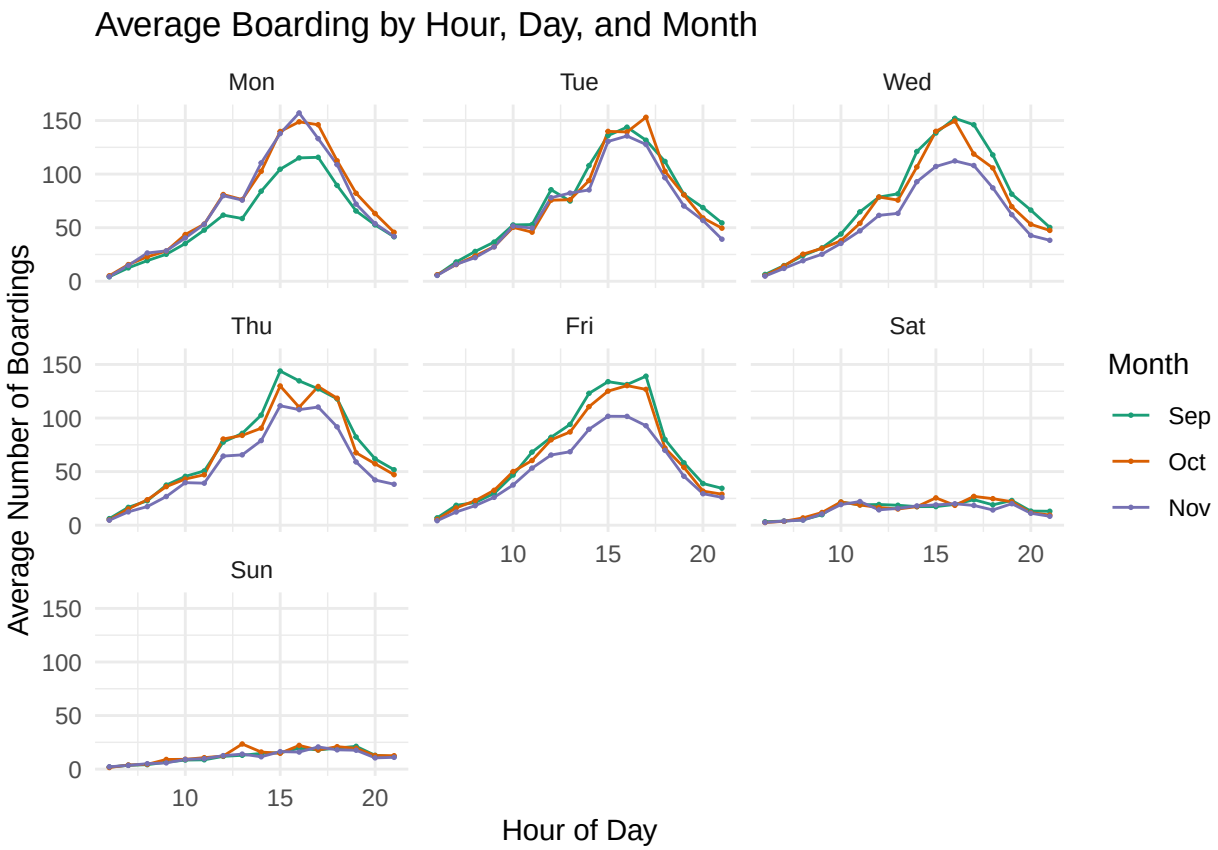
Homework_2_Writeup

Robert Petit

3/3/2022

Problem 1



\begin{figure}[H]
\caption{This figure includes the average boarding rate (in Boarding per 15 minutes) on each hour of day, by day of week (faceted) and month (by color).The peak boarding time is largely similar on weekdays, with slight variations; the main notable variation is on Monday boarding in September,and Wednesday/Thursday/Friday boarding in November. Almost certainly, this is because of Labor Day, a Monday in September, and Thanksgiving Break,which runs Wednesday-Sunday in November. Since these are month long averages, missing one day is a 25% drop in boarding on that day. The only other wide variation of note is the large drop on weekends; given this is a campus bus route, reduced traffic on weekends makes sense.} \end{figure}

