PHP2550 Project 2: Regression Analysis

Yiwen Liang

11/12/2023

Abstract

This project endeavors to develop a regression model aimed at predicting the composite outcome of tracheotomy and death, thereby offering valuable insights into the indication criteria and the optimal temporal placement of tracheotomies in neonates afflicted by severe bronchopulmonary dysplasia (sBPD). The data provided is a national database encompasses demographic, diagnostic, and respiratory parameters of infants with sBPD admitted to collaborative Neonatal Intensive Care Units (NICUs), and with known respiratory support parameters at 36 and 44 weeks post-menstrual ages (PMA). After exploratory data analysis (EDA), we construct a total of 9 models using three lasso, ridge, and best subset methods, each with all available variables, baseline information and details from week 36, as well as baseline information and details from week 44. Then we use multiple metrics to evaluate the performance of the 9 models, and the model constructed with all available variables and utilizing the best subset for variable selection emerges as the *"best"* model in this study.

Introduction

Bronchopulmonary dysplasia (BPD) constitutes a chronic pulmonary ailment characterized by inflammatory processes and lung scarring (Association, n.d.), representing the most prevalent complication arising from prematurity, particularly in its severe manifestation. Over 10,000 infants are affected by BPD each year. Severe bronchopulmonary dysplasia (sBPD) necessitates respiratory support through mechanical ventilation or oxygen therapy, often prompting tracheostomy before discharge for sustained ventilator dependence. Given the merits and drawbacks of tracheostomy placement, it becomes important to decide who needs a tracheotomy and the optimal timing for referring a patient for this procedure.

This project endeavors to develop a regression model aimed at predicting the composite outcome of tracheotomy and death, thereby offering valuable insights into the indication criteria and the optimal temporal placement of tracheotomies in neonates afflicted by sBPD. The data provided is a national database encompasses demographic, diagnostic, and respiratory parameters of infants with sBPD admitted to collaborative Neonatal Intensive Care Units (NICUs), and with known respiratory support parameters at 36 and 44 weeks post-menstrual ages (PMA).

The data used in this project is not available to the public, hence only the code used for analysis and codebook for the dataset are available at Github, https://github.com/yiwen-liang/PHP_2550_Project_2.

Methods

Analyses are performed in R v4.2.2. Variables summaries and comparisons are undertaken employing numerical (mean, median, proportion) and graphical representations. The assessment of missing data and descriptive statistics are detailed in Table 1 and Table 3. To handle missing values, the Multiple imputation (MI) technique is applied. The missing values are imputed using values generated from distributions and relationships among observed variables in the dataset. `mice()` function from `mice` package is utilized to obtain test and train (validation) dataset separately. To model and predict the composite outcome of tracheostomy and death, a novel outcome variable `Y` was created, where a value of 0 indicates the absence of tracheostomy and survival, while a value of 1 signifies the occurrence of at least one negative outcome, encompassing either tracheostomy, death, or both.

Prior to variable selection, three sets of variables are considered: all available variables, baseline and 36-week information, and baseline and 44-week information. Variable selections involve the application of three methods introduced in class: **lasso** regression, **ridge** regression, and **best subset** selection procedure. These methods aim to construct models conducive to accurate predictions. Each approach undergoes a 10-fold cross-validation, and the resulting coefficients are averaged over 5 imputed datasets to obtain the final models. The performance evaluation of these models on the the test dataset will be presented in **Result** section. The mathematical expressions defining these three methods are provided below.

- The **ridge** regression is similar to least squares, except that the ridge coefficient estimates $\hat{\beta}^R$ minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2,$$

where $\lambda \geq 0$ is called the tuning parameter.

- The **lasso** regression is an alternative to ridge regression. Unlike ridge, lasso regression is able to perform variable selection and returns a model that only involves a subset of the variables. The lasso coefficients, $\hat{\beta}_\lambda^L$ minimize the quantity

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|.$$

We observe similar formulation of equations of the two regressions, and the only difference is that β_j^2 has been replaced by $|\beta_j|$ in the lasso penalty. Besides, lasso is able to shrink the coefficient estimates toward zero when the tuning parameter is sufficiently large.

- To perform the **best subset** selection, we start with fitting separate least squares regression for all possible combination of p predictors, then identify the *"best"* model. There are established packages in R to perform these three methods. We'll mainly use `glmnet` package for lasso ($\alpha = 1$) and ridge ($\alpha = 0$), and `L0Learn` package for best subset procedure. By specifying the argument `penalty=L0`, `L0Learn` fits regularization paths for L0-regularized regression that solves

$$\min_{\beta_0,\beta} l(y_i, \beta_0 + \langle x_i, \beta\rangle) + \lambda||\beta||_0,$$

where l is the loss function.

Results

1. Exploratory Data Analysis (EDA)

The initial phase of EDA involves identification and elimination of potential duplicate records within the dataset. We notice that the patient with `record_id` is 2000824 has 4 identical records, prompting the removal of these duplicates. Subsequently, a thorough examination of missing data in the dataset is conducted. Among the 30 variables, 26 exhibited missing values. The quantitative details, including the count and proportion of missingness for each variable, are outlined in Table 1. Variables pertaining to information at the 44-week mark displayed a considerable extent of missingness. Additionally, the variable `any_surf` has more than 40% of missing values.

Table 1: Summary of Missing Values in Each Variable

Variables	N	Proportion (%)	Variables	N	Proportion (%)
center	10	1.00	ventilation_support_level.36	30	3.01
mat_race	56	5.62	inspired_oxygen.36	92	9.24
mat_ethn	57	5.72	p_delta.36	128	12.85
blength	78	7.83	peep_cm_h2o_modified.36	117	11.75
birth_hc	77	7.73	med_ph.36	30	3.01
del_method	3	0.30	weight_today.44	446	44.78
prenat_ster	35	3.51	ventilation_support_level_modified.44	424	42.57
com_prenat_ster	193	19.38	inspired_oxygen.44	448	44.98
mat_chorio	62	6.22	p_delta.44	448	44.98
gender	4	0.40	peep_cm_h2o_modified.44	446	44.78
sga	15	1.51	med_ph.44	424	42.57
any_surf	433	43.47	hosp_dc_ga	124	12.45
weight_today.36	92	9.24	Death	2	0.20

The identification of 10 missing values in the variable `center` prompts further investigation. Fortunately, no missing value is observed in the corresponding `record_id` variable. Leveraging the encoding pattern inherent in the `record_id`, it's deduced that all 10 records with missing values all come from center 1. Hence, we manually impute the missing values in `center` originated from center 1. Consequently, manual imputation is employed to fill in the missing values in `center` with the value 1. As shown in Table 2, we can see that the data is collected from 10 different centers, with considerable variations in the number of observations attributed to each center. Notably, approximately two thirds of the records are from center 2, while a mere 4 records are from center 20 and 1 record from center 21.

Table 2: Summary of Centers

center	1	2	3	4	5	7	12	16	20	21
N	65	630	57	60	40	32	69	38	4	1

Following an observation of the dataset, a disparity is identified in the coding of variable `mat_race`, compared to the specifications outlined in the codebook. Consequently, we decide to disregard the variable `mat_race` and exclude it from the subsequent model fitting processes. The final step in data processing is related to the re-coding of `ventilation_support_level.36` and `ventilation_support_level_modified.44`. Given that the chosen methods require manual calculation for predicted values, I decide to collapse these two categorical variables into binary levels after numerous attempts.

