How data availability affects the ability to learn good xG models

Pieter Robberechts[0000−0002−3734−0047] and Jesse Davis[0000−0002−3748−9263]

KU Leuven, Belgium
Department of Computer Science
{pieter.robberechts,jesse.davis}@cs.kuleuven.be

Abstract. Motivated by the fact that some shots are better than others, the expected goals (xG) metric attempts to quantify the quality of goal-scoring opportunities in soccer. The metric is becoming increasingly popular, making its way to TV analysts' desks. Yet, a vastly underexplored topic in the context of xG is how these models are affected by the data on which they are trained. In this paper, we explore several data-related questions that may affect the performance of an xG model. We showed that the amount of data needed to train an accurate xG model depends on the complexity of the learner and the number of features, with up to 5 seasons of data needed to train a complex gradient boosted trees model. Despite the style of play changing over time and varying between leagues, we did not find that using only recent data or league-specific models improves the accuracy significantly. Hence, if limited data is available, training models on less recent data or different leagues is a viable solution. Mixing data from multiple data sources should be avoided.

1 Introduction

The number of shot attempts is one of the most basic and frequently used statistics in soccer to summarize the performance of a team, where the team which accumulated the most attempts is often seen as the one that dominated the game offensively. However, the aggregate number of shots can be a misleading metric, since it does not consider the quality of the goal-scoring opportunities from which these shots arise. For example, a penalty is certainly more likely to result in a goal than a long range shot.

To overcome the pitfalls from simply using the number of shots (or even the number of shots on target), soccer analysts have introduced the notion of expected goals (xG) [8], where the main idea is to assign a quality metric on each shot. To do so, an expected-goals model assigns a value between zero and one to each shot which represents the probability that the shot will result in a goal. This is done by applying a machine learning classifier to a large historical data set of goal scoring opportunities. The classifier learns to distinguish big chances that are most likely to result in a goal from desperate shot attempts, based on features such as the shot's location, the body part used to kick the ball and the game situation (open-play, following a cross, free kick, etc.).

Allowing for a deeper and more insightful analysis of soccer players' shot attempts, xG has grown to be one of the most commonly used and best understood advanced metrics in soccer analytics. Yet, all existing research has focused on identifying the features that are indicative of a shot's quality [10, 9, 2] or interpreting the results obtained from this statistic [6, 7].

A vastly underexplored topic in the context of xG is how these models are affected by the data on which they are trained. Training data varies from a single season [6] up to six seasons of data from multiple leagues [11]. Knowing that shot locations and efficiency vary between leagues and have evolved over time [13], the question is how the choice of training data affects the xG metric. Hence, in this paper, we will look at using event stream data to answer the following four questions:

1. How much data is needed to train an accurate xG model?
2. Does data go out of date? That is, does training a model using data from more recent seasons result in improved performance compared to using data from older seasons?
3. What is the effect of training an xG model using data from multiple leagues on performance? Does training a league-specific xG model result in improved performance?
4. What is the effect of training an xG model using data from multiple data sources on performance?

2 Data

Our data set consists of match event stream data from all matches between the 2012/2013 and 2018/2019 seasons in the English Premier League, the German Bundesliga, the Spanish LaLiga, the Italian Serie A, the French Ligue 1 and the Dutch Eredivisie. This event stream data was encoded in the SPADL format [4], which was specifically designed to enable automatic data analysis. From these event streams, we extracted all shots from open play (hence omitting penalties and direct free kicks[1]) and the two actions before each shot to capture the shot's context.

3 Methodology

Our goal is not to explore the complete space of design choices to arrive at the best possible xG model, but to mimic reasonable setups. To this end, we will consider two feature sets and two standard models.

[1] Since penalties and free-kicks are relatively easy to predict, our xG models might seem less accurate than other models which include these penalty and free-kick shots.

3.1 Features

Learning an accurate xG model requires coming up with a good set of features to describe the shot attempt. In this study, we consider two features sets: a basic feature set which encodes only the location of the shot and whether it was a header or regular shot, and an advanced feature set which adds contextual information about the preceding actions.