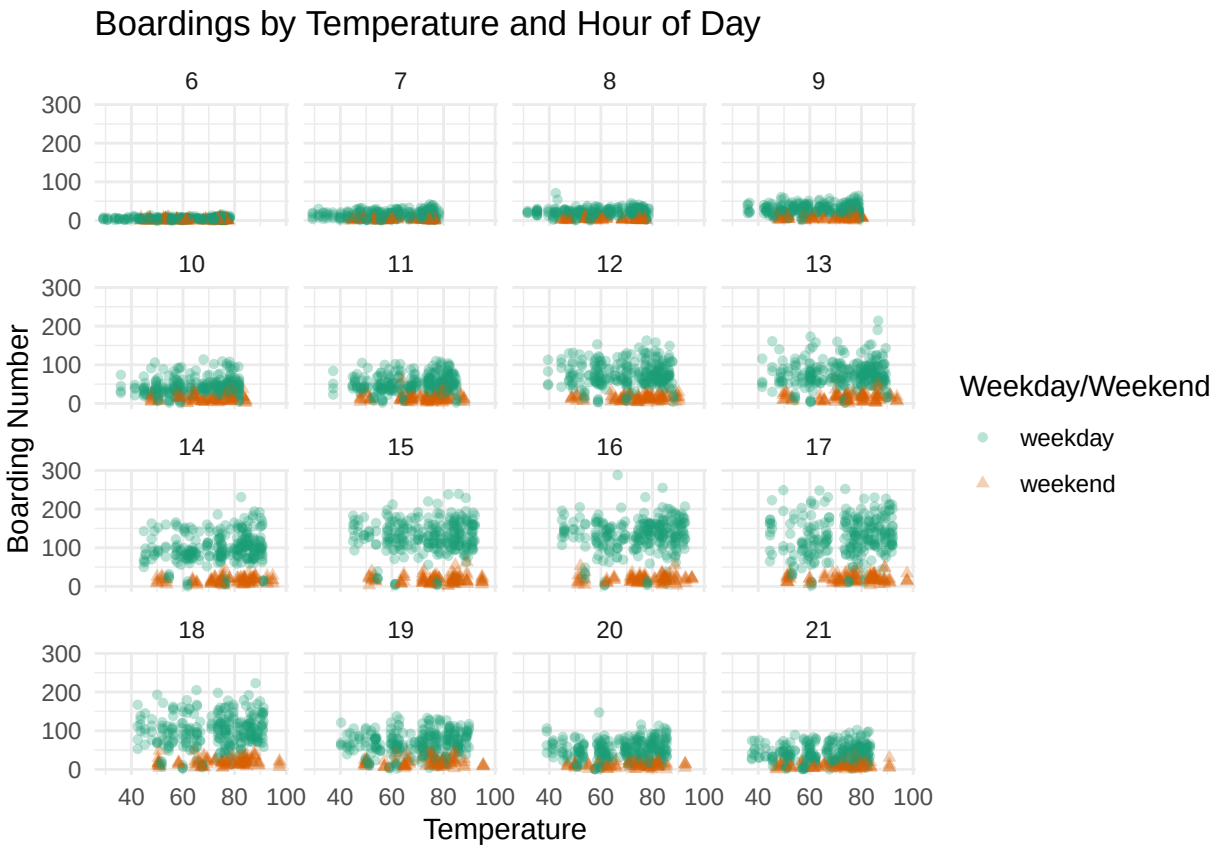


Figure 1: This figure includes the boarding in a fifteen minute window by temperature, separated by hour of the day. For the times in the middle of the day (10:00-17:00) we see some upward trends in the data This indicates that, if there is truly a trend present, it is going to be strongest in those hours. At other times of day, the relationship seems less clear.

Problem 2

This question hinges off the baseline model built in class which regresses price on all of the available *except* pctCollege, sewer, waterfront, landValue, newConstruction. This gives us the coefficients and out of sample errors (based on an 80/20 train/test split)

	Coefficients
(Intercept)	229480.116
lotSize	12899.558
age	4342.850
livingArea	117321.075
bedrooms	-22690.354
fireplaces	2858.017
bathrooms	26543.941
rooms	13694.191
heatinghot water/steam	-12480.257
heatingelectric	-7090.306
fuelelectric	-7571.854

	Coefficients
fueloil	-5955.912
centralAirNo	-19024.016

And an RMSE of 7.33404×10^4

Part A

Generating a linear model that outperforms the baseline model is simple if you select a model with a cross-validated lasso. This gives us the coefficients

	Coefficients
intercept	289308.7381
lotSize	9165.7116
age	-7285.1494
landValue	63462.5558
livingArea	84885.2219
pctCollege	0.0000
bedrooms	-7798.3775
fireplaces	337.0963
bathrooms	29827.5626
rooms	11096.9438
heatinghot air	8944.4265
heatinghot water/steam	0.0000
heatingelectric	0.0000
fuelelectric	0.0000
fueloil	-342.3981
sewerpublic/commercial	0.0000
sewernone	0.0000
waterfrontNo	-114617.3397
newConstructionNo	38442.2646
centralAirNo	-9674.9164

This has an RMSE of 5.874292×10^4. This is somewhat incomparable to the above RMSE for out class model since that RMSE is based on a single train/test split and this is the result of a 10-fold cross-validation. A 10-fold cross-validation average error of the baseline linear model is 6.635451×10^4. Note that these are based on different folds, which limits their values but does show that the lasso model is *clearly* outperforming the baseline.

Part B

In generating a good KNN model, we (unsurprisingly) do well by considering all of the non-zero vairables produced by the Lasso model. Following the standard process of finding the optimal K (and it's associated error) through cross validation.

