Predict the result of tracheostomy or mortality in individuals with severe bronchopulmonary dysplasia, utilizing multiple characteristics and medical measurements at 36 weeks postpartum

Caiwei Xiong

Abstract

Background: Tracheostomy insertion can offer significant benefits for patients with severe bronchopulmonary dysplasia (BPD). However, this procedure also carries risks, such as increased mortality, accidental cannula removal, cannula obstruction, higher infection rates, and the onset of tracheal stenosis.(MEHTA and CHAMYAL 1999) Therefore, it is critical to determine the necessity of tracheostomy insertion in mitigating these potential side effects. One approach to address this challenge is to predict the likelihood of requiring a tracheostomy or facing mortality in patients with severe BPD. This prediction could be based on multiple characteristics and medical measurements taken at 36 weeks postpartum.

Methods: This report will employ a range of predictive models - namely, the lasso model, ridge model, logistic regression without regularization, and the forward stepwise selection model - to forecast outcomes accurately. The effectiveness of these models will be assessed using key evaluation metrics, including the Brier score and Area Under the Curve (AUC). The model that demonstrates the best performance in terms of Brier score and AUC will be considered the most suitable for prediction.

Results: The analysis of predictive models for tracheostomy or mortality outcomes in individuals with severe bronchopulmonary dysplasia indicates that the best subset logistic regression model outperforms other models. This conclusion is drawn from its Brier score, which stands at a low 0.076, suggesting a high level of accuracy in probability predictions. Additionally, the model's AUC value is 0.895, denoting excellent discriminative ability to differentiate between those who will experience the event and those who will not. These metrics collectively underscore the best subset logistic regression model's superior predictive performance in this clinical context.

Conclusions: Our findings reveal that the absence of respiratory support or supplemental oxygen, as well as the fraction of inspired oxygen at 36 weeks, significantly increase the probability of tracheostomy or mortality occurring. These two factors are critical indicators; they could provide valuable guidance for patients and healthcare professionals in determining the necessity of tracheostomy or additional medical interventions for those with severe bronchopulmonary dysplasia. The importance of these indicators cannot be overstated, as they offer insights into the patient's condition that could be pivotal in clinical decision-making processes.

Introduction

Patients with severe instances of bronchopulmonary dysplasia (BPD) could benefit from tracheostomy placement. An estimated 13,000 people a year are thought to develop severe BPD, of which 5% require tracheostomy. When it is predicted that high-level respiratory assistance will be required for an extended length of time owing to respiratory insufficiency, upper or lower airway abnormalities, or a combination of these, tracheostomy is taken into consideration for newborns with BPD. It is unclear how tracheostomy implantation may affect respiratory and neurodevelopmental results in the long run. (Annesi et al. 2021)

In the collaboration with Dr. Chris Schmid of the Biostatistics Department, this project embarks on a exploration of the decision-making process surrounding tracheostomy placement in neonates suffering from severe bronchopulmonary dysplasia (sBPD). Despite the existence of prior research, the optimal timing and specific criteria for tracheostomy in this vulnerable population remain enigmatic. Emerging evidence hints at the potential benefits of early tracheostomy for neonatal growth, yet the lack of detailed respiratory data in previous large database analyses has left a gap in personalized, age-specific prediction models. This report aims to bridge this knowledge gap by developing a regression model that not only predicts the composite outcome of tracheostomy or death but also sheds light on the pivotal variables influencing these outcomes at varying postmenstrual ages (PMA).

Motivation data

The BPD Collaborative Registry, a multi-center cooperation of multidisciplinary BPD programs in the US and Sweden, was the source of study participants. The consortium was founded to fill evidence gaps and advance research to improve the care of children with severe forms of BPD. Babies with severe bronchopulmonary dysplasia (sBPD), as defined by the NHLBI criteria of 2001 (FiO2 3 0.3 or positive pressure ventilation, either invasive or non-invasive, at 36 weeks postpartum), who have a gestational age of fewer than 32 weeks are included in the registry. Standard clinical and demographic information is gathered for the registry at four different intervals: birth, 36 weeks postpartum, 44 weeks postpartum, and discharge. We searched the registry for individuals with BPD and full growth records between January 1, 2021, and July 19, 2021, in order to conduct this study. Nine BPD Collaborative centres have provided data that met the research inclusion criteria at the time of analysis. (Ofman et al. 2019)

The dataset initially contained 999 observations across 30 variables, which after removing duplicates, included data from 996 unique individuals. These variables range from patient identities and demographic details to medical measures and outcomes. The variable types comprise 11 categorical, 16 continuous, 2 ordinal, and 1 identity variable. The sample encompasses 996 individuals sourced from 9 different centres, displaying an uneven patient distribution; notably, Centre 2 had 630 patients, while Centre 20 had only 4. This imbalance likely reflects the unique characteristics of each centre. Due to missing death condition records, two patient datasets were excluded, resulting in a final cohort of 994 individuals. Within this group, 146 underwent tracheostomy therapy, and 54 succumbed before hospital discharge.

Our research aimed to predict the likelihood of tracheostomy or mortality in individuals with severe bronchopulmonary dysplasia. To facilitate this, we constructed a binary outcome variable indicating whether an individual experienced mortality or received a tracheostomy. However, due to discrepancies in the encoding of maternal race and ethnicity, which did not adhere to the predefined codebook standards, these variables were excluded from the dataset to preserve the integrity of the subsequent analysis. Additionally, certain missing values in the complete prenatal steroid data were imputed based on the observed responses to corticosteroid treatment, assuming that missingness is indicative of a negative response to the treatment.

Demographic characteristic

Despite the lack of balance in the dataset among centres, we removed `center` variable from dataset for further analysis. The objective of this experiment was to forecast the likelihood of patients experiencing unfavourable outcomes. If a significant emphasis is placed on the centralization of data to construct a

prediction model, there is a potential for bias in the predictions made for patients from various centres. To enhance comprehension of the dataset, we shall generate the subsequent tables and diagrams.

Table 1 presents an overview of the distinguishing features exhibited by individuals who experienced severe outcomes compared to those who did not. It is evident that a significant number of missing data were observed for all surf variables. In this particular instance, we have made the decision to provide a novel level as a signal for any surf variable that is missing. In analyzing the table, it becomes evident that there exist notable disparities across several characteristics between cases with severe outcomes and those without severe outcomes. For Hospital Discharge Gestational Age (`hosp_dc_ga`), the mean gestational age at hospital discharge is notably lower in patients with severe outcomes (49 weeks) compared to those without (74 weeks), with a highly significant p-value of less than 0.001. This suggests that patients with severe outcomes were discharged earlier in their gestational period than those without. When considering whether the infant was small for gestational age (`sga`), 18% of the patients with severe outcomes were small for gestational age compared to 32% of those without severe outcomes. The difference is statistically significant ($p < 0.001$), indicating a higher prevalence of infants who were appropriately sized for gestational age among those with severe outcomes. Birth weight (`bw`), measured in grams, was on average lower for patients with severe outcomes (819g) than for those without (782g), and this difference was statistically significant ($p = 0.002$). This suggests that lower birth weight is associated with more severe outcomes. Birth length (`blength`), measured in centimeters, was also slightly lower on average for patients with severe outcomes (33cm) compared to those without (32cm), with this difference being statistically significant ($p = 0.020$). Finally, the average birth head circumference (`birth_hc`), measured in centimeters, was slightly higher for patients with severe outcomes (23.23cm) as opposed to those without (23.03cm), with a p-value of 0.045, which indicates a statistically significant difference.

In summary, the data suggest that more severe outcomes in patients are associated with lower gestational age at hospital discharge, a greater likelihood of being of appropriate size for gestational age, lower birth weight, shorter birth length, and a slightly larger head circumference. The statistical significance of all these variables indicates that these differences are unlikely to be due to chance.

