Tabular Foundation Models for xG: Can TabPFN Score Without Training?

Timo Martens (✉), Pieter Robberechts, and Jesse Davis

KU Leuven, Dept. of Computer Science; Leuven.AI, B-3000 Leuven, Belgium
firstname.lastname@kuleuven.be

Abstract. Expected Goals (xG) is a widely used metric in soccer analytics to provide insight into team and player performance by evaluating chance quality. At its core, xG estimation is a binary classification task. It operates on tabular data: each shot event is represented by a structured feature vector describing the shot and preceding actions, and the target is the binary outcome of whether the shot results in a goal. For this type of tabular prediction tasks, gradient-boosted tree ensembles such as XGBoost are the dominant approach. A key limitation is that they require retraining when deployed across new leagues and typically demand large training sets to converge. This becomes especially problematic when limited historical data is available. Recently, pretrained tabular models such as TabPFN have emerged as an alternative, offering fast inference and strong performance in low-data regimes. However, their behavior in real-world applications such as xG modeling remains underexplored. In this paper, we present a systematic empirical study of TabPFN on this task, comparing it against XGBoost and logistic regression across data regimes, location on the pitch and distribution shift. We find that TabPFN is highly data-efficient, achieving strong performance with substantially fewer training examples. We further observe that TabPFN is more robust to out-of-distribution settings, consistent with its smooth inductive bias.

1 Introduction

Expected Goals (xG) is one of the most widely adopted metrics in soccer analytics [9]. It quantifies the quality of a scoring opportunity as the probability that a shot results in a goal. xG models support scouting, betting, and performance evaluation [3].

To model xG from event data, the standard approach is to convert each shot into a tabular representation using hand-crafted features describing the shot's context. Gradient-boosted tree ensembles (e.g., XGBoost [1], LightGBM [7], CatBoost [12]) have long been state-of-the-art for this task [13], consistent with their strong performance on tabular data more broadly [4]. Despite their strong predictive performance, they have two key practical limitations. First, gradient-boosted tree ensembles must be retrained whenever they are applied to a new context, such as a different league or season. This requires running the full optimization

procedure again, which can be time-consuming in practice. Second, predictive performance and calibration typically improve with access to large collections of historical shots, which may not be available for lower-tier or youth competitions.

Recently, pretrained transformer-based models such as TabPFN [6] have emerged as a competitive alternative to gradient boosted tree ensembles on tabular learning benchmarks. However, evidence for their effectiveness on real-world sports analytics tasks remains limited. Hence, it is unclear whether the improvements observed on benchmark datasets translate to expected goals modeling.

In this paper, we present the first empirical study of TabPFN for expected goals modeling. Our results show that TabPFN improves predictive performance compared to gradient boosted tree ensembles, driven mainly by more discriminative probability estimates. Moreover, it is highly data-efficient, matching XGBoost with far less data. Additionally, TabPFN is more robust to out-of-distribution scenarios (e.g., changes in the spatial distribution of shots). These findings suggest that pretrained tabular models offer a promising alternative for data-scarce soccer analytics settings.

2 Background and Notation

We extract tabular data from an event stream format. This tabular data takes the form $D = \{(x_i, y_i)\}_{i=1}^{N}$, with x being a feature vector consisting of both continuous and categorical features. The binary label y is either 1, indicating a direct goal, or 0, indicating a failed goal attempt. xG is then defined as the probability of a shot x_t resulting in a goal, $P(y = 1|x_t, D)$.

2.1 TabPFN: Tabular Prior-data Fitted Network

TabPFN [6] is a foundation model for tabular data. Although technically quite complex, the core idea is simple: instead of training a new model from scratch for every dataset, a large pretrained transformer is taught the general rules of tabular supervised learning. This pretrained model is subsequently used to make predictions on new datasets by taking the training data as context, functioning in the exact same way as a prompt in a Large Language Model (LLM).