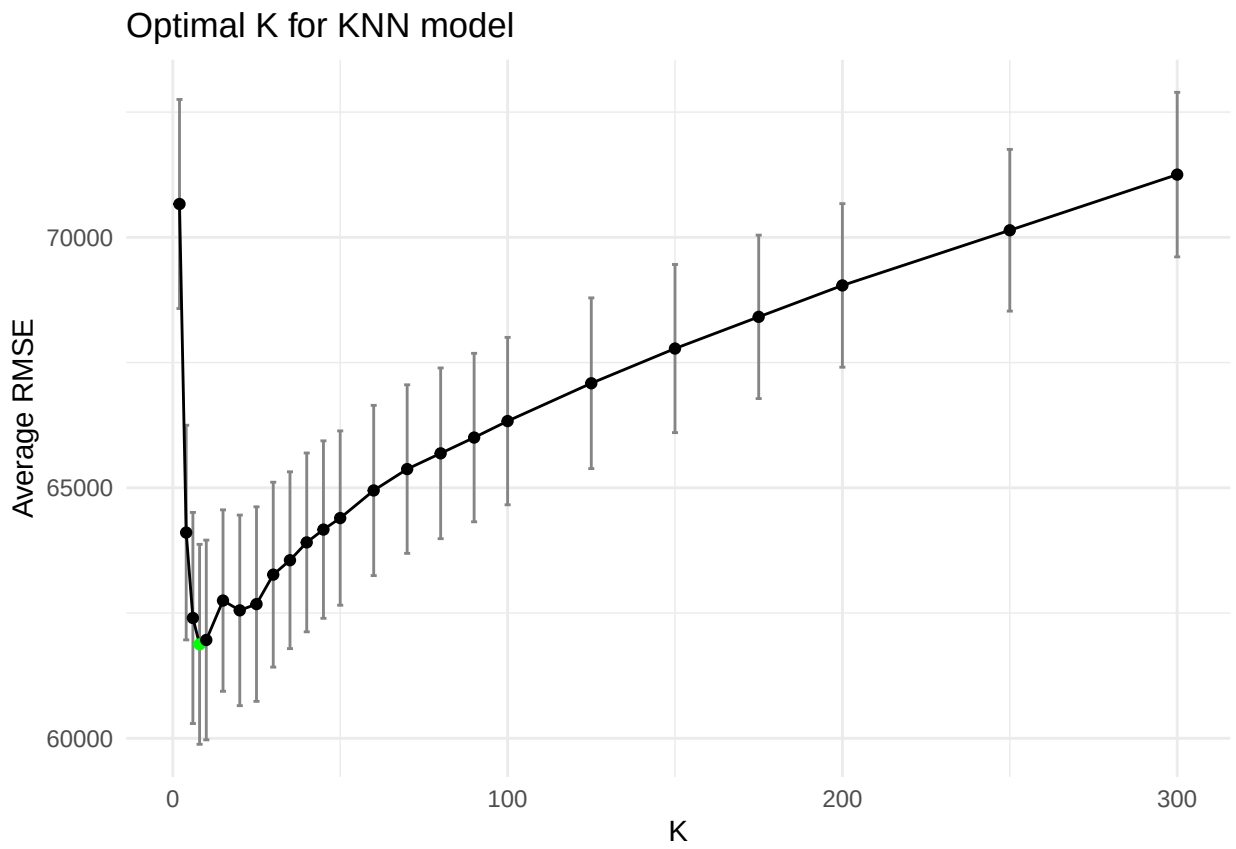


Figure 2: 10-fold cross-validated average errors for various values of K

This gives us an optimal K value of 8 which yields an RMSE of 6.187526×10^4. Notably, these are based on the same folds as the baseline model above, so the values are directly comparable. The net RMSE change $RMSE_{KNN} - RMSE_{Baseline}$ = -4479.25.

Comparison

Both the lasso and the KNN model are performing better than the baseline model in class; the only remaining question is which model performs *best*. In this case, the lasso model is substantially better, with an RMSE change of $RMSE_{Lasso} - RMSE_{KNN}$ = -3132.34. Our clear and stable performance is then $RMSE_{Baseline} > RMSE_{KNN} > RMSE_{Lasso}$. Of course, this is only one result; to verify that it is stable, we can simply repeat the same test as above multiple times.

