of this section: our goal will be to fit different models to try approximate the true probability of the events associated to Football games the most accurately possible. We will then see if our accuracy allows us to find such value bets.

2. The Problem

In this section, we aim at performing our value betting strategy on the total number of goals scored by both teams for each game. Our goal *in fine* will be to compare a so called *active* strategy to a *passive* strategy. The active strategy will be one where the outcome on which to bet will be predicted using Machine Learning algorithms, while the passive one will be implemented by only betting on the minimal odds. At the end, we will compare the performance of these two strategies to assess whether our active strategy can outperform the 'market'.

Therefore, to implement our active strategy, we are facing a regression problem. The target variable is $FTTG_i$, the Full Time Total Goals, which is simply the sum of $FTAG_i$ and $FTHG_i$. Predicting $FTTG_i$ will then allow us to test our value betting strategy on the two following bets:

- $FTTG_i > 2.5$, which means betting that the total number of goals for game i will be larger than 2.5

- $FTTG_i < 2.5$, which means betting that the total number of goals for game i will be smaller than 2.5

The features we used are the same as the ones we used for our classification approach, what is different here is of course the targets variables, which are numerical values and therefore require us to use regression models. The models we used are presented in the next section.

Each model is trained with respect to:
1. The hyperparameters proper to the momentum features (the center of mass γ and the span k)
2. The hyperparameters proper to each model (for example, the regularization parameter λ for a Ridge regression)
3. Several objective functions

The goal is then to retain the models which optimize each objective function after a cross validation. Then, if for example we considered n objective functions, we will have n optimal models. Finally, those n models will be trained again, but this time the only objective will be to yield the highest return on investment after implementing a value betting strategy.

Remember the bookmakers' margin α associated to an event E is defined by:

$$\alpha(E) = \frac{1}{Odd(E)} + \frac{1}{Odd(\overline{E})} - 1,$$

where $\overline{E}$ is the complementary event to event E.
As stated in the beginning of this chapter, the lower α, the higher the expected gain for the bettor. However, the lower α, the less the opportunities for the bettor, because a low α implies a high risk for the bookmaker. There is therefore a risk-return trade-off for the bettor. Our goal will thus be to:
1. Invest only on games where α is small enough, i.e. $\alpha \leq \delta$, where $\delta > 0$. Those games are the so-called **value bets**. Recall δ is the absolute difference between the true probability of an outcome and the estimated probability of this outcome by the bookmaker. Therefore, δ is unknown by definition. We decide to use $\hat{\delta} = 0.05$, following the findings of Kaunitz, Zhong Kreiner [**?**], which states that on average, the error associated to a bookmaker's estimated probability equals 5%.
2. After identifying the value bets, we place the bet on the outcome predicted by our models $\Psi_1, ..., \Psi_n$

3. Models

For any regression problem, the goal is to minimize the *distance* between the value of the target variable and the value of the estimated target variable. The *distance* can be measured in several ways, by different metrics. Those metrics will be presented in the next section. For now, we are interested in presenting the different algorithms that we decided to use in order to estimate the values of the target variable. The algorithms we decided to use are:

- Standard Linear Regression
- L1 Regularized Linear Regression (Lasso)
- L2 Regularized Linear Regression (Ridge)

In the standard linear regression algorithm, the goal is to fit a line with the features X and the coefficients β in order to minimize the distance between the true target variable y and the estimated target variable $\hat{y}$. The problem can be written as follows:

$$min_\beta \text{ distance}(X\beta, y) \tag{11}$$

where $X\beta = \hat{y}$
The two penalized versions of this algorithm add a penalty term to this problem, in order to avoid the algorithm to overfit, and hence to have a better generalization performance. The Lasso regression problem can be summarized as follows:

$$min_\beta \text{ distance}(X\beta, y) + \lambda|\beta| \tag{12}$$

The penalty term is simply the product of a parameter λ by the absolute value of the coefficients β. The Ridge regression differs from the lasso regression in that the penalty term is the product of the penalty parameter by the *squared magnitude* of the coefficients β:

$$min_\beta \text{ distance}(X\beta, y) + \lambda||\beta||_2^2 \tag{13}$$

These penalized algorithms allow to reduce overfitting by "forcing" the coefficients associated to irrelevant features to vanish. The real difference between the two versions is that the L1 actually sets those coefficients to zero, while the L2 version can only make them converge to zero.

Hence, when considering each metric, we will first use cross-validation to find the optimal parameter λ that gives the best algorithm, then we will be able to compare the performance of each algorithm with another. The goal will thus be to first, train each algorithm separately to obtain the version of each, then compare the performance of these best versions according to the different metrics.

4. Metrics

The *distance* between the estimated values of the target variable and the true values of the target variable can in fact be measured by several metrics. The metrics we decided to use for this regression problem are:

- The Mean Absolute Error (MAE)
- The Mean Squared Error (MSE)

When minimizing the MAE, the goal is to minimize the average of the absolute difference between the true and the estimated value of the target variable:

$$MAE(\hat{y}_n, y_n) = \frac{1}{N}\sum_{n=1}^{N} |y_n - \hat{y}_n| \tag{14}$$

Taking the absolute value of the error aims at penalizing in the same way a prediction that would over-estimate and under-estimate the true observation. One problem with this metric is that the absolute function is not continuous and derivable in every point, which can of course be a problem when the goal is to minimize such a function. MSE is a way to overcome this problem, as minimizing it consists in minimizing the average of the squared difference between predictions and observations:

$$MSE(\hat{y}_n, y) = \frac{1}{N}\sum_{n=1}^{N} (y_n - \hat{y}_n)^2 \tag{15}$$

The squared function is indeed continuous and derivable in every point, which is a desirable property for an optimization problem. However, the MSE tends to give *too much* weight to outliers, in the sense that an observation which is absurdly too far from a ”normal” observation can alterate an algorithm whose objective is to minimize the MSE. One thing to also take into consideration is that the MSE is not expressed in the same units as the variable of interest, at the contrary of MAE.

