**Can I predict the outcome of a football match (and make money)?**

Experiments in sport analytics using KNIME Analytics Platform

[[Hans Samson](https://medium.com/@hsams?source=post_page-----ec9996d3092f--------------------------------)](https://medium.com/@hsams?source=post_page-----ec9996d3092f--------------------------------)

[[Low Code for Data Science](https://medium.com/low-code-for-advanced-data-science?source=post_page-----ec9996d3092f--------------------------------)](https://medium.com/low-code-for-advanced-data-science?source=post_page-----ec9996d3092f--------------------------------)

[Hans Samson](https://medium.com/@hsams?source=post_page-----ec9996d3092f--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F39f33a086d51&operation=register&redirect=https%3A%2F%2Fmedium.com%2Flow-code-for-advanced-data-science%2Fcan-i-predict-the-outcome-of-a-football-match-and-make-money-ec9996d3092f&user=Hans+Samson&userId=39f33a086d51&source=post_page-39f33a086d51----ec9996d3092f---------------------post_header-----------)

Published in

[Low Code for Data Science](https://medium.com/low-code-for-advanced-data-science?source=post_page-----ec9996d3092f--------------------------------)

·

11 min read

·

Jun 26

307

3



Photo by [Chaos Soccer Gear](https://unsplash.com/@chaossoccergear?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral)

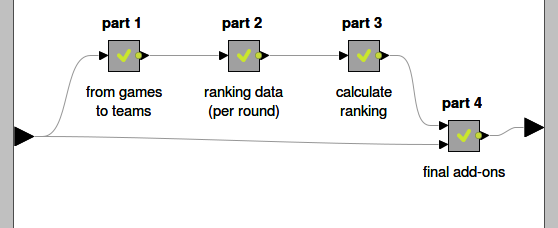
To enhance my skills in applying data science concepts and tools, I create small projects for myself, emphasizing a hands-on learning approach. One such project was the development of a football prediction model, which evolved into an accumulation of several smaller projects.

For all these projects, I utilize [KNIME](https://www.knime.com/knime-analytics-platform?utm_source=medium&utm_medium=organic&utm_term=&utm_content=&utm_campaign=com_journal), a low-code, no-code data science platform. KNIME offers a combination of standard KNIME nodes by default and the ability to incorporate code (Python/R/Java) when necessary, making it my go-to solution. Additionally, [KNIME Analytics Platform](https://www.knime.com/knime-analytics-platform?utm_source=medium&utm_medium=organic&utm_term=&utm_content=&utm_campaign=com_journal) is available at no cost. You can download it [for free here](https://www.knime.com/downloads?utm_source=medium&utm_medium=organic&utm_term=&utm_content=&utm_campaign=com_journal).

Among these projects, the one that provided me with the most enjoyment (although I wish it brought me money) involves predicting the outcome of football matches without the need for coding.

**How it got started**

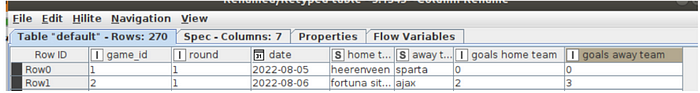
It all began with my exploration of Tableau. I started creating various visualizations centered on matches in the Dutch competition (Eredivisie), which can be found at my [Tableau profile](https://public.tableau.com/app/profile/hanssamson/viz/Eredivisie2018-2019/behaaldepunten). However, prior to visualizing the data, I developed a KNIME workflow for data preparation.



Data preparation steps.

The primary purpose of the workflow is to modify the match results after each round of the competition, enabling the calculation of team rankings. This process facilitates the determination of team standings within the competition.

So from 1 row per game (the result of 306 matches in one season) to 1 row per team (the place in the ranking), with a set of new columns (per team, per round, per season, resulting in 612 records in total).





From 1 row per game (top) to 1 row per team (botom).

