Predicting Tracheostomy and Mortality in Neonates with Severe Bronchopulmonary Dysplasia

Liangkang Wang

2023-11-12

Objective: This study aims to compare the performance of Lasso and Stepwise logistic regression models in predicting critical outcomes in neonatal care, focusing on variable importance and model calibration.

Methods: We applied Lasso and Stepwise logistic regression models to a clinical dataset, evaluating their predictive accuracy through ROC-AUC plots, Brier Scores, and Hosmer-Lemeshow tests. Variable importance was assessed by examining the models' coefficients and p-values. The Multiple Imputation by Chained Equations (MICE) method addressed missing data, ensuring robust parameter estimation.

Results: Both models demonstrated high discriminative power with AUC values of 0.94. The Stepwise model identified several significant predictors, including levels of ventilation support and birth weight, with p-values less than 0.05. The Lasso model produced a more parsimonious set of predictors due to its regularization process. Reliability diagrams indicated good calibration across most predicted probability ranges, with areas for potential improvement identified in both models.

Conclusions: The Lasso and Stepwise models exhibited strong predictive capabilities and were well-calibrated to the observed data. The Lasso model's penalization favored simplicity and generalizability, while the Stepwise model provided a more detailed predictor set. Both models' emphasis on respiratory support parameters and birth weight aligns with clinical expectations, indicating their potential utility in informing neonatal care decisions. The study highlights the importance of balancing model complexity and interpretability in clinical predictive analytics.

Introduction

Bronchopulmonary dysplasia (BPD), the most common and severe complication of prematurity, impacts between 10,000 to 15,000 infants annually, posing a significant challenge to

neonatal intensive care units worldwide. Characterized by lung fibrosis, metaplasia, and a consequential "simplified lung" architecture, BPD's etiology is multifactorial with a notable influence of individual genetic and epigenetic susceptibilities. The management of severe bronchopulmonary dysplasia (sBPD) remains a contentious issue in neonatology, particularly concerning the timing and indications for tracheostomy. While tracheostomy may enhance growth and development by alleviating respiratory workload, its optimal timing is still debated. Prior research has attempted to predict tracheostomy placement or mortality through demographic and clinical diagnosis variables, but these have not encompassed the detailed respiratory parameters critical to informed decision-making.

In collaboration with Dr. Chris Schmid from the Biostatistics Department, this study proposes to develop a predictive model that integrates comprehensive demographic, diagnostic, and nuanced respiratory parameters to predict the likelihood of tracheostomy or death in neonates with sBPD. By offering predictions at varying postmenstrual ages (PMA), this model endeavors to support clinical decisions, aid in family counseling, and potentially improve the timing of tracheostomy procedures.

This paper details the development and validation of a regression model designed for such predictions, emphasizing the impact of key variables and the interpretation of model outcomes for different patient subsets. In doing so, it aims to contribute significantly to the body of knowledge surrounding sBPD management and to foster an evidence-based approach to the care of these vulnerable neonates.

Methods

Data Collection

Data Source and Recruitment Population

This study utilized a comprehensive and robust dataset from the BPD Collaborative Registry, a multi-center consortium across the United States and Sweden. The inclusion criteria for the study were strictly defined: infants with a gestational age of less than 32 weeks and diagnosed with sBPD, as specified by the 2001 NHLBI criteria. This criteria included the need for either a fraction of inspired oxygen (FiO2) greater than 0.3 or some form of positive pressure ventilation at 36-weeks postmenstrual age (PMA). The collected data encompassed vital demographic and clinical information at key developmental milestones - birth, 36 weeks PMA, 44 weeks PMA, and at discharge. This study specifically extracted data for infants with complete growth records and a diagnosis of BPD, during the period from January 1 to July 19, 2021. At the time of this study, data from 10 BPD Collaborative centers were analyzed.

Data Description

The summary of variables definition can be found in @table-variable. We separated the 30 variables into 5 groups, which is easy to make regression later.

Demographic Variables: These include patient and maternal characteristics that are fixed, such as race, ethnicity, and potentially the medical center if it's related to the patient's location. `record_id`: A unique identifier for each patient. `center`: The medical center where the patient was treated, which may reflect geographic or institutional differences in patient populations and care practices. `mat_race`: The race of the mother, which may be relevant in studying health disparities or genetic factors that could influence patient outcomes. `mat_ethn`: The ethnicity of the mother, which, like race, can be important for understanding health disparities.

Baseline Medical Variables: Variables that are related to the patient's condition at birth, such as birth weight, gestational age, birth length, and head circumference. `bw`: Birth weight in grams, a critical indicator of neonatal health and a predictor of future medical needs. `ga`: Gestational age in weeks, which is central to understanding the development stage of the newborn and potential complications. `blength`: Birth length in centimeters, which can be another indicator of neonatal health. `birth_hc`: Head circumference at birth, which is used to assess fetal growth and development.

Medical Intervention Variables: These cover interventions or treatments the patient may have received, such as the method of delivery or prenatal steroids. `del_method`: The method of delivery (e.g., vaginal, cesarean), which can influence the risk of complications for both the mother and the infant. `prenat_ster`: Indicates whether prenatal steroids were administered, which can be a factor in lung development and other outcomes for premature infants.

Physiological Measurements The dataset contains variables with appended time indicators such as .36 and .44, which denote measurements taken at specific times.

`weight_today`: Patient's weight at specified time. `ventilation_support_level_modified`: Level of ventilation support at specified time. `inspired_oxygen`: Fraction of inspired oxygen at specified time. `p_delta`: Peak Inspiratory Pressure (cmH2O) needed at specified time. `peep_cm_h2o_modified`: Positive end exploratory pressure (cm H2O) needed at specified time. `med_ph`: Medication for pulmonary hypertension at specified time.

Outcome Variable The dataset under consideration includes two critical variables: `Trach`, indicating whether the infants underwent tracheostomy, and `Death`, denoting mortality status. For the purposes of this project, these variables will be amalgamated to create a composite outcome measure `Outcome`. This derived binary variable will serve as the dependent variable in our predictive models, enabling us to ascertain the likelihood of either tracheostomy or death in the infant population under study.

Indicator Variable We transformed `hosp_dc_ga`, denoting the infants' discharging time from hospital, into an indicator variable `discharge_after_44`. If infants were discharged before 44 weeks, then it is 0. Otherwise, we set it equals to 1.

Data Preprocessing

Outliers and Data Cleaning

An initial examination of the dataset revealed four duplicate entries. These were removed, resulting in a dataset comprising 996 unique observations suitable for our analysis. Furthermore, `center` 21 contained only a single patient observation, which was excluded to ensure robust categorical analysis of the `center` variable, a critical factor discussed further in following section. The `center` variable's stratification is essential due to its potential influence on the infants' measurements and treatment outcomes. Additionally, the dataset included three infants discharged before reaching 36 weeks of gestational age, yet they had recorded measurements at the 36-week mark. This suggests follow-up visits, which is plausible given the critical developmental stage at this age. Consequently, these cases have been categorized with infants discharged before 36 weeks.

