## The methods used in the study

### 1. Regularized Logistic Regression

The logistic regression method requires the observations to be independent of each other. Therefore, we need to remove “multiple births” from the dataset. Another assumption of the logistic regression is little to no collinearity among the features. To deal with this assumption and also avoid overfitting, we use regularization.

Least Absolute Shrinkage and Selection Operator (LASSO) and ridge regression are the two most common models for feature selection and handling the correlated features, respectively [1]. These methods can also be implemented to reduce the prediction error variance. The LASSO regression penalizes the objective function by adding the $l_1$ penalty, which is the sum of the absolute coefficients $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$. Consider that we have data $(X^i, y_i)$, $i = 1, 2, \ldots, N$, where $X^i = (x_{i1}, ..., x_{ip})^T$ and $y_i$ are the independent and response for the $i$th observation, respectively. The objective function is to minimize

$$\sum_{i=1}^{N} \left( y_i - \sum_j x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \qquad (1)$$

In equation (1), $\lambda$ is the parameter that can control the sparsity—or complexity— of the model. A larger $\lambda$ results in the greater the amount of shrinkage. The ridge regression has the same structure, except that it penalizes the second norm ($l_2$) in the objective function, which is he sum of the squares of the model coefficients: $\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2$.

The two penalties have different effects in the presence of correlated variables. The $l_2$ penalty shrinks coefficients toward each other simultaneously. On the other hand, the $l_1$ penalty tends to select only one of the correlated coefficients and sets the other one to zero. To keep the balance between these two behaviors, we use elastic net regularization. The elastic net combine the two

penalties with argument $\alpha$, which is a value between [0,1], in the same objective function [1]. This will result in a balanced sparsity and stability in the final model. To tune the hyperparameters and select the best $\alpha$ and $\lambda$, we use a grid search.

## 2. Decision Trees

Decision tree (DT) is another classification technique for building predictive models. This machine learning technique involves dividing the feature space into sub-regions $R_j, j = 1, 2, ..., J$, where it starts with a single node, which branches into $J$ terminal nodes of the trees [2]. Then, it fits a simple model, like constant $\gamma_j$ in each region $j$ as

$$x \in R_j \Rightarrow f(x) = \gamma_j$$

Decision tree methods are simple and useful for interpretation [2]. In addition, they are non-parametric methods that can handle both categorical and non-categorical variables. We use three types of decision trees in this paper for building the predictive models, random forest (RF), gradient boosting machines (GBM), and light GBM. We select these methods because they have been widely used in the classification of healthcare related problems and they outperform the other DT methods.

### *2.1. Random forest*

The main idea of random forest (RF) is bagging. Random forest, which is also called wisdom of crowds, is the procedure of first building a large number of trees $\{T(x, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and then aggregating their votes for the popular class at input $x$ [3]. This method has two important parameters for tuning, the number of trees and the depth of each tree that govern the model complexity and accuracy.

Random forest has three characteristics that are particularly useful in the classification of preterm birth in this paper. First, random forest generates variable importance plots that are useful in exploring the independent effect of predictors and making inferences. Second, it can work with multicollinearity in the data and explore the effect of many different interactions depending on the depth of trees. Third, the method does not overfit and the predictions become more stable if we grow sufficiently large number of trees and perform cross-validation [2, 3]. Despite the useful

features of random forest, gradient boosting machines (GBM) performs better in the presence of imbalanced data.

*2.2. Gradient boosting machines*

Boosting is the algorithm of combining many simpler models to fit a highly accurate model [4, 5]. This method is similar to random forest or bagging methods in being a committee method. However, these committee of weak learners improve in each iteration consecutively, and the members cast a weighted vote. These weights are adjusted in a way that the errors in the previous iterations get larger weights.

To illustrate the mathematical steps in a simple way, consider the current estimate of GBM for input $x$ as a sum of trees, or weak learners

$$f_M(x) = \sum_{m=1}^{M} T\left(x; \Theta_m\right),$$

At each step in a stage-wise procedure, the purpose is to find the best parameters, $\hat{\Theta}_m$, by solving an optimization problem to minimize the loss function $L$

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} L\left(y_i, f_{m-1}\left(x_i\right) + T\left(x_i; \Theta_m\right)\right)$$

This optimization is for the region set and constants $\Theta_m = \left\{R_{jm}, \gamma_{jm}\right\}_1^{Jm}$ of the next tree, considering the current model as $f_{m-1}(x)$. Defining the regions is difficult, but if we consider a given region $R_{jm}$, finding the optimal constants in each region, $\gamma_{jm}$ is straightforward. We also grow multiple parallel classification trees sequentially and at the end, take the popular vote as the estimate for each input $x$.

In this paper, we use two different methods of gradient boosting machines for finding the best regions and constants, gradient descent and light GBM method [5, 6]. The gradient descent has long been used in predictive modeling in the field of healthcare [5]. In 2014, XGBoost was introduced as a new GBM that was one of the fastest algorithms among gradient boosted trees with a run time ten times faster than previous popular methods on a single machine [7]. In 2017,

Ke, Meng [6] introduced the light GBM, which returns more accurate outcomes in a shorter training time.

The iterative nature of the gradient boosting machines (GBM) makes it a better classifier for the purpose of this paper, which is increasing the performance of classifier in prediction of the preterm birth as the minority class while keeping the accuracy in prediction of full-term deliveries high. In addition to the good performance, the GBM provides both variable importance plot (VIP) and partial dependence plots (PDP) that are useful tools for interpretation of the results.

Two of the major problems in the training process of the decision trees are overfitting and long training times. As the dimension of the problem increases, the computation resources needed to train the model increases and finding the optimal hyperparameters become harder. Using a subset of observations or features is one of the techniques for reducing the training time, which also reduce the overfitting to the noise and improve generalization. The other common approaches to reduce the training time and prevent overfitting are random and grid search. However, using these methods are not efficient enough.

**References**


1. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological). 1996;58(1):267-88.

2. Friedman J, Hastie T, Tibshirani R. The elements of statistical learning: Springer series in statistics New York, NY, USA:; 2001.

3. Breiman L. Random forests. Machine learning. 2001;45(1):5-32.

4. Schapire RE, Freund Y. Boosting: Foundations and algorithms. Kybernetes. 2013.

5. Friedman JH. Greedy function approximation: a gradient boosting machine. Annals of statistics. 2001:1189-232.

6. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al., editors. Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems; 2017.

7. Chen T, Guestrin C, editors. Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining; 2016: ACM.