Nevertheless, all these metrics will be used to train our different algorithms. We will thus obtain one optimal algorithm per metric. Then, we will train our value bet strategy with these three algorithms, and we will retain the one that gives the best performance. This algorithm will then be tested on a test set, and the performance of the value bet strategy implemented with it will be compared to the one yielded by the bookmaker benchmark strategy.

5. Finding the optimal models

The first problem consists in training each algorithm with respect to each metric. For a given metric, we thus test three models. For each of these three models, we optimize with respect to:

- The momentum features γ and k
- The penalty parameter λ (which is zero in the case of the standard linear regression)

Then, we use cross-validation to compare the performance of the algorithms for a given metric. The plots below illustrate the test errors of the three optimal algorithms as a function of γ and k:

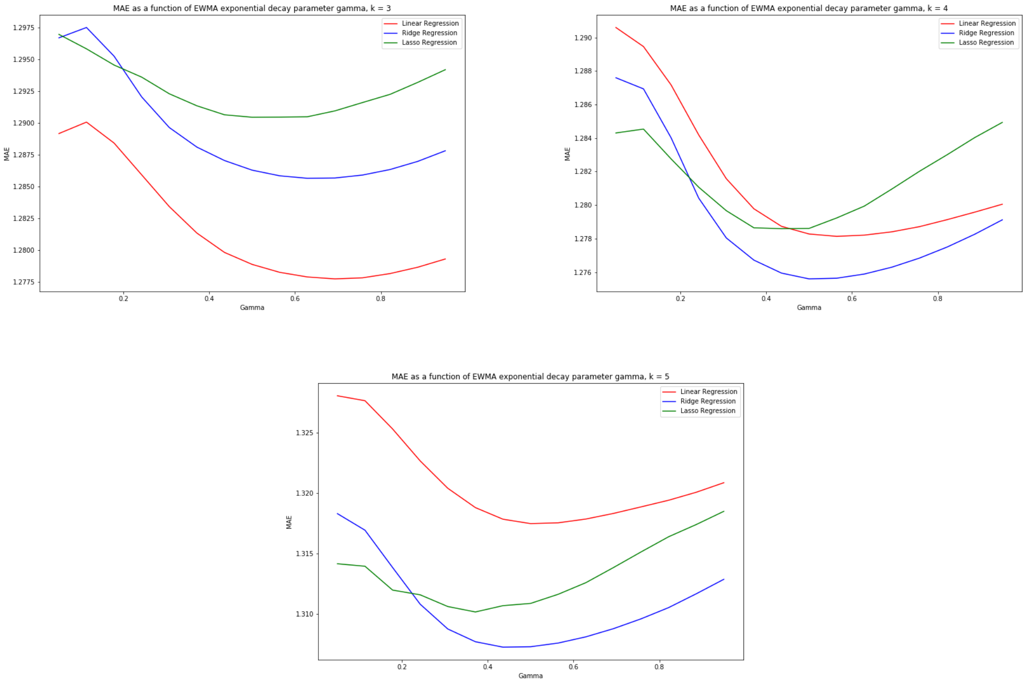


Figure 10: MAE as a function of EWMA parameter γ for the three linear regression algorithms, for k = 3, 4, 5

Figure 10 allows one to see that the error reflected by the MAE is indeed impacted by the momentum parameters γ and k.

The same procedure is done to find the optimal algorithm when the considered cost function is MSE:

