Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

Annales UMCS Informatica AI XI, 1 (2011) 7–13
DOI: 10.2478/v10065-011-0027-2

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

A strategy in sports betting with the nearest neighbours search and genetic algorithms

Damian Borycki[1]*

[1] *Jagiellonian University, Faculty of Physics, Astronomy and Applied Computer Science, Reymonta 4, 30-059 Kraków, Poland*

Abstract – The point of sports betting is not merely to correctly predict the outcome of a game, but to actually win on a bet. We propose a model of sports betting that uses the nearest neighbours search and genetic algorithms to do the job. It uses data on the teams playing, their respective formations, individual players, results of previous games, as well as odds offered by bookmakers. The model has been trained using the data from the seasons 2002/03 until 2008/09 of the English Premier League and tested against the already played games of the seasons 2009/10 and 2010/11.

1 Introduction

Sports betting is becoming more and more popular, as evidenced by an increasing number of registered players in the online bookmakers. Interest in this area is also growing in the scientific community. The literature on forecasting results of sporting events is vast.

Of all sports, football enjoys the greatest interest in Europe. Many papers on predicting results in this discipline have been published, see for example [**1, 2, 3**]. [**3**] presents a model that uses the offensive and defensive strengths of the teams participating in the game to predict the result, with the data on previous games providing the input. The model presented in [**2**] uses fuzzy logic optimized by a genetic algorithm and a neural network. The results of the last five games of each team and two direct matches between them, to a total of 12 games, serve as the input. The model presented in [**1**] is based on the rule-based reasoning and Bayesian networks. In addition to the historical data on previous games, it also uses expert knowledge to construct the priors.

*damianborycki@gmail.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

All these models focus on correctly predicting the result of a game, without paying attention to the financial outcome of a bet. This may be misleading since the payoff of a string of "obviously winning" bets can be easily offset by a couple of unfortunate ones.

The model presented in this paper uses much more data than just the historical results. In contrast to **[3]** it does not use any expert knowledge or subjective information. The principal novelty of the present approach is that it tries to optimize the actual profits from betting, not just correctly predict results of a series of games.

2 Data

Analyzing the chances of Team A winning against Team B, it is important to take into account as many factors that may influence the result as possible. Analyzing the results of previous games is not enough. For example, players change teams and somebody who played for Team A in 2005 may be playing for Team B in 2010. For this reason, the analysis should include information on individual players. It is also important to consider which players actually participate in a given game, whether in the starting lineup or on the bench, with a stress on the starting lineup. Random and unexpected weaknesses of various formations also provide valuable information, as well as the current disposition of the whole team, as judged from the results of a set of games immediately preceding the current one, results of direct matches between both teams etc. Humans consider all sorts of such information before placing a bet, and a betting algorithm should do the same.

In the model presented here, each game is represented by 134 parameters. To test the model, all games from the seasons 2002/03 until 2009/10 of the English Premier League, or the total of 3040 games, have been analyzed.

3 The model

The basic strategy employed in our model is the nearest neighbours search [**4**]. The probability of winning a bet is calculated on the basis of how many "similar" bets, found by the KNN search, have been won. The objective is not to predict the result of a game, but to optimize the profits from the actual bet.

Bets

The bets that our model admits are:

- t_1 –home win,
- t_x – tie,
- t_2 – away win,
- t_{over} – total goals above 2.5,
- t_{under} – total goals under 2.5.

The model consists of four modules:

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

Forecasting

This module calculates the probability of winning each of the bets considered. We use the following notation:

$\overline{m} = [a_1, 2_2, \ldots, a_{134}]$ is the game, the result of which we try to predict. $a_{1\ldots134}$ are the explanatory variables

$M = \{m : m = [a_1, 2_2, \ldots, a_{134}, y]$, is the set of all matches completed before. y is the result of the match

M_i is the set of i^{th} attributes of matches from the set M

$n_i = \max(M_i) - \min(M_i)$

w_i – the weighti^{th} attribute (to be discussed in Section 3.4)

d_m – the distance of m from $\overline{m}$

$$d_m = \sum_{i=1}^{134} \frac{|a_i^m - a_i^{\overline{m}}|}{n_i} \tag{1}$$

d_m serves as a metric in the KNN algorithm, with the size of the neighbourhood, k, set at 15.