**Predicting the outcome of a football match**

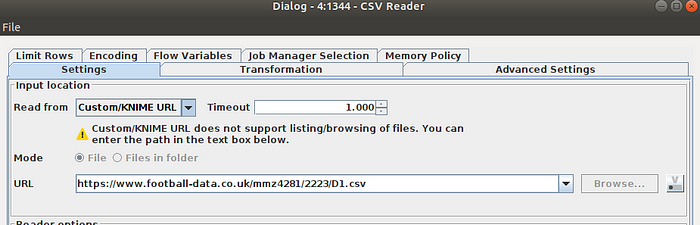
After a few seasons, I accumulated a substantial dataset, leading me to ponder over the following questions:

1. Can I predict the outcome of a match (win/loss/draw) by utilizing the information from previous matches of both teams and other relevant factors? In other words, can I create a formula where the result of the match (win/loss/draw) is determined by a function that considers home team characteristics, away team characteristics, common characteristics, and date-related characteristics?
2. If I can develop a prediction model using this dataset, does the model outperform bookmakers? And (or?) is it possible to generate profits with my model?

With these questions in mind, my project progressed to the next phase. It is important to note that achieving the described result was not a linear process. Similar to any other data science project, it involved a cycle of trial, reflection, adaptation, and improvement — facilitated so well by KNIME.

**Input data set**

To train the model, I required additional data, as unexpected outcomes can occur in football. Therefore, I incorporated not only the results of matches from the Dutch competition (Eredivisie), but also results from the past seven seasons of competitions in Germany (Bundesliga), France (Ligue 1), Spain (Primera Division), England (Premier League), and Italy (Serie A). Useful datasets were obtained from [football-data.co.uk](https://www.football-data.co.uk/), and with the KNIME CSV Reader node, the data was effortlessly loaded.



Screenshot from KNIME CSV Reader selecting data straight from the website.

By merging all of this data with the workflow described above, a dataset with a size of approximately 13,000 matches was created.

**Feature creation, engineering and selection**

The next step involved deriving features from the input dataset. It was crucial to ensure that only features based on historical data preceding the match were included. These features should not contain any future-oriented information influenced by the outcome of the match being predicted.

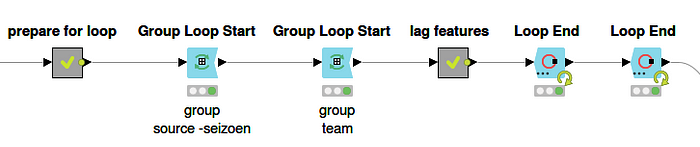
Additionally, I decided to create features only from round 5 onwards in the competition. This choice was made because some features required looking back a certain number of matches, such as the number of goals scored in the last 3 matches. Moreover, the team rankings, which served as a feature in the model, may not be “realistic” at the beginning of the season, as you may have noticed from the initial visualizations in this blog.

With these considerations in mind, I generated approximately 80 features from the dataset. I also explored other potential features to enrich the dataset, including:

* Team characteristics from the FIFA PlayStation game
* Team value and transfer value
* Mix of nationalities in the squad
* Information on injured or suspended players
* Results of previous matches in other competitions (e.g., Champions League, FA Cup)
* Tenure of the current team manager/coach in terms of months/years

However, incorporating this additional data posed a significant challenge, as it would require manual work to gather and include it for all matches across various competitions and years. Despite conducting experiments with some of these features, such as the FIFA game data, which demonstrated promising predictive power, I ultimately decided to maintain the input dataset’s consistency, ensuring its quality. Nonetheless, I encourage you to explore these features and try them yourself!

To create these features, I made extensive use of the KNIME Lag Column nodes within a Group Loop block.



Feature creations.

The next step in the process involved combining the team-level features into match-level features. The match result (win/loss/draw) was then determined as a function of the home team features, away team features, common features, and features associated with the date of the match.

