The following project is my use of the package CoxnetSurvivalAnalysis from scikit-survival to implement a Cox proportional hazards model using a machine learning approach with a L1 (lasso/absolute value) cost. Survival modeling allows me to model survival time when the final event is not observed (censored) for many records. In my experience, it's uncommon to use a machine learning approach with survival modeling.

Unlike most machine learning implementations, my purpose will not be to make predictions but rather to undertake an exploratory analysis to determine which variables are potentially predictive or explanatory of survival. This sort of purpose is generally served by inferential statistics. However, employing inferential statistics is challenging when there are a large number of potential predictors. Putting hundreds of variables in a model often leads to issues of multicollinearity. Further, using an algorithm such as forward or stepwise selection becomes computationally difficult requiring an extreme number of model runs. Hence, some sort of pre-screening criteria is generally used when employing a statistical model with a large number of potential predictors.

In this project, I will create three faux datasets with two continuous and two binary variables to compare the performance of the lasso proportional hazards model to standard inferential statistical testing in the same model. These datasets will be:

1. One continuous and one binary variable are predictors. The other two are uncorrelated with survival or the predictors.
2. One continuous and one binary variable are predictors. The other continuous variable is LIGHTLY correlated with the continuous predictor. The other binary variable is LIGHTLY correlated with the binary predictor.
3. One continuous and one binary variable are predictors. The other continuous variable is HIGHLY correlated with the continuous predictor. The other binary variable is HIGHLY correlated with the binary predictor.

After the previous three comparisons are made, I then add to #3 996 uncorrelated predictors as a proof of concept for high dimensionality applications motivating this project.

Case 1:

I use numpy random to create a faux dataset with 100,000 observations. In this case, there are four variables, but only two are predictors of survival. One of these predictors is continuous while the other is binary. The other two variables are independent of survival, the predictors, and eachother. In this case, it should be simple to determine which of the four variables are predictors using any technique (including bivariate analysis).

Comparison Model:

For comparison I fit a Cox proportional hazards model with a chisquare test for each of the four coefficients. From the 'p' column we see that the statistical tests correctly identify 'var2' and 'var4' as the predictors. The other uncorrelated variables have very high p-values (0.60 and 0.76). While simply dropping all insignificant variables from the model in batch does not always result in the correct answer, in this case it would.

model	lifelines.CoxPHFitter
duration col	'dur'
event col	'event'
baseline estimation	breslow
number of observations	100000
number of events observed	50057
partial log-likelihood	-517201.54
time fit was run	2023-01-05 03:46:46 UTC

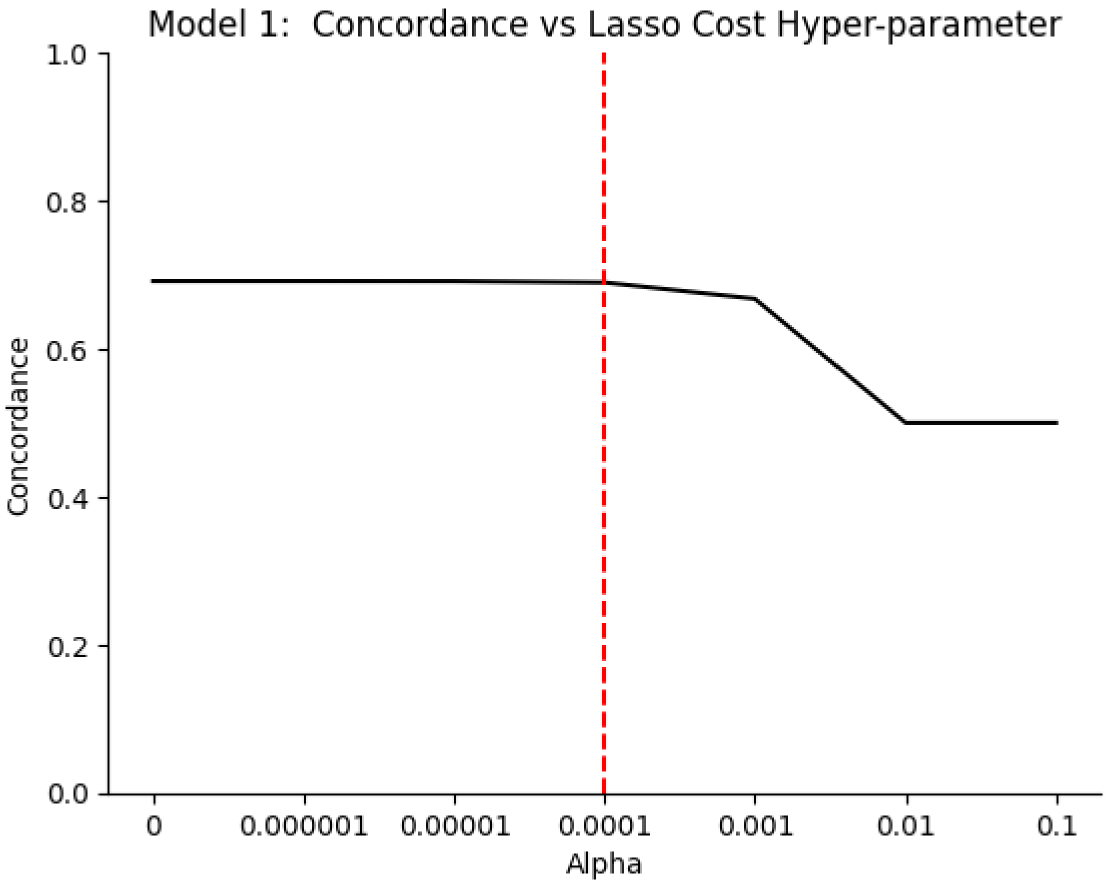
	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	p	-log2(p)
index	-0.00	1.00	0.00	-0.00	0.00	1.00	1.00	0.00	-1.28	0.20	2.33
var1	-0.02	0.98	0.02	-0.05	0.01	0.95	1.01	0.00	-1.09	0.27	1.87
var2	2.32	10.19	0.02	2.29	2.36	9.84	10.54	0.00	132.92	<0.005	inf
var3	-0.00	1.00	0.01	-0.02	0.02	0.98	1.02	0.00	-0.41	0.68	0.55
var4	-0.51	0.60	0.01	-0.52	-0.49	0.59	0.61	0.00	-54.89	<0.005	inf