The `mat_race` variable was excluded from our analysis due to discrepancies with the codebook records. However, the `mat_ethn` variable is retained as it similarly reflects maternal race information. We also deleted `any_surf` column because the percentage of missing value in this column is over 45%, which leads to the bad performance in multiple imputation.

Handling Missing Values

The visualization of missing data via a heatmap Figure 1 unveiled a discernible pattern of missingness, particularly among variables measured at 44 weeks post-menstrual age (PMA). A closer investigation into the dataset indicated that a substantial proportion of infants were discharged prior to reaching 44 weeks PMA. This early discharge accounts for the absence of data at the 44-week measurement interval, as it is not applicable to infants who have already left the care facility. Of the 313 infants discharged before 44 weeks PMA, few had any records for the variables related to this time point, which suggests that these scant data entries can be disregarded for the purpose of analysis. Within this particular cohort, there are minimal missing values pertaining to variables associated with the 36-week mark, indicating a higher completeness of data for this earlier period in the dataset.

In response to these insights, we have segmented the dataset based on the variable `hosp_dc_ga`, delineating records with values less than 43.9 weeks from those with values greater than or equal to 43.9 weeks. Furthermore, for cases with missing `hosp_dc_ga` values, we utilized the absence of data in 44-week-specific variables to categorize them into groups of less than 44 weeks or greater than or equal to 44 weeks. Pre-segmentation, the mean missingness across measurement variables at the 44-week interval stood at approximately 45% (as detailed in @table-variable). Post-segmentation, the cohort discharged after 44 weeks evidenced a reduced missingness rate of approximately 15%, rendering it more amenable for imputation via the `mice` package. Conversely, the early-discharge cohort evidenced an escalated missingness rate of 97%.

Considering the huge difference between two groups of infants, then we transformed `hosp_dc_ga` into a new variable `discharged_after_44` which contains 0 or 1, and dropped the `hosp_dc_ga`. It is a indicator variable which functions a lot in our following model. We utilize the `mice` package for multiple imputation to address the missing data within each group of patients. In the `discharged_after_44`equals to 0 group, we dropped all of the measurement variables at 44 weeks. But in the other group, we retained these variables. After getting 10 imputed data from mice (5 for each group of infants), we fuse the two groups together and finally get 5 completed datasets to fit our model.

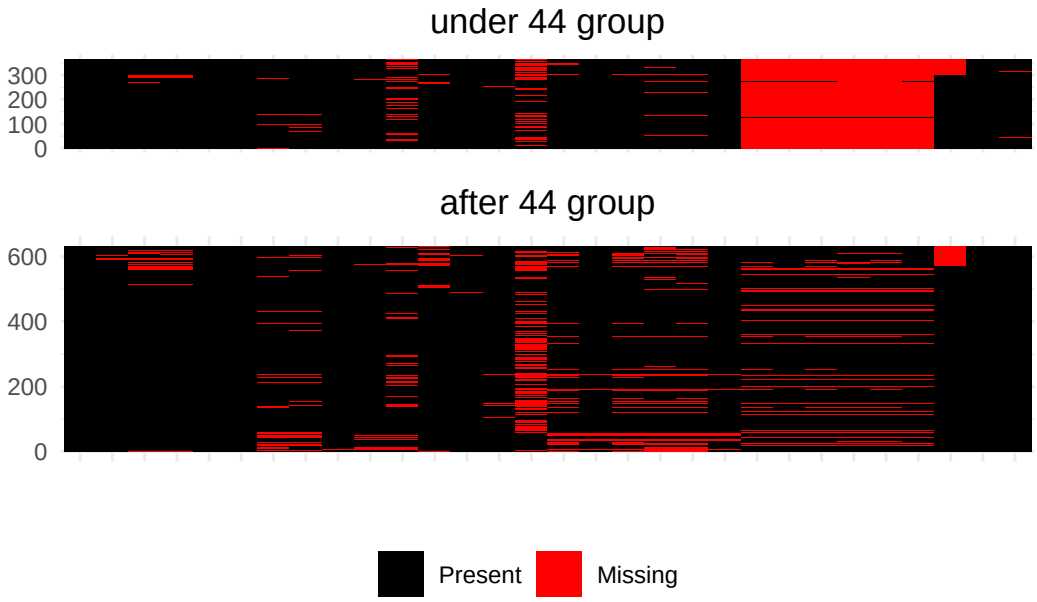

Figure 1: Heatmap of missing values

Handling of Different Sites

There are 10 different sites in our dataset. However, the `center` 21 has only 1 observation, and the `center` 20 has 4 observations. Considering the fact that we will split the dataset to do a cross-validation in model fitting process, we delete these 5 observations to make we have supplenty of observations in detecting the impact of different sites.

Model Development and Evaluation

Our objective is to construct a robust logistic regression model that enables us to predict the likelihood of infants requiring a tracheostomy or facing mortality. To ensure the model's

validity, we will undertake a thorough validation process using a test dataset. This step is crucial to assess the model's generalizability. Additionally, to mitigate the risk of overfitting, we will implement cross-validation techniques.

In our methodology, the model is independently applied to each dataset following multiple imputation. For the LASSO model, coefficients from each model are pooled, and a combined evaluation and calibration are conducted on the five imputed validation sets. Conversely, with the stepwise forward selection model, varying coefficients are selected across the five imputed datasets, complicating the pooling of coefficients. Therefore, we utilize all five models to predict on the aggregated validation sets, averaging these predictions for our final outcome. This method, treating the five models as integral components of the overall model, enhances the reliability and accuracy of predictions. Each new observation is input into these five models, and the average of these predictions is used. This approach, particularly suitable for stepwise forward selection where predictor variables may vary, offers greater flexibility and applicability across different models. This results in a robust final model, leveraging averaged results from the five datasets.

Variable Selection

Following data cleaning, we identified 25 variables for consideration. These were selected based on the patterns of missing data and the variable transformations discussed in the previous section. Our comprehensive model formula is presented below:

$$
\begin{aligned}
\log\left(\frac{p}{1-p}\right) = {} & \beta_0 + \beta_1 \times \text{center} + \beta_2 \times \text{mat\_ethn} + \beta_3 \times \text{bw} + \\
& \beta_4 \times \text{ga} + \beta_5 \times \text{blength} + \beta_6 \times \text{birth\_hc} + \\
& \beta_7 \times \text{del\_method} + \beta_8 \times \text{prenat\_ster} + \beta_9 \times \text{com\_prenat\_ster} + \\
& \beta_{10} \times \text{mat\_chorio} + \beta_{11} \times \text{gender} + \beta_{12} \times \text{sga} + \\
& \beta_{13} \times \text{weight\_today.36} + \beta_{14} \times \text{ventilation\_support\_level.36} + \\
& \beta_{15} \times \text{inspired\_oxygen.36} + \beta_{16} \times \text{p\_delta.36} + \beta_{17} \times \text{peep\_cm\_h2o.36} + \\
& \beta_{18} \times \text{med\_ph.36} + \beta_{19} \times \text{I(discharge\_44)} \times \text{weight\_today.44} + \\
& \beta_{20} \times \text{I(discharge\_44)} \times \text{ventilation\_support\_level.44} + \\
& \beta_{21} \times \text{I(discharge\_44)} \times \text{inspired\_oxygen.44} + \\
& \beta_{22} \times \text{I(discharge\_44)} \times \text{p\_delta.44} + \\
& \beta_{23} \times \text{I(discharge\_44)} \times \text{peep\_cm\_h2o.44} + \\
& \beta_{24} \times \text{I(discharge\_44)} \times \text{med\_ph.44} + \beta_{25} \times \text{discharge\_44}
\end{aligned} \tag{1}
$$

To streamline our model and focus on the most influential variables, we employed two variable selection techniques: lasso regression and stepwise forward selection. The effectiveness of these methods was evaluated using measures of discrimination and calibration.