To enhance the feature set, I created additional features by comparing the values of certain features between the home team and the away team. For instance, calculating the difference in points and the difference in position in the ranking. Here is a selection of the features I created:

* Points ahead in the ranking
* Probability of a win if the previous match was lost
* Goal difference
* Points collected at home/away
* Series of goals scored/conceded
* Day of the week

To improve the prediction accuracy and reduce computation time, I performed feature selection. Ultimately, around 30 features were selected as inputs for training the models.

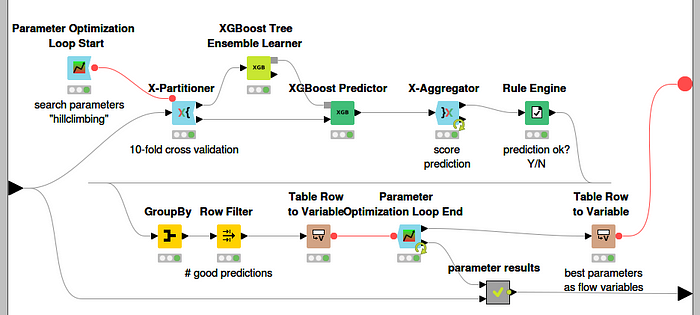
**Classification Algorithm**

The prediction of the outcome of a football match (win/loss/draw) is a multi-class classification problem. The goal is to make predictions as to whether the match will end in a win, loss or draw, based on a set of features. This differs from a regression problem, where predictions are made about the number of goals scored or conceded in a match. In this case, the focus is on determining the match result (win/loss/draw). For my prediction models, I ended up with two models: a Keras Neural Network solution and an XGBoost model.

**XGBoost**

Initially, I created my first model using the Random Forest Learner node. During the initial stages, I didn’t pay much attention to the configuration of this node. My primary objective was to determine if it was possible to train a model with sufficient predictive power. This experiment not only confirmed that it was indeed possible to train a reliable model but also inspired me to derive new features from the dataset.

Once I had a more finalized set of features, I shifted my focus to improving the model by selecting and fine-tuning the algorithms. Eventually, I switched to the XGBoost algorithm, a popular choice for its performance. KNIME played a crucial role in helping me optimize the configuration and parameter settings of the Learner node, allowing me to fine-tune the model effectively with the Parameter Optimization Loop nodes.

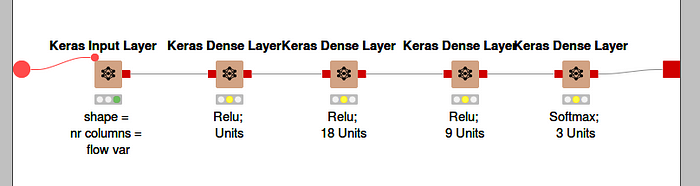


XGBoost with parameter optimization.