More technically, a transformer model is trained on synthetic datasets generated from a space of hypotheses Φ. Each hypothesis $\phi \in \Phi$ is a data generating mechanism from which data instances can be sampled. At inference time, given a test instance $\mathbf{x}_t$ and train set D, TabPFN makes a probabilistic prediction as

$$P(y = 1|x_t, D) \propto \int_{\Phi} \underset{(1)}{P(y = 1|x_t, \phi)} \underset{(2)}{P(D, \phi)} \underset{(3)}{P(\phi)} d\phi,$$

where (1) is the probability that hypothesis ϕ predicts the instance to be positive, (2) is the likelihood of D being generated by hypothesis ϕ, and (3) is the prior distribution over the hypothesis space Φ. Hence, training (i.e., fitting to the task at hand) and prediction happen in one forward pass. As a result, the pretrained model can easily be applied to multiple contexts without retraining.

3 Methodology

Models. We compare TabPFN version 3[1] to two standard approaches for modeling xG: (1) a logistic regression model [11] using a basic feature set, containing the shot's distance, goal angle and body part used; and (2) a fine-tuned XGBoost model [1], using advanced features [14] such as the location of the opposing keeper and defenders, the assist, etc. This model performs on par with the best commercial models. We use the same extended feature set for TabPFN. More details on the exact features and parameter tuning procedure can be found in Appendix B and C respectively.

Data. All models are trained on open-play shots (i.e., penalties and set-pieces are removed) from the 2023/24 season across 10 European leagues (English Premier League, Spanish LaLiga, German Bundesliga, Italian Seria A, French Ligue 1, Dutch Eredivisie, Belgian Pro League, Portuguese Liga NOS, Austrian Bundesliga and the Greek Super League) and evaluated on the corresponding 2024/25 season. This results in 49,124 training shots and 47,377 test shots.

Evaluation Metrics. We evaluate predictive performance using the Brier score, a proper scoring rule for probabilistic predictions. Given an evaluation set of M shots, the Brier score is calculated as $\frac{1}{M}\sum_{i=1}^{M}(\mathrm{xG}_i - y_i)^2$. It captures both calibration (how well predicted probabilities (i.e., xG) match empirical scoring rates) and discrimination (the model's ability to assign higher xG to goals than to non-goals). The Brier score ranges from 0 (i.e., perfect predictions) to 1.

To put the Brier score in context we will also evaluate calibration and discrimination separately by reporting respectively Expected Calibration Error (ECE) (10 equal-count bins) and Area Under the ROC (AUROC). The former reveals potential over- /underconfidence of a model, while the latter represents the probability that a randomly selected goal and non-goal are ranked correctly according to their xG values.

4 Experimental Results

In this section, we perform a systematic evaluation of TabPFN, analyzing model behavior across data availability, location on the pitch, and distribution shift.

4.1 TabPFN Outperforms the State-of-the-Art

Table 1 shows the performance metrics achieved by the different models considering the full train and test sets. TabPFN achieves the lowest Brier score among the three models, indicating improved overall probabilistic performance over the current state-of-the-art models for xG. To confirm that these results are not due to sampling noise, we perform a bootstrap test (1,000 resamples) at the match

[1] A brief discussion on the difference with v2.5 can be found in Appendix A.

level (i.e., all shots in one match are sampled together to account for within match level correlation). Table 1 provides the resulting confidence intervals on the performance metrics. For every pair of methods, we also compare the difference in Brier score on each resample. These paired differences are statistically significant (two-sided $p < 0.001$).

Table 1. The Brier score, Expected Calibration Error (ECE) and AUROC calculated over the full test set (i.e., shots from the 2024-2025 season over ten European leagues) for Logistic Regression (LR), XGBoost and TabPFN. Brackets denote 95% confidence intervals from a bootstrap test (1,000 resamples), resampled by match to account for within-match correlation between shots. **Bold** values indicate a statistically significant improvement over the next-best model (two-sided $p < 0.001$, computed as twice the proportion of resamples in which the worse-performing method beats the better-performing method).

	Brier score (↓)	ECE (↓)	AUROC (↑)
LR	0.0833 [0.0812, 0.0852]	0.0099 [0.0074, 0.0123]	0.7398 [0.7325, 0.7469]
XGBoost	0.0770 [0.0751, 0.0789]	0.0042 [0.0024, 0.0062]	0.7899 [0.7834, 0.7964]
TabPFN	**0.0763** [0.0744, 0.0782]	0.0041 [0.0023, 0.0061]	**0.7953** [0.7882, 0.8019]