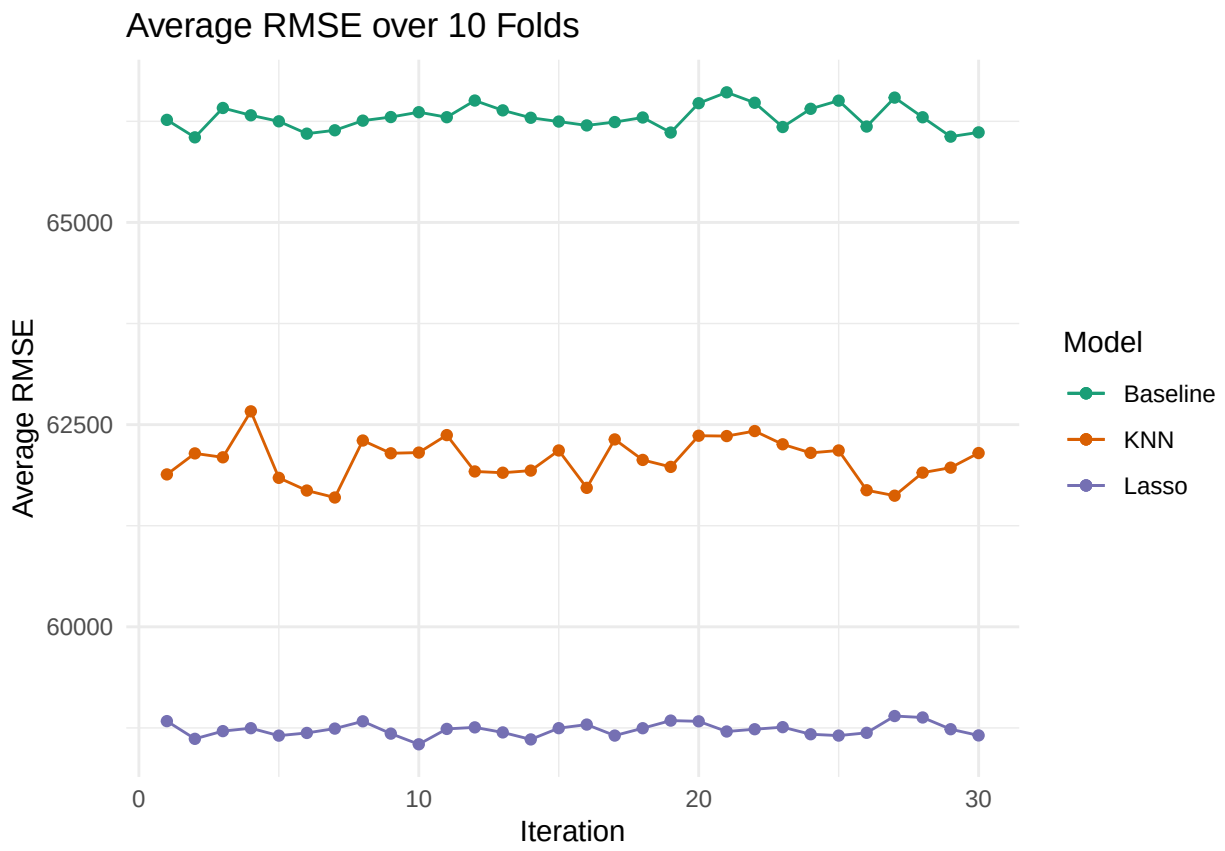


Figure 3: The average RMSE here is given by calculating the RMSE for each of 10 folds of the data (9 in, 1 out) and averaging the ten sets together. This is repeated 30 times with different folds; each fold is constant throughout the iteration. This shows clearly the difference between the models is non-trivial and constant.

Problem 3

A simple visualization of the data shows some surprising results:

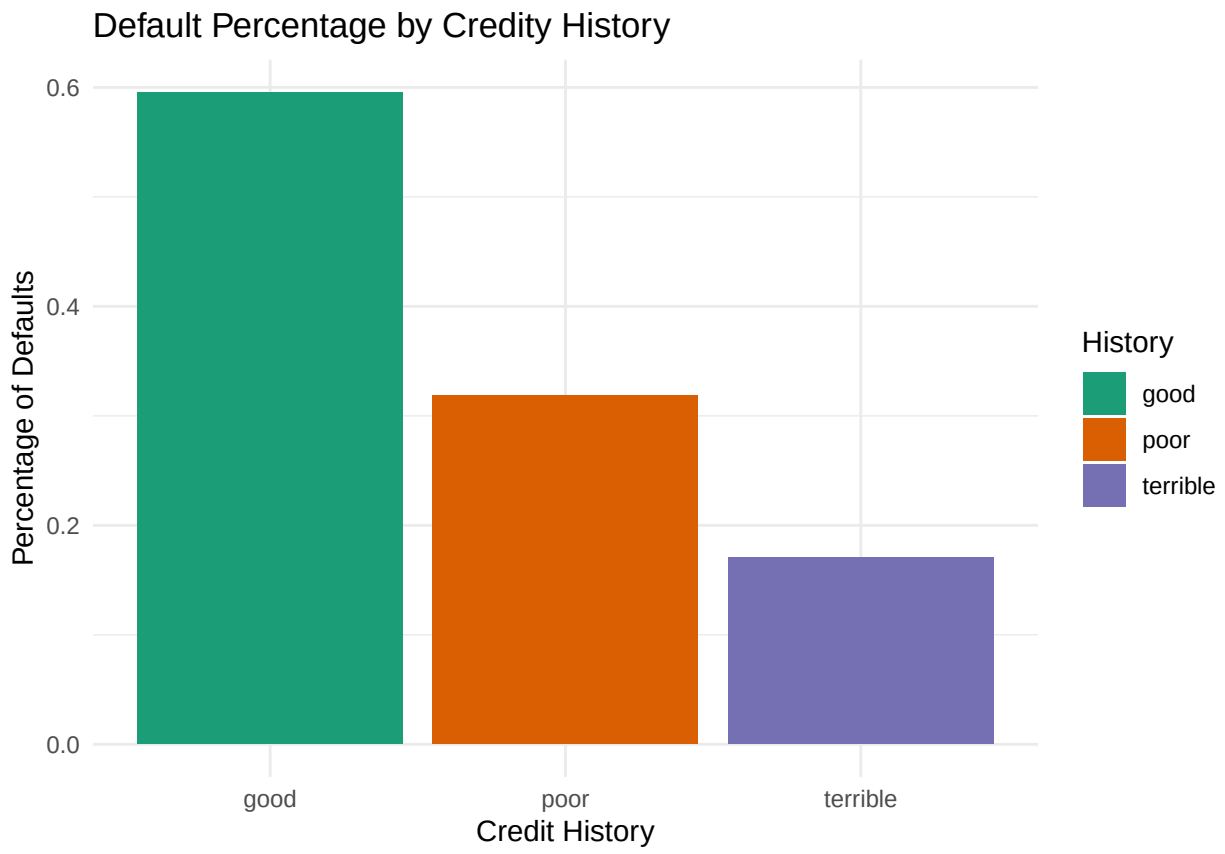


Figure 4: This shows the percentage of loans that defaulted for each given credit history. This gives a pretty clear idea of the effect of oversampling defaulted loans on the outcome liklihood.

With a binomial regression we can get an idea of the individual effects of each variable on the default likelihood. From the regression of $Default = \beta_0 + \beta_1 duration + \beta_2 amount + \beta_3 installment + \beta_4 age + \beta_5 history + \beta_6 purpose + \beta_7 foreign$ we get obtain the coefficients:

	Coefficients
Intercept	-0.71
Duration	0.03
Amount	0.00
Installment	0.22
Age	-0.02
History: Poor	-1.11
History: Terrible	-1.88
Purpose: Education	0.72
Purpose: Goods/Repair	0.10
Purpose: New Car	0.85
Purpose: Used Car	-0.80
Foreign: No	-1.26

Obviously, the interesting point of note is the reduced likelihood to default given a poor or terrible credit history. This might seem weird, but is the inevitable result of the over sampled defaults in the data set. Since

they sample the defaulted loans then matched them with similar loans, you can imagine that there are a ton of loans that were obviously not going to default included, and a substantial number of them are held by debtors with a good credit history. This would result in an inflated chance of defaulting for those groups and a relatively deflated chance for those with better chances of defaulting initially.

Problem 4

Model Building

There are four models of interest here; two baseline models that are given (referred to as baseline 1 and baseline 2) and then a generalized linear model built with a collection of single variables, no interactions, that is the result of several iterations of model building; this model is referred to as "Custom" in all of the figures. The Custom model also includes two engineered features: $ArrivalWDay$, a collection of indicators for the day of the week of the arrival and an additional indicator, $ArrivedWeekend$ for whether the arrival was on a Friday or a Saturday. This variable is helpful, presumably because couples with children are less likely to be travelling during the week than business travelers. Finally, there is a model generated by a lasso regression on all single and pairwise interactions, including engineered features, in the model except for the arrival date; this model is referred to as "Lasso" in the figures and tables. The lasso model is notably larger than any of the other models in our set, with 246 coefficients being considered.

Our first goal is to train these models on a standard train/test split built without using our valuation data. We can generate an accuracy by calculating the number of correct categorizations over the total possible answers.

	Accuracy
Null Model	0.923
Baseline 1	0.923
Baseline 2	0.9387
Custom	0.9387
Lasso	0.9293

On this test/train split, the Baseline 2 and the Baseline 1 model are performing very similarly; this is not surprising since there is very little difference between the two models. The Baseline 1 model is performing identically to the null model: at threshold $t = 0.5$ the Baseline 1 model predicts no children at all, so it is identically the null. The Lasso is performing slightly worse than the other two model on the test set, but the difference is small, so it is unclear if this is due to the lasso not producing a good model in this case, or if the Custom model is over fit for the train data.

After training and initially evaluating out models, we can begin on our evaluation data. To pick a functional threshold we can look to the ROC curves seen in the ROC Curve graph

