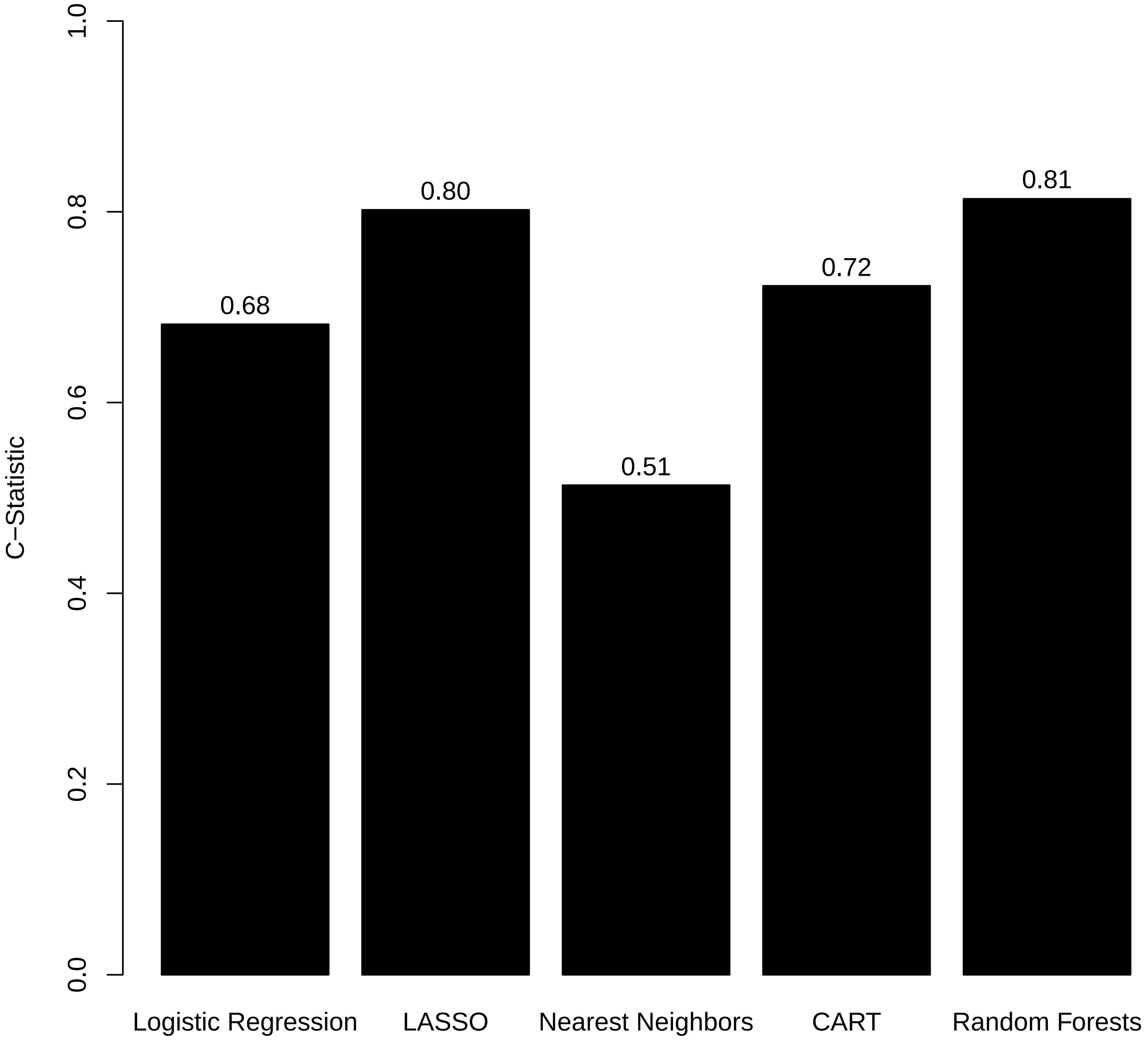