The Brier score alone does not reveal which aspect of the model has improved, since it is jointly influenced by both calibration and how well the predictions discriminate between shots with different outcome probabilities. To better understand this improvement, we look at Murphy's calibration-resolution-uncertainty[2] decomposition [10] of the Brier score for XGBoost and TabPFN. Splitting the predictions into 10 equal-count bins, we find that the difference in calibration is negligible ($\Delta < 0.00001$) and that the difference in resolution ($\Delta = 0.0006$) accounts for the difference in Brier score. A better resolution indicates that TabPFN discriminates better between goals and non-goals. The uncertainty component of the decomposition is irrelevant here as it is model-independent. These results align with the AUROC and ECE results from Table 1. We find that TabPFN achieves the largest AUROC ($p < 0.001$) and is not significantly worse than XGBoost in terms of ECE ($p = 0.926$).

Moreover, Figure 1 shows the predicted probability (i.e., xG value) distributions for TabPFN and XGBoost. TabPFN concentrates more mass in the $[0, 0.1]$ range than XGBoost, which improves resolution. Since most of these shots are not converted, this leads to smaller squared errors, which lowers the Brier score.

[2] Brier = calibration - resolution + uncertainty

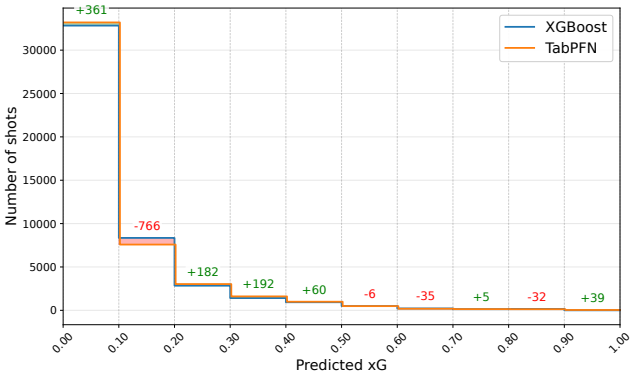


Fig. 1. Histogram of predicted xG values for XGBoost and TabPFN. Relative to XGBoost, TabPFN shifts probability mass toward lower values, moving predictions from the $[0.1, 0.2]$ range into the $[0, 0.1]$ range.

Figure 2 zooms in on the predictive performance in the $[0, 0.2]$ interval and confirms this effect. TabPFN assigns more shots to the $[0, 0.1]$ range, which slightly increases error there. This is more than compensated by a larger error reduction in the $[0.1, 0.2]$ range, resulting in an overall improvement vs. XGBoost.

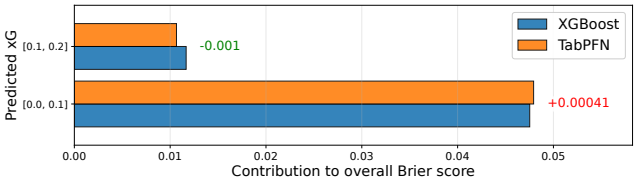


Fig. 2. Contribution to the Brier score by predicted probability bin (higher values indicate worse performance), comparing TabPFN and XGBoost.

Next, we investigate whether this advantage is concentrated in specific areas of the pitch. Figure 3 divides the offensive half of the pitch into eight playing zones and shows the Brier score and ECE for each zone. TabPFN achieves a lower Brier score in almost all zones, except for the right wing (Zone 7) and the midfield third (Zone 8). Both zones contain relatively few shots (i.e., 40 and 408). The largest contributions to the overall score come from central areas inside the penalty box (excluding the goal area) (Zone 3) and central midfield (Zone 5), as these contain the highest number of shots. In Zone 3, the Brier score improvement is partly driven by better calibration. Contrarily, in Zone 5, the Brier score improves despite a higher calibration error, indicating that the gain comes from increased resolution. This finding is consistent with our earlier observations:

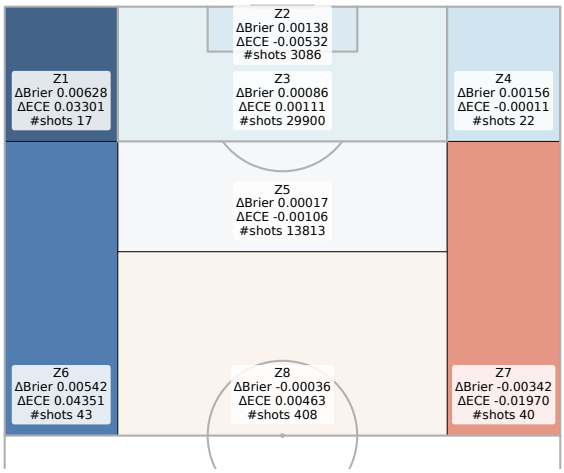


Fig. 3. Performance metrics compared across eight attacking zones (XGBoost - TabPFN). A positive delta indicates an improvement (i.e., a lower metric score) of TabPFN over XGBoost. Darker blue (red) regions indicate zones where TabPFN provides the largest relative improvement (decrease) in Brier score over XGBoost.

Zone 5 contains many low-xG shots, and TabPFN systematically pushes these predictions closer to zero, producing more discriminative probability estimates.

Our hypothesis is that this difference stems from how the two models partition the feature space. Because XGBoost groups data into discrete axis-aligned hyperrectangles, its predictions in low-data-density regions are often bound to a "floor"—the average historical conversion rate of the shots within that partition. With common constraints like a minimum number of samples per leaf node, XGBoost effectively averages over many shots in sparse areas, making it less discerning. TabPFN, in contrast, can learn completely smooth, continuous decision boundaries. This makes the probabilities smoothly go down towards zero for highly improbable shots, rather than getting stuck at a conservative average.

A potential practical consequence is that TabPFN will suffer less from "shot-volume inflation." A team that takes 20 terrible, low-hope shots can artificially rack up 1.0 to 1.5 cumulative xG, potentially skewing post-match analysis. Because TabPFN is more fine-grained in low-probability zones, it effectively filters out some of this noise, giving analysts a truer reflection of match dominance.

4.2 TabPFN is Data-Efficient

An important practical question in soccer analytics is: how much data do we actually need to build a good model? In many real-world settings—such as lower leagues, youth competitions, or (unfortunately) women's soccer—historical shot data is limited. This makes data-efficient models particularly valuable.

Like other foundation-style models, TabPFN is designed to perform well in low-data regimes. To evaluate this property, we compare the learning curves of TabPFN, XGBoost, and logistic regression as the number of training examples increases (or, in the case of TabPFN, as more shots are provided as in-context examples). For each training set size, we draw 10 independent random samples from the full training pool and report the resulting mean and standard deviation.

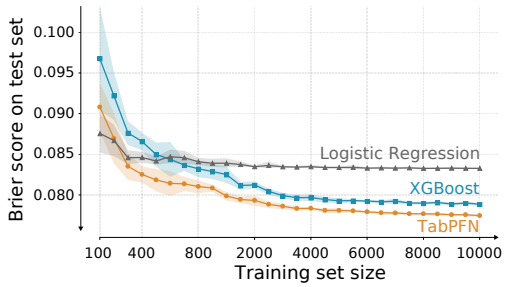


Fig. 4. The Brier score for all open-play shots in the 2024/25 seasons as a function of the number of training shots sampled from the 2023/24 seasons. For each sample size, we randomly sample training sets and repeat this procedure 10 times, reporting mean ± standard deviation.

From 100 shots onwards, TabPFN consistently achieves the lowest Brier score across all three model types. While the performance gap to XGBoost narrows as more data becomes available, we do not observe any regime in which XGBoost clearly outperforms TabPFN in terms of probabilistic accuracy. Yet, TabPFN still does not appear fully saturated within the largest training sizes considered, suggesting that further performance gains may still be possible with additional data. However, it should be noted that increasing the amount of data also negatively affects inference time and that TabPFNv3's reported context size is strictly limited to x rows and y features, where $x \times y \leq 2$ million.

In general, more complex models only become beneficial once sufficient data is available [4]. However, our results show that this transition occurs at substantially different data scales depending on the model class. Figure 4 illustrates how TabPFN achieves performance comparable to a converged logistic regression model with as few as 400 shots (approximately 25 games), whereas XGBoost requires substantially more data to reach a similar level.

4.3 Where do TabPFN and XGBoost Disagree?

Figure 5 shows the predictions for all test shots, with on the x-axis the XGBoost predictions and on the y-axis the TabPFN predictions. While most shots' predictions are well-aligned, some strong disagreements exist.