Decision-making

After calculating the probability of each of the five types $(t_1, t_x, t_2, t_{\text{over}}, t_{\text{under}})$ this module decides whether a particular type of bet should be placed or not. This decision depends on whether the type is worth the risk. Here we use the following notation:

forecast_t – probability of winning the bet t resulting from the forecasting

bookieOdd_t – odds on bet t set by the bookmaker

forecastOdd_t – odds on bet t resulting from the forecasting

$\text{forecastOdd}_t = \frac{1}{\text{forecast}_t}$

Under the assumption that we have correctly predicted the result of the game, the bet t is worth the risk if

$$\text{bookieOdd}_t \geq \text{forecastOdd}_t$$

However, as the forecast might not be perfect, a bet is placed only if

$$\text{bookieOdd}_t \geq 1.5 * \text{forecastOdd}_t$$

Assessment

This module is designed to evaluate the performance of the algorithm at a fixed time interval. To this end each match with the time interval is subjected to the typing. Then the suggested types are compared with the actual results. On this basis the values of *yield* and *maxloss* are determined. We use the following notation:

T – the set of types with a fixed time interval

$T = [t_1, t_2, \ldots]$, $\text{time}(t_i) \leq \text{time}(t_{i+1})$

$\#T$ – the number of types in the set T

$$\text{win}(t) = \begin{cases} \text{bookieOdd}_t - \text{If the type was successful,} & t \in T \\ 0 & otherwise \end{cases}$$

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

$\text{win}(T) = \sum_{i=1}^{\#T} \text{win}(t_i)$

$\text{yield}(T) = \frac{\text{win}(T) - \#T}{\#T}$

maxloss(T) – length of the longest substring of zeros in the sequence win(t_1), win$(t_2), \ldots,$win$(t_{\#T})$.

In the end, the algorithm is assessed with

$$f(T) = \left(1 + \frac{\text{yield}(T)}{\text{maxloss}(T) + 1}\right)^{\#T}.$$

For example, let us

- invest 100 units
- $\#T = 20$
- maxloss$(T) = 9$
- yield$(T) = 0.2$.

Because maxloss > 0, it follows we cannot invest all available capital in each type because we will eventually lose. We should not invest $1/9$ of the capital, either, because keeping stakes after each loss we may also go bankrupt. We invest $1/10$ of the available capital in each type. The yield shows that, on average, we gain 20% of the funds invested. If we substitute these to the formula above, we can see that after investing in 20 types, we gain, under the assumptions specified, $0.02^{20} \approx 1.49$ of our initial investment. Our net gain is therefore 49 units.

Optimization

Optimization is crucial for a good performance of the algorithm presented here. The point of the optimization is an appropriate selection of data used to calculate the distance. The optimization is based on a genetic algorithm [**5**].

Coding: A chromosome consists of 134 genes. Each gene of the chromosome bears the information on the explanatory variable it represents, and its weight, w, set to either 0 or 1. The weight $w = 0$ means that this particular explanatory variable is not used in calculating the distance.

Selection was based on the roulette method.

Crossing: Crossing occurs with the probability of 70% and if so it does on a randomly chosen site.

Mutation: Weights can switch randomly from 0 to 1 and vice versa with the probability 0.1%.

The fitness function is the same $f(T)$ that was used in the Assessment module.

The size of each generation was taken to equal 10 and there were 45 generations.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

4 Results

Finding the optimal strategy

The model was tested three times against the results of the English Premier League, the seasons 2007/08 and 2008/09.

Fig. 1 shows the average fitness of each of the 45 generations in each of the three tests. Note the logarithmic scale of the vertical axis. One can see that starting with 20 generations, the tests bring the returns of the whole generation. One can also see how quickly the model improves. Fig. 2 shows the fitness of the best chromosome in each generation. A strategy that gives profits has been found already after the 4^{th} generation of the genetic algorithm. It can be also seen that the algorithm gets stuck in a local optimum. The fitness of the best strategy identified during the test equals 843.

Verification

Very promising results obtained during the test have been verified against the games of the season 2009/10, for 7 best strategies only. These results are presented in Table 1. The best strategies identified during the tests were actually the worst, but some strategies still brought profits. It appears that in-sample training, or optimization of strategies to the results of the seasons 2007/08 and 2008/09 performed poorly out-of-sample, as some external conditions might have changed.

The four strategies that brought profits for the season 2009/10 have been further verified against the games of the 2010/11 season that have been completed until January 1, 2011. The results are presented in Table 2. These four strategies also profit in the 2010/11 season.