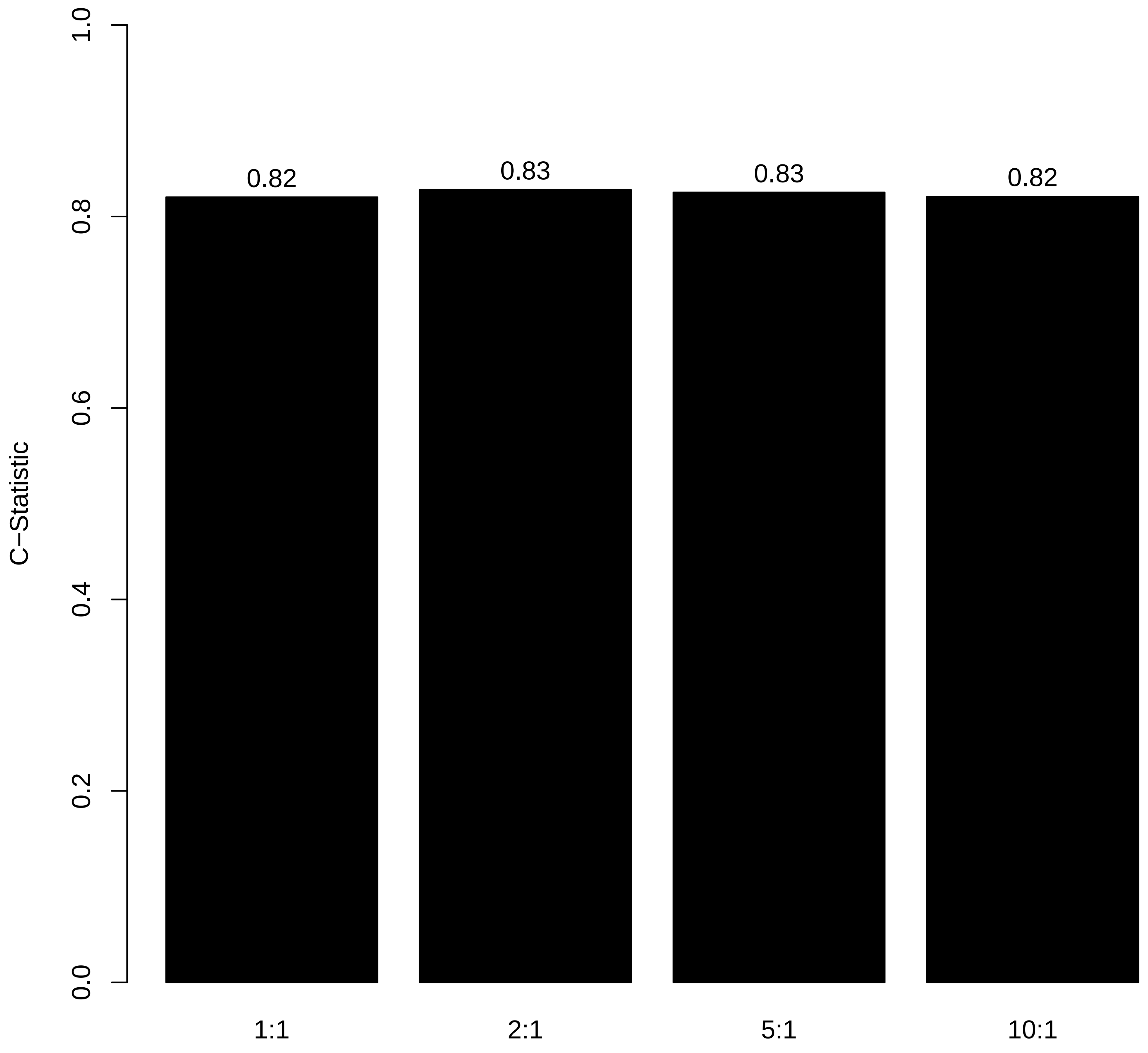