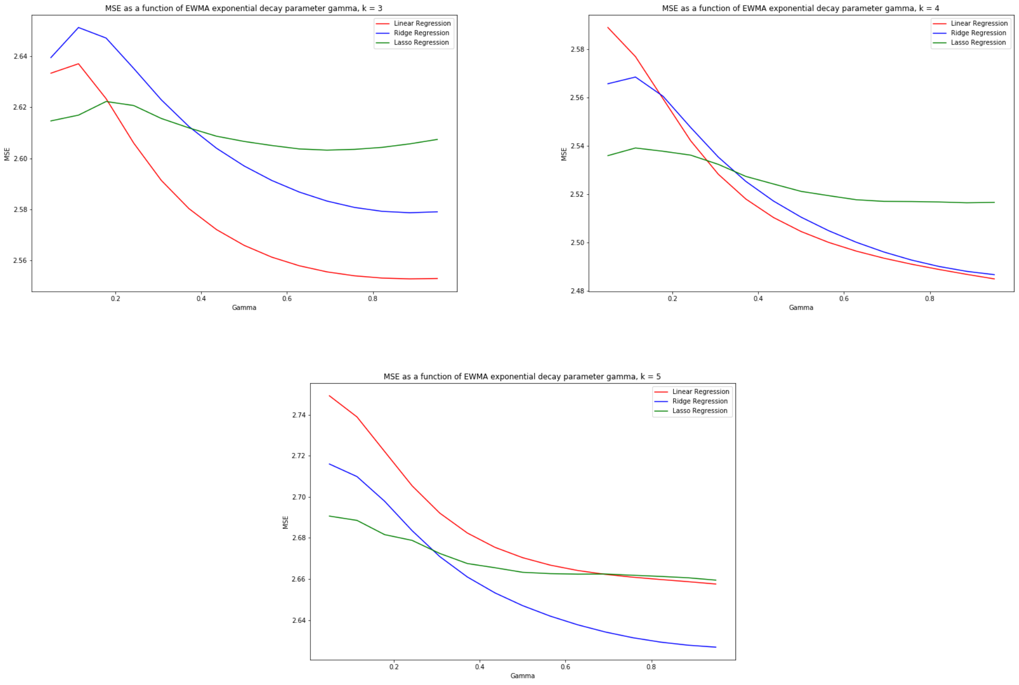


Figure 11: MSE as a function of EWMA parameter γ for the three linear regression algorithms, for k = 3, 4, 5

We can see that, similarly to what was observed with MAE, the mean squared error mainly has a negative relation with the parameter γ. Recall this parameter gauges how much weight should be given to recent games: the higher this parameter, the less relevant the observations far in time are to explain current observations. In other words, it seems that what can be translated as the *confidence* of a team changes rapidly, while the fact that the optimal k is equal to 4 can be interpreted as the fact that the results of a team more than 4 games ago have little to no importance to explain their performance for the next game.

The results of the best algorithms for the two different metrics are summarized in the table below:

Cost function	**Best algorithm**	**Min. error**	γ^*	k^*
MAE	Linear Regression	1.277	0.8857	3
MSE	Linear Regression	2.4850	0.95	4

Table 3: Best algorithm per metric

Hence, we have two optimal models on which to train our value bet strategy.

6. Training the value bet strategy

Now that we have two optimal models to predict the Full Time Total Goals of a game, we now want to find which one is the best when implementing a value bet strategy. Recall equation (6), which states that a value bet strategy occurs when α, the bookmaker's margin for a given bet, is smaller than δ, the difference between the true probability of an outcome, and the bookmaker's estimation of this probability. δ is thus by definition unknown, and we decide to use an approximation $\hat{\delta} = 0.05$, following the findings of Launitz et al. [5]. Therefore, our decision rule is the following: for every bet, if $\alpha < 0.05$, we place a proportion w of our bankroll on the outcome predicted by our two best models. We then compare the performances of each of our models, and retain the one that yields the best performance to use it on a test set.

We decide to implement a variant allocation strategy, meaning the weight w of our bankroll that we invest for each bet must be proportional to the risk/benefit associated to each bet. For each bet, we decide to invest the proportion w^* of our bankroll, such that:

$$w_i^* = \frac{a}{\alpha_i Odd_i}$$

where a is a parameter associated to risk-aversion. The higher α_i, the higher the expected gain of the bookmaker; hence, the less weight should be put on this bet. Similarly, the less weight should be put to bets with high odds, as they are riskier. Finally, the higher a, the more aggressive the strategy. Hence, when training the algorithms on the value bet strategy, we aim at finding the optimal value for a, and the goal is to retain the value of this parameter that gave us the highest performance.

We decide to use the Sharpe Ratio as a measure of performance, which is defined by:

$$Sharpe = \frac{E(r)}{\sigma(r)}, \tag{16}$$

where $E(r)$ denotes the average return of bets for a given strategy, while $\sigma(r)$ is the standard deviation of the returns associated to this strategy. The plots below illustrate the relationship between risk-aversion $1/a$ and different measures of performance (Sharpe Ratio and Return on Investment) as well as volatility measured by standard deviation of returns per bet.

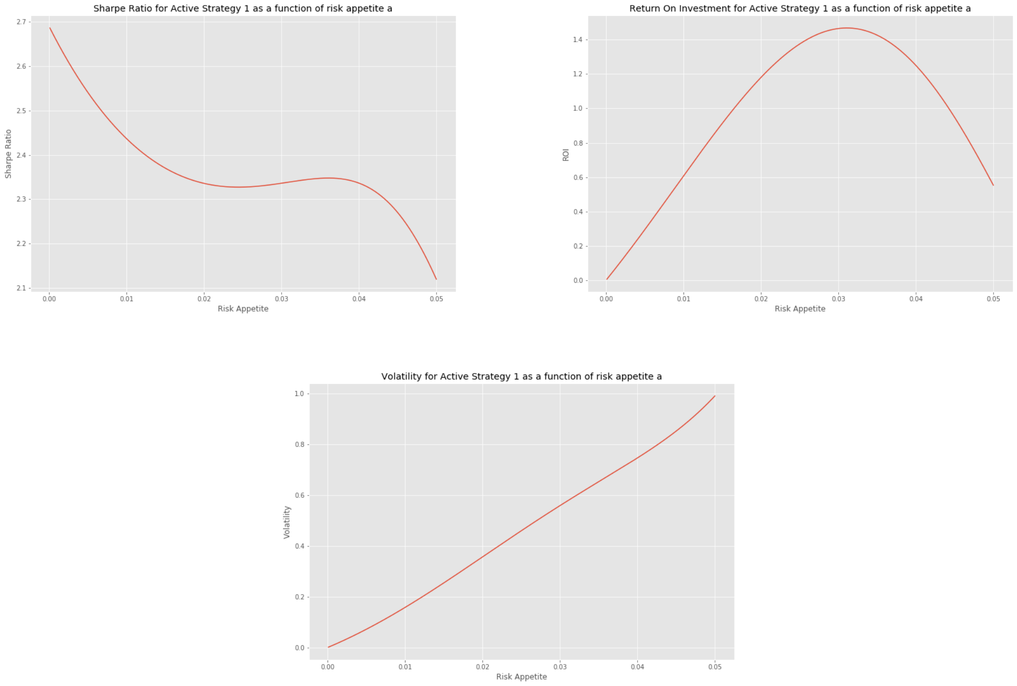


Figure 12: Performances and risk measures as a function of risk aversion for active strategy 1

