

Supplementary Methods:

Expanded Details on ML Model Logic:

The XGBoost Python package utilized to build the two unique ML models used input variables from the training dataset to construct multiple decision trees, giving each a random sample, and established a series of thresholds that split variables to maximize the information gain. Decision trees were constructed iteratively, and new decision trees were added to predict prior errors. After each model had reached the maximum allowed number of trees, or had no further improvements in performance using the validation dataset (also used for hyperparameter tuning), the ensemble of decision trees produced the final ML models that were assessed with the testing dataset.

Expanded Details on the Analysis of ML Model Performance:

As mentioned in the main body of the manuscript, the performance of the ML models was measured with the area under the receiver operating curve (AUROC) in the training, validation, and testing datasets, as well as, the recall, precision, and lift in the testing datasets.

AUROC measures the rate of true and false positives classified by the prediction model across probability thresholds.

Recall measures the rate of true positives classified by the model and is calculated as follows:

$$\text{Recall} = \frac{\text{number of true positives classified by model}}{\text{number of true positives classified by model} + \text{number of false negatives classified by model}}$$

Precision measures the positive predictive value for the model and is calculated as follows:

$$\text{Precision} = \frac{\text{number of true positives classified by model}}{\text{number of true positives classified by model} + \text{number of false positives classified by model}}$$

Lift measures the effectiveness of the model compared to random sampling and is calculated as follows:

$$\text{Lift} = \text{model precision} / \text{proportion of positives in dataset}$$