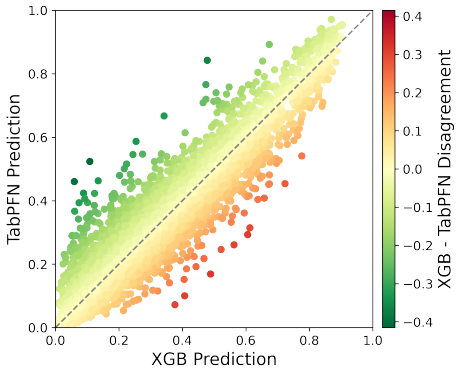


Fig. 5. Predicted xG values for XGBoost and TabPFN.

To understand where the models clash, we train a shallow decision tree (depth 4, minimum 50 samples per leaf) to classify whether a shot will result in a large disagreement (defined as the top 1% of prediction disagreements). The classifier achieved a 5-fold cross-validated ROC-AUC of 0.893 ± 0.010 (in-sample: 0.939), and the same four features dominated every fold in the same rank order: (1) whether the goalkeeper is positioned within the shot triangle (i.e., the cone between the shot location and the two goalposts), (2) the distance from the shooter's first ball contact to the shot's location, (3) whether the shot was a one-touch strike, and (4) number of defenders within the shot triangle.

Analyzing the number of large disagreement shots that end up in each leaf, 61.8% of them end up in only two leafs, which are explained by the following decision rules. (1) One-touch shots in which the player also covered some (small) distance with the ball beforehand and a well-positioned goalkeeper. Clearly, this is a logical contradiction: by definition carrying the ball first means you cannot take a one-touch shot. Investigating the raw event data and video, we conclude that these are simply one-touch shots extracted with noisy location data. (2) Shots with a mispositioned goalkeeper and a relatively small occlusion of the goal (i.e., less than 57% of the goalkeeper is within the shot triangle[3] and less than 56% of the goal face is occluded). Table 2 shows the number of total and large disagreement shots that fall in each leaf; and the resulting precision for each leaf. Moreover, the results indicate that shots contained in leaf one (two) have a mean disagreement that is 6.3 (3.7) times larger than the mean disagreement on the full test set (i.e., 0.0165).

Interestingly, the disagreement on shots differs qualitatively between the two leaves. Table 3 shows that on the first leaf's shots, TabPFN predicts a larger xG for 72% of the test shots. This is reflected in the larger mean prediction by TabPFN. Even though the decision rule leading to leaf one cannot exist in practice, we believe that TabPFN makes the correct prediction and not XGBoost.

[3] The goalkeeper is modeled as having an average wingspan of 1m60.

Table 2. Two decision rules explain most of the large disagreements (defined as the top 1% disagreements) between XGBoost's and TabPFN's predictions: (1) one-touch shots in which the player also covered some (small) distance with the ball beforehand and a well-positioned goalkeeper, and (2) shots with a mispositioned goalkeeper and relatively small occlusion of the goal (i.e., less than 57% of the goalkeeper is within the shot triangle and less than 56% of the goal face is occluded). Precision indicates the percentage of large disagreement shots within all test shots following the decision rule. Mean Δ lift denotes how much larger the mean disagreement of the shots that follow the decision rule is, compared to the average in the full test set (i.e., 0.0165).

Pattern	n (% of full test set)	Tail shots (% of tail)	Precision	Mean Δ lift
(1)	368 (0.8%)	140 (29.5%)	38.0%	6.3×
(2)	1,258 (2.7%)	153 (32.3%)	12.2%	3.7×
Combined	1,626 (3.4%)	293 (61.8%)	–	–

The scoring rate for shots that exhibit this noise in the features is significantly larger than the scoring rate for all one-touch shots (both in the training and test split). Hence, we suspect a latent variable exists that accounts for both the noise in the features and the increased scoring rate. Compared to TabPFN, XGBoost simply generalizes over this latent effect towards the scoring rate of all one-touch shots (i.e., 0.122). On the second leaf's shots, neither of the models' mean predictions deviate significantly from the true scoring rate.