Concordance	0.69
Partial AIC	1034413.08
log-likelihood ratio test	21200.16 on 5 df
-log2(p) of ll-ratio test	inf

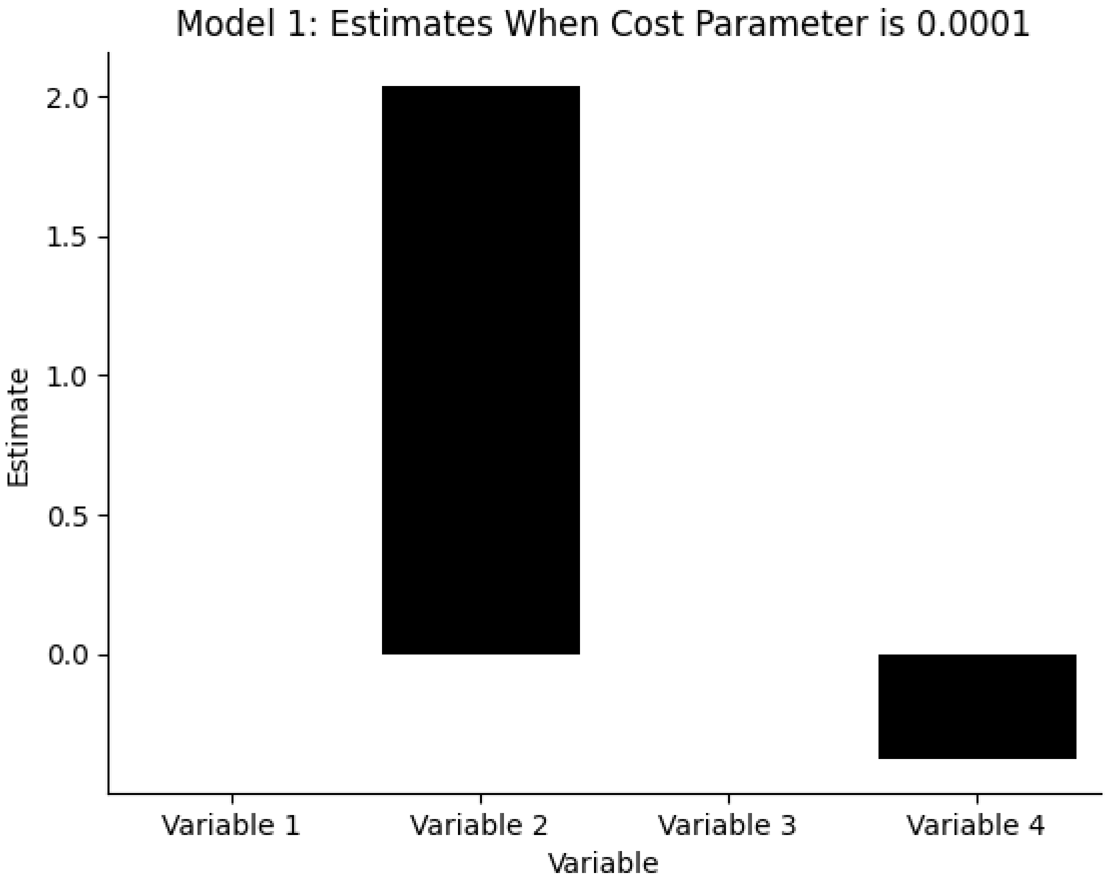
Lasso Model:

For the machine learning/lasso model I use a 70% training sample and a 30% cross validation sample. I estimate the model eight times with eight different weights (hyperparameter values) for the L1 cost parameters (0,0.000001,0.000001,0.00001,0.0001,0.001,0.01,0.1). I then evaluate the concordance rate in the cross validation dataset for each model run.

The highest concordance rate is when the hyperparameter is zero, or in other words when we run the full model with four variables. We know that this is not the correct answer. However, in the figure for Model 1 below, we see that there is no substantial change in concordance rate when we increase the hyperparameter value to 0.0001 (see red line).



Setting the hyperparameter to 0.0001, we can see that the estimates for Variable 1 and Variable 3 are zero. This is the correct answer. (The lasso cost will set the estimate for irrelevant variables to zero.)



Case 2:

Case 2 is setup very similarly to Case 1. However, in Case 1 Variable 1 and Variable 3 were independent of survival, the two predictors, and eachother. In Case 2, the continuous Variable 1 is set to be slightly correlated with Variable 2 (the other continuous variable). The binary Variable 3 is set to be slightly correlated with Variable 4 (the other binary variable). This case is more difficult as generally bivariate analysis would find that survival is correlated to Variable 1 and Variable 3 due to their correlation with the correct predictors Variable 2 and Variable 4. However, a model controlling for the predictors would generally still generate the correct solution even in face of this light multicollinearity.

Comparison Model:

Similarly to Case 1, a Cox proportional hazards model is run and p-values from chisquare tests correctly identify the predictors as Variable 2 and Variable 4.

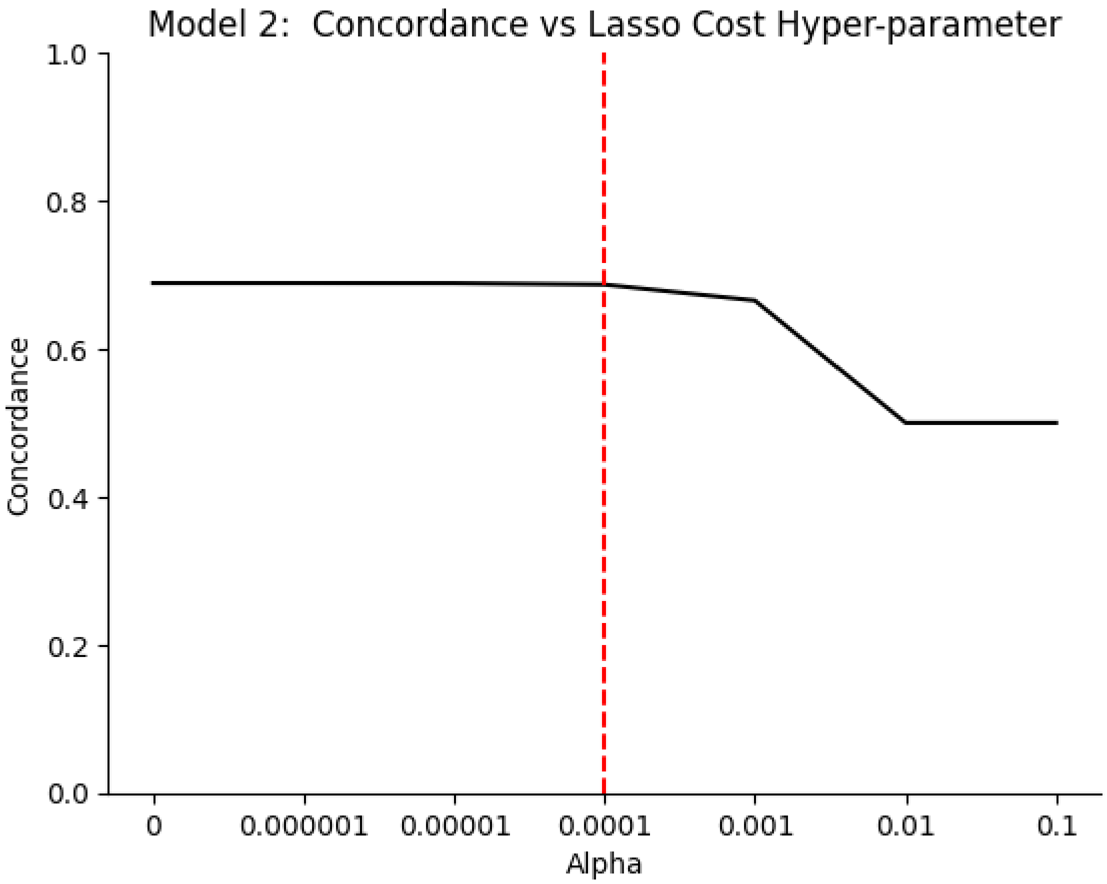
model	lifelines.CoxPHFitter
duration col	'dur'
event col	'event'
baseline estimation	breslow
number of observations	100000
number of events observed	49865
partial log-likelihood	-515086.64
time fit was run	2023-01-05 03:58:59 UTC

