OPEN

Can Hyperparameter Tuning Improve the Performance of a Super Learner?

A Case Study

Jenna Wong,[a] *Travis Manderson,*[b] *Michal Abrahamowicz,*[a] *David L Buckeridge,*[a] *and Robyn Tamblyn*[a]

Background: Super learning is an ensemble machine learning approach used increasingly as an alternative to classical prediction techniques. When implementing super learning, however, not tuning the hyperparameters of the algorithms in it may adversely affect the performance of the super learner.

Methods: In this case study, we used data from a Canadian electronic prescribing system to predict when primary care physicians prescribed antidepressants for indications other than depression. The analysis included 73,576 antidepressant prescriptions and 373 candidate predictors. We derived two super learners: one using tuned hyperparameter values for each machine learning algorithm identified through an iterative grid search procedure and the other using the default values. We compared the performance of the tuned super learner to that of the super learner using default values ("untuned") and a carefully constructed logistic regression model from a previous analysis.

Results: The tuned super learner had a scaled Brier score (R^2) of 0.322 (95% [confidence interval] CI = 0.267, 0.362). In comparison, the untuned super learner had a scaled Brier score of 0.309 (95% CI = 0.256, 0.353), corresponding to an efficiency loss of 4% (relative efficiency 0.96; 95% CI = 0.93, 0.99). The previously-derived logistic regression model had a scaled Brier score of 0.307 (95% CI = 0.245, 0.360), corresponding to an efficiency loss of 5% relative to the tuned super learner (relative efficiency 0.95; 95% CI = 0.88, 1.01).

Conclusions: In this case study, hyperparameter tuning produced a super learner that performed slightly better than an untuned super learner. Tuning the hyperparameters of individual algorithms in a super learner may help optimize performance.

Keywords: antidepressants; grid search; hyperparameters; prediction; super learning; treatment indications

(*Epidemiology* 2019;30: 521–531)

Predictive modeling has many important applications in public health, clinical practice, and epidemiologic research. Prediction algorithms can help identify target populations for health interventions, improve clinical decision making, and facilitate confounding control in observational studies.[1] The increasing amount of healthcare data being generated could help to improve the accuracy with which we can predict health outcomes,[2] but it is no trivial task sorting through the masses of data to separate the signal from the noise.

Standard epidemiologic approaches to prediction typically involve using parametric regression methods where the optimal set of covariates is identified through such procedures as performing stepwise variable selection, exploring different functional forms for continuous variables, and testing for interactions between main effects. Such model-building practices are necessary because the probability estimates from regression models may be biased if the model is incorrectly specified.[3] However, as the dimensionality of the dataset grows, researchers may find that these standard model-building procedures become cumbersome and difficult to implement properly.

Because of these challenges, there has been a growing interest to use more flexible prediction techniques from the machine learning literature that can automatically learn associations in high-dimensional data.[4,5] To allow researchers

Submitted August 1, 2018; accepted March 29, 2019.

[a]Department of Epidemiology, Biostatistics, and Occupational Health, McGill University, Montreal, Canada; and [b]School of Computer Science, McGill University, Montreal, Canada.

Supported by the Vanier Canada Graduate Scholarship, the Max E. Binz Fellowship (Faculty of Medicine, McGill University); a graduate student fellowship from the Research Institute of the McGill University Health Centre, and the Canadian Institutes of Health Research (IOP-112675). The funders had no role in the study design, collection, analysis, and interpretation of the data, writing of the manuscript, or in the decision to submit the manuscript for publication.

The authors report no conflicts of interest.

SDC Supplemental digital content is available through direct URL citations in the HTML and PDF versions of this article (www.epidem.com).

Process for obtaining data and computing code: Study data are not available as they contain confidential patient information. A sample of our R code is included in the eAppendix 3, but readers can request additional code from the corresponding author.

Correspondence: Jenna Wong, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute, 401 Park Drive, Suite 401 East, Boston, MA 02215. E-mail: jenna_wong@harvardpilgrim.org.