Table 3: Summary of Baseline Characteristics of Infants for With and Without Negative Outcome

Variable	**0**, N = 811	**1**, N = 183	**p-value**
mat_ethn			0.3
1	64 (8.3%)	10 (5.9%)	
2	703 (92%)	160 (94%)	
bw	760 (610, 950)	670 (540, 835)	<0.001
ga	25 (24, 27)	25 (24, 27)	0.6
blength	32 (30, 35)	31 (29, 34)	0.002
birth_hc	23.00 (21.50, 25.00)	22.00 (21.00, 24.00)	0.010
del_method			0.017
1	245 (30%)	39 (21%)	
2	564 (70%)	143 (79%)	
prenat_ster	679 (86%)	154 (92%)	0.024
com_prenat_ster	499 (76%)	109 (76%)	>0.9
mat_chorio	132 (17%)	28 (17%)	0.9
gender			0.7
Female	334 (41%)	73 (40%)	
Male	473 (59%)	110 (60%)	
sga			<0.001
Not SGA	658 (82%)	118 (66%)	
SGA	142 (18%)	61 (34%)	
any_surf	374 (81%)	87 (88%)	0.095
weight_today.36	2,150 (1,880, 2,408)	1,997 (1,694, 2,260)	<0.001
ventilation_support_level.36			<0.001
0	109 (14%)	7 (4.3%)	
1	693 (86%)	155 (96%)	
inspired_oxygen.36	0.29 (0.23, 0.35)	0.45 (0.34, 0.60)	<0.001
p_delta.36	0 (0, 0)	14 (2, 24)	<0.001
peep_cm_h2o_modified.36	7 (6, 8)	8 (6, 9)	<0.001
med_ph.36	32 (4.0%)	33 (20%)	<0.001
weight_today.44	3,765 (3,299, 4,143)	3,555 (3,070, 3,995)	0.009
ventilation_support_level_modified.44			<0.001
0	261 (60%)	8 (6.0%)	
1	177 (40%)	126 (94%)	
inspired_oxygen.44	0.28 (0.25, 0.32)	0.40 (0.30, 0.60)	<0.001
p_delta.44	0 (0, 0)	19 (10, 37)	<0.001
peep_cm_h2o_modified.44	0 (0, 8)	8 (8, 10)	<0.001
med_ph.44	33 (7.5%)	66 (49%)	<0.001
hosp_dc_ga	45 (42, 52)	64 (50, 91)	<0.001

[1] n (%); Median (IQR)

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

Table 3 presents the summary of baseline information and parameters from week 36 and week 44, categorized by the outcome variable `Y`. Among all subjects in the database, it's evident that the majority (811) didn't undergo tracheostomy and survived ($Y = 0$), while only 183 infants belong to the *negative outcome* group ($Y = 1$). Infants in the negative outcome group tend to exhibit smaller size, lower birth weights (`bw`), shorter birth lengths (`blength`), smaller birth head circumference (`birth_hc`), and a higher likelihood of being small for gestational age.

Besides, notable differences are observed in measures at week 36 and week 44 between the two groups. Infants with negative outcomes continue to be of lighter weight, more prone to requiring ventilation support and medication, exhibiting a higher proportion of inspired oxygen, and being discharged at an older age compared to those who didn't undergo tracheostomy placement and survived.

We conduct Rosner's tests to identify potential outliers in numeric variables, assuming that the data without any outliers follows a normal distribution. The number of outliers of each variable, as determined by the results of Rosner's tests, is presented in Table 4. We suspect that these 9 variables have outliers.

Table 4: Summary of Outliers Using Rosner's Test

Variable	Number of Outliers	Variable	Number of Outliers	Variable	Number of Outliers
bw	8	inspired_oxygen.36	26	weight_today.44	1
blength	1	p_delta.36	2	inspired_oxygen.44	13
birth_hc	7	peep_cm_h2o_modified.36	1	hosp_dc_ga	46

2. Model Built Upon All Variables

In lasso regression, the coefficient estimates for multiple variables (`bw`, `ga`, `blength`, etc.) are 0, which suggests that the they're dropped from the model. However, since the ridge regression is not capable of shrinking coefficients to zero, all coefficient estimates are non-zero. The model using best subset also possibly has covariates with coefficients of 0.

In Table 5, we see that most of the covariates that are not included in the lasso model also have coefficients with small magnitude in other two models. Though they're not excluded from the model, their effects are quite small.

Table 5: Comparison of The Coefficients of Three Models

	Lasso	Ridge	Best Subset		Lasso	Ridge	Best Subset
(Intercept)	-3.5033	-2.8659	-3.0633	mat_chorioYes	0.0420	0.0775	0.0519
center2	-0.4299	-0.2274	-0.4393	genderMale	0.0000	-0.0240	-0.0440
center3	-0.6781	-0.3259	-0.8860	sgaSGA	0.0654	0.1464	0.1905
center4	0.0000	0.0038	-0.1313	any_surfYes	0.1246	0.0183	0.1999
center5	0.0000	-0.0003	-0.1428	weight_today.36	-0.0001	-0.0002	-0.0002
center7	-0.5521	-0.3354	-1.2963	ventilation_support_level.361	0.0802	0.1185	0.3054
center12	1.4251	0.7109	1.7490	inspired_oxygen.36	0.5905	0.9452	0.4888
center16	-0.0960	-0.1954	-0.2614	p_delta.36	0.0181	0.0146	0.0181
center20	0.0000	-0.3728	-1.1333	peep_cm_h2o_modified.36	0.0000	0.0170	-0.0274
center21	1.4376	1.2012	5.2932	med_ph.36	0.1863	0.3363	0.2081
mat_ethn2	0.2407	0.1405	0.3622	weight_today.44	-0.0004	-0.0001	-0.0006
bw	0.0000	-0.0001	0.0003	ventilation_support_level_modified.441	0.0379	0.2917	-0.3019
ga	0.0000	0.0005	-0.0620	inspired_oxygen.44	0.5765	0.9099	0.3020
blength	0.0000	-0.0074	0.0093	p_delta.44	0.0494	0.0206	0.0595
birth_hc	0.0052	0.0025	0.0325	peep_cm_h2o_modified.44	0.1095	0.0419	0.1769
del_method2	0.2811	0.1463	0.5038	med_ph.44	1.1752	0.5924	1.4471
prenat_sterYes	0.4459	0.1745	0.6333	hosp_dc_ga	0.0077	0.0046	0.0101
com_prenat_sterYes	0.1086	0.0703	0.0756				

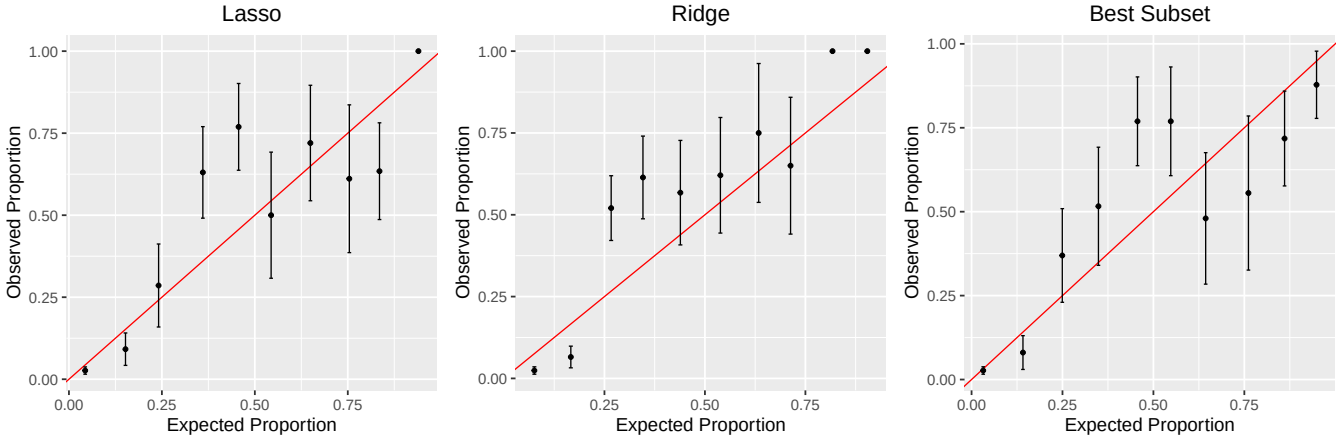
Then we use multiple metrics to evaluate the performance of the three models, as shown in Table 6. The best subset model has the lowest Breir scores than the ridge model, indicating that it has the highest predictive accuracy among the three. Table 6 also presents the thresholds that maximize the sum of specificity and sensitivity, respectively. All thresholds range between 0.17 and 0.22, the values of specificity lie between 0.88 and 0.9, and the values of sensitivity are also between 0.85 and 0.9.