	coef	**exp(coef)**	**se(coef)**	**coef lower 95%**	**coef upper 95%**	**exp(coef) lower 95%**	**exp(coef) upper 95%**	**cmp to**	**z**	**p**	**-log2(p)**
index	-0.00	1.00	0.00	-0.00	0.00	1.00	1.00	0.00	-0.38	0.71	0.50
var1	0.00	1.00	0.02	-0.04	0.04	0.96	1.04	0.00	0.12	0.91	0.14
var2	2.35	10.45	0.02	2.31	2.38	10.09	10.83	0.00	131.25	<0.005	inf
var3	-0.01	0.99	0.01	-0.03	0.01	0.97	1.01	0.00	-0.91	0.36	1.47
var4	-0.50	0.61	0.01	-0.52	-0.48	0.60	0.62	0.00	-53.65	<0.005	inf

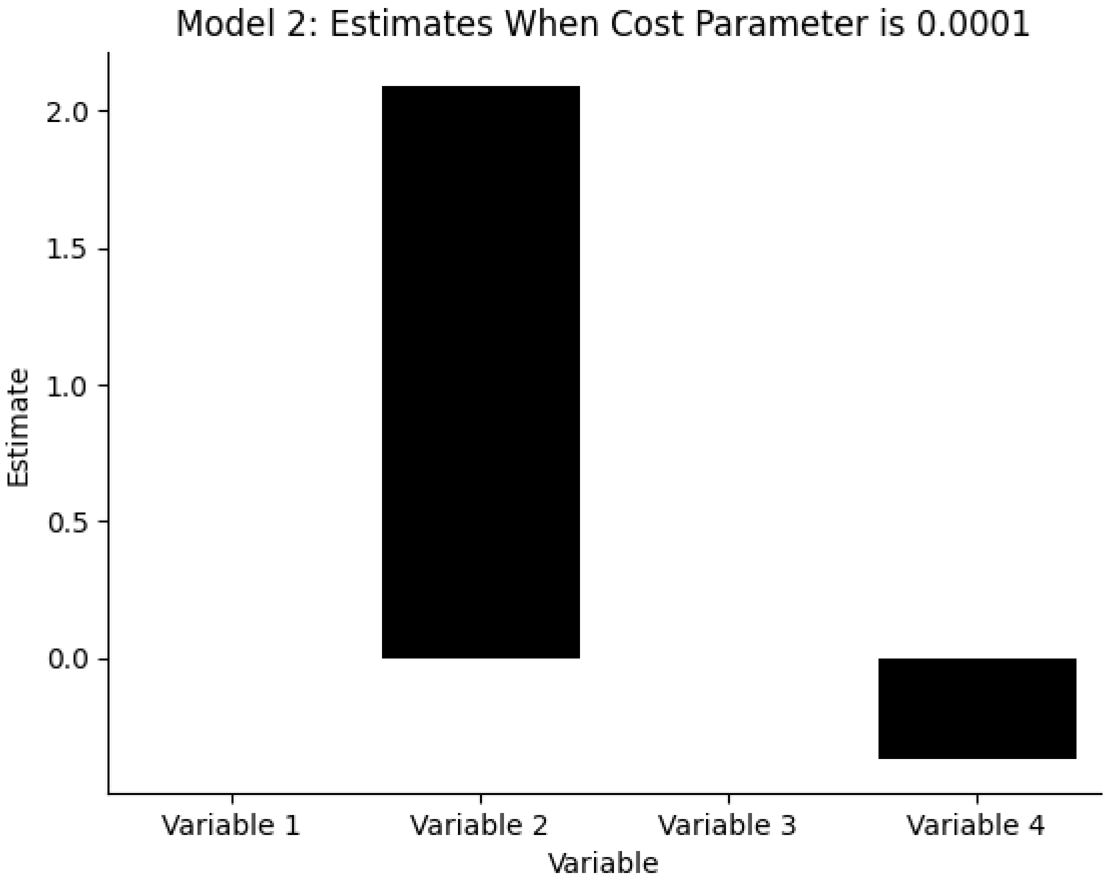
Concordance	0.69
Partial AIC	1030183.28
log-likelihood ratio test	21694.88 on 5 df
-log2(p) of ll-ratio test	inf

Lasso Model:

In the exact same manner as for Case 1, the lasso model is run eight times and the same hyperparameter (0.0001) is selected from the chart below.