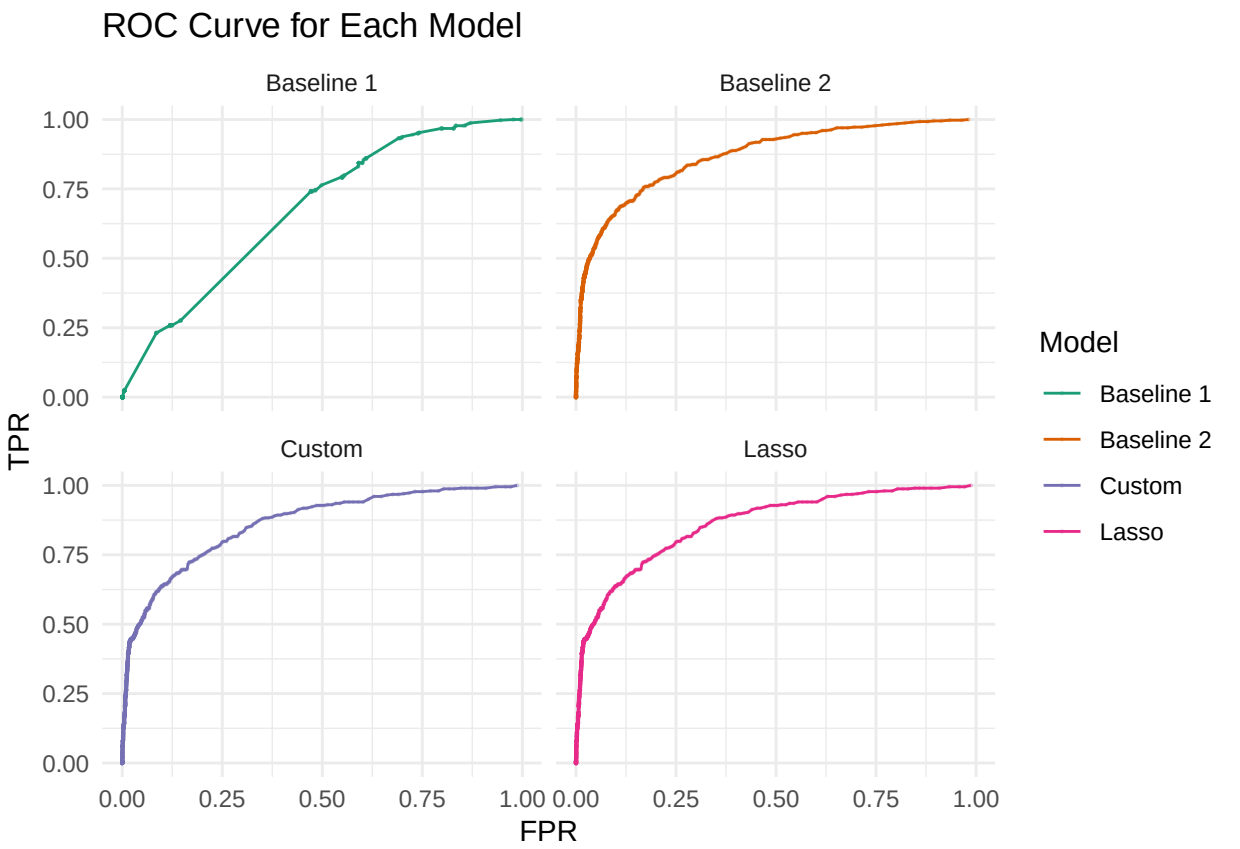


Figure 5: ROC Curve by Model, FPR is calulated as the rate of incorrect positive gueses over total true negatives. TPR is the amount of correct positive guesses over total true positives. Each point represents a different threshold used for calculation; there are 1000 thresholds evenly distributed between 0 and 1.

Interestingly, the Baseline 1 produces almost no correct guesses at all, regardless of the threshold. Each of the points visible on these curves represents a different threshold value, so in addition to the steep line, we can clearly see that there are no accurate thresholds. The other ROC curves are very similar to one another, so choosing a threshold for this model should not require to many judgment decisions. In this case, we choose the threshold that gives us the nearest FPR to 0.25 since that seems to be a sound middle ground; this results in a threshold of 0.0679321. The following results do not substantially change for a number of thresholds around that choice.

We can then split our valuation data into 20 folds and predict the number of visits with children for each of those groups, then compare that to the actual number of visits in each of the groups. By doing this we can generate a "miss number", how many children we are off for a given number of visits.