1. Basic features This simply consists of the following 5 features about the shot attempt: the x and y location of the shot, the distance to the center of the goal, the angle to the goal, and the body part used to take the shot (i.e., head or foot). The angle to the goal is measured as the width of goal mouth available to the shooter:

$$\theta = \arctan\left(\frac{\text{goal\_width} * x}{x^2 + y^2 - (\text{goal\_width}/2)^2}\right), \tag{1}$$

where x and y are the coordinates of the shot.

2. Advanced features This consists of 47 features constructed using the shot itself plus the previous two actions. These features encode the velocity of the possession, whether the assist was a through ball or a corner, whether the ball changed possession, etc. Specifically, we compute the following features for each shot and the two preceding actions. Features labelled with an asterisk are not computed for the shot itself, since they would leak information about its outcome.

Action type* The type of the action (pass, cross, shot, dribble, . . .)
Body part The body part used to perform the action (foot or head)
Result* The result of the action (success or fail)
Start location The x, y location where the action starts
End location* The x, y location where the action ends
Start polar The polar coordinates of the location where the action starts
End polar* The polar coordinates of the location where the action ends
Team Whether the same team had possession during the previous action(s)
Space delta Displacement during the previous action(s)
Time delta Elapsed time since the previous action(s)
Speed Displacement during the previous action(s), normalized for elapsed time
Goal angle The angle to the goal, as defined in equation 1.

3.2 Evaluation metrics

We believe that the primary objective of an expected goals model should be to produce calibrated probability estimates [5]. That is, the predicted probabilities should correspond to what is expected to happen in reality: when a shot is given an xG value of 0.3, this essentially means that if that one shot was taken a hundred times, it is expected to result in a goal 30 times. However, this cannot be assessed for a single shot, since each shot is taken only once. Therefore, one typically groups shots with similar xG values in bins and calculates the fraction of shots in each bin that actually resulted in a goal. Ideally, in the bin containing xG values of about x%, about x% of the shots should have resulted in a goal. This is reflected in the probability calibration curve in Figure 1.

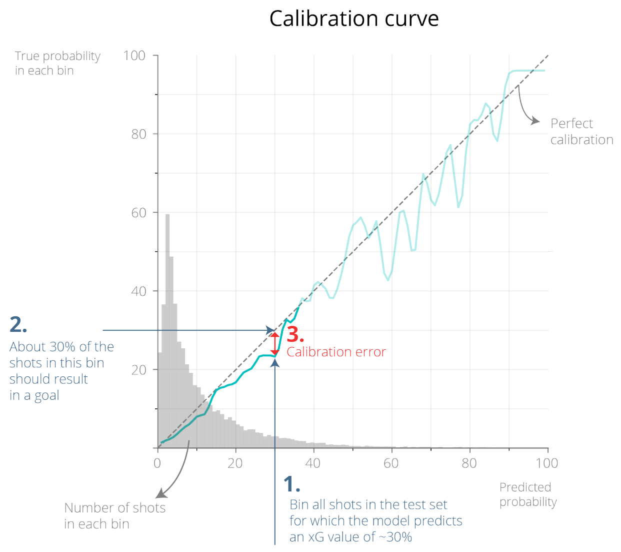

Fig. 1. Probability calibration curve of an xG model.

Many past works do not evaluate the calibration of their models and report the area under the ROC curve (AUROC). However, AUROC only considers the relative ranking of examples (i.e., whether one shot is more or less likely to result in a goal than another shot) and ignores the actual predicted probabilities. This means that a classifier can be poorly calibrated, yet still achieve an excellent AUROC. In contrast, we will report the Brier score [1][2]:

$$\text{Brier} = \frac{1}{\#\text{shots}} \sum_{i=1}^{\#\text{shots}} (\text{xG}_i - O_i)^2 \tag{2}$$

where xG_i is the predicted xG value of shot i and O_i is 1 if shot results in a goal and 0 otherwise. This is a proper scoring rule that can only be minimized by reporting well-calibrated probabilities. While we would like to stress that we do not find AUROC to be an appropriate choice for evaluating xG models, we

[2] This version of the Brier score is only valid for binary classification. The original definition by Brier is applicable to multi-category classification as well.

will report it as many past works have used AUROC to evaluate xG models so it may help place what we have done in context.