Table 3. Direction of the disagreement on the test data for the two main patterns: (1) one-touch shots in which the player also covered some (small) distance with the ball beforehand and a well-positioned goalkeeper, and (2) shots with a mispositioned goalkeeper and relatively small occlusion of the goal (i.e., less than 57% of the goalkeeper is within the shot triangle and less than 56% of the goal face is occluded). XGBoost/TabPFN columns show mean predicted probability; the observed scoring rate is the actual empirical outcome rate for shots following that pattern. Additionally, we indicate the scoring rate of shots in which TabPFN predicts a larger xG than XGBoost.

Pattern	Mean XGBoost	Mean TabPFN	Scoring rate	TabPFN xG ↑
(1)	0.166	0.226	0.231	72%
(2)	0.463	0.457	0.460	46%
Overall test set	0.102	0.102	0.102	–

From a practical standpoint, this is relevant in the context of xG modeling, where rare situations—such as unusual shot angles or atypical assists—do occur in practice. Even if these cases are infrequent, large prediction errors in such moments can strongly affect how analysts and practitioners perceive the reliability of a model, and may ultimately reduce trust in its outputs.

4.4 TabPFN Predicts the Base Rate if It Lacks Training Data

While the previous paragraph looked at instances individually, we now investigate what happens to the models' predictions when entire regions of the feature space are systematically underrepresented.

To investigate this, we construct an out-of-distribution setting by restricting training (in-context) data to shots within 20 meters of the goal and evaluating on shots beyond 20 meters. This induces a clear spatial distribution shift, allowing us to test generalization beyond the training regime.

Table 4 shows in-distribution and out-of-distribution performance. Within training distribution, TabPFN performs slightly better than XGBoost across all metrics. Out-of-distribution, both models decrease their Brier score while TabPFN keeps a slight edge over XGBoost. However, that result is mainly driven by a lower scoring rate for shots taken beyond 20m. Hence, we compare the Brier Skill Score[4] (BSS) instead, which corrects for this by normalizing against a naive baseline: predicting the test set's scoring rate for every shot. TabPFN achieves a score close to 0, indicating similar performance to the naive baseline. XGBoost has a negative BSS, indicating that it completely collapses under a spatial shift. This is reflected by their AUROC and ECE scores: XGBoost's AUROC of 0.55 is close to random predictions (0.5) while TabPFN achieves 0.64.

Table 4. In-distribution (ID) and out-of-distribution (OOD) predictive performance.

	Brier score (↓)	BSS (↑)	ECE (↓)	AUROC (↑)
TabPFN (ID)	0.085904	0.165181	0.002261	0.787454
XGBoost (ID)	0.088462	0.140328	0.014024	0.769745
TabPFN (OOD)	0.029755	0.005249	0.006836	0.638646
XGBoost (OOD)	0.032260	-0.078473	0.026226	0.547897

From the decision surfaces of the models (fitted on the full training set) in Figure 6, one can see why this happens. Once XGBoost has to interpolate to a nearly empty region of the input space, it will predict something similar to the closest in-distribution instance, which, in the setting above would be biased towards a higher scoring rate. TabPFN, comparatively, regresses towards its learned prior in the out-of-distribution regions.

Interestingly, the same figures show a counter-intuitive pattern for both models: shots from wide positions receive higher xG values than central shots. We hypothesize that this is a data artifact and related to out-of-distribution prediction rather than being model-specific. Very few shots are taken from these wide, far positions on the pitch (i.e., the training data contains only 45 shots taken at more than 25m from the goal line and less than 10m from the sideline). Nonetheless, the pattern is different in both models. TabPFN regresses towards

[4] $\text{BSS} = 1 - \text{Brier}(\hat{y}_{\text{method}}, y)/\text{Brier}(\hat{y}_{\text{baseline}}, y)$

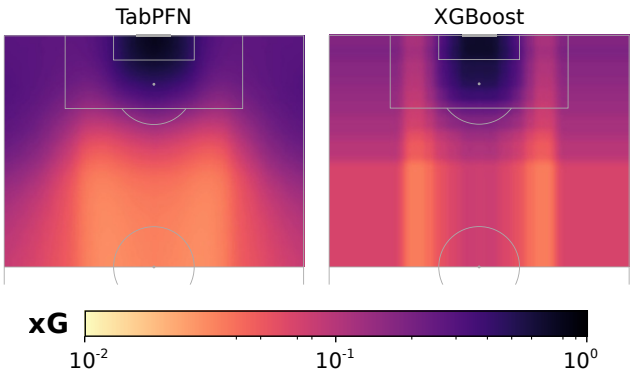


Fig. 6. The decision surfaces of TabPFN (left) and XGBoost (right) for expected goals. The surfaces are generated by sweeping the shot's location on the pitch, keeping the goalkeeper on their goal line and no defenders blocking the shot. The model counter-intuitively assigns wide shots a larger xG than central shots.

the base rate of goals in the training data ≈ 0.1, similar to a Bayesian inference approach. XGBoost, on the other hand, seems to overfit on the few samples in those regions that might be biased (e.g., a player will likely only shoot from those positions in case of a poorly-positioned goalkeeper).