Training Process

Lasso Model Training: We utilized the `glmnet` package for implementing the lasso model. A key step in this process involved using cross-validation, integrated within the `cv.glmnet` function, to ascertain the optimal lambda value. This value was then applied to train the lasso model using the entire training dataset. As a result of this approach, we obtained five slightly varying lasso models. Each of these models was used to generate predictions on validation sets, which were subsequently averaged. The aggregated output of these models constitutes the final model presented in this study.

Stepwise Forward Selection: For the stepwise forward selection, we employed the `stats` and `MASS` packages. Each imputed dataset underwent a ten-fold cross-validation process. During each iteration, we calculated the Brier score to evaluate model performance. The model yielding the smallest Brier score was selected for its superior variable set, which was then used to fit a new logistic regression model to the entire training dataset. This procedure was repeated across all imputed datasets, resulting in five distinct models, analogous to our approach with the lasso models. Predictions from these models were made on the validation sets and then averaged, leading to our final predictive model. This methodology ensures a comprehensive and robust approach to model training, blending insights from multiple imputations and variable selection strategies.

Model Evaluation

Evaluation Metrics

ROC-AUC curve: The ROC curve illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. AUC is a single number summary of the ROC curve, ranging from 0 to 1.

Calibration Plot (Reliability Diagram): A calibration plot shows the relationship between predicted probabilities and actual outcomes. It typically involves grouping predicted probabilities into bins and plotting the mean predicted probability against the observed frequency of outcomes for each bin. For a well-calibrated model, the points should fall on or near the diagonal line.

Hosmer-Lemeshow Test: The Hosmer-Lemeshow test is to determine the goodness of fit for logistic regression models. It groups observations into deciles based on their predicted probabilities and then compares the number of observed and expected events in each decile. A high p-value (typically >0.05) indicates that the model's predictions are not significantly different from the observed outcomes, suggesting good fit.

Brier Score: The Brier score is a measure of the accuracy of probabilistic predictions. It calculates the mean squared difference between the predicted probabilities and the actual

binary outcomes (0 or 1). The score ranges from 0 for a perfect model to 1 for the worst model.

$$\text{Brier Score (BS)} = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2 \tag{2}$$

where, N is the total number of predictions (or the size of your dataset). $f_i$is the predicted probability for instance $o_i$is the actual outcome for instance i, typically 0 or 1 in binary classification.

Generalization of the Model

For assessing the generalization capability of our model, we divided our imputed dataset into a training set (70% of the data), and a validation set (the remaining 30%).We employed tools such as the ROC-AUC curve analysis, Calibration Plot (Reliability Diagram), Hosmer-Lemeshow Test, and Brier Score, specifically applied to the validation dataset. These methods provided a comprehensive evaluation of the model's performance, as detailed in the final paragraph of our paper.

Results

Model Performance

Lasso regression

In our study, we utilized cross-validation to identify the optimal lambda value for our lasso regression model, as depicted in Figure 2. This figure illustrates the binomial deviance, a measure of cross-validated error, plotted against the logarithm of lambda values. The optimal lambda is thus selected at the juncture where the deviance is minimized, ensuring the best fit for the model without incurring the penalties of underfitting or overfitting.

Stepwise forward selection

In our study, we utilized cross-validation in fitting the stepwise forward selectionof logistic regression model. In our assessment of five logistic regression models, each subjected to 10-fold cross-validation, we analyzed their predictive accuracy using the mean and the lowest Brier Scores. The mean CV Brier Score represents the average performance across the folds, while the lowest Brier Score indicates the model's best performance in any single fold.

Table 1: Variables Definition

Variable	Definition	Missing
record_id	Patient ID	0%
center	Medical Center	1%
mat_race	Maternal Race 1, American Indian or Alaskan Native | 2, Asian | 3, Black or African American | 4, Native Hawaiian or Other Pacific Islander | 5, White | 6, Other	5.62%
mat_ethn	Maternal Ethnicity 1, Hispanic or Latino | 2, Not Hispanic or Latino	5.72%
bw	Birth weight (g)	0%
ga	Obstetrical gestational age	0%
blength	Birth length (cm)	7.83%
birth_hc	Birth head circumference (cm)	7.73%
del_method	Delivery Method 1, Vaginal delivery | 2, Cesarean section	0.3%
prenat_ster	Prenatal Corticosteroids 1, Yes | 2, No | 3, Unknown	3.51%
com_prenat_ster	Complete Prenatal Steroids 1, Yes | 2, No | 3, Unknown	19.38%
mat_chorio	Maternal Chorioamnionitis 1, Yes | 2, No | 3, Unknown	6.22%
gender	Gender 1, Male | 2, Female | 3, Ambiguous	0.4%
sga	Was the infant small for gestational age?	1.51%
any_surf	Did the infant receive surfactant at any point in the first 72 hours? 1, Yes | 2, No | 3, Uknown	43.47%
weight_today.36	Weight at 36 weeks	9.24%
ventilation_support_level.36	Ventilation support level at 36 weeks 0=No respiratory support or supplemental oxygen; 1 = Non-invasive positive pressure; 2 = Invasive positive pressure	3.01%
inspired_oxygen.36	Fraction of Inspired Oxygen at 36 weeks	9.24%
p_delta.36	Peak Inspiratory Pressure (cmH2O) at 36 weeks	12.85%
peep_cm_h2o.36	Positive and exploratory pressure (cm H2O) at 36 weeks	11.75%
med_ph.36	Medication for Pulmonary Hypertension at 36 weeks	3.01%
weight_today.44	Weight at 44 weeks	44.78%
ventilation_support_level.44	Ventilation support level at 44 weeks 0=No respiratory support or supplemental oxygen; 1 = Non-invasive positive pressure; 2 = Invasive positive pressure	42.57%
inspired_oxygen.44	Fraction of Inspired Oxygen needed at 44 weeks	44.98%
p_delta.44	Peak Inspiratory Pressure (cmH2O) needed at 44 weeks	44.98%
peep_cm_h2o.44	Positive end exploratory pressure (cm H2O) needed at 44 weeks	44.78%
med_ph.44	Medication for Pulmonary Hypertension at 44 weeks	42.57%
hosp_dc_ga	Hospital Discharge Gestational Age	12.45%
Trach	Trachoestomy 0 = no; 1 = yes	0%
Death	Death No, Yes	0.2%

