Applications in Machine Learning to Predict Coronary Heart Disease

Zachary Katz (zak2132), Hun Lee (sl4836), and Tucker Morgan (tlm2152)

5/12/2022

Contents

Introduction

Heart disease accounts for roughly 695,000 fatalities annually in the United States alone, with known risk factors that include high cholesterol, smoking, and blood pressure. In this project, we aim to utilize observations from the Framingham cohort study to predict whether a particular study subject will or will not develop coronary heart disease over one decade's follow-up. Our data subset of longitudinal observations come from Kaggle; its cleaning generally entailed converting appropriate variables to factors (and re-leveling where needed), and renaming and recoding binary 1/0 variables with more descriptive "yes" and "no" for ease of interpretation. While our midterm projects discarded observations lacking one or more features, this iteration attempts to overcome that limitation by using K Nearest Neighbors imputation. The full set of data contained 4,240 total observations across 16 variables, which are:

7 categorical predictors:

- `sex` (self-reported subject sex that takes values "male" or "female")
- `education` (study participant's education that takes ordinal, mutually exclusive values "some_HS" (some high school completed), "HS_grad" (completed high school but did not attend college), "some_college" (attended college but did not graduate), and "college_grad" (graduated university))
- `current_smoker` (binary "yes" or "no" indicating whether the participant was a smoker at the time of physical examination)
- `bp_meds` (binary "yes" or "no" indicating whether the participant was using anti-hypertensive medications at the time of physical examination)

- `prevalent_stroke` (binary "yes" or "no" indicating whether the participant had experienced stroke by the time of physical examination)
- `prevalent_hyp` (binary "yes" or "no" indicating whether the participant was being treated for active hypertension at the time of physical examination)
- `diabetes` (binary "yes" or "no" indicating whether the participant was diagnosed as diabetic according to pre-specified criteria at the time of physical examination).

8 continuous numeric predictors:

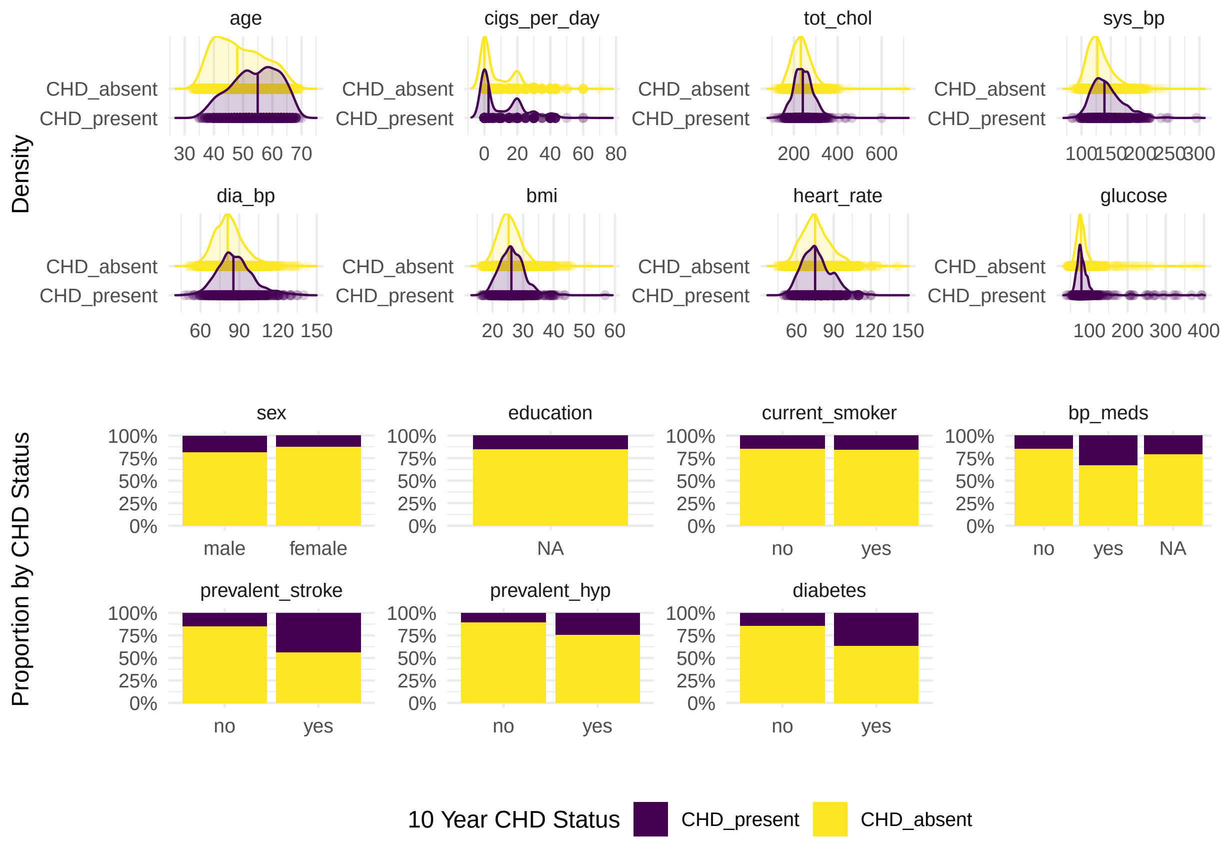
- `age` (age in years at the time of medical examination)
- `cigs_per_day` (average number of cigarettes smoked each day at the time of medical examination, notably not conditioned on smoking status (i.e. for those with `current_smoker` status as "no", should take the value 0))
- `tot_chol` (total blood cholesterol in mg/dL at the time of physical examination)
- `sys_BP` (systolic blood pressure in mm Hg at the time of physical examination)
- `dia_BP` (diastolic blood pressure in mm Hg at the time of physical examination)
- `bmi` (body mass index in kg/m^2 in mm Hg at the time of physical examination)
- `heart_rate` (resting heart rate in beats per minute at the time of physical examination)
- `glucose` (blood glucose level in mg/dL at the time of physical examination)

Finally, beyond our 15 covariates lies our outcome (response) variable `ten_year_chd`, which is a binary indicator for the presence or absence of coronary heart disease (CHD) at 10 years of follow-up. Of our 4,240 observations, 3,596 (84.8%) exhibit absence of CHD, whereas 644 (15.2%) have CHD present – a notable class imbalance.