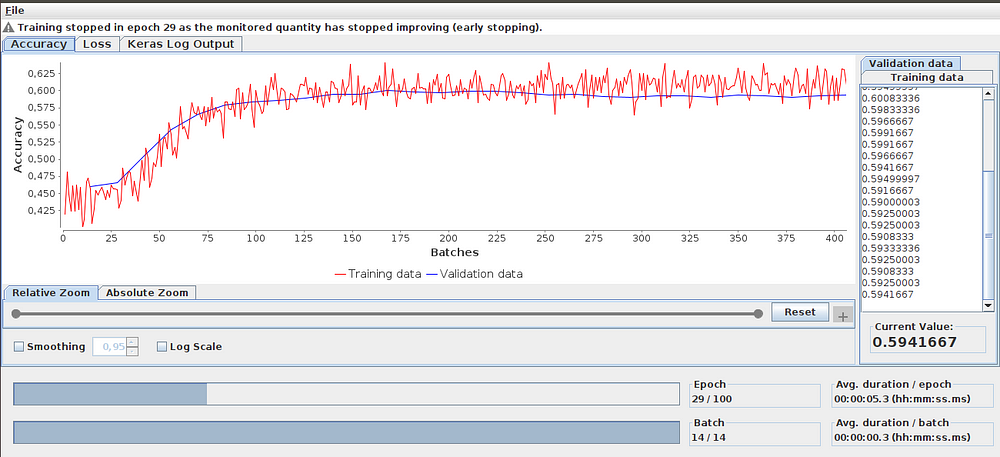
**Keras**

Initially, my plan did not include using another algorithm alongside XGBoost. However, the release of the book “[Codeless Deep Learning with KNIME](https://www.knime.com/blog/codeless-deep-learning-with-knime?utm_source=medium&utm_medium=organic&utm_term=&utm_content=&utm_campaign=com_journal)” by Kathrin Melcher & Rosaria Silipo, along with the multiple workflows on this topic on the KNIME Community Hub, presented a valuable opportunity for me to delve into training and scoring models created with a Keras Neural Network.

Before training the Keras model, certain adjustments had to be made to the features. This included handling missing values, normalizing the data, and designing the architecture of the neural network, such as determining the number of layers and neurons. Additionally, configuring the Keras Learner node required some iterations to find the optimal settings for achieving satisfactory results. One of the aspects I found particularly beneficial about the Keras Learner was the insights provided by the Learning Monitor, which provided a valuable perspective on the training process.



Neural network definition using the nodes of the Keras Integration in KNIME.



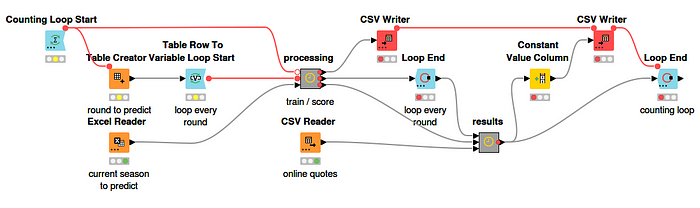
Learning Monitor View from the Keras Network Learner.

By leveraging the resources provided by the book and the workflows on the KNIME Community Hub, I was able to effectively utilize Keras Neural Networks within KNIME and further enhance the predictive capabilities of my models.

**How well do the models predict?**

To assess the effectiveness of predicting the outcome of football matches based on the set of features, I set up a test scenario. The models were trained on a dataset football matches from various seasons and competitions. I then used these models to predict the outcomes of matches in the Eredivisie for the 2022–2023 season, which were not included in the training dataset.

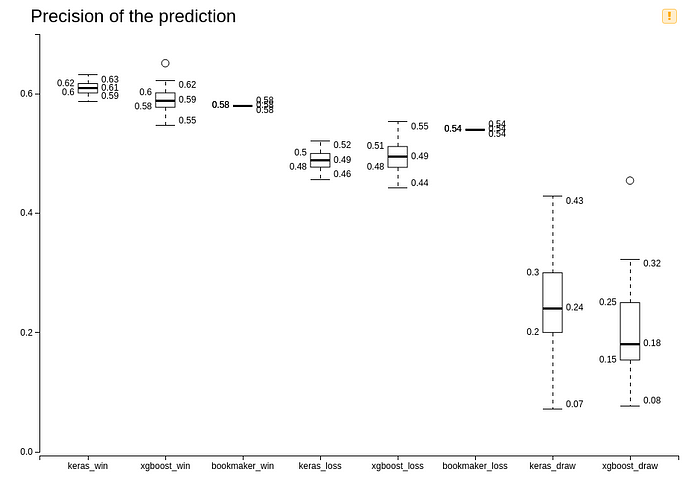
To evaluate the performance of the models, I analyzed the overall accuracy and precision of the predictions. During the training of a model, the dataset was divided into a training and test set. This partitioning is random. To get a good understanding of the models’ performance, I trained each model automatically using KNIME Loops 50 times (each time with a random assignment to the training and test set).



Training the model and scoring every match in the Eredivise (2022–2023) 50 times.