The AUC (area under the ROC curve) is utilized to evaluate the discrimination. Discrimination, in context of a predictive model, refers to its ability to distinguish between individuals with and without the specified outcome. In the case of binary outcomes, the c-statistic is equivalent to AUC. A higher AUC value signifies superior model discrimination, indicating a heightened ability to correctly classify individuals with and without the outcome. All three models seen to have goood discrimination, and the best subset model has the highest AUC of 0.9247.

Table 6: Specifity and Sensitivity at Thresholds, and AUC

	Brier.Scores	AUC	Threshold	Specificity	Sensitivity
Lasso	0.0778	0.9247	0.1857	0.8854	0.88
Ridge	0.0891	0.9085	0.2108	0.8915	0.85
Best Subset	0.0770	0.9251	0.1752	0.8905	0.87

The calibration plots are used to assess the calibration of models. Calibration refers to the agreement between the observed frequency of the events and the predicted probabilities across various percentiles. The visualization eludicates the deree of concordance between predicted and actual distributions, and close alignment with $x = y$ line indicates good calibration. The calibration plots for the three models exhibit few differences, and none of them demonstrates significantly superior calibration.



3. Model Built Upon Baseline and Week 36 Information

A model is also fitted using data comprising baseline information and details from week 36. Achieving accurate predictions based solely on baseline and 36 weeks of information holds the potential to expedite the determination of the following treatment. This early prediction enables timely communication with the patient's family, affording them more time to contemplate and make informed decisions regarding the patient's care.

In Table 7, we can see that there are more zeros from the best subset model, and the covariates whose coefficient estimates are zero also have coefficients with small magnitude in other models.

Table 7: Comparison of The Coefficients of Three Models

	Lasso	Ridge	Best Subset		Lasso	Ridge	Best Subset
(Intercept)	-3.7738	-2.5160	-3.0665	del_method2	0.3440	0.1673	0.2373
center2	-0.7590	-0.2528	-0.6023	prenat_sterYes	0.4672	0.1885	0.3659
center3	-1.6458	-0.4367	-1.6223	com_prenat_sterYes	0.1197	0.0727	0.1037
center4	-0.1161	-0.0145	-0.1791	mat_chorioYes	0.0288	0.0687	0.0000
center5	-0.1406	-0.0690	-0.2277	genderMale	-0.0349	-0.0268	0.0000
center7	-1.4176	-0.4149	-1.1626	sgaSGA	0.2012	0.2060	0.0000
center12	1.1997	0.7765	1.5085	any_surfYes	0.1349	0.0479	0.2725
center16	-0.5447	-0.2826	-0.4109	weight_today.36	-0.0006	-0.0003	-0.0010
center20	-0.5382	-0.4598	-1.0820	ventilation_support_level.361	0.2128	0.1564	0.0000
center21	1.0411	1.1654	0.0000	inspired_oxygen.36	2.5410	1.6198	2.7410
mat_ethn2	0.2060	0.1107	0.0000	p_delta.36	0.0503	0.0242	0.0574
bw	0.0000	-0.0001	0.0000	peep_cm_h2o_modified.36	0.0045	0.0254	0.0000
ga	0.0000	0.0066	0.0000	med_ph.36	0.5252	0.5168	0.3322
blength	0.0000	-0.0133	0.0000	hosp_dc_ga	0.0192	0.0066	0.0238
birth_hc	0.0023	-0.0002	0.0000				

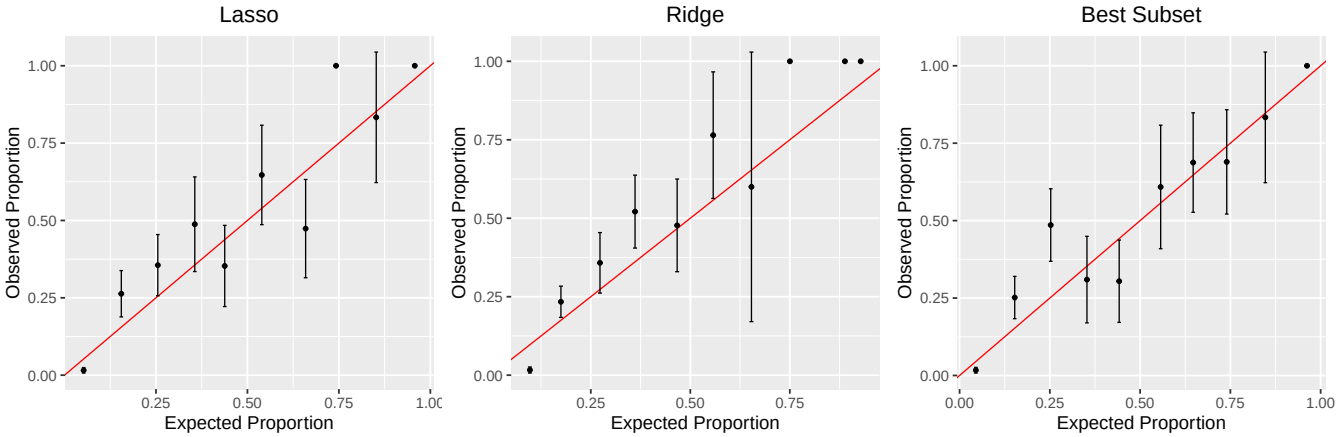
Then we use multiple metrics to evaluate the performance of the three models, as shown in Table 8. The thresholds maximize the sum of specificity and sensitivity, respectively. All thresholds range between 0.1 and 0.16, the values of specificity lie between 0.75 and 0.78, and the values of sensitivity are also between 0.9 and 0.93. Compared to the models encompassing all available variables, these models have higher Breir score (less predictive accuracy), lower specificity and higher sensitivity.

The AUC (area under the ROC curve) is utilized to evaluate the discrimination. All three models seen to have goood discrimination, and the best subset model has the highest AUC of 0.8901.

Table 8: Specifity and Sensitivity at Thresholds, and AUC

	Brier.Scores	AUC	Threshold	Specificity	Sensitivity
Lasso	0.0890	0.8867	0.1313	0.7789	0.92
Ridge	0.1028	0.8613	0.1577	0.7508	0.90
Best Subset	0.0882	0.8901	0.1175	0.7698	0.93

The calibration plots are used to asses the calibration of models. The calibration plots for the lasso and best subset models exhibit few differences, while ridge regression shows slightly worse calibration. None of them demonstrates significantly superior calibration.



4. Model Built Upon Baseline and Week 44 Information

I endeavored to construct a model exclusively relying on baseline information and data from the 44th week. This approach aimed to accommodate the potential missingness in data arising from hospital transfers and variations in data collection across different healthcare institutions. Table 9 shows the coefficient estimates from three different models.