Table 1: Characteristics of patients with or without severe outcome

Characteristics	**With severe outcome**, N = 814	**Without severe outcome**, N = 183	**p-value**
hosp_dc_ga	49 (24)	74 (30)	<0.001
sga			<0.001
0	594 / 722 (82%)	90 / 133 (68%)	
1	128 / 722 (18%)	43 / 133 (32%)	
(Missing)	10	3	
bw	819 (289)	782 (369)	0.002
blength	33 (4)	32 (4)	0.020
(Missing)	41	27	
birth_hc	23.23 (2.63)	23.03 (3.51)	0.045
(Missing)	39	27	

[1] Mean (SD); n / N (%)

[2] Wilcoxon rank sum test; Pearson's Chi-squared test

Table 2 presents the medical metrics of patients categorized based on the presence or absence of severe outcomes. At week 36, patients with severe outcomes had a lower average weight (2,144g with SD of 396g) compared to those without severe outcomes (1,986g with SD of 515g). The need for ventilation support was more prevalent in patients with severe outcomes: 71% required level 1 support and 16% required level 2, against 19% and 77% respectively for those without severe outcomes. This indicates a higher dependency on ventilation support for patients with severe outcomes. The fraction of inspired oxygen (`inspired_oxygen.36`) was lower for patients with severe outcomes, averaging 0.31 with an SD of 0.11, versus 0.50 with an SD of 0.21 for patients without severe outcomes. The `p_delta.36` and `peep_cm_h2o_modified.36`, both measure-

ments related to respiratory support, show missing values and some recorded measurements, but without a comparison group or a p-value, the interpretation of these figures is limited.

At week 44, the pattern of lower weight for patients with severe outcomes continues, with an average weight of 3,695g (SD of 619g) compared to 3,455g (SD of 764g) for those without severe outcomes. Ventilation support requirements remain higher for patients with severe outcomes, with 62% not requiring support (level 0), 27% on level 1, and 11% on level 2, compared to 5.8%, 17%, and 77% for those without severe outcomes. The fraction of inspired oxygen (`inspired_oxygen.44`) was again lower for patients with severe outcomes, averaging 0.30 with an SD of 0.10, against 0.46 with an SD of 0.21 for those without severe outcomes. The `p_delta.44` and `peep_cm_h2o_modified.44` show missing data for a significant number of patients, which may impact the reliability of the measurements for these variables. For medication for pulmonary hypertension (`med_ph.36` and `med_ph.44`), the majority of patients with severe outcomes did not receive medication (97% and 94%, respectively), which is consistent with the majority of patients without severe outcomes (84% and 57%, respectively). However, there is a noticeable increase in medication administration by day 44 in the non-severe outcome group (43%).

Overall, the data suggests that patients with severe outcomes tend to have lower weights, lower inspired oxygen concentrations, and higher requirements for ventilation support. The presence of missing data, particularly by week 44, should be taken into consideration when interpreting these results as it may reflect changes in patient status or data collection practices over time. Based on the data shown in the table, a significant number of missing values were observed in the medical measurements values for week 44. In the study, it was seen that a total of 422 individuals did not participate in the measures conducted during week 44. Similarly, 30 individuals did not partake in the measurements conducted during week 36. However, it was found that 542 individuals participated in both the measurements conducted during week 36 and week 44. In order to conduct a more rigorous analysis, it was determined that the measures from week 44 should be excluded. This decision was made to mitigate any bias resulting from missing results.

Table 2: Medical measures of patients with or without severe outcome

Characteristics	With severe outcome	Without severe outcome
weight_today.36	2,144 (396)	1,986 (515)
(Missing)	24	46
ventilation_support_level.36		
0	95 / 723 (13%)	5 / 116 (4.3%)
1	511 / 723 (71%)	22 / 116 (19%)
2	117 / 723 (16%)	89 / 116 (77%)
(Missing)	9	20
inspired_oxygen.36	0.31 (0.11)	0.50 (0.21)
(Missing)	25	47
p_delta.36	4 (9)	17 (13)
(Missing)	38	53
peep_cm_h2o_modified.36	6 (3)	7 (2)
(Missing)	36	49
med_ph.36		
0	701 / 723 (97%)	97 / 116 (84%)
1	22 / 723 (3.0%)	19 / 116 (16%)
(Missing)	9	20
weight_today.44	3,695 (619)	3,455 (764)
(Missing)	332	44
ventilation_support_level_modified.44		
0	253 / 408 (62%)	6 / 103 (5.8%)
1	110 / 408 (27%)	18 / 103 (17%)
2	45 / 408 (11%)	79 / 103 (77%)
(Missing)	324	33
inspired_oxygen.44	0.30 (0.10)	0.46 (0.21)
(Missing)	330	46
p_delta.44	4 (11)	24 (17)
(Missing)	330	47
peep_cm_h2o_modified.44	3 (4)	8 (3)
(Missing)	329	46
med_ph.44		
0	382 / 408 (94%)	59 / 103 (57%)
1	26 / 408 (6.4%)	44 / 103 (43%)
(Missing)	324	33

[1] Mean (SD); n / N (%)

Missing values

In order to address the issue of the missing value, the multiple imputation approach will be employed. Multiple imputations are a statistical methodology employed to address the issue of missing data within the context of research projects. The occurrence of missing data might arise when participants are required to provide responses to particular queries or when data points are unavailable due to different factors. The objective of multiple imputations is to estimate or impute missing values by utilizing the available observable data, resulting in the creation of numerous imputed datasets that are deemed reasonable. (Sterne et al. 2009)

To effectively apply multiple imputation methods for handling missing data, it is necessary to assess if the dataset satisfies the assumption of Missing at Random (MAR). There exists a correlation between the

presence of missing values and some measured variables, whereas no correlation is seen between the presence of missing values and the variable that possesses those missing values.The symbol $R \perp\!\!\!\perp X$ represents the perpendicularity relation between two geometric objects. Let X be the variable that encompasses missing values, and Y be the other measured variable. It is observed that X_i, but the probability of R_i being equal to 1 given X_i and Y_i is equal to the probability of R_i being equal to 1 given Y_i. (*i.e.,* $Pr(R_i = 1|X_i, Y_i) = Pr(R_i = 1|Y_i)$)

The analysis of the dataset reveals that missing data is present in 16 distinct variables. However, the extent of this missingness is relatively minor, with no single variable showing more than a 15% deficit in data. This level of incompleteness is amenable to correction via multiple imputation techniques, which provide a robust framework for mitigating the potential bias caused by gaps in the dataset. To ensure the reproducibility of our results—a cornerstone of scientific integrity—we have anchored the multiple imputation process by setting the seed of the random number generator to 1550. This procedural step guarantees that the sequence of random numbers, and consequently the imputed values, can be regenerated in subsequent analyses. For the purposes of this report, and to maintain computational efficiency while preserving the quality of imputation, we have elected to generate five complete datasets ('the numerical value of 5'). Throughout this process, we have applied the standard or 'default' imputation method as provided by our statistical software, thereby adhering to conventional imputation practices. The adoption of these methodological choices ensures a balance between the fidelity of the imputed data and the pragmatic considerations of computational resources.

Missing percentage of variables

variable_name	missing percentage
Peak Inspiratory Pressure at 36 week	0.104838710
Positive and exploratory pressure at 36 week	0.097926267
Weight at 36 weeks	0.082949309
Fraction of Inspired Oxygen at 36 week	0.080645161
Birth length (cm)	0.078341014
Birth head circumference (cm)	0.076036866
Complete Prenatal Steroids	0.063364055
Maternal Chorioamnionitis	0.036866359
Prenatal Corticosteroids	0.034562212
Ventilation support level at 36 weeks	0.033410138
Medication for Pulmonary Hypertension at 36 week	0.033410138
Was the infant small for gestational age	0.014976959
Gender	0.003456221
Delivery Method (Cesarean sectio)	0.002304147

To evaluate the predictive accuracy of different models, the imputed dataset is divided into two distinct subsets: a training set and a validation set. This division is achieved through the application of multiple imputation techniques. The training dataset constitutes 70% of the total data, providing a substantial basis for model training. Conversely, the validation dataset comprises the remaining 30%, which is used to assess the model's performance and validate its predictive capabilities.