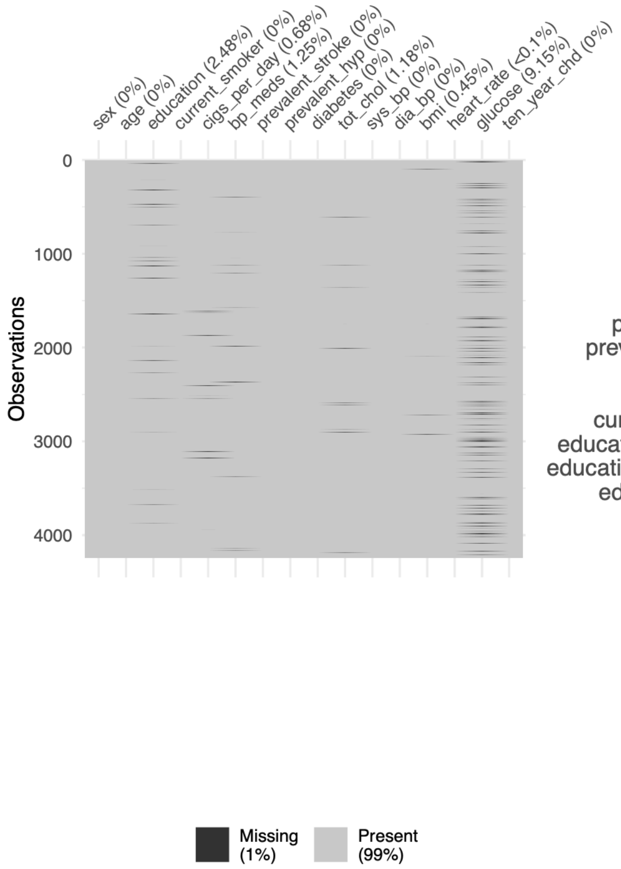
Exploratory Data Analysis

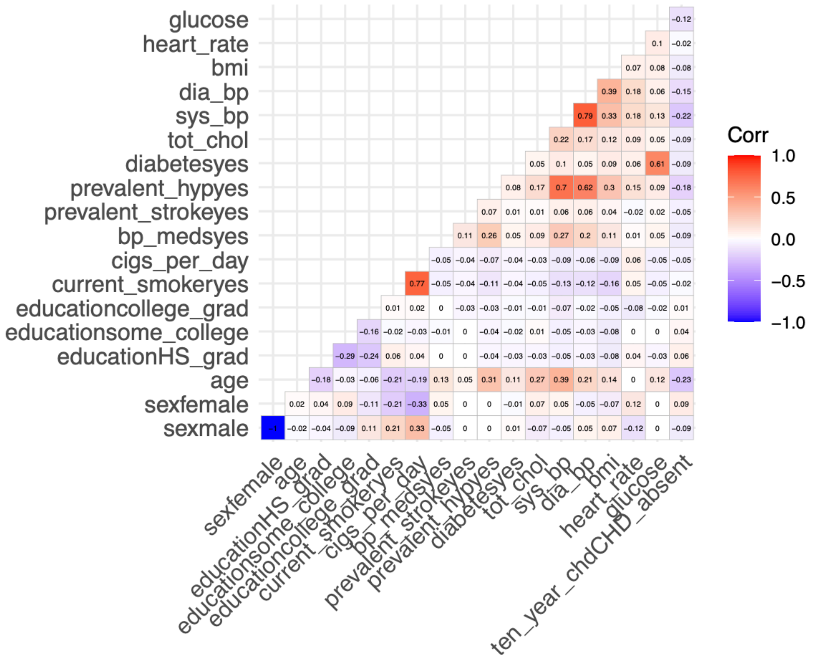
Notably, 582 rows (13.7% of our observations) lacked information for at least one feature, with `glucose` accounting for the plurality of missing data (9.15% missing rate). Moving forward, we assume that our data is at least missing at random, and potentially completely at random; in other words, our missingness depends only on the observed data, and consequently it is sensible to construct a KNN imputation step (using five nearest neighbors) into our preprocessing functionality, which also includes centering, scaling, and BoxCox transformations in all possible cases.

Fig.1: Distributions of Predictors By Outcome Class



When we stratify the distributions of our continuous predictors by outcome status, we find the most substantial differences in median age, systolic blood pressure, glucose, and daily cigarette consumption for those with and without CHD at 10 years. Looking at the proportions that have CHD present or absent across levels of our factor variables, there appears to be the most substantial differences for stroke history, diabetes status, blood pressure medication status, and hypertension prevalence.





We observe no major multicollinearities, with the highest correlations (all sub-0.80) found between systolic and diastolic blood pressure, cigarette smoking and cigarettes smoked per day, prevalent hypertension and blood pressure, and glucose levels and diabetes comorbidity. In addition, CHD status is most correlated with age, systolic blood pressure, and prevalence of hypertension, which is unsurprising given our prior exploratory visualizations stratified by CHD class.

Finally, we conducted checks for various parametric model assumptions, including homogeneity of variance and normality of residuals. Overall, we found that though there were no influential outliers in the data and there was lack of significant collinearity, several of these assumptions were not quite met, which we hypothesized would mean that non-parametric methods or regression methods with regularization, such as Lasso, might be more optimal for prediction.