Model 1 demonstrated a mean CV Brier Score of 0.09329984 and an impressive lowest Brier Score of 0.0389261897802554, suggesting a high potential for accuracy under optimal conditions despite less consistent average performance. Model 2, with a mean CV Brier Score of 0.08827815 and a lowest Brier Score of 0.0527195190114325, showed a balance between consistency and peak performance. Model 3 revealed a solid performance, having a mean CV Brier

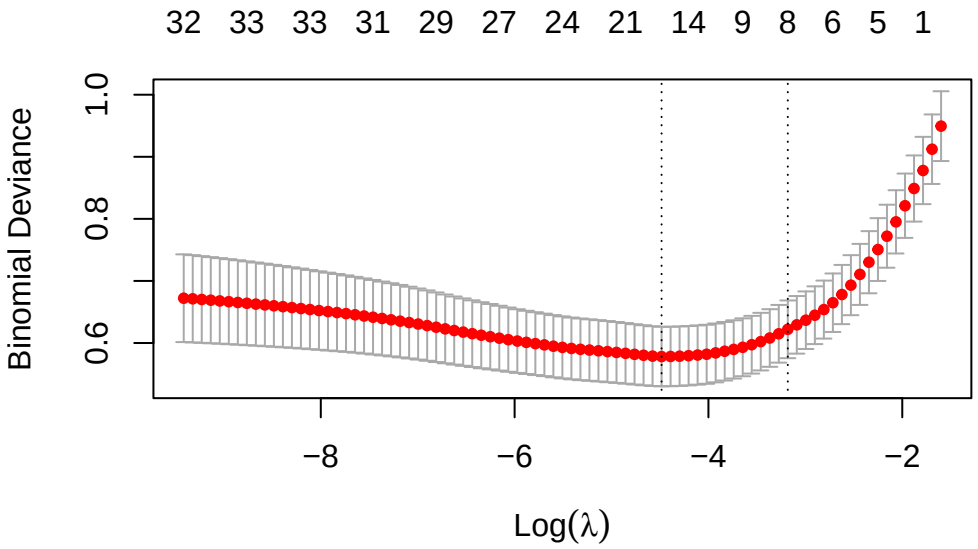

Figure 2: Cross-Validation Plot for Lambda Selection

Score of 0.09230342 and a notably lower lowest Brier Score of 0.045121433789308, indicating effective accuracy with potential for exceptional outcomes. Model 4, although exhibiting the highest mean CV Brier Score at 0.09469706, displayed promising potential with a lowest Brier Score of 0.0517892821333567. Lastly, Model 5 had a moderate mean CV Brier Score of 0.09075372 and the highest lowest Brier Score of 0.0708738243258727, suggesting it was less consistent and had lower peak performance compared to the other models.

These results highlight varying levels of consistency and peak accuracy among the models. Lower mean CV Brier Scores generally indicate better average performance, while lower lowest Brier Scores reveal the models' capabilities for high accuracy in the most favorable scenarios. The ideal model would excel in both metrics, combining reliable performance across different conditions with the potential for high accuracy.

Validation Set Results

In the comparative performance analysis section, the Lasso and Stepwise logistic regression models were evaluated on their predictive capabilities and the congruence between their estimated probabilities and observed outcomes.

The ROC-AUC plot Figure 3 reveals that both models possess strong discriminative abilities, each achieving an AUC of 0.9, showcasing their proficiency in distinguishing between the posi-

tive and negative classes. This high level of accuracy is reflected in their ability to consistently rank predictions with a high true positive rate while maintaining a low false positive rate.

The model evaluation metrics @table-metrics further substantiates their performance. The Lasso model, with a Hosmer-Lemeshow p-value of 0.3893497, and the Stepwise model, with a p-value of 0.3293648, both indicate well-fitting models. The chi-squared statistics, 8.465696 for Lasso and 9.155303 for Stepwise, are in close proximity to each other, suggesting that the differences in the predicted probabilities and the actual outcomes are not significant. Additionally, the Brier Scores, 0.0769623 for Lasso and 0.0767501 for Stepwise, confirm the accuracy of the models, with the Stepwise model displaying a marginally better score.

The Reliability Diagrams Figure 4 add depth to our understanding of model calibration. For the Lasso model, the diagram depicts a good alignment with the ideal calibration line across most predicted probabilities, with some notable exceptions. These exceptions point towards opportunities for improving the model's estimates. The Stepwise model's diagram exhibits a similar pattern, with generally well-aligned predictions but with oscillations in the mid-probability range that suggest variability in the model's certainty.

Collectively, the analyses from the ROC-AUC plot, evaluation metrics table, and Reliability Diagrams provide a thorough evaluation of the models. Both models demonstrate high levels of predictive accuracy, and the reliability diagrams indicates that while there is room for enhancement.

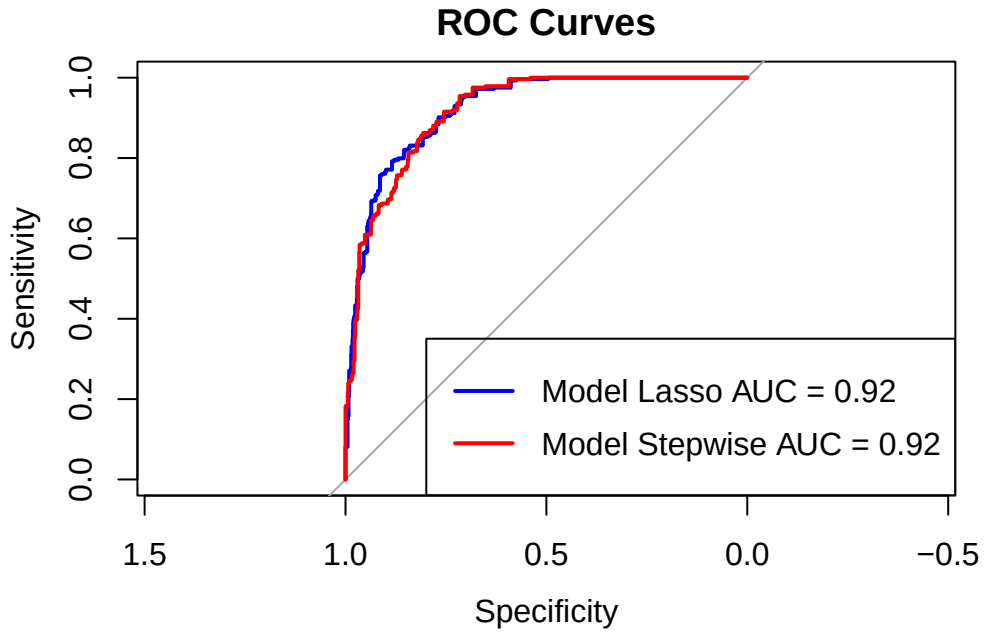

Figure 3: ROC-AUC plot for lasso and stepwise models

Table 2: Hoslem Test and Brier Score

Model	P_Value	Statistic	Brier_Score
Lasso	0.0007302	26.91878	0.0859788
Stepwise	0.0001178	31.42955	0.0894086