ISSN: 1044-3983/19/3004-0521
DOI: 10.1097/EDE.0000000000001027

to simultaneously consider multiple machine learning techniques rather than just one, many studies[6–15] have implemented "super learning"[16]–an ensemble machine learning approach that determines the optimal weights for combining the predictions from a collection of machine learning algorithms to yield a final super learner prediction function that performs at least as well as any of its component algorithms.

Despite the potential benefits of using the super learning methodology, few studies have conducted a head-to-head comparison of a super learner with a carefully constructed regression model. Furthermore, to our knowledge, no applications of super learning thus far have included formalized efforts to tune the hyperparameters of the machine learning algorithms in the super learner. Hyperparameters refer to parameters whose values are typically set by the user manually before an algorithm is trained and can impact the algorithm's behavior by affecting such properties as its structure or complexity.[17] Although the super learning methodology itself does not dictate what hyperparameter values investigators should use for their machine learning algorithms, most investigators appear to use the default values in the statistical packages used to implement super learning.[6–15] This observation is concerning given that the performance of an algorithm can be sensitive to the value of its hyperparameters.[17,18] In the machine learning literature, hyperparameters are commonly tuned using an iterative procedure called *grid search* whereby an algorithm's cross-validated performance is repeatedly assessed over a grid of possible hyperparameter values to identify the best one.[17,19] If this tuning process is not carried out (e.g., default values are used), then machine learning algorithms–and thus super learners–may not perform as well.

In this study, we applied super learning to a high-dimensional dataset from a previous study[20] that used multivariable logistic regression methods with classical model-building techniques to predict when antidepressants were prescribed for indications other than depression. This prediction task is important because the medical reasons for drug use ("treatment indications") are not routinely documented in structured electronic health data, thus creating challenges when using these data to study antidepressant use for depression and other (e.g., off-label) indications.[21,22] This study had two main objectives: (1) to compare the performance of a "tuned" super learner (fit using tuned hyperparameter values) to that of an "untuned" super learner (fit using default values) and (2) to compare the performance of the tuned super learner to that of the final logistic regression model derived in the previous study.[20]

METHODS

Data Source

The Medical Office of the XXIst Century (MOXXI) is an indication-based electronic prescribing and drug management system used by over 185 consenting primary care physicians at community-based clinics around two major urban centers in the Canadian province of Quebec.[23] The MOXXI system requires physicians to document at least one treatment indication for every prescription using either a drop-down menu containing on-label and off-label indications (without distinction) or by typing the indication(s) into a free-text field. Treatment indications in the MOXXI system were previously validated against a blinded, post hoc, physician-facilitated chart review where they had excellent sensitivity (98.5%) and high positive predictive value (97.0%).[24] Health services data on all MOXXI patients are available through the system's integration with Quebec's health insurance agency (Régie de l'assurance maladie du Québec) and hospital discharge summary database (MED-ECHO). These data sources provide information on patient demographics, diagnoses, hospitalizations, and medical services received.

This study included MOXXI prescriptions for all drugs approved for depression in Canada written between January 2003 and December 2012. We excluded drugs with fewer than 120 prescriptions written during the study period. The unit of analysis was the antidepressant prescription. All patients gave informed consent to have their information used for research purposes.

This study was approved by the McGill institutional review board.

Study Variables

The outcome being predicted was a binary variable indicating whether an antidepressant had been prescribed for an indication other than depression. The outcome was measured using the physician-documented treatment indications in the MOXXI system.