3.3 Model learning

We trained two machine learning models: a logistic regression model [12] and a gradient boosted tree model [3]. From a technical perspective, any model that returns a probability would be suitable, but these two models are most commonly used in practice. The key difference between a logistic regression model and a gradient boosted tree model is that the latter model can represent more fine-grained differences between goal scoring opportunities.

The hyperparameters of both models were tuned on the 2012/2013 and 2013/2014 seasons of the top-5 European leagues, using a grid search with 5-fold cross-validation. This hyperparameter tuning was done separately for both feature sets. The parameters which resulted in the best Brier score are listed in appendix A. The source code and trained models are available at `https://github.com/ML-KULeuven/soccer_xg`.

4 How much data is needed to train an accurate xG model?

Figure 2 plots the Brier score for all shots in the 2018/2019 Premier League season as a function of the number of shot attempts included in the training set. This training set is constructed by randomly sampling shots from the 2012/2013 to 2017/2018 Premier League seasons. Because the training set differs for each sample size, we repeat this process 10 times and report the mean ± std Brier score for each sample size.

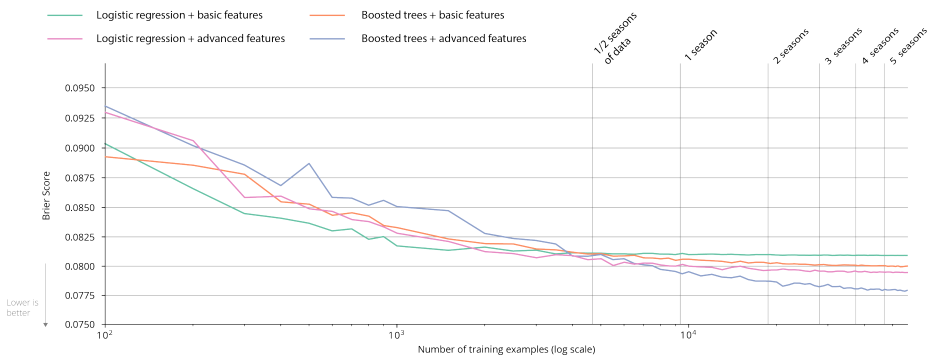

Fig. 2. Mean Brier scores on the 2018/2019 season of the English Premier League using random samples of increasing amounts of training data from the 2012/2013 to 2017/2018 seasons of the same league. The more complex models with the more expressive feature sets require more data, but eventually reach a better Brier score too.

The performance of a logistic regression model using the basic feature set converges after around 6,000 shots which is about 2/3 of a season of data. In contrast, the more complicated feature set requires about three times more data to converge. Of course, the more expressive feature set also results in better performance. Similarly, the more expressive gradient boosting model is more accurate and needs more data than the logistic regression model to converge. On the advanced feature set, it still slightly improves after five seasons of data.

From a machine learning perspective, these findings correspond to common knowledge. Typically, you want more examples than features so when you restrict the size of the feature set you need less data to train an accurate model. Similarly, training more complex models like gradient boosted ensembles requires more data than simpler models like logistic regression.

5 Does data go out of date?

Conventional wisdom is that more recent data is more valuable than older data. Moreover, data may eventually go out of date. In soccer, the style of play changes over time. Thus, it is possible that the types and quality of shot attempts may vary over time. For example in the Premier League, both the number of shots and the average distance of shots has decreased in the past six seasons [13]. Moreover, a player's skill or ability to convert the attempts may also change.

In this vein, we now evaluate the effect of using older data to train an xG model on its performance. In this experiment, the shots from the 2018/2019 Premier League season serve as the test set. We use two consecutive seasons of Premier League data to form the training set and vary the years used, progressively making them older. The below table shows the Brier scores for all four of the models on this experiment:

Table 1. Brier scores on the 2018/2019 season of the English Premier League using pairs of increasingly older seasons of the same league as training data. Using old data results in only a negligible performance hit.

	Basic Features		**Advanced Features**	
Training seasons	Logistic Regression	XGBoost	Logistic Regression	XGBoost
2016/17, 2017/18	0.0812	0.0806	0.0783	0.0783
2015/16, 2016/17	0.0812	0.0804	0.0786	0.0786
2014/15, 2015/16	0.0813	0.0803	0.0790	0.0790
2013/14, 2014/15	0.0813	0.0803	0.0789	0.0789
2012/13, 2013/14	0.0813	0.0803	0.0802	0.0802

Interestingly, using old data results in only a negligible performance hit. Perhaps we would need a much longer historical window to see larger changes in performance.