As in Case 1, the use of the hyperparameter 0.0001 results in the correct model with only Variable 2 and Variable 4 having non-zero estimates in Case 2.



Case 3:

Case 3 is exactly the same as Case 2 except instead of Variable 1 and Variable 3 having light correlation with the predictors, they are now highly correlated with the predictors (though not perfectly). Higher levels of multicollinearity are a problem for bivariate analysis and both the proportional hazards model with statistical tests and the proportional hazards model with lasso. Therefore, the following will determine how the two models still perform.

Comparison Model:

In the same manner as the first two cases, a cox proportional hazards model is run and chisquare tests are evaluated. The model still easily identifies Variable 2 and Variable 4 as the correct predictors.

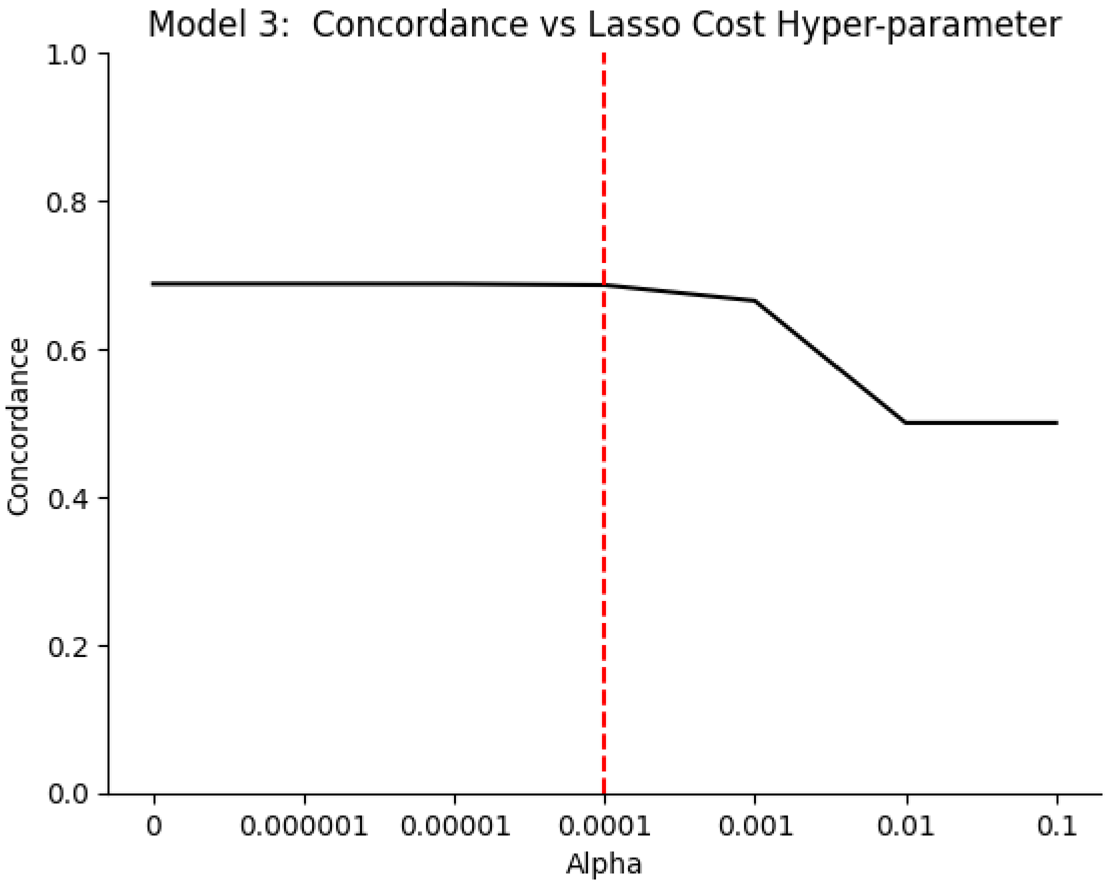
model	lifelines.CoxPHFitter
duration col	'dur'
event col	'event'
baseline estimation	breslow
number of observations	100000
number of events observed	49885
partial log-likelihood	-515490.57
time fit was run	2023-01-05 04:09:47 UTC