Table 9: Comparison of The Coefficients of Three Models

	Lasso	Ridge	Best Subset		Lasso	Ridge	Best Subset
(Intercept)	-3.2189	-2.5915	-2.2729	del_method2	0.3142	0.1846	0.3743
center2	-0.5217	-0.2718	-0.2401	prenat_sterYes	0.4882	0.2024	0.6593
center3	-0.9161	-0.4092	-0.5164	com_prenat_sterYes	0.1183	0.0781	0.0719
center4	0.0000	-0.0017	-0.1009	mat_chorioYes	0.0520	0.0891	0.0415
center5	0.0000	0.0476	-0.1034	genderMale	-0.0179	-0.0329	-0.0498
center7	-0.9051	-0.4677	-1.2880	sgaSGA	0.1317	0.1844	0.1892
center12	1.4620	0.8534	1.9187	any_surfYes	0.1004	0.0287	0.1278
center16	-0.2843	-0.2805	-0.3751	weight_today.44	-0.0005	-0.0002	-0.0007
center20	-0.1293	-0.4893	-1.1967	ventilation_support_level_modified.441	0.0265	0.3687	-0.2359
center21	2.1735	1.7488	4.1913	inspired_oxygen.44	0.7488	1.1482	0.3597
mat_ethn2	0.3363	0.2027	0.1368	p_delta.44	0.0600	0.0291	0.0669
bw	0.0000	-0.0001	0.0003	peep_cm_h2o_modified.44	0.1226	0.0540	0.1729
ga	-0.0112	-0.0027	-0.0570	med_ph.44	1.2162	0.7377	1.5369
blength	0.0000	-0.0111	0.0059	hosp_dc_ga	0.0090	0.0059	0.0112
birth_hc	0.0131	0.0029	0.0181				

Then we use multiple metrics to evaluate the performance of the three models, as shown in Table 10. The thresholds maximize the sum of specificity and sensitivity, respectively. All thresholds range between 0.2 and 0.4, the values of specificity lie between 0.88 and 0.91, and the values of sensitivity are also between 0.8 and 0.85. Compared to

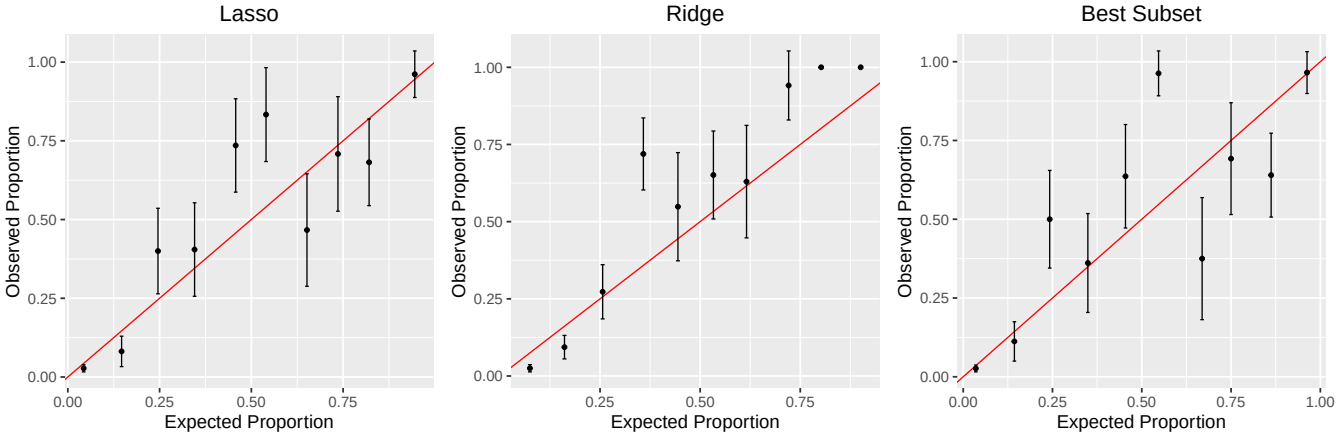
the models encompassing all available variables, these models have approximately simialr Breir score (comparable predictive accuracy), higher specificity and lower sensitivity.

The AUC (area under the ROC curve) is utilized to evaluate the discrimination. All three models seen to have goood discrimination, and the lasso model has the highest AUC of 0.9221.

Table 10: Specifity and Sensitivity at Thresholds, and AUC

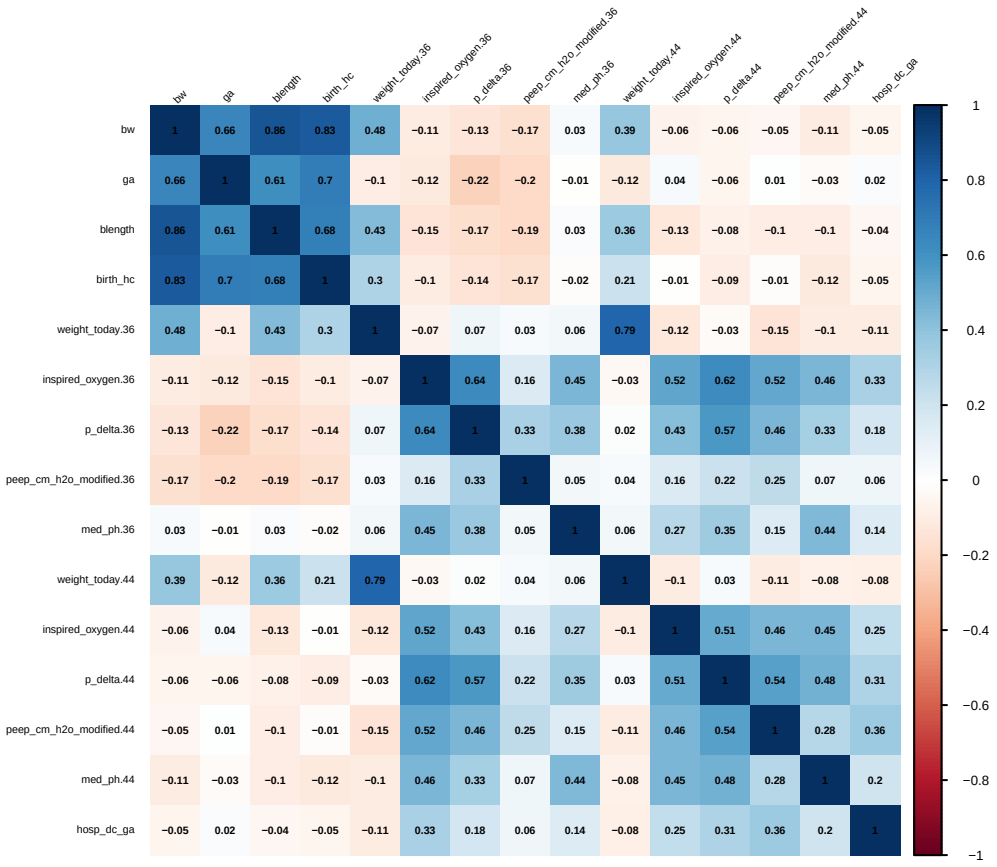
	Brier.Scores	AUC	Threshold	Specificity	Sensitivity
Lasso	0.0771	0.9221	0.2078	0.9005	0.840
Ridge	0.0871	0.9045	0.2325	0.8894	0.805
Best Subset	0.0790	0.9198	0.2095	0.9045	0.835

The calibration plots are used to asses the calibration of models. The calibration plots for the lasso and ridge models exhibit few differences, while best subset regression shows slightly worse calibration. None of them demonstrates significantly superior calibration.