6 Are xG models league-specific?

Different leagues have different styles of play. However, given that more data enables learning more accurate models, there is a tendency to combine data across different leagues when training a model. The question is what effect does this have? Would training a league-specific model result in better performance?

To answer this question, we consider data from the top-5 European leagues and the Dutch league. We create one test set for each league that consists of data from the 2018/2019 season and vary the data in the training set, considering three types of models:

1. A league-specific model that uses only data from the same league as the test set
2. A mixed model that uses data from the league in the test set plus other leagues, which is the standard approach
3. A model that uses only data from other leagues

To ensure that these training sets are of equivalent size, we always take a random sample of 14,460 shot attempts from 2 seasons of data, which is the largest common subset among these leagues.

In contrast to what others have found [2], the league on which the models are trained does not seem to have a big influence on the models' accuracy (Figure 3). The league-specific model, mixed model and model trained on data from the other leagues perform equally on all leagues. Although these leagues definitely have different styles of play, these seem not to affect the scoring probabilities of a shot. This might be different in lower-level leagues where players have less intrinsic qualities, or in women's soccer.

Which league is in the test set is much more significant: shots in the Dutch league seem much harder to predict than shots in the Serie A. In terms of AUROC scores, the Dutch league is even a clear outlier. Perhaps scoring is more affected by luck in the Dutch league, being a lower-level league and having less skilled players.

7 Are results affected by the data source?

It is worth noting that each data source has its own data quality problems and uses different definitions for labeling shots. These data quality issues are related to the tracking of the events by human annotators who have to determine the right location and type of each event. This is hard to do, especially in a near real-time setting. Therefore, the locations of shots can vary up to a couple of meters between data providers.

These differences may affect the classifiers and the conclusions derived from their outputs. Therefore, we repeated the earlier experiments with a second data provider. Although the data accuracy and event definitions can have a small impact on the reported Brier and AUROC score, the general conclusions in terms of data needed, recency and league effects remain valid. However, mixing data

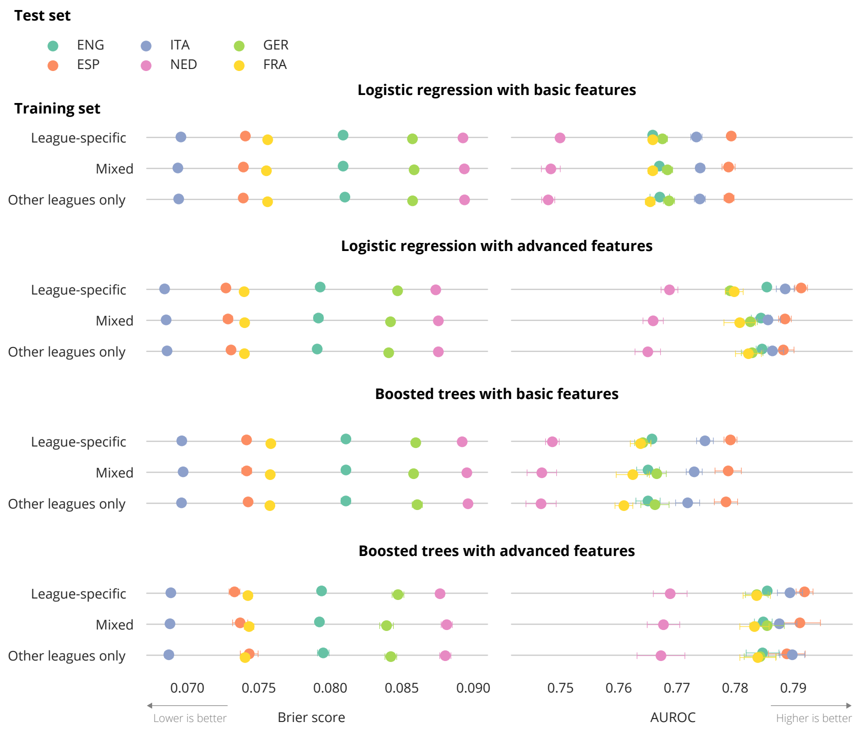

Fig. 3. Mean ± std Brier and AUROC scores on the 2018/2019 season of the English, Spanish, Italian, Dutch, German and French top leagues using 10 equal size random samples of training data from the same league (league-specific), all leagues (mixed) and other leagues only. The league on which the models are trained does not have a significant effect on the models' accuracy.