Fig. 12 shows us the relationship between risk aversion and Sharpe Ratio, Return On Investment and Volatility for Active Strategy 1 (i.e. strategy where the predictions are computed with the best model for MAE, listed on Table 1). We can see that there is a clear increasing linear relationship between the degree of risk appetite a and volatility. Our goal is thus to find the right balance between an a that would generate a high Sharpe Ratio, but also a large return on investment (referred to as ROI). The optimal a^* is therefore the one that maximizes the Sharpe Ratio, for a minimum ROI of 5%, as we believe any strategy yielding a ROI below this on the train set is not worth testing.

The results of our two models are listed below:

Best Model	**Sharpe Ratio**	**ROI**	a^*
Model 1	2.659494	5.08%	0.0009
Model 2	1.279966	5.21%	0.0014

Table 4: Perform

The best model is thus Model 1, i.e. the model defined by:

- Linear Regression with $\gamma = 0.95$, $k = 3$, $a^* = 0.0009$

We then test the performance of this model on a new dataset it has never been trained on, and compare the performance of the value bet strategy with a benchmark performance, i.e. the performance if we always bet on the minimal odd, for any value bet detected by $\hat{\delta}$ and for the same level of risk appetite $a^* = 0.0009$.

7. Testing the value bet strategy

We test our value bet strategy on a dataset consisting of 2,261 La Liga Spanish Championship games, between 2013 and 2018. We first transform the features using γ^* and k^*, before implementing the value bet strategy, for:

- An active strategy, i.e. the one where the games are predicted by the optimal Linear Regression
- A passive strategy, i.e. by betting on the minimal odds proposed by the bookmaker

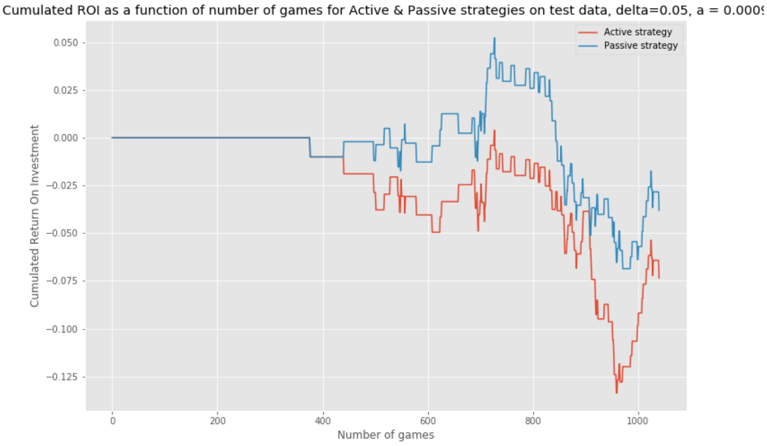


Figure 13: Cumulative ROI for Active & Passive Value Bet Strategies on test dataset, with $\hat{\delta} = 0.05$ and $a^* = 0.0009$

As one can see, both strategies did not perform well on the test set, but one can see the benchmark strategy performed better than our active strategy. In any case, one cannot conclude that any of these strategies is profitable.

The weakness of our value bet strategy can have many sources. The main that comes to mind is the fact that we only bet on the odds proposed by one bookmaker, which severely limited our possible bets. One issue might also come from the choice of the parameter $\hat{\delta}$.

III. Arbitrage Opportunities strategy

1. Be close to your friends and closer to your ennemies

In order to arbitrage the football betting market, we must first understand how the odds offered by the bookmakers are constructed. This is actually quite simple. Given the probability P of an event, the fair odd for this event is

$$\pi_{fair} = \frac{1}{P}$$

such that the expected value of a bet B on that event is a martingale:

$$E[B] = B.P.\pi_{fair} + B.(1-P).0 = B$$

So, once the bookmakers have the probabilities, they obtain the fair odds and then they fix their odds below this fair odds:

$$\pi = \pi_{fair} - \alpha \tag{17}$$

this way αP is their expected commision and the expected value of a bet becomes a supermartingale:

$$E[B] = BP\pi = BP(\pi_{fair} - \alpha) = B(1-\alpha P) < B$$

The house always wins, this is the basic principle of all gambling institution. For instance when betting on reds at the roulette[7] the probability of winning is $P = \frac{36}{2.36+1} = 0.493$, which gives a fair odd of $\pi_{fair} = 1/P = 2.027$, but the odd offered by the casino is only of 2, hence the casino will take an expected commission of $\alpha P = 1.4\%$ on every bet.

2. Divide to better reign

Now it is important to understand two key differences between a casino and a bookmaker.

The first difference is that contrary to a casino, probabilities are not fixed and are unknown. The probabilities of the match outcomes must be determined through models such as the one we developed in this project. A first ambitious but arrogant approach would be to think that we can construct a model that outperform those of the bookmakers. Arrogant indeed, because these firms hires full teams of data experts that have access to extensive data that track every details, from the injuries of the players to the weather forecast. Arrogant but also not necessary. This lead us to the next point.