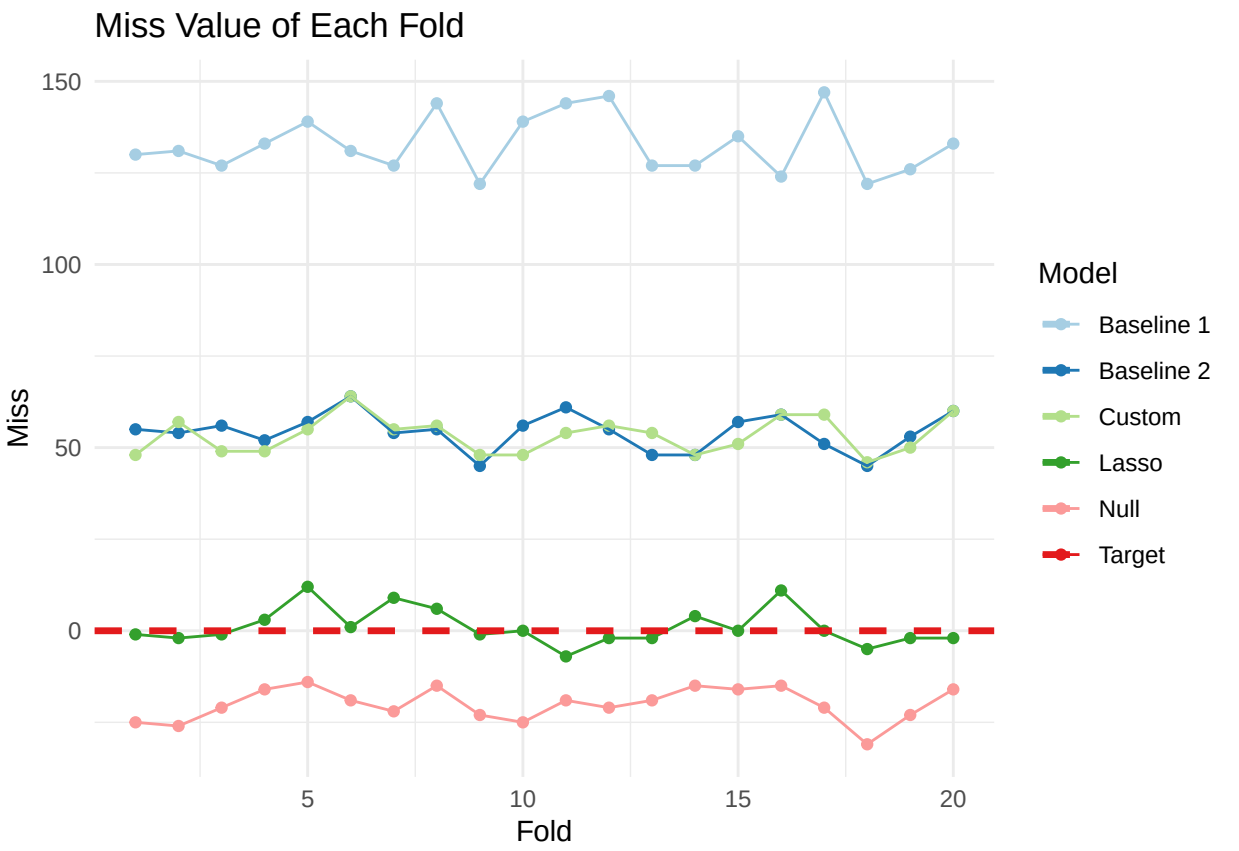


Figure 6: This gives the number of misses by fold. Misses are calculated as the predicted number minus the true number of visits with children. A negative value indicates too few predicted children, a positive number indicates too many predicted children.

This draws a pretty clear relationship between the models. The Lasso model is performing extremely well compared to the others, whereas the Baseline 2 and Custom model, though both are worse than the null model. The only difference is that they over predict where the null model (by design) under predicts. The Baseline 1 model substantially worse here than any other model, including not guessing at all.

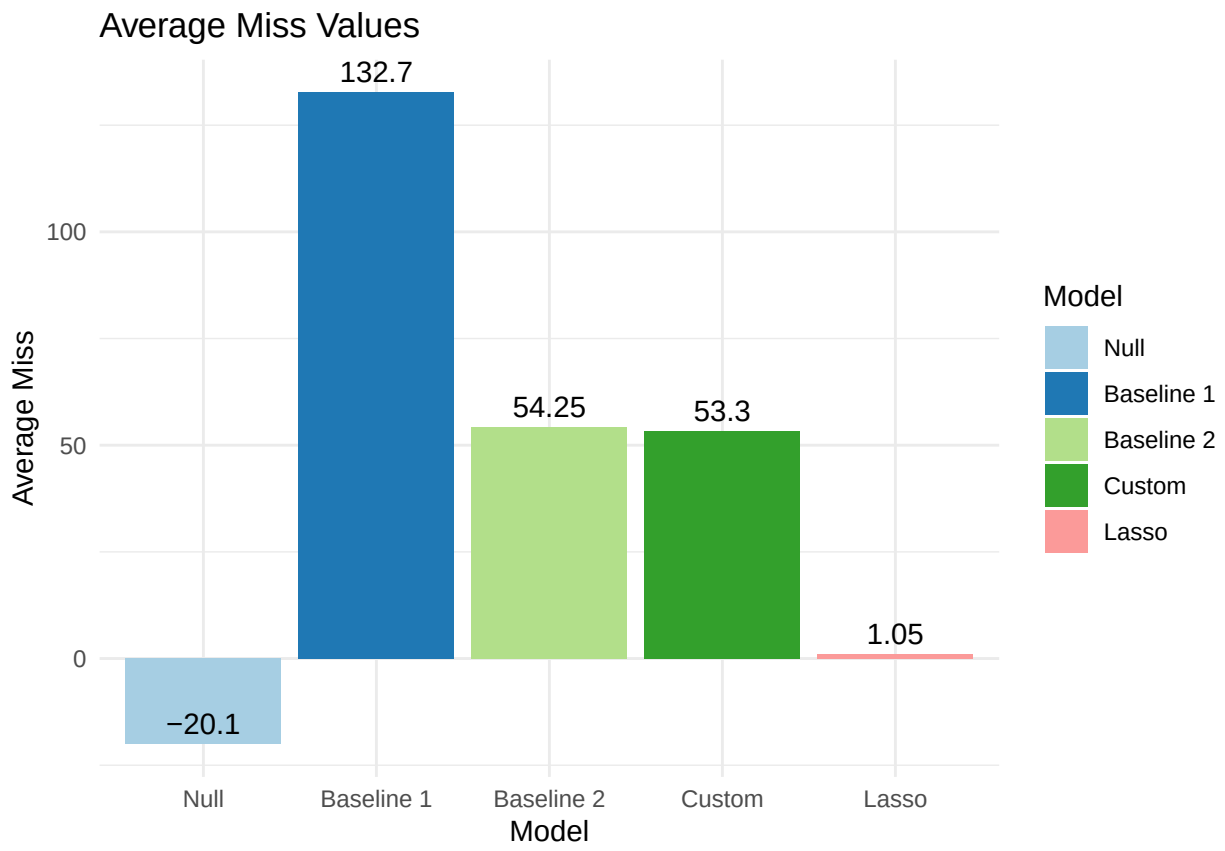


Figure 7: The average number of misses for each model. The lasso with pairwise interactions is clearly the best performer here.