Discussion

In this project, we construct a total of 9 models using three different approaches, each with three different sets of initial variables. Although the coefficient estimates are not identical, the direction and magnitude of the effects of the covariates are closely aligned. `center` exhibit to be a significant factor, indicating that the composite outcomes of patients from different hospitals are different while controlling for all other covariates. Most covariate effects are consistent with the findings from EDA, including factors such as delivery method, prenatal corticosteroids, the proportion of inspired oxygen and medication for pulmonary hypertension. Furthermore, there is no discernible difference between males and females regarding the composite outcome of tracheostomy and death. However, a few discrepancies are observed. Contrary to expectations from EDA, where birth weight and birth length are anticipated to be significant in explaining the outcome, variables `bw` and `blength` are consistently excluded in all three lasso models. A potential explanation for this discrepancy may be found in the correlation matrix, which indicates a correlation between birth weight, weight at week 36 and weight at week 44. This correlation might account for the observed exclusion of `bw` and `blength` in the models.



All models exhibit exceptional predictive accuracy, as evidenced by Brier scores below 0.11. Furthermore, they demonstrate commendable discrimination, with AUC values greater than 0.86. The chosen cutoff points, which maximize the sum specificity and sensitivity, are notably low, below 0.25. In the context of this study, such low cutoff points are desirable as they align with the objective of minimizing the risk of overlooking any negative outcomes. However, the calibration of all models are not ideal. Considering multiple evaluation metrics, it can be concluded that the model constructed with all available variables and utilizing the best subset for variable selection emerges as the *"best"* model in this study.

The development of predictive models with high predictive accuracy and discrimination holds promising clinical implications. These models provide clinicians with valuable tools for early prognosis of the composite outcome of tracheostomy and death in infants with severe bronchopulmonary dysplasia. The ability to give accurate predictions allows for timely-decision-making and informed discussions with the patient's family. Additionally, the coefficient estimates offer nuanced understanding of relevant covariates, enhancing the clinician's capacity to tailor interventions strategies and improve patient care.

The strength of this study is that we try multiple approaches and different initial variable sets, and we employ numerous metrics to assess the performance of models in different aspects. However, our analysis also has a few limitations. First is that we don't include any interaction terms while fitting models. As we can see in the correlation matrix, multiple pairs of numeric variables are correlated. Besides, due to the lack of relevant medical knowledge, some variables that may have mechanistic associations. Secondly, given that the data is collected from 10 different centers, it's natural for us to consider it a multilevel data, and fit mixed effect models. However, the sample sizes in center 20 and center 21 are not sufficiently large.

Conclusion

In conclusion, this study successfully develop predictive models for the composite outcome of tracheostomy and death in infants with severe bronchopulmonary dysplasia. The identified *"best"* model in this study is the one constructed with all available variables and utilizing the best subset for variable selection. These models offer clinicians valuable tools for early prognosis, enabling informed discussions with the families of infants at risk. However, due to our

prior knowledge and the cluster-size imbalance, future research endeavors should test whether any interaction term should be included in the model, and whether mixed effects model would have a better performance for this dataset.

References

Association, American Lung. n.d. "Bronchopulmonary Dysplasia." *American Lung Association*. https://www.lung.org/lung-health-diseases/lung-disease-lookup/bronchopulmonary-dysplasia#:~:text=BPD%20is%20associated%20with%20inflammation,development%20of%20BPD%20more%20likely.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)

# Load the packages
library(knitr)
library(tidyverse)
library(kableExtra)
library(reshape2)
library(gt)
library(gtsummary)
library(corrplot)
library(cowplot)
library(ggplot2)
library(ggpubr)
library(janitor)
library(mice)
library(nnet)
library(bestglm)
library(glmnet)
library(EnvStats)
library(LOLearn)
library(MASS)

# Load the data
df <- read.csv("project2.csv")

# 1 record_id has 4 records in the dataset (3 duplicated)
# sum(duplicated(df))
df <- unique(df)

# 26 out of 29 variables have missing value
descript1 <- df %>%
  summarise(
    N = colSums(is.na(df)),
    prop = round(colMeans(is.na(df))*100, 2)) %>%
  mutate(Variables = colnames(df)) %>%
  filter(N != 0) %>%
  as.data.frame()

descript1 <- descript1[,c(3,1,2)]

# Display missing data summary table using kable
knitr::kable(
  list(descript1[1:13,], descript1[14:26,]),
  caption = "Summary of Missing Values in Each Variable",
  col.names = linebreak(c("Variables", "N", "Proportion (%)")),
  row.names = FALSE,
  booktabs = TRUE,
  escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'))

# Impute missing in center with 1
df$center[is.na(df$center)] <- 1
```

```r
# Data was collected from 10 centers
# 2/3 are from center 2
df %>%
  group_by(center) %>%
  summarize(N = n()) %>%
  t() %>%
  kable(caption = "Summary of Centers",
        row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'), font_size = 7)

df <- df %>%
  dplyr::select(-c(record_id, mat_race))

df$Y <- ifelse(df$Death=="No" & df$Trach==0, 0, 1)
df$Y <- as.factor(df$Y)

# Format the variables
df$mat_ethn <- as.factor(df$mat_ethn)
df$del_method <- as.factor(df$del_method)
df$prenat_ster <- as.factor(df$prenat_ster)
df$com_prenat_ster <- as.factor(df$com_prenat_ster)
df$mat_chorio <- as.factor(df$mat_chorio)
df$gender <- as.factor(df$gender)
df$sga <- as.factor(df$sga)
df$any_surf <- as.factor(df$any_surf)
df$ventilation_support_level.36 <-
  ifelse(df$ventilation_support_level.36==0, 0,
         ifelse(df$ventilation_support_level.36 %in% c(1,2), 1, NA))
df$ventilation_support_level.36 <- as.factor(df$ventilation_support_level.36)
df$ventilation_support_level_modified.44 <-
  ifelse(df$ventilation_support_level_modified.44==0, 0,
         ifelse(df$ventilation_support_level_modified.44 %in% c(1,2), 1, NA))
df$ventilation_support_level_modified.44 <- as.factor(df$ventilation_support_level_modified.44)
df$Trach <- as.factor(df$Trach)
df$Death <- as.factor(df$Death)

p_corr <- cor(select_if(df, is.numeric) %>% dplyr::select(-center),
              use = "complete.obs")

# Group by outcome Y
df %>%
  dplyr::select(-c(center, Trach, Death)) %>%
  tbl_summary(
    by = Y,
    missing = "no") %>%
  add_p() %>%
  modify_header(label = "**Variable**") %>%
  modify_caption(caption = "Summary of Baseline Characteristics of Infants
                 for With and Without Negative Outcome") %>%
  as_kable_extra(booktabs = TRUE) %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'),
                font_size = 8)

# Outliers: tests
```

```r
temp <- data.frame(
  Variable = c("bw", "blength", "birth_hc", "inspired_oxygen.36",
               "p_delta.36", "peep_cm_h2o_modified.36",
               "weight_today.44", "inspired_oxygen.44", "hosp_dc_ga"),
  Outliers = c(sum(rosnerTest(df$bw, k = 10)$all.stats$Outlier),
               sum(rosnerTest(df$blength, k = 10)$all.stats$Outlier),
               sum(rosnerTest(df$birth_hc, k = 10)$all.stats$Outlier),
               sum(rosnerTest(df$inspired_oxygen.36, k = 30)$all.stats$Outlier),
               sum(rosnerTest(df$p_delta.36, k = 10)$all.stats$Outlier),
               sum(rosnerTest(df$peep_cm_h2o_modified.36, k = 10)$all.stats$Outlier),
               sum(rosnerTest(df$weight_today.44, k = 10)$all.stats$Outlier),
               sum(rosnerTest(df$inspired_oxygen.44, k = 20)$all.stats$Outlier),
               sum(rosnerTest(df$hosp_dc_ga, k = 50)$all.stats$Outlier)))

knitr::kable(
  list(temp[1:3,], temp[4:6,], temp[7:9,]),
  caption = "Summary of Outliers Using Rosner's Test",
  col.names = linebreak(c("Variable", "Number of Outliers")),
  row.names = FALSE,
  booktabs = TRUE,
  escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'hold_position'), font_size = 7)