Table 1 lists all variables that were considered as potential predictors of the outcome. There were a total of 373 variables related to characteristics of the prescription, patient, or prescribing physician. Prescription-related variables (n = 4) included the molecule name, the prescribed dose, whether the drug was prescribed on a take-as-needed basis, and the number of other drugs concurrently prescribed with the antidepressant. Patient-related variables (n = 362) captured information on demographics, socioeconomic status, diagnostic codes for plausible antidepressant treatment indications and other morbidities, health services use (e.g., previous hospitalizations, outpatient visits, emergency room visits, medical services received), and drugs prescribed in the past year. Finally, physician-related variables (n = 7) included physician sex, place of medical training, level of clinical experience, size of patient workload, and scores from a survey[25] that measured physicians' attitudes towards new information about good clinical practices. Further details on the creation of these variables are included in eAppendix 1; http://links.lww.com/EDE/B513 and were described in the earlier article.[20]

Of the 373 variables, 13 were continuous, two were multicategorical, and the remaining 358 were binary. Each categorical variable was expressed using dummy coding,

Table 1. Candidate Predictors of Antidepressant Prescriptions for Indications Other than Depression (n = 373)

Variable[a]	Variable Type	Variable Levels
Prescription-related factors (n = 4)		
Molecule name	Categorical[b]	Amitriptyline, Bupropion, Citalopram, Clomipramine, Desipramine, Desvenlafaxine, Doxepin, Duloxetine, Escitalopram, Fluoxetine, Fluvoxamine, Imipramine, Mirtazapine, Nortriptyline, Paroxetine, Sertraline, Trazodone, Trimipramine, and Venlafaxine
Prescribed dose (mg/day)	Continuous	
Drug prescribed on a take-as-needed basis	Binary	Yes vs. No
No. other drugs concurrently prescribed	Continuous	
Patient-related factors (n = 362)		
Demographics and socioeconomic status		
Sex	Binary	Male vs. Female
Age (years)	Continuous	
Household income (CAD)	Continuous	
Less than university education (%)	Continuous	
Unemployment rate (%)	Continuous	
Type of drug insurance	Binary	Public vs. Private
Diagnostic codes in the past year		
13 plausible antidepressant treatment indications		
Around the index prescription date (±3 days)	13 Binary variables	Yes vs. No
Before the index prescription date (−4 to −365 days)	13 Binary variables	Yes vs. No
Chronic conditions in the Charlson comorbidity index	17 Binary variables	Yes vs. No
Other morbidities	86 Binary variables	Yes vs. No
Health services use in the past year		
No. outpatient visits	Continuous	
No. outpatient physicians seen	Continuous	
Continuity of care with the prescribing physician (%)	Continuous	
Previous hospitalization	Binary	Yes vs. No
Previous day surgery	Binary	Yes vs. No
Previous ER visit	Binary	Yes vs. No
Medical services	52 Binary variables	Yes vs. No
In-hospital procedures	70 Binary variables	Yes vs. No
Drugs prescribed in the past year	99 Binary variables	Yes vs. No
Physician-related factors (n = 7)		
Sex	Binary	Male vs. Female
Place of medical training	Binary	Canada/United States vs. Other
Experience (years in practice)	Categorical[b]	24+ years, 15–23 years, and <15 years
Workload (average no. patients per working day)	Continuous	
Factors affecting physician response to new information on good clinical practice		
Evidence score	Continuous	
Nonconformity score	Continuous	
Practicality score	Continuous	

[a]Further details on these variables can be found in eAppendix 1; http://links.lww.com/EDE/B513.
[b]Expressed using n − 1 binary variables, where n represents the number of variable levels.
CAD indicates Canadian dollars; ER, emergency room.

yielding a final covariate matrix with 391 columns. Because some of the algorithms in the super learner required prior scaling of the inputs, we standardized each continuous variable by subtracting the variable's mean and dividing by twice the variable's SD.[18]