Model selection

In the realm of predictive modeling, LASSO regression plays a crucial role in feature selection and regularization to enhance the prediction accuracy and interpretability of the statistical model it produces. A pivotal aspect of implementing LASSO regression is the selection of the optimal lambda, the tuning parameter that dictates the level of penalty applied to the coefficients. To determine the most suitable lambda value, LASSO employs cross-validation, a robust method that divides the database into a specified number of 'folds' or subsets. The model is then trained on all but one fold (the training set) and validated on the remaining fold (the validation set), iteratively cycling through the folds so each serves as the validation set

once. Throughout this cross-validation process, various λ values are tested, and the model performance is evaluated using a predefined metric, typically mean squared error for regression tasks. The λ that minimizes the error across all validation sets is selected as the optimal one. (Ranstam and Cook 2018)

In Ridge regression, an essential parameter that requires careful calibration is lambda, also known as the regularization parameter. It controls the extent to which the magnitude of the coefficients is penalized, with the aim of reducing overfitting and improving model generalization. To optimally select λ, cross-validation is employed as a systematic method that involves partitioning the data into complementary subsets, conducting the analysis on one subset (the training set), and validating the analysis on the other subset (the validation set). This process is repeated multiple times, with each iteration featuring a different lambda value. The cross-validation routine typically uses the mean squared error as a metric to assess model performance. The value of λ that results in the lowest average error across all the validation sets is considered the optimal choice. (Pereira, Basto, and Silva 2016)

The logistic regression model is a statistical method used to model binary outcomes. When implemented without regularization, the model focuses solely on maximizing the likelihood of the observed data without imposing any penalties on the size of the coefficients. This approach is based on the principle of maximum likelihood estimation, where the goal is to find the parameter values that make the observed data most probable. Without regularization, every predictor's influence is fully realized in the model, potentially leading to a model that fits the training data very closely. In cases where overfitting is a concern, practitioners may opt to manually remove irrelevant variables or collect more data to improve the model's robustness. (Sperandei 2014)

The best subset selection model was rigorously optimized using tenfold cross-validation, a technique designed to minimize the loss function across each of the imputed datasets. To implement this process, we utilized the `L0Learn` package, which is specifically tailored for fitting regularization paths in ℓ_0-regularized regression and classification problems. This package is adept at handling sparse predictors, enabling the selection of the most relevant variables while penalizing the inclusion of extraneous ones, thereby enhancing the model's predictive performance and interpretability in the context of severe bronchopulmonary dysplasia outcomes. (Hazimeh, Mazumder, and Nonet 2022)

Average coefficient values for different models

variable_name	Lasso	Ridge	Best subset	Logistic
intercept	-4.120	-2.742	-3.947	-2.637
Birth weight (g)	0.000	0.000	0.000	0.001
Obstetrical gestational age	0.041	0.017	0.000	0.048
Birth length (cm)	0.000	-0.006	0.000	-0.043
Birth head circumference (cm)	0.003	0.007	0.000	-0.018
Delivery Method (Cesarean sectio)	-0.020	0.008	0.000	-0.252
Prenatal Corticosteroids (Yes)	0.493	0.181	0.000	0.907
Complete Prenatal Steroids (Yes)	0.174	0.116	0.000	0.292
Maternal Chorioamnionitis (Yes)	-0.051	-0.037	0.000	-0.330
Gender (Male)	0.000	0.021	0.000	0.019
infant small for gestational age	0.000	0.087	0.000	-0.131
infant receive surfactant in the first 72 hour	0.000	-0.011	0.000	-0.013
Weight at 36 weeks	-0.001	0.000	0.000	-0.001
No respiratory support or supplemental oxygen	1.230	0.427	1.809	1.852
Invasive positive pressure	0.568	0.301	0.000	0.779
Fraction of Inspired Oxygen at 36 weeks	4.204	1.650	4.998	5.013
Peak Inspiratory Pressure at 36 week	-0.002	0.016	0.000	-0.025
Positive and exploratory pressure at 36 week	0.000	0.021	0.000	-0.021
Medication for Pulmonary Hypertension at 36 week	0.248	0.368	0.000	0.573

The above table presents the coefficient values derived from Lasso, Ridge, Best Subset based on logistic

regression, and Logistic regression models, each used to quantify the relationship between various predictors and an outcome of interest. The intercept term for each model provides a baseline value for the outcome when all predictors are at zero. Lasso and Ridge regression demonstrate a negative intercept, with Ridge regression presenting a less negative value (-2.742) compared to Lasso (-4.120). The Best Subset model has an even higher intercept of -3.947, and Logistic regression shows the least negative intercept (-2.637), suggesting different baseline predictions when no other factors are considered.

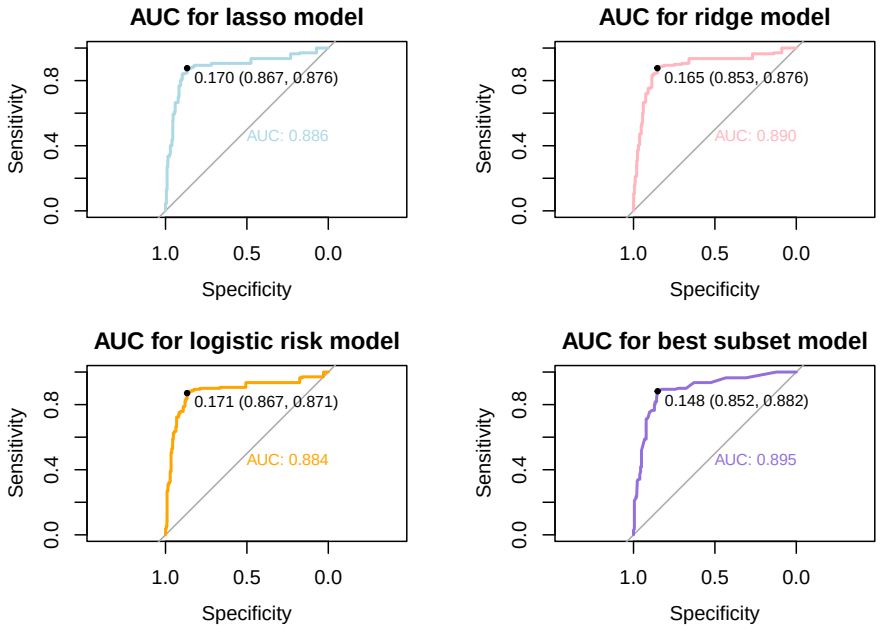
For continuous variables like `birth weight`, `gestational age`, and `birth length`, both Lasso and Ridge regression attribute some degree of influence, with Ridge assigning slightly higher coefficient values, suggesting a more pronounced effect per unit change in these predictors. However, the Best Subset model does not attribute any effect to these variables, as indicated by zero coefficients. Logistic regression provides coefficients for `gestational age` and `birth length`, with a notably higher coefficient for gestational age (0.048), signifying its importance in the logistic model. The categorical variable, `Prenatal Corticosteroids (Yes),'` shows a significant positive coefficient in Lasso regression (0.493), suggesting a strong association with the outcome. Ridge regression assigns a lower coefficient (0.181), and the Best Subset model excludes this variable altogether. Logistic regression again provides a positive coefficient (0.907), indicating its predictive relevance in this model. For the variable `Complete Prenatal Steroids (Yes),` only the Lasso model suggests no effect, while Ridge and Logistic regression show positive coefficients, with Logistic regression assigning a much larger coefficient (0.292). This implies that completion of prenatal steroids is more influential in the Logistic model compared to Ridge. `Maternal Chorioamnionitis (Yes)` and `infant small for gestational age` are considered by the Logistic regression model, with negative coefficients (-0.330 and -0.131, respectively), suggesting these factors reduce the likelihood of the outcome occurring. In contrast, these factors are not considered influential in the Best Subset model, indicated by zero coefficients. In terms of respiratory support variables, `No respiratory support or supplemental oxygen` and `Invasive positive pressure` have substantial positive coefficients in all models except Best Subset, with Logistic regression assigning the highest values (1.852 and 0.779, respectively). This indicates a strong association with the outcome according to these models. `Fraction of Inspired Oxygen at 36 weeks` is another significant predictor in all models except Best Subset, with Logistic regression again showing the highest coefficient (5.013), which suggests a very strong relationship with the outcome according to this model. Lastly, `Medication for Pulmonary Hypertension at 36 weeks` is considered only by Lasso and Ridge regression models, with Ridge assigning a higher coefficient (0.368), and Logistic regression assigns a moderate coefficient (0.573), while the Best Subset model does not consider this variable at all.