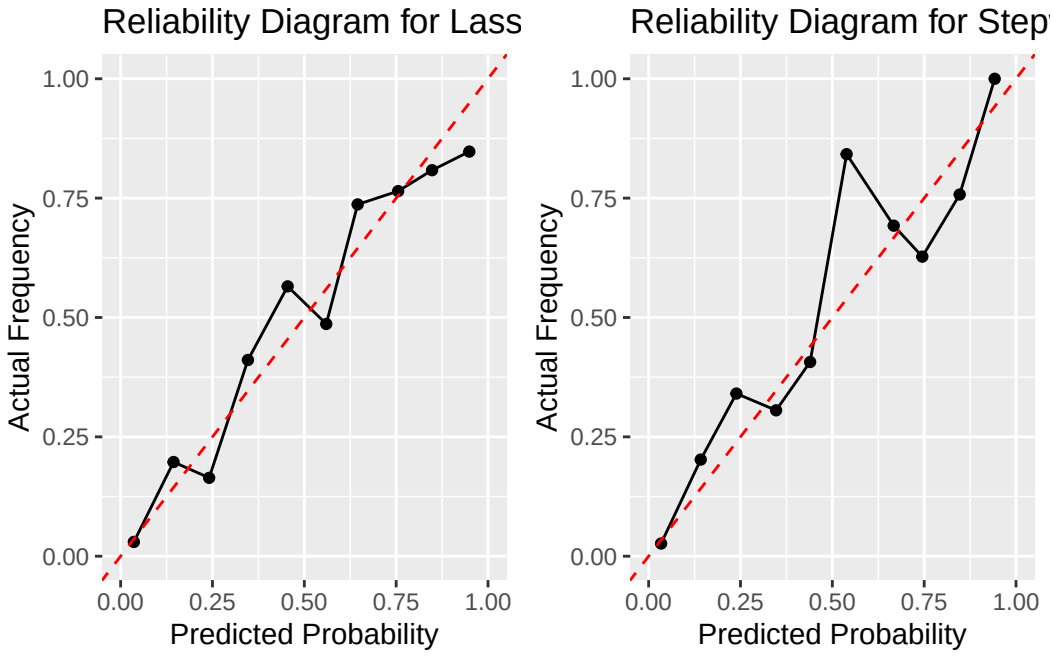

Figure 4: Reliability Diagram

Variable Importance & Model Interpretation

The variable importance analysis for the Lasso and Stepwise logistic regression models, as summarized by their coefficients and significance levels (can be found in the appendix), reveals the factors most influential in predicting the outcome variable.

For the Stepwise model, several predictors stand out as statistically significant, as indicated by the p-values. Notably, 'ventilation_support_level.362', 'center2', 'center3', 'inspired_oxygen.36', 'sgaSGA', and 'bw' show a strong association with the outcome, with p-values less than 0.05, denoting that these variables play a crucial role in the model. The interaction term 'discharge_44:ventilation_support_level.442' is particularly significant, suggesting that the effect of ventilation support level at discharge at 44 weeks is a key predictor.

In contrast, the Lasso model's coefficients, which are the result of penalized regression, show a sparser selection of variables, indicating a more parsimonious model. Notable non-zero coefficients include a strong negative intercept, which sets the baseline log odds of the outcome, and positive coefficients for 'ventilation_support_level.362', 'inspired_oxygen.36', and the interaction 'discharge_44:ventilation_support_level.442'. These variables are also identified as important in the Stepwise model, reaffirming their significance.

The convergence on key variables by both models underscores their importance. The 'ventilation_support_level' in both its forms (361 and 362) and the 'inspired_oxygen' at 36 weeks are consistent across models, highlighting the relevance of respiratory support parameters. The variable 'bw' (birth weight) also emerges as a common significant predictor, reflecting the clinical importance of birth weight in the outcome being studied.

Interestingly, while both models identify similar variables as important, the Lasso model, by nature, tends to enforce sparsity, which might explain why some variables that appear significant in the Stepwise model do not show up in the Lasso model. This could be due to the Lasso's regularization process, which penalizes the less important variables more heavily.

In summary, the Stepwise model emphasizes the importance of various centers, levels of ventilation support, and birth weight, among others, whereas the Lasso model corroborates the importance of these variables while presenting a more streamlined set of predictors. This analysis not only guides the understanding of the factors most predictive of the outcome but also aids in model selection based on the complexity and interpretability of the variables involved.

Discussion

The comparison between the Lasso and Stepwise models reveals differences in model complexity and variable selection, influenced by the penalization inherent to Lasso regression. This penalization often results in a more parsimonious model, which can be advantageous in terms

of model interpretability and generalizability, but may sacrifice some detail that Stepwise selection retains.

Both models' findings have important implications. The significant variables identified align with clinical expectations, affirming the models' potential utility in a healthcare setting. However, the presence of significant interaction terms indicates complex relationships between variables that require careful consideration when applying these models in practice.

The results from the Lasso and Stepwise models have practical implications for neonatal care, particularly in understanding the factors that influence critical outcomes. The identification of key predictors can inform clinical decision-making and the allocation of healthcare resources. Furthermore, the models' emphasis on certain variables, like ventilation support, suggests areas for targeted intervention and research.

In summary, the Stepwise model offers a detailed, albeit potentially more complex, view of the factors influencing outcomes, while the Lasso model provides a streamlined and potentially more generalizable perspective. The choice between models should consider the specific clinical context and the trade-offs between complexity and interpretability.ts