############################
#### Multiple Imputation ####
############################

# MI
set.seed(2550)
ignore <- sample(c(TRUE, FALSE), size = 25, replace = TRUE, prob = c(0.3, 0.7))
traindata <- df[!ignore, ]
testdata <- df[ignore, ]

# Train dataset
imp.train <- mice(traindata, m=5, print=FALSE, seed=2550)
train_imp <- vector("list", 5)
for (i in 1:5){
  train_imp[[i]] <- mice::complete(imp.train, i)
  train_imp[[i]]$Y <- ifelse(train_imp[[i]]$Death=="No" & train_imp[[i]]$Trach==0,
                             0, 1)
  train_imp[[i]]$Y <- as.factor(train_imp[[i]]$Y)
  train_imp[[i]]$center <- as.factor(train_imp[[i]]$center)
  train_imp[[i]] <- train_imp[[i]] %>% dplyr::select(-c(Trach, Death))
}

# Test dataset
imp.test <- mice.mids(imp.train, newdata=testdata, print=FALSE, seed=2550)
test_long <- mice::complete(imp.test, action="long")
test_long$Y <- ifelse(test_long$Death=="No" & test_long$Trach==0, 0, 1)
test_long <- test_long %>% dplyr::select(-c(Trach, Death))

test_long_wide <- test_long %>%
  pivot_wider(names_from = center, names_glue = "center{center}",
              values_from = center,  values_fill = 0) %>%
  mutate(center21 = 0) %>%
```

```r
  dplyr::select(-center1)

test_long_wide[,29:37] <- ifelse(test_long_wide[,29:37]!=0, 1, 0)
test_long_wide <- test_long_wide[,c(1:2, 32:36, 29:31, 37, 3:28)]

x_vars <- model.matrix(Y~. , test_long_wide)[,-c(2,3)]

# Only using week 36
train_imp_36 <- vector("list", 5)
for (i in 1:5){
  train_imp_36[[i]] <- train_imp[[i]] %>%
    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))
}

test_long_36 <- test_long_wide %>%
  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))

x_vars_36 <- model.matrix(Y~. , test_long_36)[,-c(2,3)]

# Only using week 44
train_imp_44 <- vector("list", 5)
for (i in 1:5){
  train_imp_44[[i]] <- train_imp[[i]] %>%
    dplyr::select(-c(weight_today.36, ventilation_support_level.36,
                     inspired_oxygen.36, p_delta.36,
                     peep_cm_h2o_modified.36, med_ph.36))
}

test_long_44 <- test_long_wide %>%
  dplyr::select(-c(weight_today.36, ventilation_support_level.36,
                   inspired_oxygen.36, p_delta.36,
                   peep_cm_h2o_modified.36, med_ph.36))

x_vars_44 <- model.matrix(Y~. , test_long_44)[,-c(2,3)]

################
#### Lasso ####
################

lasso <- 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(Y ~., data = df)[,-1]
  y.ord <- as.matrix(df$Y)

  # Generate folds
  k <- 10
  set.seed(1)
  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, nfolds = 10, alpha = 1,
                      family = "binomial",
                      lambda = lasso_mod_cv$lambda.min)

  # Get coefficients
  coef <- coef(lasso_mod)
  return(coef)
}

# Find average lasso coefficients over imputed datasets
lasso_coef1 <- lasso(train_imp[[1]])
lasso_coef2 <- lasso(train_imp[[2]])
lasso_coef3 <- lasso(train_imp[[3]])
lasso_coef4 <- lasso(train_imp[[4]])
lasso_coef5 <- lasso(train_imp[[5]])
lasso_coef <- cbind(lasso_coef1, lasso_coef2, lasso_coef3,
                    lasso_coef4, lasso_coef5)
avg_coefs_lasso <- apply(lasso_coef, 1, mean)

###############
#### 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(Y ~., data = df)[,-1]
  y.ord <- as.matrix(df$Y)

  # Generate folds
  k <- 10
  set.seed(1)
  folds <- sample(1:k, nrow(df), replace=TRUE)

  # Ridge model
  ridge_mod <- cv.glmnet(x.ord, y.ord, nfolds = 10, foldid = folds,
                         family = "binomial", alpha = 0)

  # Get coefficients
  coef <- coef(ridge_mod, lambda = ridge_mod$lambda.min)
  return(coef)
}

# Find average ridge coefficients over imputed datasets
ridge_coef1 <- ridge(train_imp[[1]])
ridge_coef2 <- ridge(train_imp[[2]])
ridge_coef3 <- ridge(train_imp[[3]])
ridge_coef4 <- ridge(train_imp[[4]])
ridge_coef5 <- ridge(train_imp[[5]])
```

```
ridge_coef <- cbind(ridge_coef1, ridge_coef2, ridge_coef3,
                    ridge_coef4, ridge_coef5)
avg_coefs_ridge <- apply(ridge_coef, 1, mean)

######################
#### Best Subset ####
######################

L0Learn_func <- 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(Y ~., data = df)[,-1]
  y.ord <- as.matrix(df$Y)

  # Ridge model
  L0Learn_mod <- L0Learn.cvfit(x.ord, y.ord, nFolds = 10, seed = 1,
                               loss = "Logistic", penalty = "L0")

  # Get coefficients
  min_index <- which.min(L0Learn_mod$cvMeans[[1]])
  coef <- coef(L0Learn_mod, lambda = L0Learn_mod$fit$lambda[[1]][min_index])
  return(coef)
}

# Find average ridge coefficients over imputed datasets
L0Learn_coef1 <- L0Learn_func(train_imp[[1]])
L0Learn_coef2 <- L0Learn_func(train_imp[[2]])
L0Learn_coef3 <- L0Learn_func(train_imp[[3]])
L0Learn_coef4 <- L0Learn_func(train_imp[[4]])
L0Learn_coef5 <- L0Learn_func(train_imp[[5]])

L0Learn_coef <- cbind(L0Learn_coef1, L0Learn_coef2, L0Learn_coef3,
                      L0Learn_coef4, L0Learn_coef5)
avg_coefs_L0Learn <- apply(L0Learn_coef, 1, mean)

# Predicted probabilities
test_long$p_lasso <- plogis(x_vars %*% avg_coefs_lasso)
test_long$p_ridge <- plogis(x_vars %*% avg_coefs_ridge)
test_long$p_L0Learn <- plogis(x_vars %*% avg_coefs_L0Learn)

coefficients <- round(cbind(avg_coefs_lasso, avg_coefs_ridge, avg_coefs_L0Learn),4)

knitr::kable(
  list(coefficients[1:18,], coefficients[19:35,]),
  caption = "Comparison of The Coefficients of Three Models",
  col.names = linebreak(c("Lasso", "Ridge", "Best Subset")),
  row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'),
                font_size = 7)