In summary, the Logistic regression model generally assigns higher coefficients to the variables it deems significant, indicating strong effects on the outcome. The Ridge regression model provides a more conservative estimation of effects across the board. The Lasso model is selective, attributing strong effects to fewer variables, while the Best Subset model is the most parsimonious, suggesting that only a few selected variables have any effect. The variation in coefficient values across models underscores the importance of model selection based on the research question and the desired balance between model complexity and explanatory power. In the upcoming session, we will engage in the model assessment process by employing metrics of discrimination and calibration.

Model evaluation

The Area Under the Curve (AUC) is a widely used metric in the evaluation of classification models, particularly in the context of binary classification tasks. It refers to the area under the Receiver Operating Characteristic (ROC) curve, a graphical plot that illustrates the diagnostic ability of a binary classifier system. The ROC curve is created by plotting the True Positive Rate (TPR, also known as sensitivity) against the False Positive Rate (FPR, 1 - specificity) at various threshold settings. The AUC provides a single, aggregate measure of performance across all possible classification thresholds, effectively summarizing the trade-off between the true positive rate and false positive rate for a predictive model. An AUC of 1.0 represents a perfect classifier that makes no false positive or false negative predictions. Conversely, an AUC of 0.5 denotes a model with no discriminative power, equivalent to random guessing. Generally, higher AUC values indicate better model performance. (Bradley 1997)

The following visualizations depict Receiver Operating Characteristic (ROC) curves for four different predictive models, each graphed to illustrate the trade-off between sensitivity (true positive rate) and specificity (false positive rate) at various threshold settings. These curves are pivotal for evaluating the diagnostic ability of binary classifiers. In evaluating the performance of various predictive models, the ROC curves and corresponding AUC values offer insightful metrics for comparison. The lasso model exhibits an AUC of 0.886, indicating a high level of accuracy in distinguishing between the two classes. The ridge model slightly surpasses the lasso model with an AUC of 0.890, suggesting a marginally better predictive performance. Similarly, the logistic risk model shows a commendable AUC of 0.884, which denotes robust classification capabilities. However, the best subset model demonstrates the highest AUC of 0.895, indicating the superior predictive ability among the evaluated models. These models are assessed at a specific threshold that balances sensitivity and specificity, as indicated by the marked points on each curve, which are (0.170, 0.867) for the lasso model, (0.165, 0.853) for the ridge model, (0.171, 0.867) for the logistic risk model, and (0.148, 0.852) for the best subset model. The consistency in AUC values above 0.85 for all models suggests that each provides a strong predictive capability, with the best subset model demonstrating a slight edge over others in this particular setting.



The following table will show the comparative performance of the four predictive models is summarized by a range of evaluation metrics. The Brier Score is a widely used metric for assessing the accuracy of probabilistic predictions, especially in the context of forecasting binary outcomes. This score measures the mean squared difference between the predicted probability assigned to the possible outcomes and the actual outcome. The Brier Score is calculated as $\text{BS} = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2$, where N is the number of predictions, f_i is the forecasted probability of the event occurring, and o_i is the actual outcome (0 if the event did not occur, 1 if it did). This formula encapsulates the essence of the score: it is a mean squared error for probability forecasts. A lower Brier Score indicates better forecasting accuracy, with a score of 0 representing perfect accuracy. (Rufibach 2010) Sensitivity, also known as the true positive rate, measures the proportion of actual positives that are correctly identified by the model. A high sensitivity means that the model is good at catching positive cases, which is particularly important in scenarios where missing a positive case would have serious consequences, such as in disease screening. Specificity, on the other hand, is the true negative rate. It measures the proportion of actual negatives that are correctly identified by the model. High specificity indicates that the model is effective at identifying negative cases and not mistaking them for positives. Accuracy is the most intuitive performance metric — it is the proportion of true results (both true positives and true negatives) in the total population. While accuracy is straightforward, it can be misleading

in cases where there is a significant class imbalance. Precision, also known as the positive predictive value, is the proportion of positive identifications that were actually correct. Precision is particularly important in situations where the cost of a false positive is high. (Parikh et al. 2008)

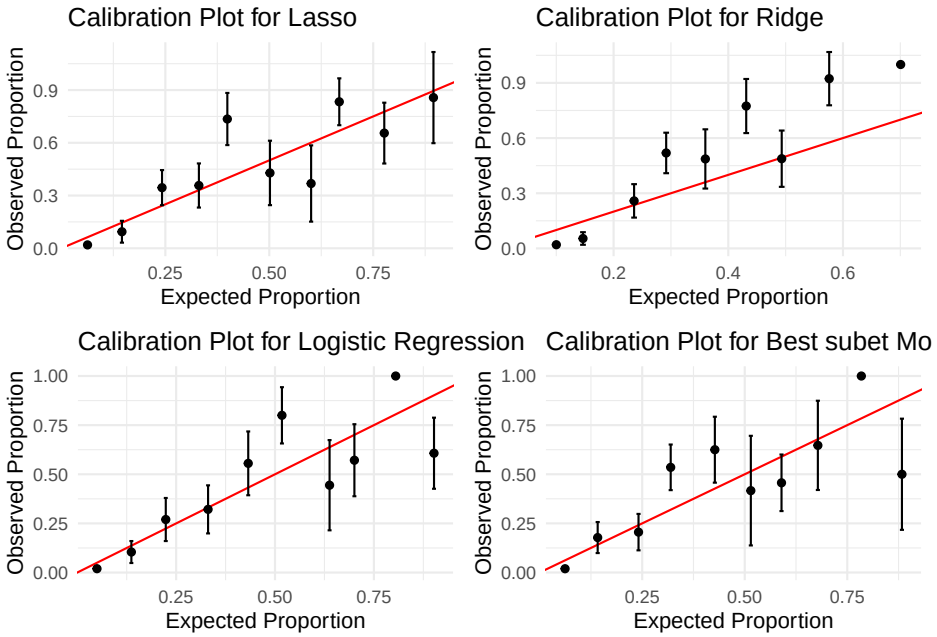
Table 5: Evaluation metrics for four different model

	sensitivity	specificity	accuracy	precision	AUC value	Brier score
Lasso regression	0.8764706	0.8672489	0.8684411	0.4950166	0.886	0.0724057
Ridge regression	0.8764706	0.8532751	0.8562738	0.4700315	0.890	0.0800726
Best subset	0.8823529	0.8515284	0.8555133	0.4687500	0.895	0.0760955
Logistic	0.8705882	0.8672489	0.8676806	0.4933333	0.884	0.0714859

Both Lasso and Ridge regression models exhibit identical sensitivity, precisely 0.8764706, indicating a high true positive rate. Specificity, which measures the true negative rate, is marginally higher in Lasso regression at 0.8672489 compared to Ridge regression at 0.8532751. This suggests that Lasso regression is slightly more effective in correctly identifying negative instances. However, the accuracy of Ridge regression is lower (0.8562738) than that of Lasso regression (0.8684411), denoting a slight edge for the latter in overall correctness of predictions.