Contrary to casinos, bookmaker must face the risk of an unbalanced accumulation of bets. Indeed contrary to casinos where only a limited number of player can place a bet on the same outcome, bookmakers must face an accumulation of bets that may start up to a week before the game. That represents thousands of bets. Then at the end of the match they must pay all winning bets at the same time. As a result in order to reduce the risk of incurring large losses when settling highly unbalanced betting accounts, bookmakers are sometimes ready to sacrifice

[7] There is 36 number in red, 36 number in black plus the zero that has no color.

some of their expected profits. Indeed by increasing odds on one outcome they can increase the demand of bets for this outcome. So the idea is to take advantage of the deviation[8] in odds due to the pressure of large irrational gambler crowds. We will also make use of the fact that these deviations will not be the same for all bookmakers and our ability to find arbitrage opportunity will highly depend on the number of different bookmakers odds we consider. But first let us define what we hear by an arbitrage opportunity.

An arbitrage opportunity is a situation where the expected profit is positive and the probability of losing money is zero. Assuming that the sum of the inverse odds for the three outcomes is smaller than one, we can show that we can always construct an arbitrage opportunity. Indeed, suppose that we bet a proportion w_A, w_D and w_H of our total bet value B, respectively on the outcomes A, D and H. The expected value of our bet is given by:

$$E[B] = B \sum_{i=\{A,D,H\}} w_i \pi_i P_i \tag{18}$$

If we set $w_i \pi_i = C$, where C is a constant it means that no matter the outcome we will win the same amount BC.

Using the fact that $\sum_{i=\{A,D,H\}} w_i = 1$, we can write C as a function of the π_i:

$$\begin{aligned} 1 = \sum_{i=\{A,D,H\}} w_i = \sum_{i=\{A,D,H\}} \frac{C}{\pi_i} = C \sum_{i=\{A,D,H\}} \frac{1}{\pi_i} \\ \Leftrightarrow C = \frac{1}{\sum_{i=\{A,D,H\}} \frac{1}{\pi_i}} > 1 \quad \text{if} \quad \sum_{i=\{A,D,H\}} \frac{1}{\pi_i} < 1 \end{aligned}$$

In this case (18) becomes:

$$\begin{aligned} E[B] &= B \sum_{i=\{A,D,H\}} w_i \pi_i P_i = BC \sum_{i=\{A,D,H\}} P_i \\ &= BC > B \quad \text{if} \quad C > 1 \end{aligned}$$

Hence if $\sum_{i=\{A,D,H\}} \frac{1}{\pi_i} < 1$ this is an arbitrage opportunity since the expectation of the bet is a subrmartingale, and furthermore, no matter the outcomes we will win the same amount $BC > B$, it is therefore not possible to lose any money. This is the particularity of arbitrage opportunities, they present no risk[9], but still offer a positive return. This is also why they usually do not exist.

Now the question is, how often this arbitrage condition $C = \frac{1}{\sum_{i=\{A,D,H\}} \frac{1}{\pi_i}} > 1$ is fulfilled and with which value of C? Is it even ever fulfilled?

Our dataset contains a collection of betting odds from eight different bookmakers over 8824 football games between 2012 and 2018. Over these 8824 match, we detected 5397 arbitrage opportunities. In order to establish a coherent betting strategy, when multiple arbitrage occurred the same day[10], we considered only the highest arbitrage. This leave us with 872 arbitrage opportunities, distributed between 2012 and 2018 as shown in Fig.14. The simple return over

[8] Hopefully we look for odds for which the α in (17) is negative

[9] Their Sharp ratio is infinite $SR = \frac{R}{\sigma} = \frac{R}{0} = \infty$.

[10] Multiple match can be organised the same day.

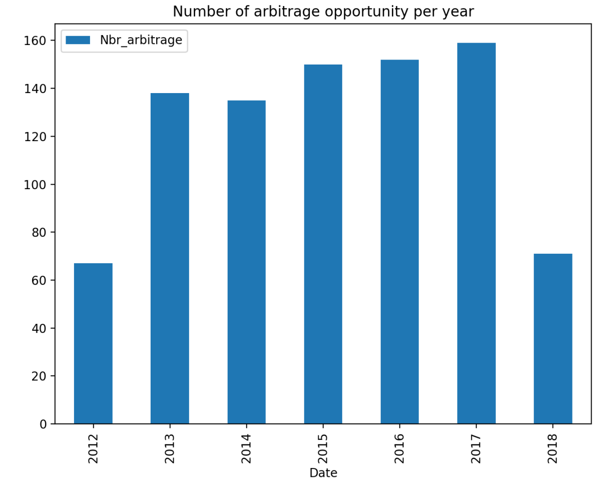


Figure 14: Distribution of the arbitrage opportunities over the period.