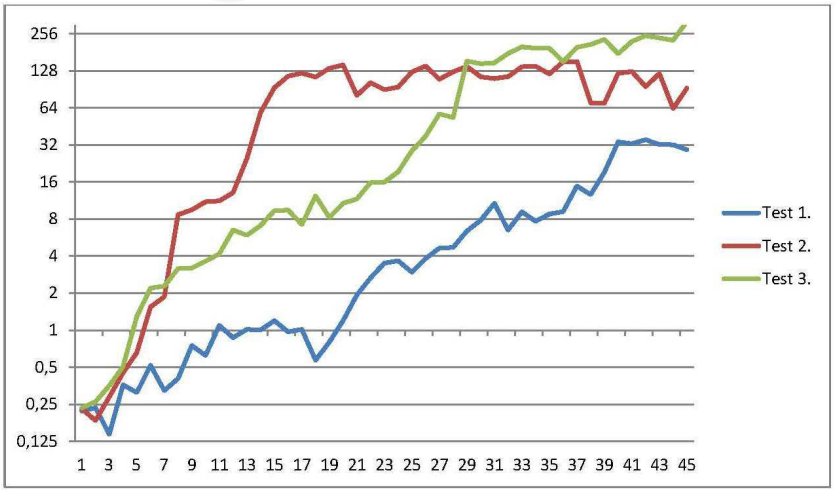


Fig. 1. The average value of the match in the generation.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

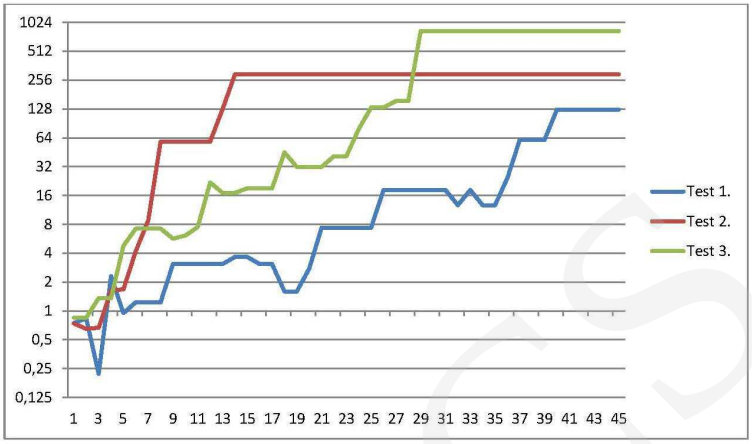


Fig. 2. The highest value of matches in the generation.

Table 1. Verification the 2009/2010 season.

Fitness 07/08, 08/09	Fitness 09/10	yield	wins	loses	maxloss
833.96	0.09	-0.19	51	134	14
438.38	0.66	-0.02	63	132	11
417.72	4.27	0.09	54	132	11
396.42	3.35	0.07	52	130	10
349.96	0.53	-0.04	55	127	13
309.54	1.74	0.03	52	131	10
292.55	3.34	0.08	55	125	12

Table 2. Verification season 2010/2011.

Fitness 07/08, 08/09	Fitness 10/11	yield	wins	loses	maxloss
417.72	2.88	0.14	37	59	12
396.42	2.44	0.11	36	60	11
309.54	2.70	0.15	38	59	14
292.55	3.65	0.19	40	60	14

5 Conclusions

An original, fully automatic algorithm for investing in sports betting has been presented. Unlike most previous studies on similar subjects, the presented model uses a

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 11/11/2026 23:18:52

lot more publicly available information on football matches. It also focuses not merely on correctly predicting results of games, but rather on profiting from actual bets. If a bet is deemed not to be profitable, it is not taken, and the gambler does not sustain losses, at the price of abstaining from minute winnings. Our results show that, contrary to the popular opinion, a profitable strategy in investing in sporting bets is possible.

Several possibilities of improving our model appear. For example, introducing the non-Euclidean metric, separate strategies for different types of bets and changing the weights from 0-1 to fractional values can help identify even more profitable strategy. Several programming issues need to be solved in order to run the appropriate software faster. Eventually we want to be able to use our model in different leagues and in sports other than football.

References

[1] Min B., Kim J., Choe C., Eom H., McKay R. I., A Compound Framework for Sports Prediction: The Case Study of Football, Knowledge-Based Systems, 21(7) (2008): 551.

[2] Rotshtein A. P., Posner M., Rakityanskaya A. B., Football Predictions Based on a Fuzzy Model with Genetic and Neural Tuning, Cybernetics and Systems Analysis 41(4) (2005): 619.

[3] Rue H., Salvesen O., Prediction and retrospective analysis of soccer matches in a league, The Statistician 49(3) (2000): 399.

[4] Larose D. T., Discovering Knowledge in Data. An Introduction to DATA MINING, Chapter 5.

[5] Larose D. T., Data mining methods and models, Chapter 6.