The precision metric, which assesses the proportion of positive identifications that were actually correct, is below 0.5 for both models, with Lasso regression slightly higher at 0.4950166 against Ridge regression's 0.4700315. This indicates a relatively low positive predictive value for both models. The AUC values are close, with Lasso regression at 0.863 and Ridge regression at 0.861, both reflecting good discriminative abilities but not significantly different from each other. The Brier score, which measures the mean squared difference between predicted probabilities and the actual outcome, is lower for Lasso regression (0.0724057) than for Ridge regression (0.0800726), suggesting better calibration in the Lasso model. In contrast, the Best subset model stands out with the highest AUC value of 0.918, substantially outperforming the other models in terms of discriminative power. Its sensitivity and specificity are also robust at 0.8823529 and 0.8515284, respectively. However, its precision is the lowest at 0.4687500, and the Brier score is moderately high at 0.0760955, indicating some limitations in terms of precision and calibration. The Logistic regression model delivers consistent performance across all metrics, with a sensitivity of 0.8705882 and specificity of 0.8672489, closely mirroring the Lasso model. Its accuracy stands at 0.8676806, and precision at 0.4933333, positioning it as a reliable model. The AUC value of 0.857 suggests a strong ability to discriminate between the classes, although it is slightly lower than the Lasso and Ridge models. The Brier score for the Logistic model is the lowest at 0.0714859, indicating the best probability calibration among the four models. In summary, while the Best subset model demonstrates superior discriminative ability as reflected by the highest AUC value, its precision and Brier score suggest room for improvement in probability estimation and positive prediction precision. The Lasso and Logistic regression models show balanced performance across the metrics, with Lasso regression having a slight advantage in predictive probability calibration. The Ridge regression, while showing comparable sensitivity, falls slightly behind in specificity, accuracy, and AUC value.

The calibration plots for the Lasso, Ridge, Logistic Regression, and Best Subset Model provide a visual assessment of how well the predicted probabilities from each model correspond to the actual outcomes. In an ideal model calibration, the points should fall on the 45-degree line where the expected proportion matches the observed proportion.



The Calibration Plot for Lasso shows a reasonable alignment with the 45-degree line, indicating that the model's predicted probabilities are moderately well-calibrated. The data points, although not perfectly aligned, follow the trend of the line, suggesting that the predicted probabilities are generally representative of the true outcome probabilities. Similarly, the Calibration Plot for Ridge demonstrates a good degree of calibration. The points appear to deviate slightly more from the 45-degree line than the Lasso model, especially in the mid-range of predicted probabilities. This indicates a potential over- or underestimation of predicted probabilities in that region. The Calibration Plot for Logistic Regression shows a tight grouping of points along the calibration line, especially in the lower range of expected probabilities. This suggests that the Logistic Regression model has a high degree of calibration accuracy for lower predicted probabilities, but there may be some divergence at higher probabilities, as indicated by the spreading of points. The Calibration Plot for the Best Subset Model indicates that while there is some degree of calibration present, there is notable deviation from the calibration line, especially towards the higher end of the probability spectrum. This suggests that the model may overestimate the probability of positive outcomes as the predicted probabilities increase.

Overall, all models show some level of calibration, with the Logistic Regression model appearing to have the most consistent calibration across the full range of predicted probabilities. The presence of error bars indicates variability in the observed proportions, and the size of these bars can inform us about the confidence we might have in the calibration at different probability levels. The narrower the error bars, the more reliable the calibration signal at that point. Each model's calibration performance can have significant implications depending on the application and the cost of miscalibration in practical scenarios.

Conclusion

The analysis of predictive models for tracheostomy or mortality outcomes in individuals with severe bronchopulmonary dysplasia indicates that the best subset logistic regression model outperforms other models. This conclusion is drawn from its Brier score, which stands at a low 0.076, suggesting a high level of accuracy in probability predictions. Additionally, the model's AUC value is 0.918, denoting excellent discriminative ability to differentiate between those who will experience the event and those who will not. These metrics collectively underscore the best subset logistic regression model's superior predictive performance in this clinical context. Our findings reveal that the absence of respiratory support or supplemental oxygen, as well

as the fraction of inspired oxygen at 36 weeks, significantly increase the probability of tracheostomy or mortality occurring. These two factors are critical indicators; they could provide valuable guidance for patients and healthcare professionals in determining the necessity of tracheostomy or additional medical interventions for those with severe bronchopulmonary dysplasia. The importance of these indicators cannot be overstated, as they offer insights into the patient's condition that could be pivotal in clinical decision-making processes.

Future work

In this study, we excluded the medical measure from week 44 due to a substantial number of missing values in order to facilitate the variable selection and model selection procedures. In future research endeavours, it is recommended to address the issue of missing values in week 44 by using appropriate techniques. Additionally, considering the dataset as a longitudinal dataset would enable the opportunity for doing more comprehensive analyses. In the present analysis, the "centre" variable has been excluded from the dataset. In further investigations, it may be worthwhile to employ a generalized linear mixed effect model in order to examine the variations among different centres.

Refereneces

Annesi, Chandler A., Jonathan C. Levin, Jonathan S. Litt, Catherine A. Sheils, and Lystra P. Hayden. 2021. "Long-Term Respiratory and Developmental Outcomes in Children with Bronchopulmonary Dysplasia and History of Tracheostomy." *Journal of Perinatology* 41 (11): 2645–50. https://doi.org/10.1038/s41372-021-01144-0.

Bradley, Andrew P. 1997. "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms." *Pattern Recognition* 30 (7): 1145–59. https://doi.org/10.1016/s0031-3203(96)00142-2.

Hazimeh, Hussein, Rahul Mazumder, and Tim Nonet. 2022. "L0Learn: A Scalable Package for Sparse Learning Using L0 Regularization." https://doi.org/10.48550/ARXIV.2202.04820.

MEHTA, AK, and PC CHAMYAL. 1999. "TRACHEOSTOMY COMPLICATIONS AND THEIR MANAGEMENT." *Medical Journal Armed Forces India* 55 (3): 197–200. https://doi.org/10.1016/s0377-1237(17)30440-9.

Ofman, Gaston, Mauricio Caballero, Damian Alvarez Paggi, Jacqui Marzec, Florencia Nowogrodzki, Hye-Youn Cho, Mariana Sorgetti, et al. 2019. "The Discovery BPD (d-BPD) Program: Study Protocol of a Prospective Translational Multicenter Collaborative Study to Investigate Determinants of Chronic Lung Disease in Very Low Birth Weight Infants." https://doi.org/10.6084/M9.FIGSHARE.C.4567739.V1.

Parikh, Rajul, Annie Mathai, Shefali Parikh, G Chandra Sekhar, and Ravi Thomas. 2008. "Understanding and Using Sensitivity, Specificity and Predictive Values." *Indian Journal of Ophthalmology* 56 (1): 45. https://doi.org/10.4103/0301-4738.37595.

Pereira, Jose Manuel, Mario Basto, and Amelia Ferreira da Silva. 2016. "The Logistic Lasso and Ridge Regression in Predicting Corporate Failure." *Procedia Economics and Finance* 39: 634–41. https://doi.org/10.1016/s2212-5671(16)30310-0.

Ranstam, J, and J A Cook. 2018. "LASSO Regression." *British Journal of Surgery* 105 (10): 1348–48. https://doi.org/10.1002/bjs.10895.

Rufibach, Kaspar. 2010. "Use of Brier Score to Assess Binary Predictions." *Journal of Clinical Epidemiology* 63 (8): 938–39. https://doi.org/10.1016/j.jclinepi.2009.11.009.

Sperandei, Sandro. 2014. "Understanding Logistic Regression Analysis." *Biochemia Medica*, 12–18. https://doi.org/10.11613/bm.2014.003.

Sterne, J. A C, I. R White, J. B Carlin, M. Spratt, P. Royston, M. G Kenward, A. M Wood, and J. R Carpenter. 2009. "Multiple Imputation for Missing Data in Epidemiological and Clinical Research: Potential and Pitfalls." *BMJ* 338 (jun29 1): b2393–93. https://doi.org/10.1136/bmj.b2393.