Modeling

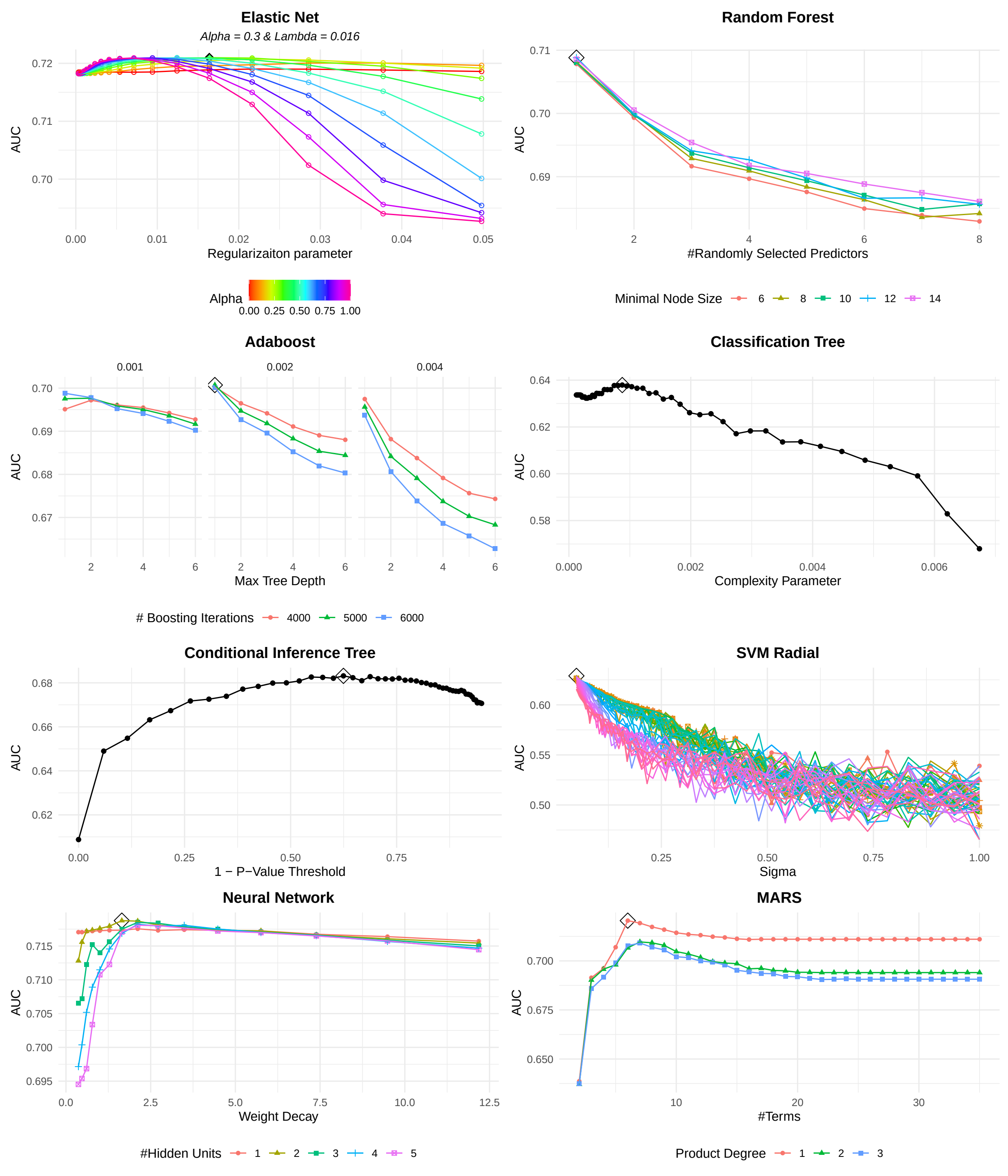
As mentioned, model training for our binary classification and prediction involved the use of all 15 available predictors. Early on, class imbalance was identified as a potential issue, which we attempted to rectify in prior iterations of this work through upsampling and downsampling. However, because this method was ineffective in improving our AUC, the focus here was on trying additional models, including ones with increased flexibility. We began by splitting the data into 80% training set and 20% test set. Given lack of major collinearity and use of KNN imputation for missing data points, all available predictors were included in each model unless a specific model selected them out during the training process. In total, 11 kinds of models were used, with a pre-processing step (imputation, centering, scaling, and Box-Cox transformations) included in each iteration of model training under 10-fold cross-validation, repeated five times. Seeds were set in each instance to ensure reproducibility of results, given the many assumptions and tuning parameters in our variety of models. All tuning parameters were selected using cross-validation in model training to find the optimal model that maximized validation AUC. We constructed:

- **Penalized logistic regression** (elastic net, binomial family, logit link), with objective function that includes loss and penalty given our high number of predictors, and tuning on α (mixing proportion between L1 and L2 penalties, associated with LASSO and ridge, respectively) and λ (a penalty term that limits the number and/or magnitude of predictor coefficients) for our final model. As with a generalized linear model (i.e. logistic regression with no tuning parameters or variable selection), we assume a linear relationship between some outcome $log\frac{\pi}{1-\pi}$, where π is a probability of 10 year CHD presence, and our predictors $X_1, ..., X_n$. Strictly speaking, parametric model assumptions, such normality of residuals may also be helpful to meet in theory, though in practice violations of these assumptions are not always seen as reasons alone not to try them, especially if the modeling process has yielded a satisfactory out-of-sample error rate.
- **Generalized additive model (GAM)** (also binomial family, logit link), with potential inclusion of flexible nonlinearities for some predictors, performed in `caret` for model comparison, and using general cross-validation to determine the smoothness of each operator $\hat{f}_j$ and to search over `GCV.Cp` given unknown scale parameter.
- **Multivariate adaptive regression splines (MARS)** using `earth`, with tuning parameters (1) degree of features (number of possible hinge functions per parameter) and (2) number of terms (may not equal the number of predictors given the possibility of multiple hinge functions), including a stepwise model building procedure involving the addition of piecewise linear models / spline bases, followed by a pruning procedure. In general, MARS models tend to be well-suited for high-dimensional data problems.
- **Linear discriminant analysis (LDA)**, which has no tuning parameters and assumes normally distributed features to classify by nearest centroid following data sphering and projection onto smaller subspaces, tends to work reasonably well with small n or well-separated classes, which doesn't appear true in our case and may make the model less robust.
- **Classification and Regression Tree (CART)**, which involves minimizing the Gini index (a measure of node purity such that a smaller value indicates a higher proportion of observations from a single class in the node) through a top-down, greedy approach that divides the feature space into a set of axis-parallel partitions, followed subsequently by cost-complexity pruning of these splits to obtain a set of optimal subtrees as a function of tuning parameter α, selected through cross-validation. Once α is chosen such that mean squared prediction error from the left out k-th fold is minimized on average, a final subtree corresponding to our chosen α is returned, representing the final tree. Although single trees tend to be less stable than ensemble methods, such as random forest, we include them regardless simply out of interest for how much the use of multiple trees actually improves prediction accuracy.
- **Conditional Inference Tree (CIT)**, a non-parametric alternative to CART, given CART's tendency towards overfitting and selection bias towards covariates with many possible splits. Although CITs continue to use recursive binary partitioning, they use a significance test to select input variables instead of choosing variables that maximize the information measure (e.g. Gini coefficient). Given that our stopping criterion is based on p-values, the tuning parameter in this case is "mincriterion" such that a split is implemented when 1 minus the p-value exceeds this parameter, in order to ensure growing the right sized tree. Unlike CART, pruning does not occur after the largest tree is created. However, stopping may occur early. As with CART, we expect this method to be less stable than ensembles, such as random forest, but include the model here for completeness and personal interest.
- **Random Forest**, an ensemble method that generates a set of bootstrapped training samples as the basis for a set of decorrelated trees, for which a random selection of predictors is chosen as the set of candidates for splitting each time a new split is considered. As with classical classification trees, we utilize a splitting rule based on the Gini index, although other methods, such as cross-entropy, are possible. Here, we do not tune over the splitting rule because we only try the Gini method. Using the `ranger` package to implement the random forest method, our first tuning parameter, `mtry`, represents the number of randomly selected predictors used as split candidates at each split, whereas our second tuning parameter, `min.node.size`, controls the size of the tree by defining the minimum number of observations that can be included in any given terminal node.
- **Boosting**, a similar approach to random forest, however trees are grown sequentially - each tree is

grown using information from previous trees. Given an initial tree, a second tree is fit to the residuals from the first. This second tree is then added into the fitted function in order to update the residuals. Iterative trees are typically small with just a few terminal nodes, determined by number of splits parameter d, leading to relatively slow learning. The shrinkage parameter λ controls the speed of this process to avoid overfitting. The total number of trees, B, is the remaining tuning parameter for this method. If B is too large, the boosted model can overfit the training data.
- **Support Vector Machine (SVM with Linear and Radial Kernels)**, which involves the construction of a hyperplane that maximizes separability, or soft margin between data points of each of the two classes, in high-dimensional feature space, thereby minimizing generalization error. Only some proportion of the training observations serve as support vectors, which lie closest to the hyperplane, or decision surface, and would change that hyperplane's location if removed. Given that the sets between which we'd like to discriminate tend not to be linearly separable, a radial kernel, which results in non-linear decision boundaries, is also used in practice. *Note: we are well aware that SVM should not generally be used to estimate probability of class membership, and that as a result, the AUC metric is not applicable given the weaknesses of Platt calibration. When checking the SVM model, we look therefore not only at the AUC metric, but also the model accuracy and kappa, ultimately determining that regardless of the metric, SVMs are weaker classifiers than some of our better models.* For the linear kernel, our tuning parameter is cost, which is selected using cross-validation and represents the penalty associated with having an observation on the incorrect side of the classification boundary. The smaller the cost, the less flexible the model (unlike, say, the comparable parameter associated with LASSO). For the radial kernel, we add a second tuning parameter, sigma, which controls model flexibility (higher gamma means more flexible model) and is also selected with cross-validation. One inherent limitation to our radial SVM is that we chose to train on only 15% of the training dataset due to computing limitations. This can improve speed of training the model while sacrificing relatively little discretionary performance because the primary focus of SVM models is the boundary between classes rather than the full scope of observations.
- **Neural Network**, which mimics a biological nervous system by modeling connections with weights between nodes. Each predictor is considered an input, which is then modified by weight and summed into a linear combination; this may occur iteratively through sets of hidden layers until one final output layer is reached. Activation functions provide non-linearity to neural networks, and for classification tasks, the softmax activation function (a generalization of logistic functions used for multiclass problems) tends to be used in order to normalize the network outputs to a multinomial probability distribution over predicted output classes. Our model has two tuning parameters set by cross-validation: decay and size. Decay, or weight decay, is a regularization technique applied to network weights in order to reduce overfitting, while size is the number of units in the hidden layer, and can be zero if there are skip-layer units.

Optimal Tuning Parameters

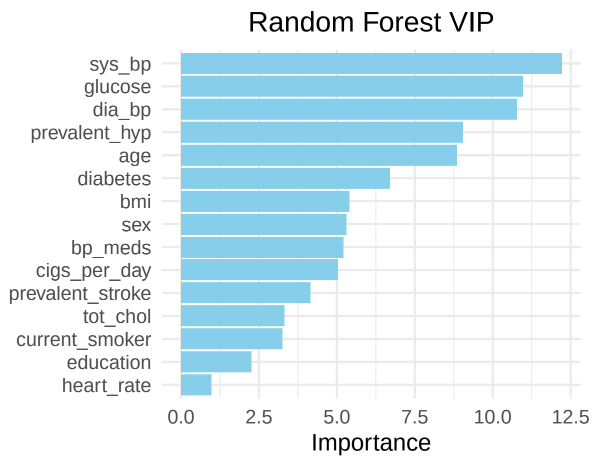
Our tuning parameters were selected via 10-fold cross-validation repeated 5 times using only the training data to maximize validation AUC. We chose AUC as an evaluation measure because it is generally invariant to classification decision boundaries. Our optimal elastic net model had parameters $\alpha = 0.3$ and $\lambda = 0.0164$, where $\alpha = 0$ indicates pure ridge regression and $\alpha = 1$ indicates pure LASSO regression, so our final model is closer to ridge. The optimal MARS model had a product degree of 1 and 'nprune' of 6. Our optimal random forest model used 1 randomly selected predictor (known as `mtry` in some contexts) and a minimum node size of 2 observations per node. Our optimal boosted model had a maximum tree depth of 1 with 5,000 boosting iterations and a shrinkage factor of 0.002. The optimal cost parameter for linear SVM was 0.00777. Our optimal radial SVM model had a σ value of 0.0498 and a cost parameter of 2.61. The optimal parameters for our neural network model were size 2 and decay of 1.65.