C−Statistic
0.0
0.2
0.4
0.6
0.8
1.0
0.68
0.80
0.51
0.72
0.81
Logistic Regression
LASSO
Nearest Neighbors
CART
Random Forests


C−Statistic
0.0
0.2
0.4
0.6
0.8
1.0
0.82
0.83
0.83
0.82
1:1
2:1
5:1
10:1

Supplemental Figure Legend

Figure S1: Cross-Validated AUC Across Different algorithms. Comparison of different machine learning algorithms using 10-fold cross-validation on the training data. The two most adaptive algorithms, LASSO and Random Forests perform best.

Figure S2: Cross-Validated AUC across different Ratios Cases to Controls. Comparison of ratios of cases to controls using 10-fold cross-validation on the training data. Adding extra controls does not improve overall model performance.

Technical Appendix

Machine learning (ML) is an increasingly popular tool in medical statistics. While traditional statistics requires the pre-specification of a working model (as in regression analysis) ML is a field of statistics that uses an algorithmic approach to *search* for the best data model. ML algorithms differ based on how they approach this search and what their underlying fitting process consists of, referred to as *basis functions*. The best algorithm for a particular data problem will depend on how these basis functions capture the true underlying relationship.

ML approaches have both strengths and limitations over traditional regression models. The primary strength is that they do not require the user to pre-specify the relationship between the predictor variables and the outcome. For example, in the given data problem it was known that clinical factors like blood pressure were likely related to sudden cardiac death, but the exact nature of the relationship was not known a priori. A ML approach allows for various non-linear effects to be modeled. In doing so, the hope is that a better approximation of the “true model” is found. Such flexibility is often not possible within regression models. The cost of such flexibility is that any fit is likely to be an overfit. This is why it is necessary to use cross-validation procedures or in our case sample splitting. Moreover, since ML algorithms are often non-parametric (very general functions), it is generally not possible to estimate the “effect” (Beta value) of a predictor on the outcome. However many procedures have *adhoc* ways of defining variable importance.

For scientific questions where the goal is to predict an outcome from a set of variables and effect estimation is not of primary importance, ML can be very useful. In this paper we utilized the ML algorithm Random Forests, introduced by Leo Breiman in 2001[1]. Figure S3 presents a schematic of RF. At its core it consists of a series of decision trees. Decision trees have been popularized by the ML algorithm CART [2]. CART is a recursive algorithm that represents data in a tree structure via a series of binary splits (Figure S4). The algorithm searches over all possible cut points of all available variables for the optimal split. The optimal split is chosen based on the minimization of a pre-specified loss function. In the example shown (simulated data), two predictor variables are represented: systolic blood pressure (SBP) and diastolic blood pressure (DBP). The outcome here is binary (e.g. death). The first split is on SBP. If a person has SBP > 163.9 they move down the right side of the tree and 97% of these people eventually die. If one's SBP < 163.9 they move down the left side of the tree. The next

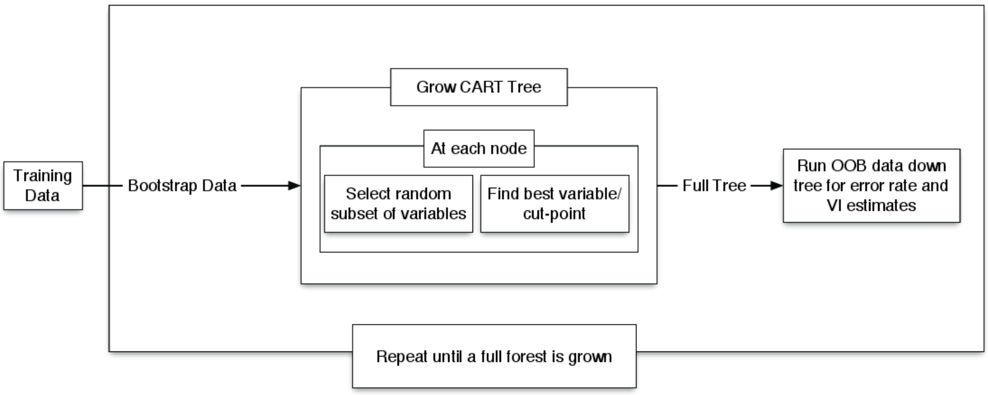

Training Data
Bootstrap Data
Grow CART Tree
At each node
Select random subset of variables
Find best variable/ cut-point
Full Tree
Run OOB data down tree for error rate and VI estimates
Repeat until a full forest is grown