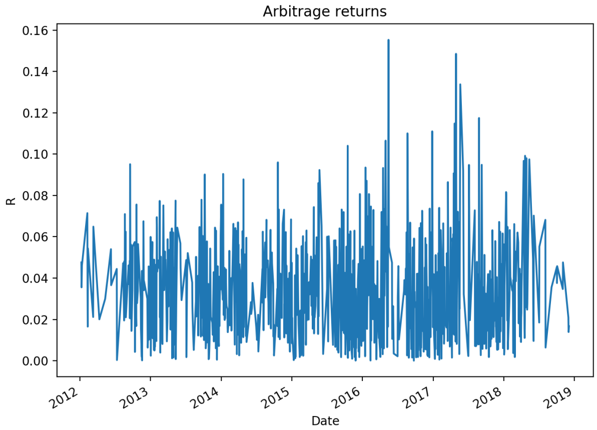


Figure 15: Return of each arbitrage.

a bet B is given by $R = \frac{BC-B}{B} = C - 1$. These are exposed in Fig.15, notice that they are all positive since they come from arbitrage opportunities. Supposing we re-invest our profits each time, our cumulative performance will be given by:

$$B_T = B_0 \prod_{t=0}^{T} (1 + R_t) \tag{19}$$

The cumulative return each year is plotted in Fig.16
Supposing we started with $B_0 = \$1$ in 2012, we would have end-up with a bit more than 2.5 trillion dollars at the end of 2018, as shown in Fig.17. Obviously this strategy is not realistic since it implies placing progressively larger and larger bet as the profits accumulates but it is hardly conceivable that a bookmaker let you place a bet of a billion USD for instance. Similar to fund capacity and liquidity pools (exchanges, dark pools, OTC desks) for exchange-traded financial assets, this betting strategy's capacity is finite, even if we spread bets across bookies.

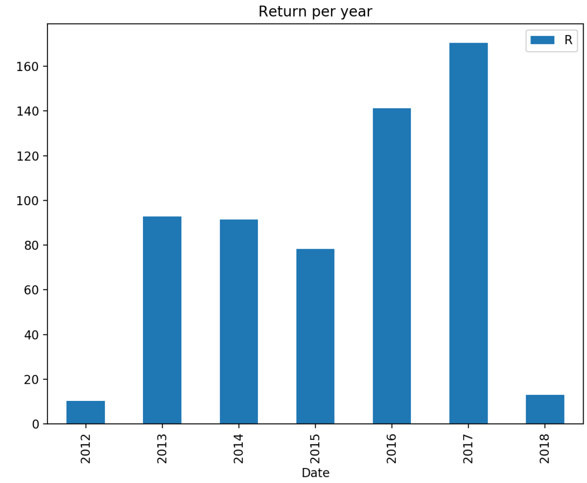


Figure 16: Cumulative return over each year (these are not given in percent).

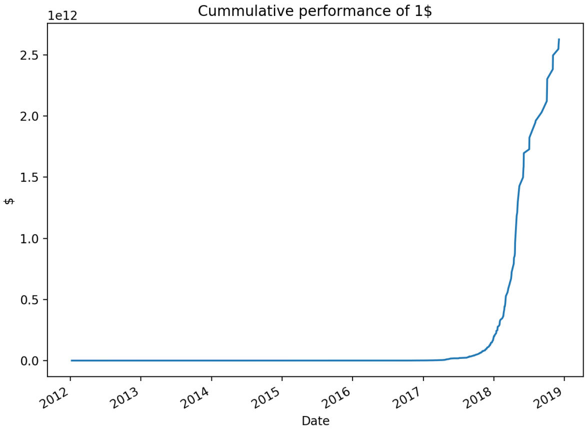


Figure 17: Cumulative performance of 1$

Nevertheless, having an idea of the betting limits it would be possible to construct a strategy where we start placing bets on the lower arbitrage opportunity[11] as well, once we would have reached the betting limit of the bookmaker.

3. Results

4. Limits

[11] Remember that we considered only one arbitrage opportunity per day as we assume that each time we place a bet B_t that correspond to all our money at this time.

Conclusion

We backtested machine learning-powered betting strategies to try and capitalize on the sports betting market, representing over \$ 100 million worth of bets in 2017 in the US. We conducted our analysis on predicting the outcome of football matches on a dataset of more than $8,000$ matches. We ran a total of ten models after data cleaning and feature engineering. Our performance measurements yielded different rankings of the best model, namely Logistic Regression for a F_1-based performance assessment, and Support Vector Machine for a Rank Probability Score-based performance assessment. We placed backtested bets based on the $FTTG_i$ feature prediction using the mean absolute error and mean squared error as performance metrics in the framework of a value betting strategy. None of the active nor passive strategies performed positive returns in the long run. At last, we backtested strategies focusing on arbitrage opportunities. Even if profitable on paper, a real-life implementation would be limited, for instance due to bet size limits. One could extend our reasoning and results across several other bookmakers, to increase the data input, as well as possibilities with respect to fund capacity.

Annex

Machine learning results

F1 score

	precision	recall	f1-score	support
A	0.444122	0.462236	0.452998	662.00000
D	0.286307	0.118966	0.168088	580.00000
H	0.511076	0.678571	0.583032	952.00000
accuracy	0.465360	0.465360	0.465360	0.46536
macro avg	0.413835	0.419924	0.401373	2194.00000
weighted avg	0.431454	0.465360	0.434103	2194.00000