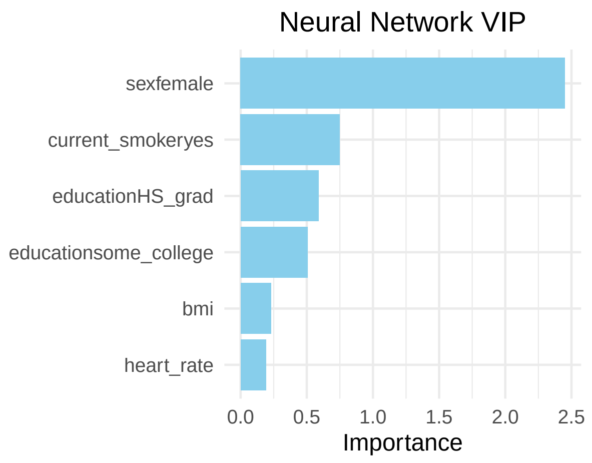


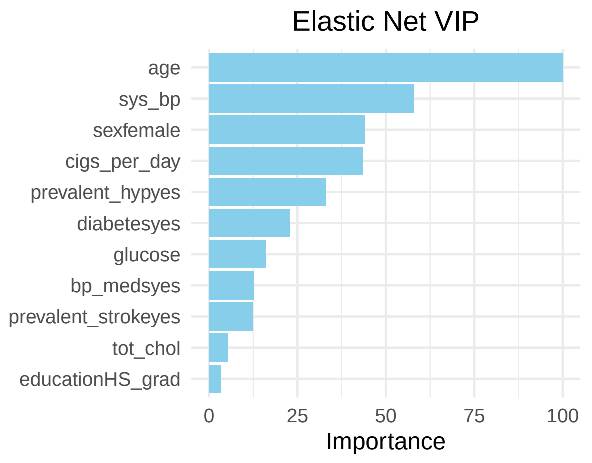
Variable Importance

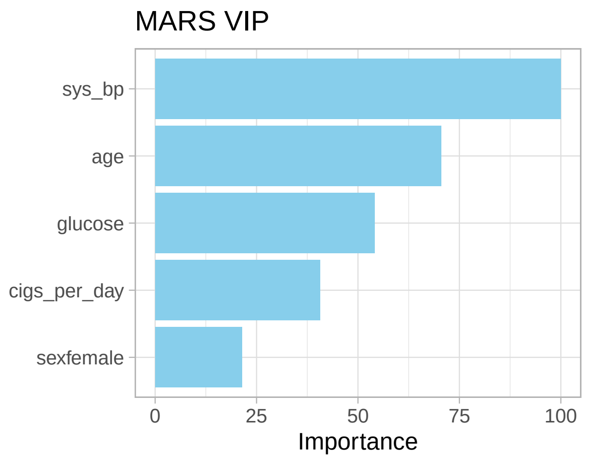
The most important variables in our elastic net model were age, systolic blood pressure, sex, and cigarettes per day. The GAM identified age, glucose, and sex as the most important variables. The most important variables in the MARS model were systolic blood pressure, age, sex, and glucose. For our random forest model, we looked at permutation and impurity methods to assess variable importance. When using the permutation method, we identified systolic blood pressure, diastolic blood pressure, prevalence of hypertension, and glucose as important variables. Similarly, when using the impurity method, we identified systolic blood pressure, age, diastolic blood pressure, and glucose. The most important variables in our boosted model were age, systolic blood pressure, glucose, and cigarettes per day. Our neural network model identified similar important variables in age, glucose, and systolic blood pressure.

Overall, we tended to see the following variables as being important across our analysis: systolic blood pressure, age, glucose, and sex. This is perhaps unsurprising given our exploratory data analysis and relatively better class separability for these features than for others.