Prediction Approaches

Classical Epidemiologic Techniques

This approach replicated our previous analysis of the same dataset[20] that used classical multivariable logistic regression methods to predict the same outcome. In the previous

analysis, we started with a baseline logistic regression model containing 26 of the 373 candidate variables, all of which were binary variables indicating whether the patient had a diagnostic code for any of 13 plausible antidepressant treatment indications recorded within two separate observation windows: (1) ±3 days around the index prescription date and (2) 4–365 days before the index prescription date. We then built upon this baseline model by considering the remaining 347 variables and applying a comprehensive suite of model-building techniques commonly used with regression methods in epidemiology. First, for all candidate continuous variables, we identified the best fitting first-degree fractional polynomial (FP1) function[26] among eight candidate FP1 functions: X^p, where the powers p were represented by the set {–2, –1, –0.5, 0, 0.5, 1, 2, 3}, and X^0 denoted log(X). Next, we used a score-based forward stepwise variable selection procedure to iteratively add covariates to the baseline model, starting with the variable that produced the greatest improvement in performance and stopping when none of the remaining variables further improved performance. Finally, we added first-order interaction terms to this main-terms-only model if they offered additional improvement. More details of this model-building procedure are available in the previous article.[20]

Super Learning

We used super learning to combine the prediction functions from five machine learning algorithms: (1) a least absolute shrinkage and selection operator (LASSO) model,[27] (2) a recursive partitioning and regression tree (hereafter referred to as simply "decision tree"),[28] (3) a random forest,[29] (4) a neural network,[30] and (5) a support vector machine.[31] We chose these five algorithms for their diverse approaches for solving prediction tasks and their popularity of use in other fields like genetics[32] and biomedicine.[33] Because of the computational time required to tune each of their respective hyperparameters, we did not consider more than five algorithms.

We implemented super learning using the SuperLearner package in the R programming language.[34] Table 3 shows the R packages we used to implement each algorithm in the super learner and the corresponding hyperparameters that we tuned. For the LASSO model, we tuned the regularization parameter lambda, where higher values imply more shrinkage of the regression coefficients. For the decision tree, we tuned the hyperparameter cp ("complexity parameter"), where higher values generally yield simpler, smaller trees. For the random forest, we tuned the number of trees in the forest (nTree) and the number of predictors randomly selected for consideration at each tree node (mTry). For the neural network, we fit a network with one hidden layer (the maximum allowed for by the nnet package) and tuned the number of nodes in this hidden layer (size). Finally, for the support vector machine, we tuned the regularization parameter C, where higher values allow for a more complex decision surface separating data points from different outcome classes. When fitting support vector machines, a common practice is to increase the dimensionality of the covariate space by applying a kernel to the predictor matrix.[18] Thus, we used one of the most commonly used kernels for support vector machines–the radial basis function kernel[35,36]–and tuned its gamma parameter, where higher values generally yield more complex decision boundaries.[35,36] The eAppendix 2; http://links.lww.com/EDE/B513 contains further details of these machine learning algorithms and their corresponding hyperparameters.

Primary Performance Metric

For all models, we used the scaled Brier score[37,38] as the primary performance metric to guide our modeling decisions during the training phase and to assess the performance of the final models during the testing phase. Similar to the R^2 statistic in linear regression,[9] we calculated the scaled Brier score using the following formula:

$$\text{Brier score}_{\text{scaled}} = 1 - \left(\frac{1}{\text{N}} \sum_{i=1}^{\text{N}} \left(\hat{Y}_i - Y_i \right)^2 \Big/ \frac{1}{\text{N}} \sum_{i=1}^{\text{N}} \left(\overline{Y} - Y_i \right)^2 \right)$$

where N represents the total number of antidepressant prescriptions in the validation set (during training) or the testing set (during testing), $\hat{Y}_i$ represents the predicted probability that prescription i was written for an indication other than depression, Y_i represents the observed outcome for prescription i (1 if the prescription was not written for depression, 0 otherwise), and $\overline{Y}$ represents the overall (marginal) observed probability of $Y = 1$ in the validation set (during training) or the testing set (during testing). Accordingly, the scaled Brier score can be interpreted as the relative reduction in the mean squared error yielded by a given algorithm relative to a noninformative (random) algorithm that assigns all prescriptions the marginal probability of having an indication other than depression.