from different sources is not a good idea. This is illustrated in Figure 4, where we repeat the experiment from Section 6 varying the data source in the training set instead of the league. Training xG models on data from the other provider, or mixing data from multiple providers results in a decreased performance.

8 Discussion and conclusions

In this study we have explored several data-related questions that may affect the performance of an xG model. These issues tend not to be extensively discussed in soccer analytics literature (at least publicly). We showed that the amount of data needed to train an accurate xG model depends on the complexity of the learner and the number of features, with up to 5 seasons of data needed to train a complex gradient boosted trees model. Despite the style of play changing over time and varying between leagues, we did not find that using only recent data or league-specific models improves the accuracy significantly. Hence, if limited data

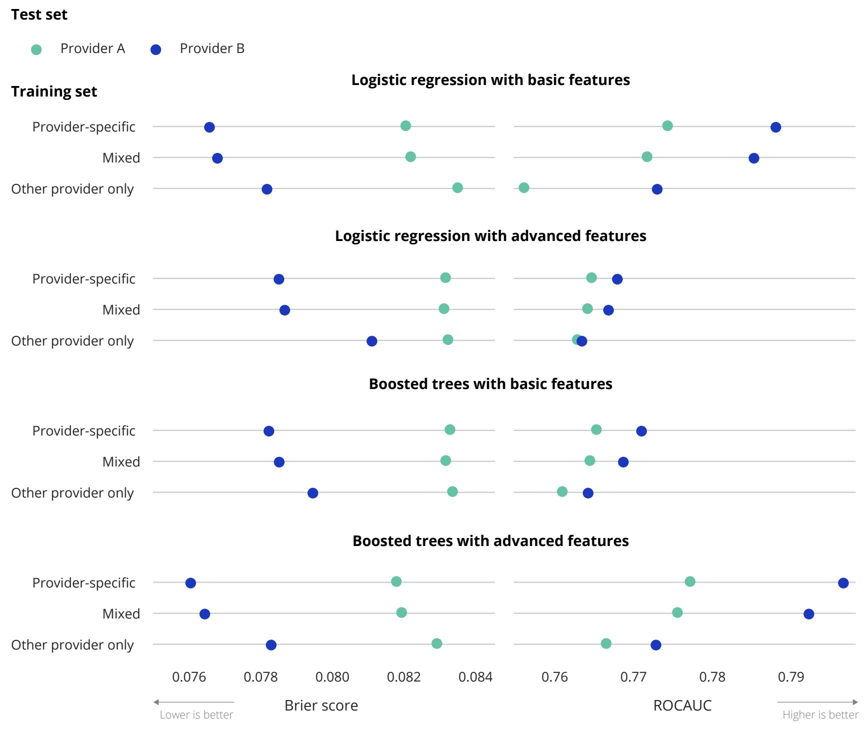

Fig. 4. Mean ± std Brier and AUROC scores on the 2018/2019 season of the English, Spanish, Italian, Dutch, German and French top leagues using 10 equal size random samples of training data from the same data provider (league-specific), both providers (mixed) and the other provider only. Training on data from a different source or mixing data from multiple sources results in decreased performance.

is available, training models on less recent data or different leagues is a viable solution.

In terms of evaluation, we advocated the use of Brier scores instead of AUROC. The applications of xG rely on the predicted probabilities, while the AUROC only measures whether the classifier is able to discriminate between failed and successful shot attempts. The Brier score, on the other hand, is a proper scoring rule that is affected by both discrimination and calibration. The importance of choosing the right metric is further illustrated by the fact that some models perform better on AUROC but worse on the Brier score and the other way around (Figure 3). First, they both measure different qualities of the model and the Brier score is simply better at capturing the qualities that we need. Second, the AUROC can be sensitive to class imbalance [14]. Hence, the AUROC scores may be affected by the fraction of shot attempts that results in a goal in each league.