Appendices

```
Call:
glm(formula = best_formula, family = binomial(), data = train_set)

Coefficients:
                                            Estimate Std. Error z value
(Intercept)                                -2.2346776  1.1525886  -1.939
ventilation_support_level.361               0.1098192  0.5711425   0.192
ventilation_support_level.362               1.4185068  0.5782754   2.453
center2                                    -1.2838331  0.5454328  -2.354
center3                                    -0.9789364  0.9186840  -1.066
center4                                     0.0200641  0.7587989   0.026
center5                                     0.0697349  0.8239315   0.085
center7                                    -1.2631264  1.3129469  -0.962
center12                                    1.3594094  0.6326741   2.149
center16                                   -1.2093511  1.1875379  -1.018
inspired_oxygen.36                          1.7889343  0.8842405   2.023
weight_today.36                            -0.0015208  0.0004079  -3.728
bw                                          0.0015014  0.0005108   2.939
mat_ethn2                                   0.9266641  0.5742729   1.614
discharge_44                               -0.7218973  0.5148502  -1.402
com_prenat_sterYes                          0.7624367  0.3374630   2.259
discharge_44:ventilation_support_level.441 -2.0900857  1.2627998  -1.655
discharge_44:ventilation_support_level.442 -0.8136166  1.3120124  -0.620
discharge_44:med_ph.441                     1.3474617  0.4086796   3.297
discharge_44:peep_cm_h2o.44                 0.3611765  0.1491446   2.422
                                           Pr(>|z|)
(Intercept)                                0.052522 .
ventilation_support_level.361              0.847523
ventilation_support_level.362              0.014167 *
center2                                    0.018583 *
center3                                    0.286611
center4                                    0.978905
center5                                    0.932550
center7                                    0.336022
center12                                   0.031660 *
center16                                   0.308503
inspired_oxygen.36                         0.043060 *
weight_today.36                            0.000193 ***
bw                                         0.003292 **
```

```
mat_ethn2                                    0.106608
discharge_44                                 0.160870
com_prenat_sterYes                           0.023864 *
discharge_44:ventilation_support_level.441 0.097900 .
discharge_44:ventilation_support_level.442 0.535173
discharge_44:med_ph.441                      0.000977 ***
discharge_44:peep_cm_h2o.44                  0.015450 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 657.56  on 693  degrees of freedom
Residual deviance: 344.02  on 674  degrees of freedom
AIC: 384.02

Number of Fisher Scoring iterations: 6


34 x 1 sparse Matrix of class "dgCMatrix"
                                                    s0
(Intercept)                                -3.8034711450
center2                                    -0.6005234442
center3                                     .
center4                                     0.2768568801
center5                                     .
center7                                     .
center12                                    1.2506818788
center16                                    .
mat_ethn2                                   0.1468583946
bw                                          .
ga                                          0.0498307464
blength                                     .
birth_hc                                    0.0118173960
del_method2                                 .
prenat_sterYes                              .
com_prenat_sterYes                          0.2147601193
mat_chorioYes                               .
genderMale                                  .
sgaSGA                                      0.0273028258
weight_today.36                            -0.0005762026
ventilation_support_level.361               .
ventilation_support_level.362               0.9339887090
```

```
inspired_oxygen.36                            1.6090778991
p_delta.36                                    .
peep_cm_h2o.36                                .
med_ph.361                                    .
discharge_44                                  .
weight_today.44:discharge_44                  .
discharge_44:ventilation_support_level.441    .
discharge_44:ventilation_support_level.442    1.1042819527
discharge_44:inspired_oxygen.44               .
discharge_44:p_delta.44                       .
discharge_44:peep_cm_h2o.44                   0.0775635234
discharge_44:med_ph.441                       1.1248834895
```

```r
library(knitr)
library(tidyr)
library(formatR)
library(kableExtra)
library(mice)
library(reshape2)
library(tidyverse)
library(gtsummary)
library(gt)
library(cowplot)
library(pROC)
library(MASS)
library(glmnet)
library(boot)
library(rjtools)
library(formatR)
library(rms)
library(ResourceSelection)


opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warning = FALSE, message = FALSE, echo = FALSE, fig.align = "center"
set.seed(123)
data <- read.csv("project2.csv")

data <- unique(data)
```

```r
colnames(data)[23] <- "ventilation_support_level.44"
colnames(data)[20] <- "peep_cm_h2o.36"
colnames(data)[26] <- "peep_cm_h2o.44"

data_under_44 <- data%>%filter(hosp_dc_ga<43.9)

data_after_44 <- data%>%filter(hosp_dc_ga>=43.9)

data_disna_44 <- data%>%filter(is.na(hosp_dc_ga))

data_disna_under_44 <- data_disna_44 %>%
  filter(is.na(weight_today.44),is.na(ventilation_support_level.44),
         is.na(inspired_oxygen.44),is.na(p_delta.44),
         is.na(peep_cm_h2o.44),is.na(med_ph.44))

data_disna_after_44 <- data_disna_44 %>%
  filter(!(is.na(weight_today.44)&is.na(ventilation_support_level.44)&
             is.na(inspired_oxygen.44)&is.na(p_delta.44)&
             is.na(peep_cm_h2o.44)&is.na(med_ph.44)))

data_under_44 <- rbind(data_under_44,data_disna_under_44)
data_after_44 <- rbind(data_after_44,data_disna_after_44)
missing_heatmap <- function(data,title){
  missing_values <- is.na(data)

# Melt the matrix for use with ggplot
missing_melted <- reshape2::melt(missing_values,
                                 id.vars = rownames(missing_values))

# Create the heatmap
g <- ggplot2::ggplot(missing_melted, aes(x = Var2, y = Var1)) +
  geom_tile(aes(fill = value)) +
  scale_fill_manual(name = "", labels = c("Present", "Missing"),
                    values = c("black", "red")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Variables", y = "Observations",title = title)+
  theme(plot.title = element_text(hjust = 0.5))
return(g)
}
```

```r
# Create the heatmaps for each data subset without any themes yet
g1 <- missing_heatmap(data_under_44, "under 44 group")
g2 <- missing_heatmap(data_after_44, "after 44 group")

# Add a blank theme for the plots to remove axis text and titles
# which will be added later with cowplot
g1 <- g1 + theme(
  legend.position = "none",