Figure S3: Schematic of the Random Forests Procedure

split is also SBP. Among people with SBP < 148.6 only 1% of people die. If one's SBP > 148.6 (but less than 163.9) the next split to consider is DBP. 5% of people with DBP < 70.83 die while 70% above die. This toy example illustrates how trees represent a complex relationship, particularly non-linear or interaction effects. Moreover, a CART tree can serve as a predictor for a future observation. To predict, one simply passes a person down the tree and observes which terminal node they land in. For example, a person with SBP of 150 and DBP of 60 would fall in the second terminal node and have a predicted probability of death of 5%.

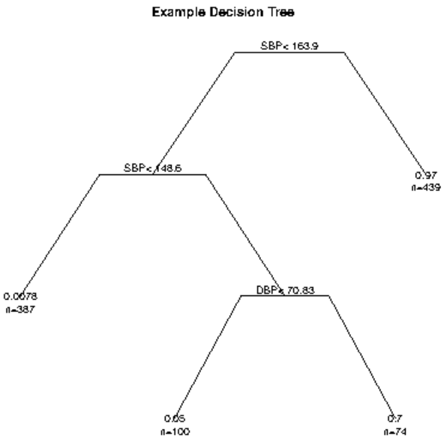


Figure S4: A sample decision tree with SBP and DBP as predictors.

RF is an extension of CART, where instead of "growing" one tree, one grows many trees. In our analysis we grew 2,000 such trees. The number of trees to grow is referred to as a "tuning parameter." Some analyses may require fewer trees while others may require more (see Goldstein et al. [3] for discussion of the different tuning parameters and how to select them). Since growing a tree is a deterministic process, RF injects randomization into the process by selecting bootstrap samples of the data at each iteration and by searching over only subset of the variables at each node. This produces enough variability in the trees, that once combined together create a more stable and robust predictor (see [3] for more theoretical justification). One additional advantage of the bootstrapping process is that in each iteration a subset of the data is left out (approximately 37%). The "out-of-bag" sample serves as an independent validation set for the given tree. Over the "forest" of trees this allows for a measure of fit, similar to cross-validation.

While RF results in a better predictor, the cost is a more obscured model. It is easy to interpret one tree but it is much harder to interpret multiple trees. To aid in the identification of important variables, various metrics have been proposed. The most common metric, and that used in our paper, is the permutation importance. Conceptually, the permutation importance measures the importance of a variable to the prediction model. After the "forest" of trees is created, each predictor variable is permuted. The observed decrease in prediction accuracy is the permutation importance. This importance measure does not have any real-world interpretation (like a Beta value would in a regression model) and to this point there are not even any statistical tests. However, it does provide a relative ranking of which variables are most important.

In our analysis a series of RF models were fit. For each model, we grew 2,000 trees. This value was chosen based on minimization of the out-of-bag error rate. The other primary tuning parameter is the number of variables to search over at each node. The standard value is the square-root of the number of predictors. We used this value, also after exploring the out-of-bag error rate. Other tuning parameters were similarly chosen. For variable importance we used the permutation importance and examined the top 6 variables. This was an arbitrary choice though further examination did not change the overall impression of important variables.

1. Breiman L. Random Forests. *Machine Learning*. 2001;45:5–32.

2. Breiman L, Friedman J, Olshen R, Stone C. *Classification and Regression Trees*. New York: Chapman & Hall; 1984.

3. Goldstein BA, Polley, E.C., Briggs, F. B.S. Random Forests for Genetic Association Studies. *Statistical applications in genetics and molecular biology*. 10(1):32.