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	p	-log2(p)
index	-0.00	1.00	0.00	-0.00	0.00	1.00	1.00	0.00	-0.14	0.89	0.17
var1	-0.01	0.99	0.08	-0.16	0.14	0.85	1.15	0.00	-0.10	0.92	0.12
var2	2.31	10.08	0.06	2.18	2.44	8.88	11.43	0.00	35.89	<0.005	934.82
var3	-0.00	1.00	0.01	-0.03	0.02	0.97	1.02	0.00	-0.39	0.70	0.52
var4	-0.49	0.61	0.01	-0.51	-0.47	0.60	0.62	0.00	-46.03	<0.005	inf

Concordance	0.69
Partial AIC	1030991.15
log-likelihood ratio test	21055.71 on 5 df
-log2(p) of ll-ratio test	inf

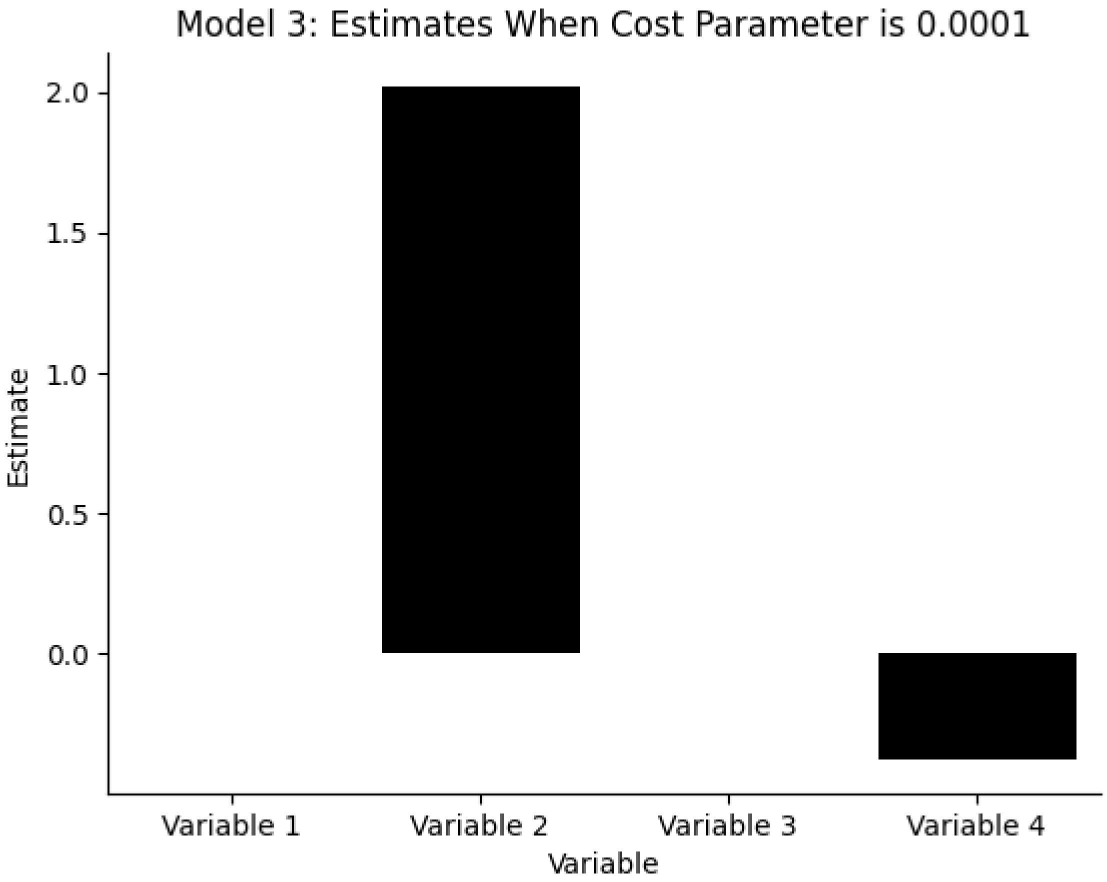
Lasso Model:

Like the previous two cases, the hyperparameter 0.0001 is selected for the model.



Like the previous two cases, only Variable 2 and Variable 4 have non-zero estimates.

