

# An Empirical Model for Live Sports Betting Odds Calculation

Sebastian Krämer

1132361

s.kramer@warwick.ac.uk

February 1<sup>st</sup>, 2012

## 1. Abstract

Sports betting is a hobby for a lot of people and a main income for a few. During the last years, due to the growing population of broadband internet connections, online bookmakers began to operate and meanwhile the main part of the sports betting turnover worldwide is done online.

With the online bookmakers coming up, there was also a new way of betting introduced: live betting. Customers are now able to place bets while the actual game is still running, using the websites of the online bookmakers.

This development also forced bookmakers to develop new formulas for calculating their odds for the live events, considering new factors like the time that was played so far, the goals that were scored so far and others. However, almost all bookmakers copy their odds for football events of big Asian bookmakers and partly betting exchanges. There, odds are driven by single users betting high amounts. The details of why an odd is as high or low as it is in reality is far behind the scope of this paper, because there are other important factors like the possibility of arbitrage betting influencing the market. Nevertheless, it is fact that either if it's the bookmakers to set the odds or if it's the odds driven by single "high rollers", there is definitely someone calculating probabilities and therefore odds and making the market.

This project tries to model the calculation of live betting odds using the Empirical Modelling approach. Empirical modelling allows developers to create flexible models, using spread sheet like techniques enabling the user to change so called "observables" while the program is running. This is very useful for a model of sports betting odds calculation, because the main issue in calculating the odds and developing formulas for it is the influence of the single factors like goals scored so far, time played so far etc. and as well their weighting within the formulas. Flexible "observables" in this case enable the user to change these factors and their weighting "on-the-fly", helping them to possibly find "good", working formulas for the odds calculation.

In the project, there are only odds for the amounts of goals scored in total, e.g. over/under 2,5 goals. This is an Asian notation that avoids "draws" and therefore "cash backs" (e.g. when it would be over/under 2 goals, there is the possibility of exactly two goals scored, meaning there must be a third outcome, a "draw"). The notation is also quite common in European bookmakers. It was chosen to only calculate this odds and leaving e.g. the "who wins" market aside, as there are more than two outcomes in that market and finding a decent formula for calculating would be even harder. However, the formula used in this model have been developed especially for this model, there is no reference around as well as there is only very few common knowledge available for this kind of calculations, as an exact formula for this would be a very valuable asset, because it would enable its owner to beat the market.

## **2. The model**

### **2.1. Running the model**

The model is written in the scout, donald and eden notations. It consists of the files run.e, windows.s, proct.e and func.e.

Run.e is including the other files.

Windows.s is describing the windows that appear in the GUI as well as their attributes.

Proct.e holds all variables and the most of the functionalities, e.g. it catches mouse clicks and runs the actions triggered by them. In this file there are as well the formulas for the calculations of the odds stored.

Func.e holds additional functions, like increasing time and also a few mouse events.

The model was developed and tested using eden 1.73 under Windows XP 32bit.

After loading the model, the user can interact using the yellow buttons. Before starting the actual game, he can change the skill of the teams if he wants to. He also can already change the weighting of the four factors in the formula used for the calculation (time, goals, players, skills).

After starting the game, the “go ahead” button adds a random number of minutes between zero and five to the time played. The user can trigger goals and red cards whenever he likes during the game, using the designated yellow buttons. Whenever a goal is scored or a red card is given, there will also be the time played increased as described above. There can be only four red cards for every team, as football rules only allow a minimum number of seven players on the pitch. Moreover, the user can still amend the skill and weighting factors while the game is running.

The odds in the centre part of the screen will change regarding the settings done by the user. The odds are given in continental European decimal notation.

### **2.2. Weaknesses and possible improvements**

The model currently calculates quite realistic odds. However they are not perfect, especially in extreme situations (e.g. two goals scored after five minutes) the odds are sometimes too low or too high. Also the difference between the markets (e.g. over/under 1,5 and 2,5) is sometimes too low. Nevertheless, finding the perfect balance of factors was not and cannot be the objective of the project. Instead the focus was on building a framework that demonstrates how odds are changing during a live football game and that enables the user to experiment with the factors in order to possibly find the optimal balance between them.

There are a lot of possible improvements that would be very interesting to implement. First of all, including more factors (injuries, weather, historical performance statistics of the teams or popularity of the teams playing).

A relatively easy thing to implement would be the bookmaker margin. It is not included yet, so the odds shown are “fair” odds, overall returning all stakes placed.

The probably most interesting but as well most complicated thing to implement would be an input stream of real live odds. For example, there could be an xml feed of the odds of the betting exchange “Betfair” implemented. Combined with statistical data of the particular game that takes place, it could be possible compare the odds calculated by the model with the real world odds and therefore to use the model to find patterns and dependencies in the market movement, and finally maybe find a formula that really represents the real world live odds calculation or even beat the market. However, this really is beyond the scope of what was possible to implement considering the time that was given.

### **3. Conclusion**

#### **3.1. Relation to Empirical Modelling**

The project was very interesting. After having certain doubts on the sense of building the model in EDEN in the beginning and struggling with the notations for quite some time, I finally found the empirical modelling approach very useful for models like this one. It was highly practical to be able to change the variables while running the model, it made it possible to build a formula step by step (which nevertheless remained complicated) and see how the change of different factors or amendments in the formula affect the actual odds. The EDEN framework is probably not the best choice for straight-forward programs, where developers know what they are going to build before they start the coding, however it appears to be a great choice for building models with a lot of dependencies and observables, models in which the core calculations yet has to be elaborated and build, like this one. I can also imagine that it is be a great framework for building mathematical models, as they naturally have a lot of variables and formulas.

I also really appreciated the way the GUI is built, even if I didn’t use the “best practise”. I realized too late that even the GUI and its windows can have dependent variables, instead I defined all windows with static points.

On the negative side, there is insufficient documentation. The documentation given is only very small and it was obviously written by the developers themselves, resulting in a documentation that is hard to understand for people who start using Empirical Modelling from scratch. This is only referring to Scout, Donald and Eden, as this were the only notations used in the project. It would be great to have an extended documentation with lots of examples, maybe a possible future project for someone taking the Empirical Modelling module.

#### **3.2. References**

There were almost no references used. The model was built without copying any existing code or adopting a model from past projects. Those models were only used to figure out how the notations work, in particular the models “waterLim2006” and “roomvieweryung1991”.

The formula for calculating the odds was developed for this model. The knowledge about the betting market and the odds calculation is my personal knowledge learnt over years.