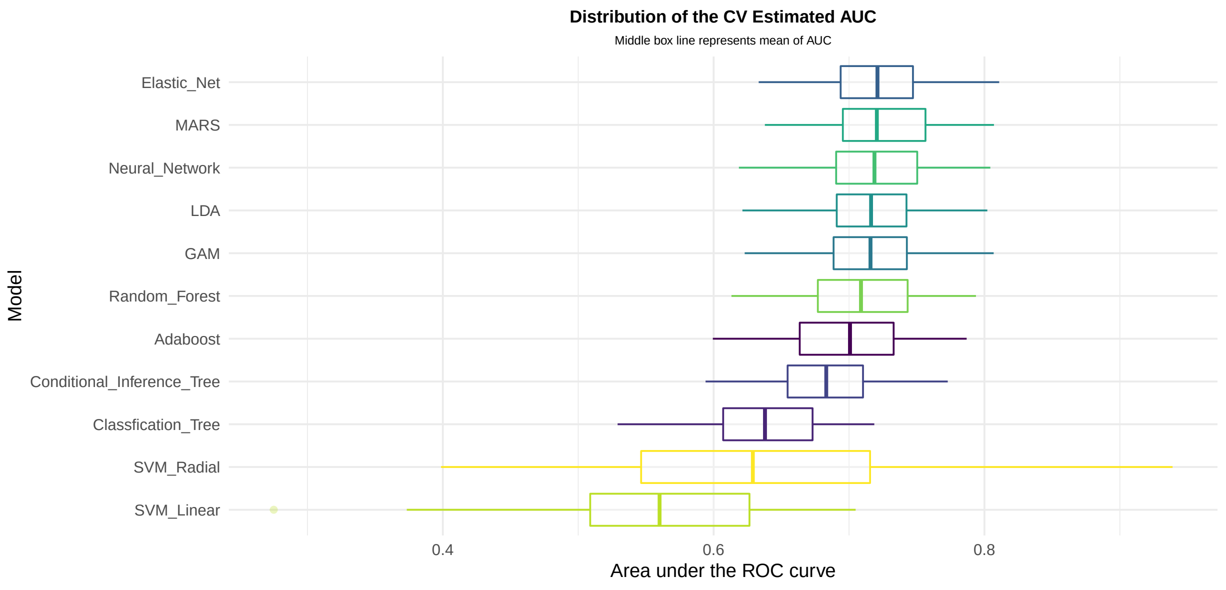


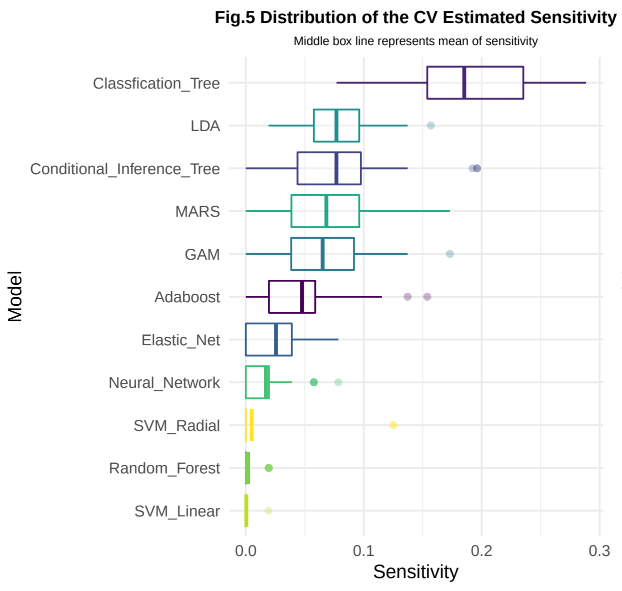


Resampling Results and Model Selection

Having trained and fit both more and less flexible models, we then explore the resampling AUC across models and find generally limited stratification with a couple of notable exceptions – namely, significantly lower performance from support vector machines in particular, but also from our tree models (classification tree especially). Although the SVM with a radial kernel did indeed provide significant improvement over the linear SVM as hypothesized, the performance was still far below our other models. This may be in part due to the fact that given the limitations of our machine power, we could only train the radial SVM on a much smaller proportion of our training observations, which leaves room for future exploration. Again, we also note that SVMs cannot quite be compared on AUC given the use of Platt's probabilities; regardless, we check other metrics, such as accuracy and kappa, and find that these, too, underperform other models – so we would not select SVM models anyway.

Our top 5 models (elastic net, MARS, neural network, LDA, and GAM) are all extremely close in resampling AUC, falling in the 0.716 to 0.721 range. In general, we prefer to use the simplest, most easily interpretable model, which luckily also happens to be our highest performing model on this metric: the elastic net model. Our elastic net model has an AUC of 0.721 and is closer to the ridge model than to LASSO. This AUC means that even our best model is only "okay" or "decent"; its accuracy of just over 85% is also not significantly above the No Information Rate of 0.849, which means we see only marginal improvement once we take into account the inherent class imbalance found in our data. As noted, upsampling or manual changes to the probability threshold for classification did not ultimately lead to significant improvement, hence its omission here. It's also worth noting that though none of our models achieves a particularly high sensitivity, the elastic net model also achieves a middle ground in the sensitivity/specificity tradeoff, falling in the middle of the pack compared to other models on both metrics.