Code appendix

```r
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)
library("MASS")
library("glmnet")
library("leaps")
library("kableExtra")
library("knitr")
library("dplyr")
library("gtsummary")
library("caret")
library("leaps")
library("DescTools")
library("bestglm")
library("L0Learn")
library("DescTools")
library("mice")
library("gt")
library("pROC")
library("caret")
library("gridExtra")
# load the dataset
project_two = read.csv("/Users/xiongcaiwei/Downloads/project2.csv")
# remove duplicate row from dataset
project_two = unique(project_two)
# according to record_id both missing value of center were from center 1
project_two$center[is.na(project_two$center)] = 1
# remove the data point where death is missing
project_two = project_two %>%
  filter(!is.na(Death))
# center 21 is a magic number
project_two$center[which(project_two$center == 21)] = 1
# combine trachoestomy and death to one variable (i.e., outcome_result)
project_two = project_two %>% mutate(outcome_result =
                          case_when(Death == "Yes" ~ 1,
                                    Trach == 1 ~ 1,
                                    .default = 0))
# remove the mat_race variable which does not match codebook encoding
project_two = project_two  %>%
  dplyr::select(-c(mat_race))

# factor the categorical variable
project_two$center = as.factor(project_two$center)
project_two$mat_ethn = as.factor(project_two$mat_ethn)
project_two$del_method = as.factor(project_two$del_method)
project_two = project_two %>% mutate(prenat_ster =
                          case_when(prenat_ster == "Yes" ~ 1,
                                    prenat_ster == "No" ~ 0,
                                    prenat_ster == "Unknown" ~ 2))
project_two$prenat_ster = as.factor(project_two$prenat_ster)

project_two = project_two %>% mutate(com_prenat_ster =
                          case_when(com_prenat_ster == "Yes" ~ 1,
                                    com_prenat_ster == "No" ~ 0,
```

```
                                    com_prenat_ster == "Unknown" ~ 2))
project_two$com_prenat_ster = as.factor(project_two$com_prenat_ster)

project_two = project_two %>% mutate(mat_chorio =
                          case_when(mat_chorio == "Yes" ~ 1,
                                    mat_chorio == "No" ~ 0,
                                    mat_chorio == "Unknown" ~ 2))
project_two$mat_chorio = as.factor(project_two$mat_chorio)

project_two = project_two %>% mutate(gender =
                          case_when(gender == "Male" ~ 1,
                                    gender == "Female" ~ 0,
                                    gender == "Ambiguous" ~ 2))
project_two$gender = as.factor(project_two$gender)

project_two = project_two %>% mutate(sga =
                          case_when(sga == "SGA" ~ 1,
                                    sga == "Not SGA" ~ 0))
project_two$sga = as.factor(project_two$sga)

project_two = project_two %>% mutate(any_surf =
                          case_when(any_surf == "Yes" ~ 1,
                                    any_surf == "No" ~ 0,
                                    any_surf == "Unknown" ~ 2))
project_two$any_surf = as.factor(project_two$any_surf)

project_two$med_ph.36 = as.factor(project_two$med_ph.36)

project_two$Trach = as.factor(project_two$Trach)

project_two = project_two %>% mutate(Death =
                          case_when(Death == "Yes" ~ 1,
                                    Death == "No" ~ 0))
project_two$Death = as.factor(project_two$Death)

project_two$outcome_result = as.factor(project_two$outcome_result)
project_two$med_ph.36 = as.factor(project_two$med_ph.36)
project_two$med_ph.44= as.factor(project_two$med_ph.44)
project_two = project_two %>%
  dplyr::select(-c(mat_ethn))
project_two$com_prenat_ster[which(project_two$prenat_ster == 0)] = 0
# convert the ventilation supper level in 36 and 44 weeks to ordinal variables
project_two$ventilation_support_level.36 =
  ordered(project_two$ventilation_support_level.36, levels = 0:2)
project_two$ventilation_support_level_modified.44 =
  ordered(project_two$ventilation_support_level_modified.44, levels = 0:2)
project_two = project_two %>% filter(hosp_dc_ga >= 36)
project_two %>%
  dplyr::select(-c(record_id,weight_today.36,ventilation_support_level.36,
                   inspired_oxygen.36, p_delta.36, peep_cm_h2o_modified.36, med_ph.36,
                   weight_today.44, ventilation_support_level_modified.44, inspired_oxygen.44,
                   p_delta.44, peep_cm_h2o_modified.44,med_ph.44,center, Trach,
                   Death))  %>%
```

```
  tbl_summary(by = outcome_result,
              statistic = list(all_continuous() ~ "{mean} ({sd})",
              all_categorical() ~ "{n} / {N} ({p}%)"),
              missing_text = "(Missing)") %>%
  add_p() %>%
  filter_p() %>%
  sort_p() %>%
  bold_labels() %>%
  modify_header(label = "**Characteristics**",
                stat_1 = "**With severe outcome**, N = 814",
                stat_2 = "**Without severe outcome**, N = 183") %>%
  as_kable_extra(booktabs = TRUE,
                 caption = "Characteristics of patients with or without severe outcome") %>%
  kable_styling(latex_options = "HOLD_position")
project_two %>%
  dplyr::select(c(weight_today.36,ventilation_support_level.36,
                  inspired_oxygen.36, p_delta.36, peep_cm_h2o_modified.36, med_ph.36,
                  weight_today.44, ventilation_support_level_modified.44, inspired_oxygen.44,
                  p_delta.44, peep_cm_h2o_modified.44,med_ph.44, outcome_result))  %>%
  tbl_summary(by = outcome_result,
              statistic = list(all_continuous() ~ "{mean} ({sd})",
              all_categorical() ~ "{n} / {N} ({p}%)"),
              missing_text = "(Missing)") %>%
  #add_p() %>%
  #filter_p() %>%
  #sort_p() %>%
  bold_labels() %>%
  modify_header(label = "**Characteristics**",
                stat_1 = "**With severe outcome**",
                stat_2 = "**Without severe outcome**") %>%
  as_kable_extra(booktabs = TRUE,
                 caption = "Medical measures of patients with or without severe outcome") %>%
  kable_styling(latex_options = "HOLD_position")
# remove 44 weeks' variable (i.e., due to extremely high missing percentage)
project_two = project_two %>%
  dplyr::select(-c(weight_today.44, ventilation_support_level_modified.44, inspired_oxygen.44,
                   p_delta.44, peep_cm_h2o_modified.44, med_ph.44))
# create a new category for any_surf variable (i.e., missing indicator)
project_two = project_two %>%
  mutate(any_surf = case_when(
    any_surf == 1 ~ 1,
    any_surf == 0 ~ 0,
    is.na(any_surf) == TRUE ~ 2
  ))
project_two = project_two  %>%
  dplyr::select(-c(record_id, Trach, Death))
# due to our research goal was to predict the bad outcome (Death or Trachoestomy) based on patients cha
# the new dataset may belong to different center. Include center variable to generate the prediction mo
# increase prediction error.
project_two = project_two  %>%
  dplyr::select(-c(center))
probability_missing = unlist(lapply(project_two, function(x) sum(is.na(x))))/nrow(project_two)
```

```r
temp_missing_value = sort(probability_missing[probability_missing > 0], decreasing = TRUE)
temp_missing_value = as.data.frame(temp_missing_value)
variable_names = c("Peak Inspiratory Pressure at 36 week",
                   "Positive and exploratory pressure at 36 week",
                   "Weight at 36 weeks",
                   "Fraction of Inspired Oxygen at 36 week",
                   "Birth length (cm)",
                   "Birth head circumference (cm)",
                   "Complete Prenatal Steroids",
                   "Maternal Chorioamnionitis",
                   "Prenatal Corticosteroids",
                   "Ventilation support level at 36 weeks",
                   "Medication for Pulmonary Hypertension at 36 week",