Acknowledgements

This research received funding from the KU Leuven Research Fund (C14/17/070) and the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly weather review **78**(1), 1–3 (1950)
2. Caley, M.: Premier League Projections and New Expected Goals (2015 (accessed May 27, 2020)), https://cartilagefreecaptain.sbnation.com/2015/10/19/9295905/premier-league-projections-and-new-expected-goals
3. 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. KDD '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785
4. Decroos, T., Bransen, L., Van Haaren, J., Davis, J.: Actions speak louder than goals: Valuing player actions in soccer. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1851–1861 (2019)
5. Decroos, T., Davis, J.: Interpretable prediction of goals in soccer. In: Proceedings of the AAAI-20 Workshop on Artifical Intelligence in Team Sports (Dec 2019)
6. Fairchild, A., Pelechrinis, K., Kokkodis, M.: Spatial analysis of shots in mls: A model for expected goals and fractal dimensionality. Journal of Sports Analytics **4**(3), 165–174 (2018)
7. Gelade, G.: Which team formations produce the most expected goals? (Jul 2017), http://business-analytic.co.uk/blog/which-team-formations-produce-the-most-expected-goals/
8. Green, S.: Assessing the performance of premier league goalscorers (Apr 2012), https://www.optasportspro.com/news-analysis/assessing-the-performance-of-premier-league-goalscorers/
9. Ijtsma, S.: A close look at my new Expected Goals Model (Aug 2015), http://www.11tegen11.com/2015/08/14/a-close-look-at-my-new-expected-goals-model/
10. Kullowatz, M.: Expected goals 3.0 methodology (Apr 2015), https://www.americansocceranalysis.com/home/2015/4/14/expected-goals-methodology
11. Manfredi, G.: Expected goals & player analysis (May 2019), https://www.kaggle.com/gabrielmanfredi/expected-goals-player-analysis
12. 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)
13. Statsbomb: Danish football analysis (April 2019), https://divisionsforeningen.dk/wp-content/uploads/2019/04/Superliga_Analysis.pdf
14. Webb, G.I., Ting, K.M.: On the application of roc analysis to predict classification performance under varying class distributions. Machine learning **58**(1), 25–32 (2005)

A Hyperparameters

Logistic regression + basic features

```
LogisticRegression(C=1.4174741629268048, class_weight=None, dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=10000, multi_class='auto', n_jobs=None,
                   penalty='l2', random_state=None, solver='lbfgs', tol=0.0001,
                   verbose=0, warm_start=False)
```

XGBoost + basic features

```
XGBClassifier(base_score=0.6617525794305433, booster=None,
              colsample_bylevel=0.7545715311191046, colsample_bynode=1,
              colsample_bytree=0.8620890848654332, gamma=0.8215141264974605,
              gpu_id=-1, importance_type='gain', interaction_constraints=None,
              learning_rate=0.08, max_delta_step=0, max_depth=3,
              min_child_weight=8, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=0,
              reg_alpha=0.4953861571782162, reg_lambda=9.636709165264326,
              scale_pos_weight=1.015057505195135, subsample=0.7667216094789041,
              tree_method=None, validate_parameters=False, verbosity=None)
```

Logistic regression + advanced features

```
LogisticRegression(C=0.04328761281083057, class_weight=None, dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=10000, multi_class='auto', n_jobs=None,
                   penalty='l2', random_state=None, solver='lbfgs', tol=0.0001,
                   verbose=0, warm_start=False)
```

XGBoost + advanced features

```
XGBClassifier(base_score=0.7590179091419386, booster=None,
              colsample_bylevel=0.8558659606677331, colsample_bynode=1,
              colsample_bytree=0.9437207946618666, gamma=0.2986963828079735,
              gpu_id=-1, importance_type='gain', interaction_constraints=None,
              learning_rate=0.08, max_delta_step=0, max_depth=4,
              min_child_weight=6, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=0,
              reg_alpha=0.7495409923361984, reg_lambda=7.956623710511393,
              scale_pos_weight=1.0211256886100497, subsample=0.7790327445095349,
              tree_method=None, validate_parameters=False, verbosity=None)
```