These results suggest that the two models behave differently in regions of the feature space that are sparsely covered by the training data. XGBoost shows a larger deterioration in calibration in such regions, which can lead to unrealistic probability estimates in rare game situations. Although TabPFN degrades more gracefully, it barely outperforms a naive baseline (i.e., the BSS is close to 0).

5 Conclusion

Tabular foundation models offer a promising direction for sports analytics. A pre-trained TabPFN model outperforms a finely tuned XGBoost model and achieves comparable performance with only a fraction of the data. This enables accurate xG estimation even in low-data settings such as lower-tier leagues or specific tactical contexts. Its pretrained nature also removes the need for per-context hyperparameter tuning. Moreover, TabPFN regresses towards its learned prior when moving away from the data distribution. We further find that TabPFN and XGBoost's disagreements can mostly be attributed to two types of shots. Unfortunately, tabular foundation models still pose some practical drawbacks: TabPFN is currently limited to batch inference and does not yet scale to the largest tabular datasets. Though these remain active research directions.

Acknowledgments. This research is supported by the Flemish government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme (TM, PR & JD) and KU Leuven Research Fund (C14/24/091 to JD).

References

1. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
2. Devos, L., Perini, L., Meert, W., Davis, J.: Detecting evasion attacks in deployed tree ensembles. In: Proceedings of the 2023 Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 120–136 (2023)
3. Green, S.: Assessing the performance of premier league goalscorers. (2012), https://www.statsperform.com/ resource/assessing-the-performance-of-premier-league-goalscorers/
4. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? Advances in Neural Information Processing Systems **35**, 507–520 (2022)
5. Grinsztajn, L., Flöge, K., Key, O., Birkel, F., Jund, P., Roof, B., Manium, M., Hoo, S.B., Bühler, M., Garg, A., Safaric, D., Robertson, J., Jäger, B., Alessi, S., Hayler, A., Moroshan, V., Purucker, L., Singer, P., Arazi, A., Siems, J., Metzen, J.H., Grab, G., Erickson, N., Guo, S., Kalfon, E., Bing, S., Salinas, D., Cornu, C., Wehrhahn, L.C., Kriuchkova, D., Kaya, K., Sidhoum, L., Salmon, M., Chen, J., Hulsebos, M., LeCun, Y., Müller, S., Schölkopf, B., Gambhir, S., Hollmann, N., Hutter, F.: Tabpfn-3: Technical report (2026), https://arxiv.org/abs/2605.13986
6. Hollmann, N., Müller, S., Eggensperger, K., Hutter, F.: Tabpfn: A transformer that solves small tabular classification problems in a second. In: The 11th International Conference on Learning Representations (2023)
7. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems **30** (2017)
8. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceedings of the 8th IEEE International Conference on Data Mining. pp. 413–422 (2008)
9. Macdonald, B.: An expected goals model for evaluating nhl teams and players. In: Proceedings of MIT Sloan Sports Analytics Conference (2012)
10. Murphy, A.H.: A new vector partition of the probability score. Journal of Applied Meteorology and Climatology **12**(4), 595–600 (1973)
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
12. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems **31** (2018)
13. Robberechts, P., Davis, J.: How data availability affects the ability to learn good xg models. In: International Workshop on Machine Learning and Data Mining for Sports Analytics. pp. 17–27 (2020)
14. Vatvani, D.: Upgrading expected goals. (2022), https://blogarchive.statsbomb.com/articles/soccer/upgrading-expected-goals/

A Comparing TabPFN v2.5 and v3.