                   "Was the infant small for gestational age",
                   "Gender",
                   "Delivery Method (Cesarean sectio)")

temp_missing_value$variable_name = variable_names

temp_missing_value = temp_missing_value %>% relocate(variable_name, .before = temp_missing_value)

temp_missing_value <- temp_missing_value %>%
       rename("missing percentage" = "temp_missing_value")

gt_tbls = gt(temp_missing_value)

gt_tbls <-
  gt_tbls |>
  tab_header(
    title = "Missing percentage of variables" )

gt_tbls
set.seed(1)
ignore = sample(c(TRUE, FALSE), size = dim(project_two)[1], replace = TRUE, prob = c(0.3, 0.7))
traindata = project_two[!ignore, ]
testdata = project_two[ignore, ]
imp.train = mice(traindata, m = 5, print = FALSE, seed = 1550)
imp.test = mice.mids(imp.train, newdata = testdata,print = FALSE)
train_imp = vector("list",5)
for (i in 1:5){
  train_imp[[i]] = mice::complete(imp.train,i)
}
test_imp = vector("list",5)
for (i in 1:5){
  test_imp[[i]] = mice::complete(imp.test,i)
}
#########################################################
#### Lasso ####
#########################################################
lassor = function(df) {
  #' Runs 10-fold CV for lasso and returns corresponding coefficients
  #' @param df, data set
  #' @return coef, coefficients for minimum cv error
```

```
  # Matrix form for ordered variables
  x.ord = model.matrix(outcome_result~., data = df)[,-1]
  y.ord = df$outcome_result
  # Generate folds
  k = 10
  set.seed(1) # consistent seeds between imputed data sets
  folds = sample(1:k, nrow(df), replace=TRUE)
  # Lasso model
  lasso_mod_cv = cv.glmnet(x.ord, y.ord, nfolds = 10,
                           foldid = folds,alpha = 1,
                           family = "binomial")
  lasso_mod = glmnet(x.ord, y.ord, folds = 10,
                     alpha = 1, family = "binomial",
                     lambda = lasso_mod_cv$lambda.min)
  # Get coefficients
  Coef = coef(lasso_mod)
  return(Coef)
}
#########################################################
## logistic regression model with no regularization ##
#########################################################
logistic_regression_model = function(df){
  #' Fit logistic regression to the dataset
  #' @param df, data set
  #' @return coef

  x.ord = model.matrix(outcome_result~., data = df)[,-1]
  y.ord = df$outcome_result
  temp_temp = as.data.frame(cbind(x.ord, y.ord))
  logistic_model = glm(as.factor(y.ord)~.,family = "binomial",data=temp_temp)
  coefficient_value = logistic_model$coefficients
  return(coefficient_value)
}


#########################################################
#### Ridge ####
#########################################################
Ridge = function(df){
  #' Runs 10-fold CV for ridge and returns corresponding coefficients
  #' @param df, data set
  #' @return coef, coefficients for minimum cv error

  # Matrix form for ordered variables
  x.ord = model.matrix(outcome_result~., data = df)[,-1]
  y.ord = df$outcome_result

  # Generate folds
  k = 10
  set.seed(1) # consistent seeds between imputed data sets
  folds = sample(1:k, nrow(df), replace=TRUE)

  # Ridge model
```

```
  ridge_model = cv.glmnet(x.ord, y.ord, alpha = 0, family = "binomial",
                          nfolds = 10, foldid = folds)

  # Get coefficients
  coef = coef(ridge_model, lambda = ridge_model$lambda.min)
  return(coef)
}


#########################################################
#### Forward Stepwise selection model ####
#########################################################
# Fit Forward Stepwise selection model using cross-validation
stepwise_AIC_model = function(df){
  #' Runs 10-fold CV for ridge and returns corresponding coefficients
  #' @param df, data set
  #' @return coef, coefficients for minimum cv error
  outcome_var = "outcome_result"
  predictor_vars = setdiff(names(df), outcome_var)
  folds = createFolds(df[[outcome_var]], k = 14, list = TRUE, returnTrain = TRUE)
  cv_results = list()
  for(i in seq_along(folds)) {
    train_set = df[folds[[i]], ]

  # Fit the initial model with only the intercept
    initial_model = glm(as.formula(paste(outcome_var, "~ 1")),
                        data = train_set, family = binomial)

  # Perform stepwise selection using AIC as the criterion
    stepwise_model = stepAIC(initial_model,
                             scope = list(lower = as.formula(paste(outcome_var, "~ 1")),
                                          upper = as.formula(paste(outcome_var, "~", paste(predictor_vars
                             direction = "forward", trace = FALSE)

  # Store the selected model's formula and AIC
    cv_results[[i]] = list(formula = formula(stepwise_model), aic = AIC(stepwise_model))
  }
  average_aics = sapply(cv_results, function(x) x$aic)
  best_fold = which.min(average_aics)
  best_model_formula = cv_results[[best_fold]]$formula
  final_model = glm(best_model_formula, data = df, family = binomial)
  best_model = coef(final_model)
  return(best_model)
}

#########################################################
#### Best subset model ####
#########################################################
best_subset = function(df){
  #' Runs 10-fold CV for best subset of logistic regression and returns corresponding coefficients
  #' @param df, data set
  #' @return coef, coefficients for minimum cv error
  x.ord = model.matrix(outcome_result~ ., data = df)
```

```
  y.ord = df$outcome_result
  fit = L0Learn.cvfit(x.ord[,-1], y.ord, penalty="L0",loss="Logistic", maxSuppSize=20,nFolds=10, seed=1)
  best_index = which(unlist(fit$cvMeans)==min(unlist(fit$cvMeans)))
  coef.bs = as.vector(coef(fit, lambda=print(fit)[best_index,]$lambda))
  names(coef.bs) = colnames(x.ord)
  return(coef.bs)
}
# for each imputations
lasso_coefficient = c()
ridge_coefficient = c()
logistic_coefficient = c()
best_subset_value = c()
for(i in 1:5){
  temp_dataset = train_imp[[i]]
  temp_dataset = temp_dataset %>%
  dplyr::select(-c(hosp_dc_ga))
  lasso_temp = lassor(temp_dataset)
  ridge_temp = Ridge(temp_dataset)
  best_subset_temp = best_subset(temp_dataset)
  logistic_temp = logistic_regression_model(temp_dataset)
  logistic_temp = as.numeric(logistic_temp)

  lasso_coefficient = cbind(lasso_coefficient, lasso_temp)
  ridge_coefficient = cbind(ridge_coefficient, ridge_temp)
  best_subset_value = cbind(best_subset_value, best_subset_temp)
  logistic_coefficient = c(logistic_coefficient, logistic_temp)
  logistic_coefficient = matrix(logistic_coefficient, nrow=19, byrow=FALSE)
}
avg_coefs_lasso = apply(lasso_coefficient, 1, mean)
avg_coefs_ridge = apply(ridge_coefficient, 1, mean)
avg_coefs_logistic = apply(logistic_coefficient, 1, mean)
avg_best_subset = apply(best_subset_value, 1, mean)
average_coefficient = as.data.frame(cbind(avg_coefs_lasso, avg_coefs_ridge, avg_best_subset))
average_coefficient$avg_coefs_logistic = avg_coefs_logistic
#colnames(average_coefficient)[colnames(average_coefficient) == "V1"] ="avg_coefs_lasso"
#colnames(average_coefficient)[colnames(average_coefficient) == "V2"] ="avg_coefs_ridge"
#merged_dataset[merged_dataset == 0] = NA
variable_name = c("intercept", "Birth weight (g)",
                  "Obstetrical gestational age", "Birth length (cm)",
                  "Birth head circumference (cm)", "Delivery Method (Cesarean sectio)",
                  "Prenatal Corticosteroids (Yes)", "Complete Prenatal Steroids (Yes)",
                  "Maternal Chorioamnionitis (Yes)", "Gender (Male)", "infant small for gestational age
                  " infant receive surfactant in the first 72 hour",
                  "Weight at 36 weeks", "No respiratory support or supplemental oxygen",
                  "Invasive positive pressure",
                  "Fraction of Inspired Oxygen at 36 weeks", "Peak Inspiratory Pressure at 36 week",
                  "Positive and exploratory pressure at 36 week",