The average miss value tells the clearest story of all; the lasso model is clearly the best performing of all the models; the other models are actually worse than simply guessing no children will be present. The most notable part of this is that we have not chosen a threshold that is particularly favorable to the lasso, relative to other models. If we choose the particular value of acceptable false positives such that the Lasso is the most accurate possible (instead of some more situationally appropriate rate) we can use $FPR = 0.215$ to get a threshold of 0.0759241. By doing this we can achieve a very low miss rate for the lasso, though we do not get significantly better for the other models. Note that in order to do this we only have to shift our acceptable false positive rate by -0.035 and our threshold by 0.007992.

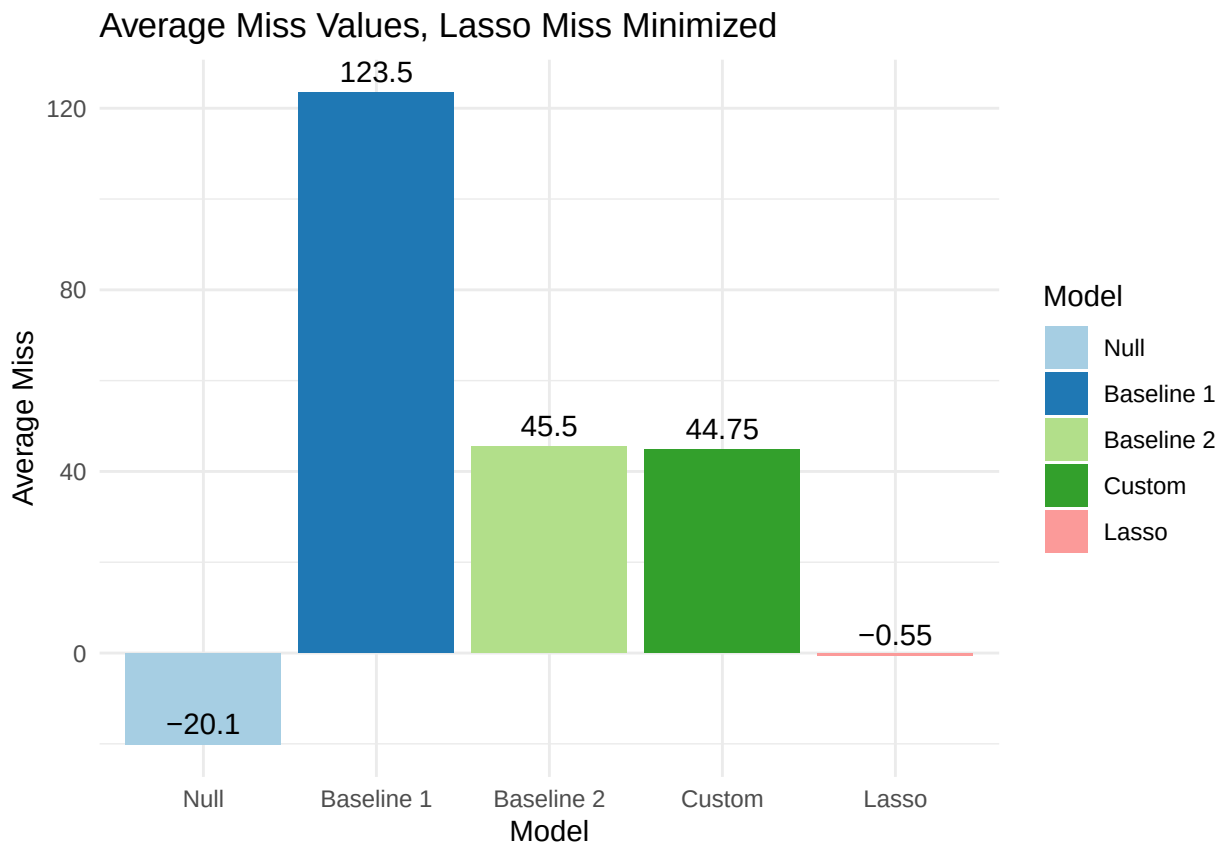


Figure 8: This shows the average miss values if we choose our threshold to intentionally minimize the miss values for the Lasso

So, in conclusion, a lasso-fitted model on all single variables and pairwise interactions is the clearly best performer and, depending on the situation of the actual model, it is possible that it is incredibly accurate.