Initially, our study used v2.5 of TabPFN. In this section, we briefly discuss the main differences we observed in the analysis that we observed between the two versions. We found that v3 achieves improved Brier score, AUROC, and out-of-distribution performance, while showing slightly different calibration and prediction behavior compared to v2.5.

Predictive performance comparison TabPFNv3 achieves marginally improved Brier score (0.0763 vs 0.0766) and AUROC (0.7953 vs 0.7932) over v2.5, while the ECE remained similar. Moreover, increased resolution remains the main driving factor of improvement across versions. Looking at the different zones on the pitch (see Figure 3), v2.5 has slightly different prediction behavior in Zone 4, 7 and 8, which can all be attributed to sampling noise due to the small sample sets in those zones.

Model disagreement with XGBoost For v2.5 we were unable to fit a small model on the disagreement between TabPFN and XGBoost. However, we found that they mainly disagreed on outliers. IsolationForest [8] and OC-score [2] values indicated that the top 20 disagreements occurred on outlier instances (i.e., examples far from other instances in the training set). This indicates that TabPFNv3 has partly improved its Brier score by improving local calibration.

Out-of-Distribution is no longer TabPFN's Achilles heel When exposed to out-of-distribution data, TabPFNv2.5 would collapse completely, similar to XGBoost. However, with the updated v3, the model has included out-of-distrubution data priors [5], along with spatial data priors. This combination has improved TabPFN's robustness to out-of-distrubution data significantly, as described in Section 4.4.

B Feature sets

Our Logistic Regression model uses a set of basic features: the relative distance to the goal (Δx and Δy) and the body part used to make the shot.

XGBoost and TabPFN use a more advanced set of features, that includes the basic features and is extended with:

- *mov_a02*: The distance the ball covered since the previous two actions.
- *speed_a02*: The speed of the ball since the previous two actions.
- *from_counterattack*: A binary feature indicating whether the shot was taken on a counterattack.
- *time_prev_shot*: The time elapsed since the previous shot in that game.
- *under_pressure*: A binary feature indicating whether the shot was taken under pressure.

- *first_touch*: A binary feature indicating whether the shot was made on first touch.
- *impact_height*: The height at which the player struck the ball.
- *carry_dist*: The distance that the ball was carried during the action that lead to this shot.
- *goalkeeper_dist_to_goal*: The distance of the goalkeeper to the goal at the moment the shot is taken.
- *goal_face_occlusion_pct*: The percentage of the goal face that is occluded/blocked by other players on the pitch.
- *nb_defenders_in_shot_triangle*: The number of defenders that are within the shot triangle (i.e., the triangle made by connecting the ball to both posts of the goal).
- *goalkeeper_in_shot_triangle*: The percentage of the goalkeeper that is within the shot triangle (i.e., the triangle made by connecting the ball to both posts of the goal).
- *type_assist*: A categorical feature indicating the type of the assist leading to this shot.
- *shot_technique*: A categorical feature indicating the technique of the shot.

C XGBoost Hyperparameter Tuning

The hyperparameters of the XGBoost model were optimized in three sequential stages, as summarized in Table C.5. We employed a grid search over the specified ranges, evaluating each configuration using 5-fold cross-validation on the training set. At each stage, the optimal parameters were selected based on the highest average Brier score on the validation folds, while all parameters not under consideration were fixed to their previously selected values. This sequential grid search procedure yielded the optimal hyperparameter configuration reported in the last column of Table C.5. Parameters not explicitly mentioned in the table were left at their default value.

Table C.5. XGBoost hyperparameter search space across the three tuning stages, with final selected values.

Stage	Hyperparameter	Search space	Selected value
1. Tree structure	`learning_rate`	0.1 (fixed)	-
	`n_estimators`	200 (fixed)	-
	`max_depth`	$2, \ldots, 10$	5
	`min_child_weight`	$1, \ldots, 15$	11
	`gamma`	$[0, 1]$	0.78
2. Regularization and sampling	`subsample`	$[0.5, 1]$	1.0
	`colsample_bytree`	$[0.5, 1]$	0.793
	`colsample_bylevel`	$[0.5, 1]$	0.861
	`reg_alpha`	$[0, 1]$	0.8
	`reg_lambda`	$[0, 10]$	8.44
3. Number of estimators	`learning_rate`	0.01 (fixed)	0.01
	`n_estimators`	$500, 600, \ldots, 2000$	1700