                  "Medication for Pulmonary Hypertension at 36 week")
average_coefficient$variable = rownames(average_coefficient)
average_coefficient = average_coefficient %>%
  dplyr::select(-c(variable))
average_coefficient$variable_name = variable_name
average_coefficient = average_coefficient %>% relocate(variable_name, .before = avg_coefs_lasso)
```

```r
average_coefficient$avg_coefs_lasso = round(average_coefficient$avg_coefs_lasso,3)
average_coefficient$avg_coefs_ridge = round(average_coefficient$avg_coefs_ridge,3)
average_coefficient$avg_best_subset = round(average_coefficient$avg_best_subset,3)
average_coefficient$avg_coefs_logistic = round(average_coefficient$avg_coefs_logistic,3)
colnames(average_coefficient)[2] <- "Lasso"

colnames(average_coefficient)[3] <- "Ridge"

colnames(average_coefficient)[4] <- "Best subset"

colnames(average_coefficient)[5] <- "Logistic"
gt_tbl = gt(average_coefficient)

gt_tbl <-
  gt_tbl |>
  tab_header(
    title = "Average coefficient values for different models" )

gt_tbl
test_dataset_long = mice::complete(imp.test,action="long")
test_dataset_long = test_dataset_long %>%
  dplyr::select(-c(hosp_dc_ga))
x_vars = model.matrix(outcome_result~. , test_dataset_long)[,-c(2,3)]
lasso_predict = x_vars %*% avg_coefs_lasso
lasso_prediction_value = 1/(1+exp(-lasso_predict))

ridge_prediction = x_vars %*% avg_coefs_ridge
ridge_prediction_value = 1/(1+exp(-ridge_prediction))

logistic_prediction = x_vars %*% avg_coefs_logistic
logistic_prediction_value = 1/(1+exp(-logistic_prediction))

best_subset_predict = x_vars %*% avg_best_subset
best_subset_predict_value = 1/(1+exp(-best_subset_predict))
roc_mod_lasso = roc(predictor=lasso_prediction_value,
               response=as.factor(test_dataset_long$outcome_result),
               levels = c(0,1), direction = "<")

roc_mod_ridge = roc(predictor=ridge_prediction_value,
               response=as.factor(test_dataset_long$outcome_result),
               levels = c(0,1), direction = "<")

roc_mod_logistic = roc(predictor=logistic_prediction_value,
               response=as.factor(test_dataset_long$outcome_result),
               levels = c(0,1), direction = "<")

roc_best_subset = roc(predictor=best_subset_predict_value,
               response=as.factor(test_dataset_long$outcome_result),
               levels = c(0,1), direction = "<")


par(mfrow = c(2, 2))
```

```r
plot(roc_mod_lasso, print.auc=TRUE, print.thres = TRUE,
     col="lightblue",main="AUC for lasso model")

plot(roc_mod_ridge, print.auc=TRUE, print.thres = TRUE,
     col="lightpink",main="AUC for ridge model")

plot(roc_mod_logistic, print.auc=TRUE, print.thres = TRUE,
     col="orange", main="AUC for logistic risk model")

plot(roc_best_subset, print.auc=TRUE, print.thres = TRUE,
     col="mediumpurple",main="AUC for best subset model")
evaluation_values = function(pred,y.test,threshold){
  #' get AUC, sensitivity, specificity, accuracy, precision and Brier scorevalues of the fitted model
  #' @param pred, the prediction values
  #' @param y.test, the labels of test dataset
  #' @param threshold, a numeric number of threshold to classify the probability to classes
  #' @return sensitivity, specificity, accuracy, and precision values


  df <- data.frame(pred=as.numeric(pred>threshold),label=as.numeric(y.test)-1)
  TP <- dim(df[(df$pred==1&df$label==1),])[1]
  TN <- dim(df[(df$pred==0&df$label==0),])[1]
  FP <- dim(df[(df$pred==1&df$label==0),])[1]
  FN <- dim(df[(df$pred==0&df$label==1),])[1]
  return(c(sensitivity=TP/(TP+FN),specificity=TN/(TN+FP),
           accuracy=(TP+TN)/(TP+TN+FP+FN),precision=TP/(TP+FP)))
}

lasso_evaluation_values = evaluation_values(lasso_prediction_value, test_dataset_long$outcome_result, t

ridge_evaluation_values = evaluation_values(ridge_prediction_value, test_dataset_long$outcome_result, t

best_subset_evaluation_values = evaluation_values(best_subset_predict_value, test_dataset_long$outcome_

logistic_evaluation_values = evaluation_values(logistic_prediction_value, test_dataset_long$outcome_resu
temp = rbind(lasso_evaluation_values, ridge_evaluation_values, best_subset_evaluation_values, logistic_

AUC_values = c(0.886, 0.890, 0.895, 0.884)

temp = cbind(temp, AUC_values)

lasso_BrierScore = mean((lasso_prediction_value-(as.numeric(as.numeric(test_dataset_long$outcome_result)

ridge_BrierScore = mean((ridge_prediction_value-(as.numeric(as.numeric(test_dataset_long$outcome_result)

logistic_BrierScore = mean((logistic_prediction_value-(as.numeric(as.numeric(test_dataset_long$outcome_

best_subset_BrierScore = mean((best_subset_predict_value-(as.numeric(as.numeric(test_dataset_long$outcor

BrierScore = c(as.numeric(lasso_BrierScore), as.numeric(ridge_BrierScore), as.numeric(best_subset_Brier

temp = cbind(temp, BrierScore)
temp = as.data.frame(temp)
```

```r
rownames(temp) = c("Lasso regression", "Ridge regression", "Best subset", "Logistic")
temp %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Evaluation metrics for four different model",
      col.names=linebreak(c("sensitivity", "specificity", "accuracy",
                            "precision", "AUC value", "Brier score")),
      booktabs=T, escape=F, align = "c") %>%
kable_styling(full_width = FALSE, latex_options = c('hold_position'))

number_bins = 10

lasso_calibration = data.frame(prob = lasso_prediction_value,
                                    bin = cut(lasso_prediction_value, breaks = number_bins),
                                    observed = as.numeric(test_dataset_long$outcome_result)-1)

lasso_calibration = lasso_calibration %>%
  group_by(bin) %>%
  dplyr::summarize(observed = sum(observed)/n(),
                   expected = sum(prob)/n(),
                   se = sqrt(observed*(1-observed)/n()))

p1 = ggplot(lasso_calibration) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion",title = "Calibration Plot for Lasso") +
  theme_minimal()



ridge_calibration = data.frame(prob = ridge_prediction_value,
                                    bin = cut(ridge_prediction_value, breaks = number_bins),
                                    observed = as.numeric(test_dataset_long$outcome_result)-1)

ridge_calibration = ridge_calibration %>%
  group_by(bin) %>%
  dplyr::summarize(observed = sum(observed)/n(),
                   expected = sum(prob)/n(),
                   se = sqrt(observed*(1-observed)/n()))

p2 = ggplot(ridge_calibration) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion",title = "Calibration Plot for Ridge") +
  theme_minimal()
```

```r
logistic_calibration = data.frame(prob = logistic_prediction_value,
                                  bin = cut(logistic_prediction_value, breaks = number_bins),
                                  observed = as.numeric(test_dataset_long$outcome_result)-1)

logistic_calibration = logistic_calibration %>%
  group_by(bin) %>%
  dplyr::summarize(observed = sum(observed)/n(),
                   expected = sum(prob)/n(),
                   se = sqrt(observed*(1-observed)/n()))

p3 = ggplot(logistic_calibration) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion",title = "Calibration Plot for Logistic Regressi
  theme_minimal()


best_subset_calibration = data.frame(prob = best_subset_predict_value,
                                  bin = cut(best_subset_predict_value, breaks = number_bins),
                                  observed = as.numeric(test_dataset_long$outcome_result)-1)

best_subset_calibration = best_subset_calibration %>%
  group_by(bin) %>%
  dplyr::summarize(observed = sum(observed)/n(),
                   expected = sum(prob)/n(),
                   se = sqrt(observed*(1-observed)/n()))

p4 = ggplot(best_subset_calibration) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion",title = "Calibration Plot for Best subet Model"
  theme_minimal()

grid.arrange(p1, p2, p3, p4, nrow = 2, ncol = 2)
```