Table 5: Classification report of clf_kNN

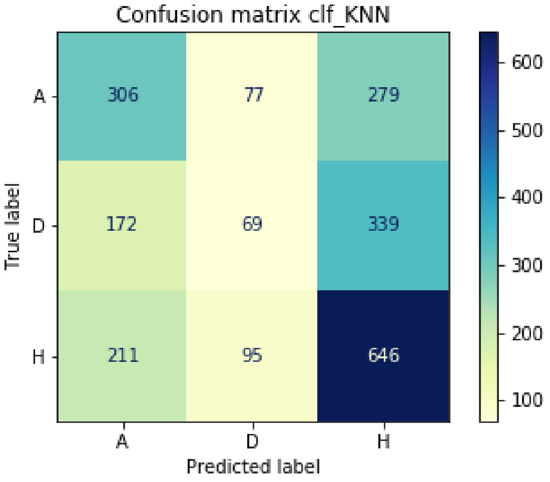


Figure 18: Confusion matrix for KNN

	precision	recall	f1-score	support
A	0.445693	0.539275	0.488038	662.000000
D	0.302100	0.322414	0.311927	580.000000
H	0.568475	0.462185	0.509849	952.000000
accuracy	0.448496	0.448496	0.448496	0.448496
macro avg	0.438756	0.441291	0.436605	2194.000000
weighted avg	0.461010	0.448496	0.450946	2194.000000

Table 6: Classification report of Support Vector Machine

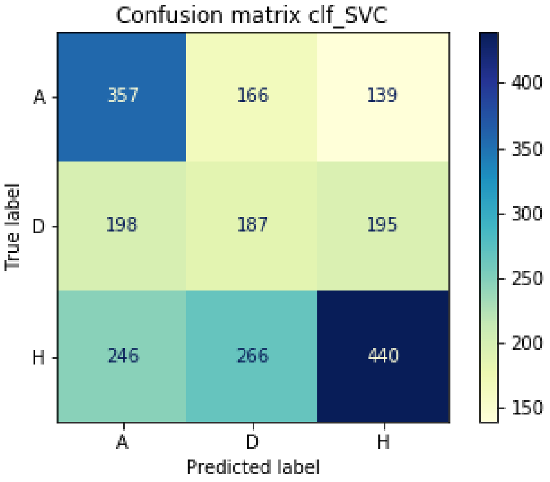


Figure 19: Confusion matrix for Support Vector Machine

	precision	recall	f1-score	support
A	0.443787	0.566465	0.497678	662.000000
D	0.310212	0.277586	0.292994	580.000000
H	0.560241	0.488445	0.521886	952.000000
accuracy	0.456244	0.456244	0.456244	0.456244
macro avg	0.438080	0.444166	0.437519	2194.000000
weighted avg	0.459006	0.456244	0.454072	2194.000000

Table 7: Classification report of Random Forest

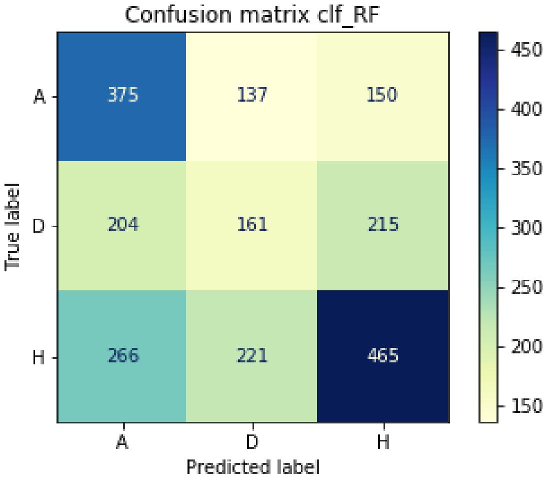


Figure 20: Confusion matrix for Random forest

	precision	recall	f1-score	support
A	0.438859	0.487915	0.462089	662.000000
D	0.318725	0.137931	0.192539	580.000000
H	0.522784	0.662815	0.584530	952.000000
accuracy	0.471285	0.471285	0.471285	0.471285
macro avg	0.426789	0.429554	0.413053	2194.000000
weighted avg	0.443516	0.471285	0.443960	2194.000000

Table 8: Classification report of Naive Bayes

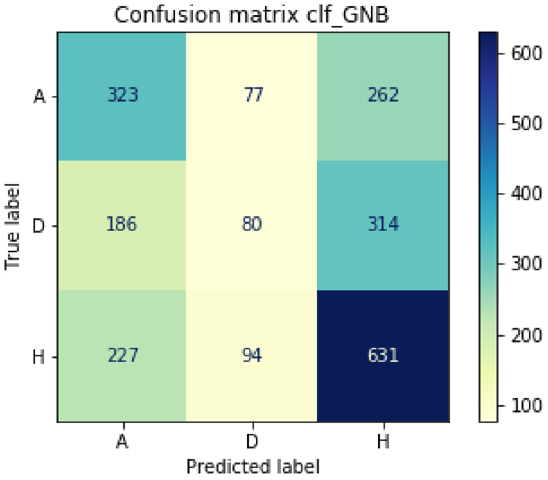


Figure 21: Confusion matrix for Naive Bayes

custom metric

	precision	recall	f1-score	support
A	0.464115	0.439577	0.451513	662.000000
D	0.353846	0.039655	0.071318	580.000000
H	0.498003	0.785714	0.609617	952.000000
accuracy	0.484047	0.484047	0.484047	0.484047
macro avg	0.438655	0.421649	0.377483	2194.000000
weighted avg	0.449669	0.484047	0.419609	2194.000000