It's important to note that given another random dataset I encountered during development, it was found that Variable 1 had a non-zero estimate. This estimate was very close to zero. Given that I have normalized all of my variables prior to running the lasso models, the variable estimates are directly comparable to eachother. It's possible that the user could have manually removed this variable due to it's small estimate and arrived at the correct result. If one wished to employ a more objective criteria than the small estimate, he could run a model similar to the comparison model on the three variables selected by the lasso stage and determine which coefficients are statistically signficant. In the case of only four potential variables, it would be pointless to run both models on the same data. But given a large number of potential predictors, running the lasso model as a first stage to narrow down the list of potential predictors only to 'guarantee' the final list with statistical testing could be a good course of action.

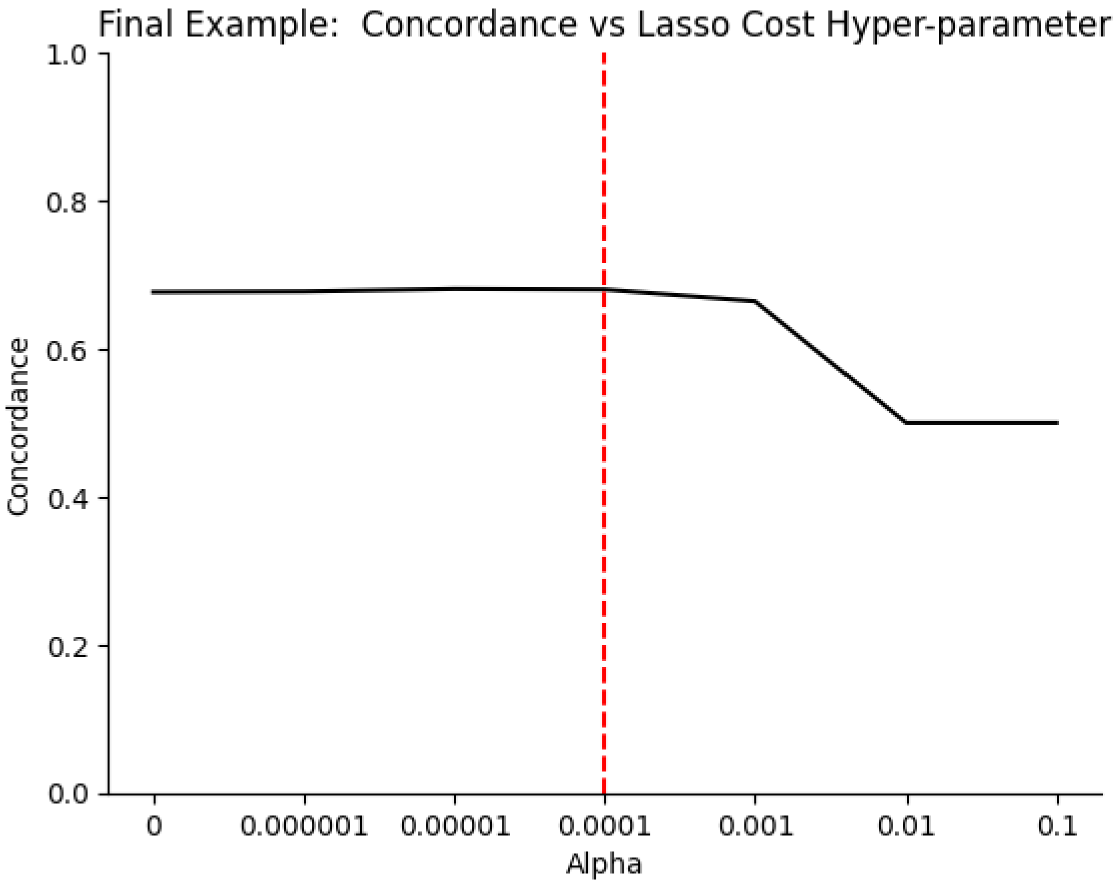


Final Example:

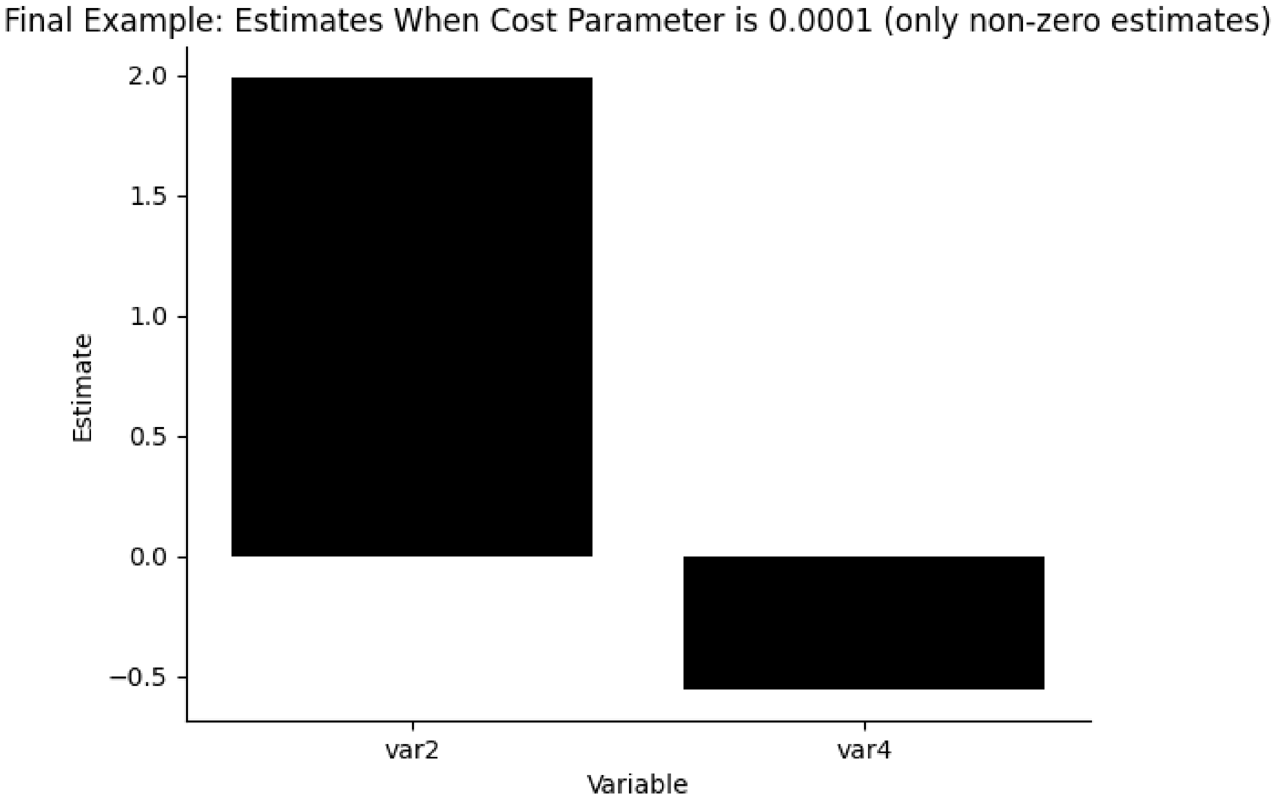
The previous three cases only have four potential predictors, of which two are correct. For these cases, there is no reason to employ a lasso type model. These were only provided as an illustration or example. Instead, a statistical model similar to the comparison models above is more correct. All variables can be added to the model and only the statistically significant variables retained. An even more correct approach would be to employ a forward, backwards, or stepwise selection algorithm that is more likely to identify the correct variables in face of multicollinearity in the model.

However, given a large number of potential predictors statistical models are more difficult to implement. Forward, backwards, and stepwise selection will require a prohibitive number of model runs. Instead, the lasso technique describe above can be used.

The below example adds 996 uncorrelated and irrelevant variables to the four variables already found in Case 3 (i.e. two predictors and two variables highly correlated to these predictors). Again, the hyperparameter value chosen is 0.0001.



Due to the high number of potential variables (1,000), only the variables with non-zero estimates are retained. These are, again, Variable 2 and Variable 4. These are the correct variables.



Conclusion:

Three cases were provided for survival modeling in a Cox proportional hazards model. Each had a small number of potential predictors. In all three cases, the lasso model identified the same (correct) predictors as the statistical model assuming a suitable hyperparameter was chosen. This hyperparameter was not chosen to maximize concordance in the cross-validation set, but rather the highest value was chosen such to not lead to a substantial decrease in concordance.

Note, the lasso approach has few benefits compared to the statistical comparison model given few potential predictors. But given the success encountered in the first three cases, the lasso method was employed to a large number of potential predictors (1,000). In this case, the statistical model would be difficult to employ without reducing the number of potential predictors using some sort of pre-screening criteria. The lasso method again quickly identified the correct predictors.

It's certainly possible that lasso could identify the incorrect predictors, just as this was observed in at least one random dataset used when developing this project. Note, this is also a possibility in statistical testing. However, the hyperparameter selection process in lasso is somewhat less objective than common statistical criteria. It's possible that a two step process could be employed where variables identified by lasso can be confirmed by statistical testing in a statistical model. Note this is similar to pre-screening frequently done to reduce dimensionality in a statistical model. However, using lasso as a pre-screen in probably more accurate than many of the bivariate methods commonly employed.