# Lasso
```

```r
test_long$Y <- test_long$Y==1
obs <- test_long$Y[order(test_long$p_lasso)]
threshold <- sort(test_long$p_lasso)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_lasso <- threshold[which.max(sens+spec)]
spec_lasso <- spec[which.max(sens+spec)]
sens_lasso <- sens[which.max(sens+spec)]
auc_lasso <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long$p_lasso,
                         bin = cut(test_long$p_lasso, breaks = num_cuts),
                         class = test_long$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_lasso <- ggplot(calib_data) +
  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") +
  ggtitle("Lasso") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# Ridge
obs <- test_long$Y[order(test_long$p_ridge)]
threshold <- sort(test_long$p_ridge)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_ridge <- threshold[which.max(sens+spec)]
spec_ridge <- spec[which.max(sens+spec)]
sens_ridge <- sens[which.max(sens+spec)]
auc_ridge <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long$p_ridge,
                         bin = cut(test_long$p_ridge, breaks = num_cuts),
                         class = test_long$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_ridge <- ggplot(calib_data) +
```

```
  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") +
  ggtitle("Ridge") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# L0Learn
obs <- test_long$Y[order(test_long$p_L0Learn)]
threshold <- sort(test_long$p_L0Learn)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_L0Learn <- threshold[which.max(sens+spec)]
spec_L0Learn <- spec[which.max(sens+spec)]
sens_L0Learn <- sens[which.max(sens+spec)]
auc_L0Learn <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long$p_L0Learn,
                         bin = cut(test_long$p_L0Learn, breaks = num_cuts),
                         class = test_long$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_L0Learn <- ggplot(calib_data) +
  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") +
  ggtitle("Best Subset") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# Table: Brier scores, AUC, thresholds, specificity, and sensitivity
df_thres <- data.frame(
  "Brier Scores" = round(c(mean((test_long$p_lasso - test_long$Y)^2),
                           mean((test_long$p_ridge - test_long$Y)^2),
                           mean((test_long$p_L0Learn - test_long$Y)^2)),4),
  AUC = round(c(auc_lasso, auc_ridge, auc_L0Learn),4),
  Threshold = round(c(thres_lasso, thres_ridge, thres_L0Learn),4),
  Specificity = round(c(spec_lasso, spec_ridge, spec_L0Learn),4),
  Sensitivity = round(c(sens_lasso, sens_ridge, sens_L0Learn),4))
rownames(df_thres) <- c("Lasso", "Ridge", "Best Subset")

kable(df_thres, row.names = TRUE, booktabs = T, escape = T,
      caption = "Specifity and Sensitivity at Thresholds, and AUC") %>%
```

```
  kable_styling(full_width = F,
                latex_options = c("striped", "HOLD_position"))

ggarrange(p_calib_lasso, p_calib_ridge, p_calib_L0Learn, ncol = 3, nrow = 1)

################
#### Lasso ####
################

lasso_coef1_36 <- lasso(train_imp_36[[1]])
lasso_coef2_36 <- lasso(train_imp_36[[2]])
lasso_coef3_36 <- lasso(train_imp_36[[3]])
lasso_coef4_36 <- lasso(train_imp_36[[4]])
lasso_coef5_36 <- lasso(train_imp_36[[5]])
lasso_coef_36 <- cbind(lasso_coef1_36, lasso_coef2_36, lasso_coef3_36,
                       lasso_coef4_36, lasso_coef5_36)
avg_coefs_lasso_36 <- apply(lasso_coef_36, 1, mean)

################
#### Ridge ####
################

ridge_coef1_36 <- ridge(train_imp_36[[1]])
ridge_coef2_36 <- ridge(train_imp_36[[2]])
ridge_coef3_36 <- ridge(train_imp_36[[3]])
ridge_coef4_36 <- ridge(train_imp_36[[4]])
ridge_coef5_36 <- ridge(train_imp_36[[5]])

ridge_coef_36 <- cbind(ridge_coef1_36, ridge_coef2_36, ridge_coef3_36,
                       ridge_coef4_36, ridge_coef5_36)
avg_coefs_ridge_36 <- apply(ridge_coef_36, 1, mean)

######################
#### Best Subset ####
######################

L0Learn_coef1_36 <- L0Learn_func(train_imp_36[[1]])
L0Learn_coef2_36 <- L0Learn_func(train_imp_36[[2]])
L0Learn_coef3_36 <- L0Learn_func(train_imp_36[[3]])
L0Learn_coef4_36 <- L0Learn_func(train_imp_36[[4]])
L0Learn_coef5_36 <- L0Learn_func(train_imp_36[[5]])

L0Learn_coef_36 <- cbind(L0Learn_coef1_36, L0Learn_coef2_36, L0Learn_coef3_36,
                         L0Learn_coef4_36, L0Learn_coef5_36)
avg_coefs_L0Learn_36 <- apply(L0Learn_coef_36, 1, mean)

# Predicted probability
test_long_36$p_lasso <- plogis(x_vars_36 %*% avg_coefs_lasso_36)
test_long_36$p_ridge <- plogis(x_vars_36 %*% avg_coefs_ridge_36)
test_long_36$p_L0Learn <- plogis(x_vars_36 %*% avg_coefs_L0Learn_36)

coefficients <- round(cbind(avg_coefs_lasso_36, avg_coefs_ridge_36, avg_coefs_L0Learn_36),4)

knitr::kable(
  list(coefficients[1:15,], coefficients[16:29,]),
  caption = "Comparison of The Coefficients of Three Models",
```

```
  col.names = linebreak(c("Lasso", "Ridge", "Best Subset")),
  row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'),
                font_size = 8)

# Lasso
test_long_36$Y <- test_long_36$Y==1
obs <- test_long_36$Y[order(test_long_36$p_lasso)]
threshold <- sort(test_long_36$p_lasso)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_lasso_36 <- threshold[which.max(sens+spec)]
spec_lasso_36 <- spec[which.max(sens+spec)]
sens_lasso_36 <- sens[which.max(sens+spec)]
auc_lasso_36 <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long_36$p_lasso,
                         bin = cut(test_long_36$p_lasso, breaks = num_cuts),
                         class = test_long_36$Y)
calib_data <- calib_data %>%
             group_by(bin) %>%
             summarize(observed = sum(class)/n(),
                       expected = sum(prob)/n(),
                       se = sqrt(observed*(1-observed)/n()))

p_calib_lasso_36 <- ggplot(calib_data) +
  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") +
  ggtitle("Lasso") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# Ridge
obs <- test_long_36$Y[order(test_long_36$p_ridge)]
threshold <- sort(test_long_36$p_ridge)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_ridge_36 <- threshold[which.max(sens+spec)]
spec_ridge_36 <- spec[which.max(sens+spec)]
sens_ridge_36 <- sens[which.max(sens+spec)]
auc_ridge_36 <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long_36$p_ridge,
                         bin = cut(test_long_36$p_ridge, breaks = num_cuts),
                         class = test_long_36$Y)
```

```
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_ridge_36 <- ggplot(calib_data) +
  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") +
  ggtitle("Ridge") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# L0Learn
obs <- test_long_36$Y[order(test_long_36$p_L0Learn)]
threshold <- sort(test_long_36$p_L0Learn)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_L0Learn_36 <- threshold[which.max(sens+spec)]
spec_L0Learn_36 <- spec[which.max(sens+spec)]
sens_L0Learn_36 <- sens[which.max(sens+spec)]
auc_L0Learn_36 <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long_36$p_L0Learn,
                         bin = cut(test_long_36$p_L0Learn, breaks = num_cuts),
                         class = test_long_36$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_L0Learn_36 <- ggplot(calib_data) +
  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") +
  ggtitle("Best Subset") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# Table: Brier scores, AUC, thresholds, specificity, and sensitivity
df_thres_36 <- data.frame(
  "Brier Scores" = round(c(mean((test_long_36$p_lasso - test_long_36$Y)^2),
                           mean((test_long_36$p_ridge - test_long_36$Y)^2),
                           mean((test_long_36$p_L0Learn - test_long_36$Y)^2)),4),
  AUC = round(c(auc_lasso_36, auc_ridge_36, auc_L0Learn_36),4),
```