  axis.text.x = element_blank(),
  axis.title.x = element_blank(),
  # axis.text.y = element_blank(),
  axis.title.y = element_blank()
)
g2 <- g2 + theme(
  legend.position = "none",
  axis.text.x = element_blank(),
  axis.title.x = element_blank(),
  # axis.text.y = element_blank(),
  axis.title.y = element_blank()
)

 legend <- get_legend(g1 + theme(legend.position = "bottom"))

combined_plot <- plot_grid(g1, g2,legend,ncol = 1,
                           align = 'v', rel_heights = c(0.37,0.63))

combined_plot
data <- read.csv("project2.csv")
data <- unique(data)

colnames(data)[23] <- "ventilation_support_level.44"
colnames(data)[20] <- "peep_cm_h2o.36"
colnames(data)[26] <- "peep_cm_h2o.44"

# Load the codebook
codebook <- readxl::read_excel("project2_codebook.xlsx")

names(codebook) <- c("Variable","Label","Type")
# Assuming your codebook has columns "Variable", "Label" and "Type"
# and that your data is a data frame called `data`
```

```r
# Create a summary table of variables
variable_summary <- codebook %>%
  transmute(
    Variable = Variable,
    Definition = ifelse(is.na(Type),Label,paste(Label,Type)) ,
    Missing = sapply(data[Variable], function(x) {
      paste0(round(sum(is.na(x)) / length(x) * 100, 2), "%")
    })
  )

# Create the table in R Markdown format
#kable(variable_summary, format = "markdown", align = 'l')
kable(variable_summary, "latex", booktabs = TRUE, escape = TRUE,
      caption = "Variables Definition") %>%
  kable_styling(full_width = TRUE, font_size = 9,
                latex_options = c("hold_position", "caption_below")) %>%
  column_spec(1, width = "4cm", extra_css = "word-wrap: break-word;") %>%
  column_spec(2, width = "9cm", extra_css = "word-wrap: break-word;")
### data drop

data$mat_race <- as.factor(data$mat_race)
data$mat_ethn <- as.factor(data$mat_ethn)
data$del_method <- as.factor(data$del_method)
data$prenat_ster <- as.factor(data$prenat_ster)
data$com_prenat_ster <- as.factor(data$com_prenat_ster)
data$mat_chorio <- as.factor(data$mat_chorio)
data$gender <- as.factor(data$gender)
data$ventilation_support_level.36 <- as.factor(data$ventilation_support_level.36)
data$ventilation_support_level.44 <- as.factor(data$ventilation_support_level.44)
data$med_ph.36 <- as.factor(data$med_ph.36)
data$med_ph.44 <- as.factor(data$med_ph.44)
data$sga <- as.factor(data$sga)


data.2 <- data[,-c(3,15)]
data.2 <- data.2[-which(data.2$center %in% c("20","21")),]


data_under_44 <- data.2%>%filter(hosp_dc_ga<43.9)

data_after_44 <- data.2%>%filter(hosp_dc_ga>=43.9)
```

```r
data_disna_44 <- data.2%>%filter(is.na(hosp_dc_ga))

data_disna_under_44 <- data_disna_44 %>%
  filter(is.na(weight_today.44),is.na(ventilation_support_level.44),
         is.na(inspired_oxygen.44),is.na(p_delta.44),
         is.na(peep_cm_h2o.44),is.na(med_ph.44))

data_disna_after_44 <- data_disna_44 %>%
  filter(!(is.na(weight_today.44)&is.na(ventilation_support_level.44)&
             is.na(inspired_oxygen.44)&is.na(p_delta.44)&
             is.na(peep_cm_h2o.44)&is.na(med_ph.44)))

data_under_44 <- rbind(data_under_44,data_disna_under_44)
data_after_44 <- rbind(data_after_44,data_disna_after_44)

data_under_44$discharge_44 <- 0
data_after_44$discharge_44 <- 1

data.new <- rbind(data_under_44,data_after_44)
data.new$Outcome <- ifelse((data.new$Death=="Yes"|data.new$Trach==1),1,0)

data.new <- data.new[,-c(26:28)]

data.new$discharge_44 <- as.factor(data.new$discharge_44)
data.new$Outcome <- as.factor(data.new$Outcome)
data.new$center <- as.factor(data.new$center)

save(data.new,file = "data.new.Rda")
load(file = "data.new.Rda")
w44_col <- colnames(data.new)[20:25]

data_under_44 <- data.new[data.new$discharge_44==0,-c(1,20:25)]
data_after_44 <- data.new[data.new$discharge_44==1,-1]

id_under_44 <- data.new[data.new$discharge_44==0,1,drop=FALSE]
id_after_44 <- data.new[data.new$discharge_44==1,1,drop=FALSE]


imputed_under_44 <- mice(data_under_44, m = 5, seed = 123)
imputed_after_44 <- mice(data_after_44, m = 5, seed = 123)
```

```r
completed_under_44 <- lapply(1:5,FUN = function(x){complete(imputed_under_44, x) }) # This
completed_after_44 <- lapply(1:5,FUN = function(x){complete(imputed_after_44, x) })

completed_data <- list()
for (i in 1:5) {
  w44 <- data.new[data.new$discharge_44==0,w44_col]
  w44[is.na(w44)] <- 0
  completed_data[[i]]=cbind(id_under_44,completed_under_44[[i]],w44)
  completed_data[[i]] <- completed_data[[i]][ names(data.new) ]
  completed_data[[i]]=rbind(completed_data[[i]],cbind(id_after_44,completed_after_44[[i]])
  completed_data[[i]]$Outcome <- as.numeric(as.character(completed_data[[i]]$Outcome))
  completed_data[[i]]$discharge_44 <- as.numeric(as.character(completed_data[[i]]$discharg
}

save(completed_data,file = "completed_data.Rda")
### select 30% as validation set, and 70% as training dataset.
load(file = "completed_data.Rda")

train_id <- sample(data.new$record_id,size = round(0.7*nrow(data.new)),replace = FALSE)
validation_id <- setdiff(data.new$record_id,train_id)

train_sets <- lapply(completed_data,FUN = function(X){X %>%
    filter(record_id %in% train_id)})
validation_sets <- lapply(completed_data,FUN = function(X){X %>%
    filter(record_id %in% validation_id)})

X_train <- train_sets[[1]]
y_train <- train_sets[[1]][,27]

variable_name <- colnames(X_train)[-c(1,27)]
interaction_term <- paste0(variable_name[19:24],":discharge_44")

formula_str <- paste0("Outcome ~ ",paste(c(variable_name[1:18],
                                           interaction_term,"discharge_44"),collapse = " +

formula <- as.formula(formula_str)

# Assuming X_train is your predictor matrix and y_train is your response vector
X_matrix <- model.matrix(object = formula, data = X_train)[,-1]
y_vector <- y_train
```

```r
cv_model <- cv.glmnet(X_matrix, y_vector, family = "binomial", alpha = 1)
plot(cv_model)

Lasso_models <- function(train_set,myformula){

  X_train <- train_set
  y_train <- train_set$Outcome

  X_matrix <- model.matrix(object = myformula, data = X_train)[,-1]
  cv_model <- cv.glmnet(X_matrix, y_train, family = "binomial", alpha = 1)
  optimal_lambda <- cv_model$lambda.min
  best_lasso_model <- glmnet(X_matrix, y_train, family = "binomial",
                             alpha = 1, lambda = optimal_lambda)
  return(best_lasso_model)
}

# Averaging coefficients
lasso_model_list <- lapply(train_sets,FUN = function(train_set){
  Lasso_models(train_set = train_set, myformula = formula)
})

lasso_coef_list <- lapply(lasso_model_list, coef)

lasso_coef_cbind <- cbind(lasso_coef_list[[1]], lasso_coef_list[[2]],
                     lasso_coef_list[[3]],lasso_coef_list[[4]], lasso_coef_list[[5]])
avg_coefs_lasso <- apply(lasso_coef_cbind, 1, mean)



Stepwise_models <- function(train_set, myformula, cv = TRUE) {

  best_formula <- NULL
  lowest_mean_cv_score <- Inf

  if (cv) {
    folds <- sample(1:10, size = nrow(train_set), replace = TRUE)
    cv_scores <- rep(NA, 10)
    for (k in 1:10) {
      # Create training and validation subsets
      train_subset <- train_set[folds != k, ]
      validation_subset <- train_set[folds == k, ]
```

```