It turned out that the Keras model achieved on average a slightly higher accuracy of 55%, while the XGBoost model achieved an accuracy of 52%. To put these results into perspective, I also compared them to the performance of a specific bookmaker (e.g., Bet365). The bookmaker demonstrated slightly better accuracy, with a rate of 57%.

Based on these findings, it can be concluded that while the models showed some predictive ability, the bookmaker’s performance was slightly better in terms of accuracy. However, it’s important to note that accuracy alone may not provide a complete picture of the models’ performance. Therefore, I also examined precision. Precision is a metric that indicates how often the model’s prediction for each individual outcome (win/loss/draw) is correct.



Precision of the predictions.

It appears that the model’s predictions for a home team victory have on average a higher precision of 59% (XG Boost model) and 61% (Keras model) compared to the bookmaker’s overall precision of 58%. This suggests that when the models predict a home team victory, they are more likely to be correct compared to the bookmaker’s predictions.

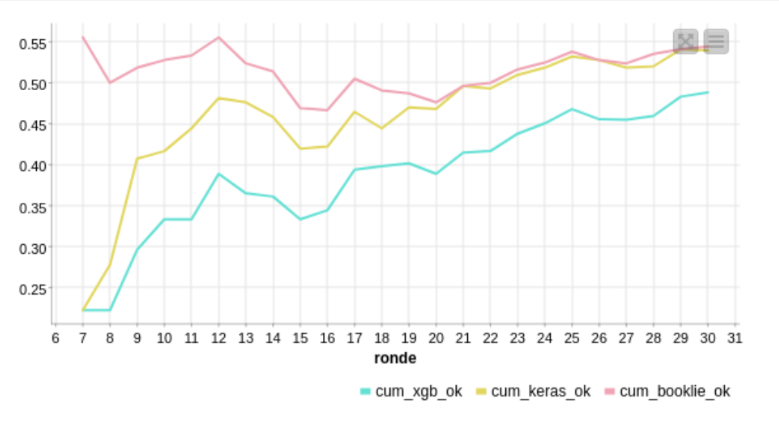
Additionally, it’s worth noting that the bookmakers do not make predictions for matches to end in a draw. This could impact the overall accuracy comparison, as the models take into account the possibility of a draw in their predictions. But you also see that for both models, it is quite challenging to make an accurate prediction for a draw. There is a low percentage correctly predicted, and there is also a large variation around this low percentage.

Considering these results, it seems that the models have demonstrated some promising predictive power, particularly in predicting home team victories.

**(Overall) accuracy over time**

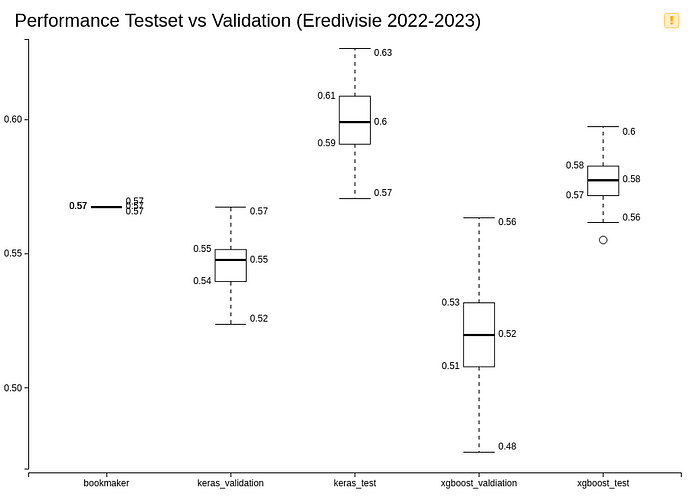
Upon analyzing the prediction results per round played, it appears that there is a trend indicating improved prediction accuracy as the season progresses. It is evident that the number of correctly predicted matches (with a maximum of 9 games per round) fluctuates, indicating some variability in prediction performance (see the figure on the left). However, the figure on the right reveals that the average cumulative number of correctly predicted matches increases steadily over the course of the season.