```
  Threshold = round(c(thres_lasso_36, thres_ridge_36, thres_L0Learn_36),4),
  Specificity = round(c(spec_lasso_36, spec_ridge_36, spec_L0Learn_36),4),
  Sensitivity = round(c(sens_lasso_36, sens_ridge_36, sens_L0Learn_36),4))
rownames(df_thres_36) <- c("Lasso", "Ridge", "Best Subset")

kable(df_thres_36, row.names = TRUE, booktabs = T, escape = T,
      caption = "Specifity and Sensitivity at Thresholds, and AUC") %>%
  kable_styling(full_width = F,
                latex_options = c("striped", "HOLD_position"))

ggarrange(p_calib_lasso_36, p_calib_ridge_36, p_calib_L0Learn_36,
          ncol = 3, nrow = 1)

###############
#### Lasso ####
###############

lasso_coef1_44 <- lasso(train_imp_44[[1]])
lasso_coef2_44 <- lasso(train_imp_44[[2]])
lasso_coef3_44 <- lasso(train_imp_44[[3]])
lasso_coef4_44 <- lasso(train_imp_44[[4]])
lasso_coef5_44 <- lasso(train_imp_44[[5]])
lasso_coef_44 <- cbind(lasso_coef1_44, lasso_coef2_44, lasso_coef3_44,
                       lasso_coef4_44, lasso_coef5_44)
avg_coefs_lasso_44 <- apply(lasso_coef_44, 1, mean)

###############
#### Ridge ####
###############

ridge_coef1_44 <- ridge(train_imp_44[[1]])
ridge_coef2_44 <- ridge(train_imp_44[[2]])
ridge_coef3_44 <- ridge(train_imp_44[[3]])
ridge_coef4_44 <- ridge(train_imp_44[[4]])
ridge_coef5_44 <- ridge(train_imp_44[[5]])

ridge_coef_44 <- cbind(ridge_coef1_44, ridge_coef2_44, ridge_coef3_44,
                       ridge_coef4_44, ridge_coef5_44)
avg_coefs_ridge_44 <- apply(ridge_coef_44, 1, mean)

####################
#### Best Subset ####
####################

L0Learn_coef1_44 <- L0Learn_func(train_imp_44[[1]])
L0Learn_coef2_44 <- L0Learn_func(train_imp_44[[2]])
L0Learn_coef3_44 <- L0Learn_func(train_imp_44[[3]])
L0Learn_coef4_44 <- L0Learn_func(train_imp_44[[4]])
L0Learn_coef5_44 <- L0Learn_func(train_imp_44[[5]])

L0Learn_coef_44 <- cbind(L0Learn_coef1_44, L0Learn_coef2_44, L0Learn_coef3_44,
                         L0Learn_coef4_44, L0Learn_coef5_44)
avg_coefs_L0Learn_44 <- apply(L0Learn_coef_44, 1, mean)

# Predicted probability
test_long_44$p_lasso <- plogis(x_vars_44 %*% avg_coefs_lasso_44)
```

```r
test_long_44$p_ridge <- plogis(x_vars_44 %*% avg_coefs_ridge_44)
test_long_44$p_L0Learn <- plogis(x_vars_44 %*% avg_coefs_L0Learn_44)

coefficients <- round(cbind(avg_coefs_lasso_44, avg_coefs_ridge_44, avg_coefs_L0Learn_44),4)

knitr::kable(
  list(coefficients[1:15,], coefficients[16:29,]),
  caption = "Comparison of The Coefficients of Three Models",
  col.names = linebreak(c("Lasso", "Ridge", "Best Subset")),
  row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'),
                font_size = 8)

# Lasso
test_long_44$Y <- test_long_44$Y==1
obs <- test_long_44$Y[order(test_long_44$p_lasso)]
threshold <- sort(test_long_44$p_lasso)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_lasso_44 <- threshold[which.max(sens+spec)]
spec_lasso_44 <- spec[which.max(sens+spec)]
sens_lasso_44 <- sens[which.max(sens+spec)]
auc_lasso_44 <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long_44$p_lasso,
                         bin = cut(test_long_44$p_lasso, breaks = num_cuts),
                         class = test_long_44$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_lasso_44 <- ggplot(calib_data) +
  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") +
  ggtitle("Lasso") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# Ridge
obs <- test_long_44$Y[order(test_long_44$p_ridge)]
threshold <- sort(test_long_44$p_ridge)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_ridge_44 <- threshold[which.max(sens+spec)]
spec_ridge_44 <- spec[which.max(sens+spec)]
```

```r
sens_ridge_44 <- sens[which.max(sens+spec)]
auc_ridge_44 <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long_44$p_ridge,
                         bin = cut(test_long_44$p_ridge, breaks = num_cuts),
                         class = test_long_44$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_ridge_44 <- ggplot(calib_data) +
  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") +
  ggtitle("Ridge") + theme_grey(base_size = 15) +
  theme(plot.title = element_text(hjust = 0.5))

# L0Learn
obs <- test_long_44$Y[order(test_long_44$p_L0Learn)]
threshold <- sort(test_long_44$p_L0Learn)
sens <- (sum(obs)- cumsum(obs))/sum(obs)
spec <- cumsum(!obs)/sum(!obs)

thres_L0Learn_44 <- threshold[which.max(sens+spec)]
spec_L0Learn_44 <- spec[which.max(sens+spec)]
sens_L0Learn_44 <- sens[which.max(sens+spec)]
auc_L0Learn_44 <- round(sum(spec*diff(c(0, 1 - sens))), 4)

# Calibration
num_cuts <- 10
calib_data <- data.frame(prob = test_long_44$p_L0Learn,
                         bin = cut(test_long_44$p_L0Learn, breaks = num_cuts),
                         class = test_long_44$Y)
calib_data <- calib_data %>%
              group_by(bin) %>%
              summarize(observed = sum(class)/n(),
                        expected = sum(prob)/n(),
                        se = sqrt(observed*(1-observed)/n()))

p_calib_L0Learn_44 <- ggplot(calib_data) +
  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") +
  ggtitle("Best Subset") + theme_grey(base_size = 15) +
```

```
  theme(plot.title = element_text(hjust = 0.5))

# Table: Brier scores, AUC, thresholds, specificity, and sensitivity
df_thres_44 <- data.frame(
  "Brier Scores" = round(c(mean((test_long_44$p_lasso - test_long_44$Y)^2),
                           mean((test_long_44$p_ridge - test_long_44$Y)^2),
                           mean((test_long_44$p_L0Learn - test_long_44$Y)^2)),4),
  AUC = round(c(auc_lasso_44, auc_ridge_44, auc_L0Learn_44),4),
  Threshold = round(c(thres_lasso_44, thres_ridge_44, thres_L0Learn_44),4),
  Specificity = round(c(spec_lasso_44, spec_ridge_44, spec_L0Learn_44),4),
  Sensitivity = round(c(sens_lasso_44, sens_ridge_44, sens_L0Learn_44),4))
rownames(df_thres_44) <- c("Lasso", "Ridge", "Best Subset")

kable(df_thres_44, row.names = TRUE, booktabs = T, escape = T,
      caption = "Specifity and Sensitivity at Thresholds, and AUC") %>%
  kable_styling(full_width = F,
                latex_options = c("striped", "HOLD_position"))

ggarrange(p_calib_lasso_44, p_calib_ridge_44, p_calib_L0Learn_44,
          ncol = 3, nrow = 1)

# Correlation plot: interaction
corrplot(p_corr, method = "shade", shade.col = NA,
         tl.col = "black", addCoef.col = "black", tl.srt = 45,
         cl.pos = "r", number.cex = 0.3, tl.cex = 0.3, cl.cex = 0.4)
```