Table 9: Classification report of clf_kNN for custom metric

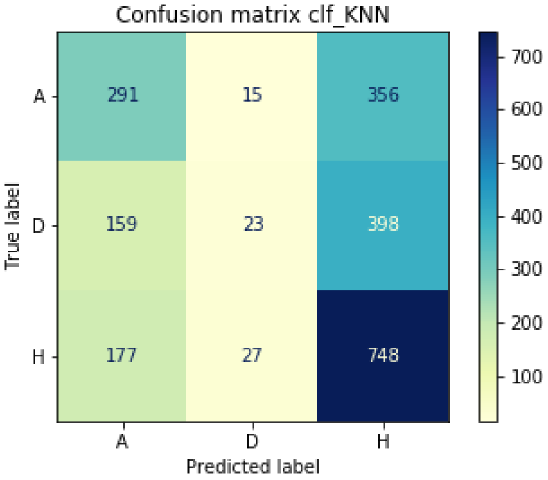


Figure 22: Confusion matrix for KNN

	precision	recall	f1-score	support
A	0.428058	0.539275	0.477273	662.0000
D	0.291242	0.246552	0.267040	580.0000
H	0.577675	0.527311	0.551345	952.0000
accuracy	0.456700	0.456700	0.456700	0.4567
macro avg	0.432325	0.437713	0.431886	2194.0000
weighted avg	0.456810	0.456700	0.453837	2194.0000

Table 10: Classification report of logistic regression for custom metric

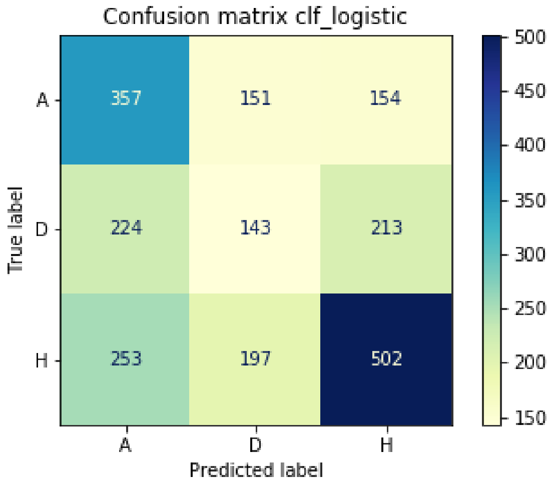


Figure 23: Confusion matrix for logistic

	precision	recall	f1-score	support
A	0.445559	0.469789	0.457353	662.000000
D	0.255172	0.063793	0.102069	580.000000
H	0.506292	0.718487	0.594008	952.000000
accuracy	0.470374	0.470374	0.470374	0.470374
macro avg	0.402341	0.417356	0.384477	2194.000000
weighted avg	0.421581	0.470374	0.422727	2194.000000

Table 11: Classification report of Random Forest

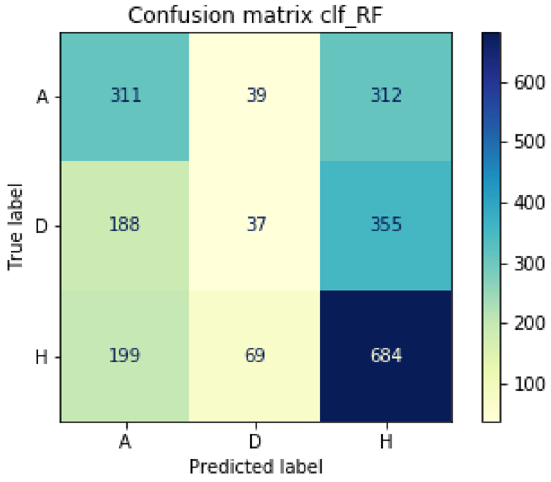


Figure 24: Confusion matrix for Random forest, custom metric

	precision	recall	f1-score	support
A	0.457778	0.466767	0.462229	662.000000
D	0.276680	0.120690	0.168067	580.000000
H	0.533175	0.709034	0.608656	952.000000
accuracy	0.480401	0.480401	0.480401	0.480401
macro avg	0.422544	0.432164	0.412984	2194.000000
weighted avg	0.442619	0.480401	0.448002	2194.000000

Table 12: Classification report of Naive Bayes custom metric

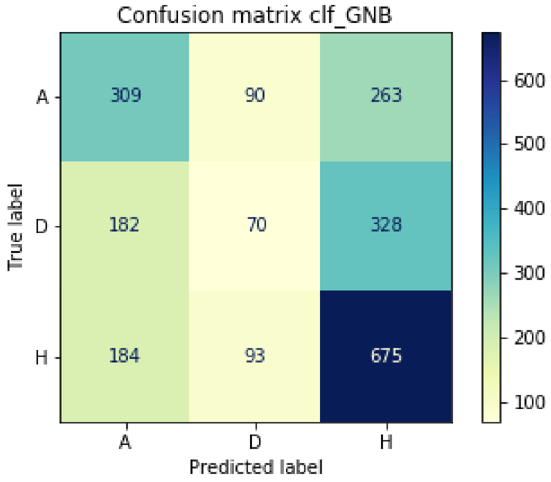


Figure 25: Confusion matrix for Naive Bayes custom metric

References

[1] Prof. Bruffaerts Christopher. EPFL 2020 class of Data Science in Practice

[2] Fama E. (1970) Efficient Capital Markets: A Review of Theory and Empirical Work, *The Journal of Finance*

[3] Graham B. (1949) The Intelligent Investor, *Harper*

[4] Constantinou A., Fenton N. (2012) Solving the Problem of Inadequate Scoring Rules for Assessing Probabilistic Football Forecast Models, *Journal of Quantitative Analysis in Sports*, 8(1).

[5] Kaunitz L., Zhong S., Kreiner J. (2017) Beating the bookies with their own numbers - and how the online sports betting market is rigged, *https://arxiv.org/abs/1710.02824*