Left: correctly predict matches per round. Right: moving average percentage predicted correctly.

Although I am quite satisfied with the way the models are predicting real-life matches, the XGBoost and Keras models do not achieve the same level of accuracy as they did when evaluated on an unseen validation set. It is noticeable that the average accuracy of both the test sets is slightly higher than the average score of the bookmaker. However, the realized accuracy on the unseen set of matches from the Eredivisie of the 2022–2023 season is slightly lower again.

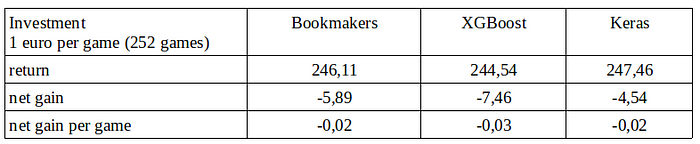


Performance test set vs validation set (Eredivisie matches 2022–2023).

There is still a noticeable difference, but there could be several explanations for these disparities. The dataset on which the models are trained consists of matches from six different competitions and multiple seasons. The question is whether this dataset accurately represents the football matches I'm trying to predict. I believe that is not the case, and there is room for improvement in constructing a training dataset.

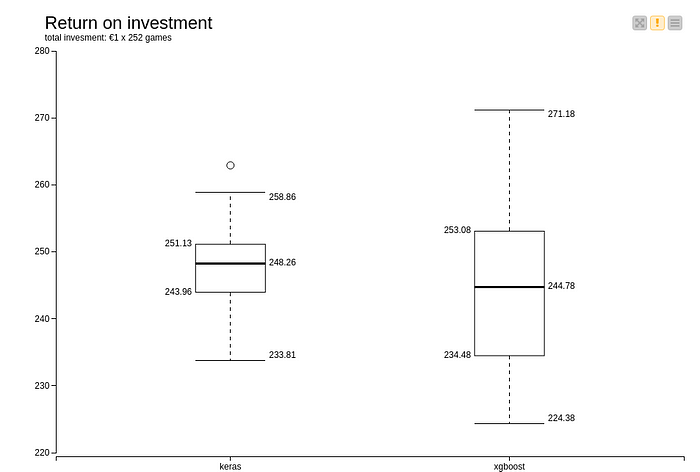
**Making money?**

As demonstrated, it is possible to predict the outcome of a football match to some extent. However, the major question remains: can you actually make money from it? The answer depends on various factors. For instance, if I had placed a 1 euro bet on each match, none of my models would on average resulted in a profit. But blindly following the lowest quote provided by the bookmakers also does not prove to be profitable.



Making money?

The data from the above table was obtained by training and scoring the models 50 times. Therefore, there is a variation around the average. This variation is slightly larger for the XGBoost model compared to the Keras model. With the Keras model, you have less risk of losing a significant amount of money but also a lower chance of a substantial profit.



Return on investment.

Betting on all matches played in a competition round solely based on the model outcome does not seem to be a viable strategy. To make money from betting on football matches, further analysis is required to gain insights into which types of matches offer a higher level of certainty and or profit in terms of prediction and result.

**Epilogue, "Close but no cigar"**

In general, the quality of the predictions could still be improved. That’s true. However, what I really appreciate is the fact that using KNIME has made it possible to predict the outcome of a football game quite easily. The journey of creating this football match prediction model, like any predictive modeling activity, is not a straightforward path to the finish line. Through experimentation and testing, I have gained valuable knowledge in model training, scoring, and utilizing KNIME. Multiple versions of the models have been developed, and I believe this current version will not be the last. As demonstrated in this post, there is always room for improvement, whether it’s through additional data, more refined features, better parameter estimation, exploring alternative algorithms or selecting specific games to bet on. Perhaps, in the end, I will be able to outperform the bookmakers?