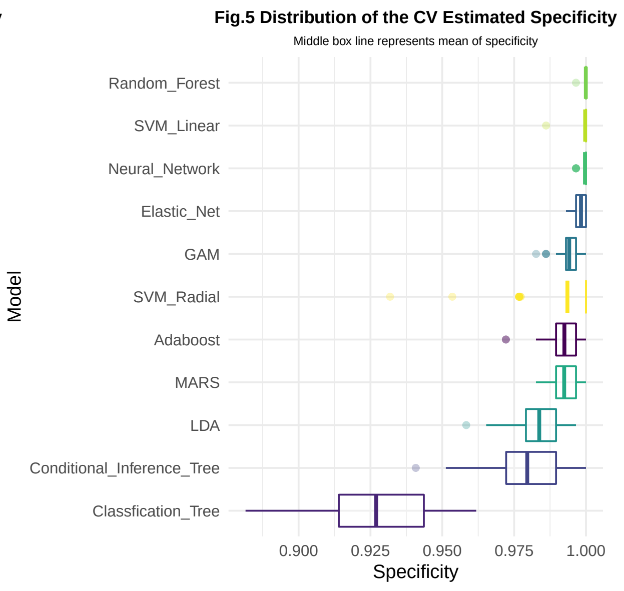
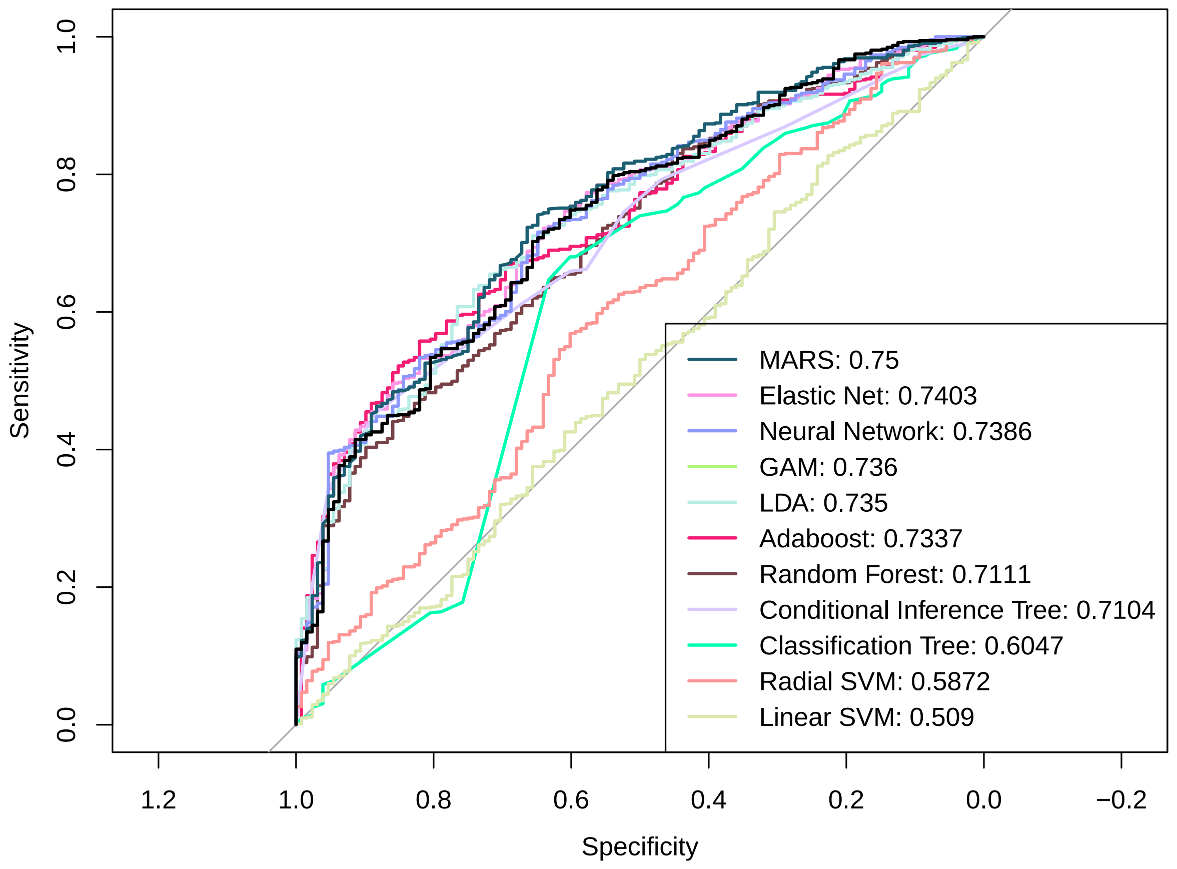


Table 1: AUC from 10-fold 5 Repeated CV Estimate

Models	Min.	X1st.Qu.	Median	Mean	X3rd.Qu.	Max.
Elastic_Net	0.633	0.694	0.727	0.721	0.747	0.811
MARS	0.638	0.695	0.723	0.721	0.756	0.807
Neural_Network	0.619	0.691	0.721	0.719	0.750	0.804
LDA	0.621	0.691	0.719	0.716	0.742	0.802
GAM	0.623	0.689	0.718	0.716	0.743	0.807
Random_Forest	0.613	0.677	0.714	0.709	0.743	0.794
Adaboost	0.599	0.664	0.708	0.701	0.733	0.787
Conditional_Inference_Tree	0.594	0.655	0.683	0.683	0.710	0.773
Classfication_Tree	0.529	0.607	0.639	0.638	0.673	0.719
SVM_Radial	0.399	0.546	0.638	0.629	0.716	0.939
SVM_Linear	0.275	0.509	0.588	0.560	0.626	0.705

Model Application to Test Data



In the ROC plot above, we see the area-under-curve of our selected model applied to our test data with a value of 0.7403, indicating middling classification performance. Despite our efforts with data imputation, it looks like our model still suffers due to class imbalance.