      # Initial and full models for stepwise selection
      initial_model <- glm(Outcome ~ 1, data = train_subset, family = binomial())
      full_model <- glm(myformula, data = train_subset, family = binomial())

      # Perform stepwise forward selection
      stepwise_model <- stepAIC(initial_model,
                                scope = list(lower = initial_model,upper = full_model),
                                direction = "forward", trace = FALSE)

      # Predict on validation data and calculate Brier score
      pred_prob <- predict(stepwise_model, newdata = validation_subset, type = "response")
      brier_score <- mean((pred_prob - validation_subset$Outcome)^2)
      cv_scores[k] <- brier_score
      # Check if this model has the lowest mean CV score
      if (brier_score < lowest_mean_cv_score) {
        lowest_mean_cv_score <- brier_score

        best_terms <- labels(terms(stepwise_model))
        best_formula <- as.formula(paste("Outcome ~", paste(best_terms, collapse = " + ")))
      }
    }
    mean_cv_score <- mean(cv_scores, na.rm = TRUE)
    #print(mean_cv_score)
    #print(paste("Lowest Mean CV Brier Score:", lowest_mean_cv_score))

    best_model_all_data <- glm(best_formula, data = train_set, family = binomial())

  } else {
    # Fit the stepwise model to the entire training set using the full dataset
    initial_model_all <- glm(Outcome ~ 1, data = train_set, family = binomial())
    full_model_all <- glm(myformula, data = train_set, family = binomial())
    best_model_all_data <- stepAIC(initial_model_all,
                                   scope = list(lower = initial_model_all,
                                                upper = full_model_all),
                                   direction = "forward", trace = FALSE)
  }
  return(best_model_all_data)
}


stepwise_model_list <- lapply(train_sets,FUN = function(train_set){
```

```
  Stepwise_models(train_set = train_set,myformula = formula,cv=TRUE)
})

coef_stepwise_list <- lapply(stepwise_model_list, coef)

# Find predicted probabilities on long imputed data (no rounding applied in this case!)
completed_validation_long <- do.call(rbind, validation_sets)

x_vars <- model.matrix(object = formula, data = completed_validation_long)
completed_validation_long$score <- x_vars %*% avg_coefs_lasso
mod <- glm(Outcome~score, data = completed_validation_long, family = "binomial")
probabilities_lasso <- predict(mod, type="response")

Probabilities <- c()
for (i in 1:5) {
      X_test <- completed_validation_long
      y_test <- X_test$Outcome

      probabilities <- predict(stepwise_model_list[[i]], newdata = X_test, type = "respons
      Probabilities <- cbind(Probabilities,probabilities)}
probabilities_stepwise <- rowMeans(Probabilities)

#
#
# Average_predict_across_models <- function(model_lists,k=5,validation_sets,model="lasso",
#
#   Probabilities <- c()
#
#   if (model=="lasso") {
#     for (i in 1:k) {
#       X_test <- validation_sets[[i]]
#       y_test <- X_test$Outcome
#
#       X_matrix.test <- model.matrix(object = myformula, data = X_test)
#       probabilities <- predict(model_lists[[i]], newx = X_matrix.test, type = "response"
#       Probabilities <- cbind(Probabilities,probabilities)
#     }
#   }else if(model=="stepwise"){
#     for (i in 1:k) {
#       X_test <- validation_sets[[i]]
#       y_test <- X_test$Outcome
```

```
#
#       probabilities <- predict(model_lists[[i]], newdata = X_test, type = "response")
#       #Probabilities <- cbind(Probabilities,probabilities)
#       Probabilities <- cbind(Probabilities,probabilities)
#     }
#   }
#   return(Probabilities)
#   #return(rowMeans(Probabilities))
# }
#
# #
# # probabilities_lasso <- Average_predict_across_models(model_lists = lasso_model_list,
# #                                                      k = 5,validation_sets = validatio
# #                                                      model = "lasso",myformula = formu
# probabilities_stepwise <- Average_predict_across_models(model_lists = stepwise_model_lis
#                                                       k = 5,validation_sets = validati
#                                                       model = "stepwise",myformula = f



y_test <- completed_validation_long$Outcome
  # Calculate ROC curve and AUC for the first model
  roc1 <- roc(y_test, probabilities_lasso)
  auc1 <- auc(roc1)

  # Calculate ROC curve and AUC for the second model
  roc2 <- roc(y_test, probabilities_stepwise)
  auc2 <- auc(roc2)

  # Plot ROC curves
  plot(roc1, main = paste0("ROC Curves"), col="blue")
  plot(roc2, add = TRUE, col="red")

  # Add legend
  legend("bottomright", legend=c(paste("Model Lasso AUC =", round(auc1, 2)),
                                 paste("Model Stepwise AUC =", round(auc2, 2))),
         col=c("blue", "red"), lwd=2)

  # Optionally, print AUC values
 # print(paste("AUC for Lasso model:", auc1))
 # print(paste("AUC for Stepwise model:", auc2))
```

```r
# Conduct the Hosmer-Lemeshow test
hoslem.test_lasso <- hoslem.test(x = y_test, y = probabilities_lasso, g = 10)
hoslem.test_stepwise <- hoslem.test(x = y_test, y = probabilities_stepwise, g = 10)

# Values from your Hosmer-Lemeshow tests and Brier Scores
p_value_stepwise <- hoslem.test_stepwise$p.value
p_value_lasso <- hoslem.test_lasso$p.value
statistic_stepwise <- hoslem.test_stepwise$statistic
statistic_lasso <- hoslem.test_lasso$statistic
BS_lasso <- mean((probabilities_lasso - y_test) ^ 2)
BS_stepwise <- mean((probabilities_stepwise - y_test) ^ 2)

# Create a data frame
results <- data.frame(
  Model = c("Lasso", "Stepwise"),
  P_Value = c(p_value_lasso, p_value_stepwise),
  Statistic = c(statistic_lasso, statistic_stepwise),
  Brier_Score = c(BS_lasso, BS_stepwise)
)

kable(results, caption = "Hoslem Test and Brier Score", align = 'c',escape = TRUE)

prepare_reliability_data <- function(probabilities, actuals, n_bins = 10) {
  data.frame(probabilities, actuals) %>%
    mutate(bin = cut(probabilities, breaks = seq(0, 1, length.out = n_bins + 1),
                     include.lowest = TRUE, labels = FALSE)) %>%
    group_by(bin) %>%
    summarise(
      predicted_probability = mean(probabilities),
      actual_frequency = mean(actuals)
    )
}

reliability_data_lasso <- prepare_reliability_data(probabilities_lasso, y_test)
reliability_data_stepwise <- prepare_reliability_data(probabilities_stepwise, y_test)

plot_reliability_diagram <- function(reliability_data, model_name) {
  ggplot(reliability_data, aes(x = predicted_probability, y = actual_frequency)) +
    geom_point() +
    geom_line() +
    geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
```

```
    xlim(0, 1) +
    ylim(0, 1) +
    xlab("Predicted Probability") +
    ylab("Actual Frequency") +
    ggtitle(paste("Reliability Diagram for", model_name))
}

# Plot for Lasso model
p_lasso <- plot_reliability_diagram(reliability_data_lasso, "Lasso")

# Plot for Stepwise model
p_step <- plot_reliability_diagram(reliability_data_stepwise, "Stepwise")

combined_plot2 <- plot_grid(p_lasso, p_step, align = "v")
combined_plot2
summary(stepwise_model_list[[1]])
coef(lasso_model_list[[1]])
```