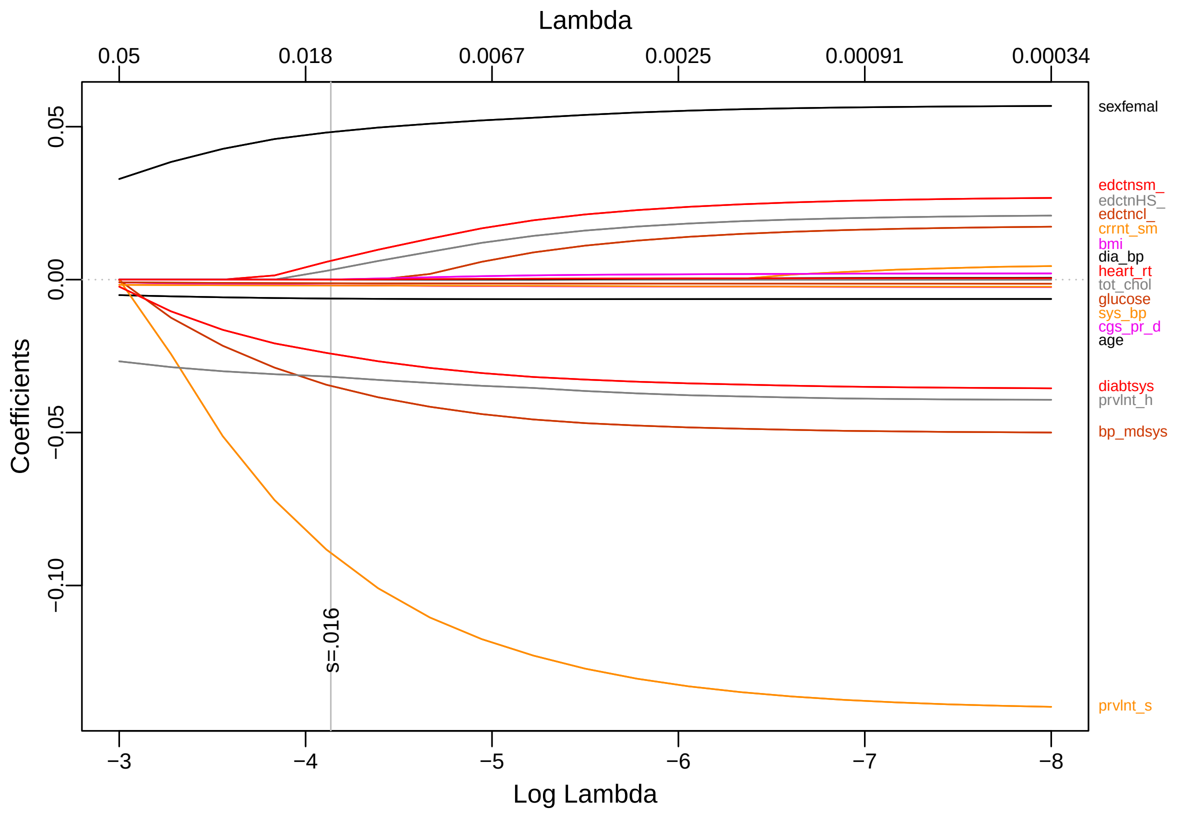
Limitations

As noted throughout, class imbalance was a difficulty for model training and performance. Despite our best efforts at upsampling the less common class (CHD presence), we could not significantly improve model performance. We would have preferred to have at least 1000 observations in each class. Another limitation here is that we used the `recipes` implementation of KNN imputation, for the most part, since this works for both categorical and continuous variables, and can easily work within `caret` model training during repeated cross-validation. However, our future work might explore whether bagged imputation might be more effective, given that our predictor with highest missingness (glucose) has one clear correlate that is essentially complete and has a clear correlational mechanism (diabetes). In addition, interaction terms are excluded from this analysis, although there may very well be some effect measure modification at play. As mentioned, if we had more computational power, we would try the radial SVM without having to train a model only on a smaller number of training observations than our machines would allow – although we'd still face the same issues in making comparisons AUC given the use of Platt's probabilities. We note as well the many violations found for parametric model assumptions, which may inherently limit the utility of models like elastic net. Our AUC may also improve with the inclusion of more and other types of predictors, such as race, that are missing from our data set, but were certainly recorded during the original Framingham Heart Study. Such features may be more closely related to and more highly predictive of our 10-year CHD follow-up outcome.

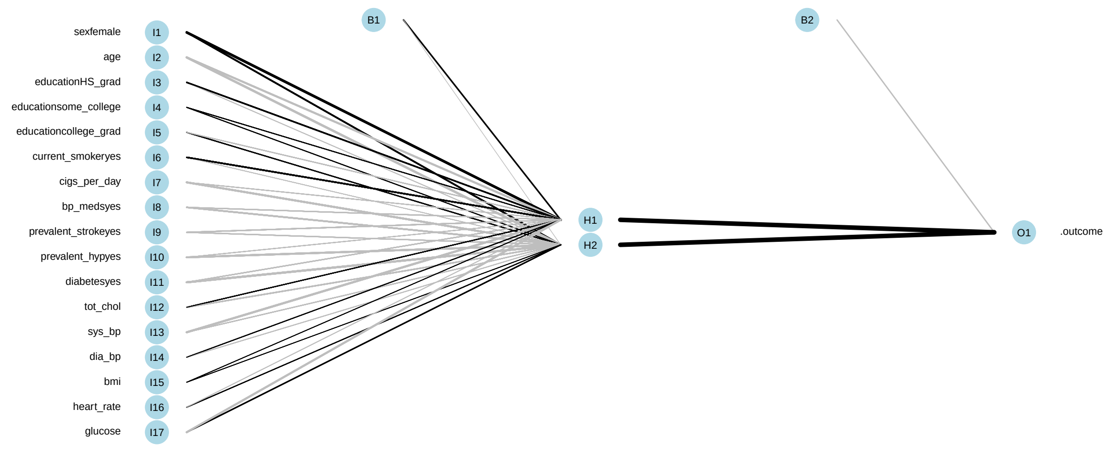
Conclusion

Overall, our modeling tells us that the most important determinants of heart disease tend to be systolic blood pressure, age, glucose, and sex, which played important roles in most of our models and jibes with our exploratory analysis. That said, other covariates, such as race and income, may ultimately have had stronger explanatory power of CHD status at 10 years had they been included. Although our attempt to find a model with AUC above 0.80 did not prevail – even with upsampling and despite the inclusion of models with varying degrees of flexibility – we settled on a reasonably-performing penalized logistic regression model, i.e. elastic net, that did a fair job when applied to our test data. Regardless, the consistency of variable importance in our parametric models, and the fact that they generally performed better than our non-parametric and/or black-box models (with a couple of exceptions, including the neural network) may ostensibly be used by physicians going forward to help identify patients who may be particularly susceptible to CHD in the future. A brief literature review confirms that these variables have been noted as important risk factors for CHD by medical professionals and researchers in the past, which lends additional credibility to our findings.

```r
training_df <- training_df %>% na.omit()

x_train = model.matrix(ten_year_chd ~ ., training_df)[, -1]

y_train = training_df$ten_year_chd %>% as.numeric()

mod <- glmnet::glmnet(x = x_train, y = y_train, alpha = 0.3, lambda = exp(seq(-8, -3, length = 19)))

plot_glmnet(mod,label = TRUE, s = 0.016)
```



```
plotnet(nnet_fit_impute$finalModel)
```


B1
B2
sexfemale
age
educationHS_grad
educationsome_college
educationcollege_grad
current_smokeryes
cigs_per_day
bp_medsyes
prevalent_strokeyes
prevalent_hypyes
diabetesyes
tot_chol
sys_bp
dia_bp
bmi
heart_rate
glucose
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
I16
I17
H1
H2
O1
.outcome