Analytic Procedure

The Figure illustrates the flow of the study analysis. Only antidepressant prescriptions with complete data for all covariates were used in the main analysis (~95% of all eligible prescriptions). All prescriptions with complete data were randomly divided into a "training set" versus "testing set" using a 3:1 split. Because prescriptions were clustered within patients, who in turn were nested within physicians, we assigned a randomly selected 75% of physicians (rather than individual prescriptions) to the training set and the remaining 25% of physicians to the testing set. Thus, all prescriptions from the same physician and patient were limited to either the training or testing set. To ensure that patients and prescriptions were also divided approximately 3:1 between the training and testing sets, we first divided physicians into four strata by the number of their patients and then randomly sampled physicians separately within each stratum.

We used the training set (Figure Box A) to build, tune, and fit the final models. The testing set (Figure Box B) was used only to evaluate the performance of the final models–it

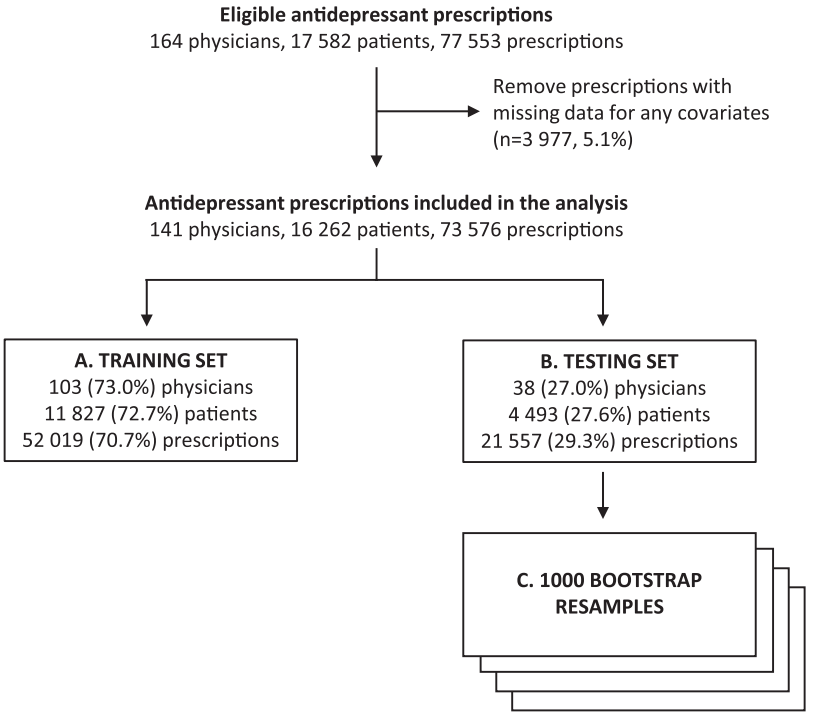

FIGURE. Flowchart of the study analysis. We assigned all antidepressant prescriptions in the analysis to either the training set (Box A) or testing set (Box B). Physicians and patients were mutually exclusive between the training and testing sets. We used the training set to build, tune, and fit the final logistic regression model and two super learners. We assessed the performance of these final models in the testing set, which had not been used during any part of the training process. To measure the statistical uncertainty around our performance estimates in the testing set, we bootstrapped the testing set using a two-stage cluster bootstrap[40] to account for multilevel clustering of prescriptions within patients, who in turn were clustered within physicians. For each performance estimate, the reported 95% CI corresponds to the values of the 2.5th and 97.5th percentiles of the distribution across 1000 bootstrap resamples of the testing set (Box C).

was not used in any part of the training process so that the final algorithms would be tested on completely independent data.

Cross-validation During the Training Phase

To reduce the risk of overfitting our final models to the training data, we split the training set into three mutually exclusive subsets using the same stratified randomization procedure as before. We used these three subsets to calculate a cross-validated estimate of the scaled Brier score whenever it was used to make a modeling decision. To compute the cross-validated scaled Brier score for a candidate algorithm, we fit the algorithm on two of the three training subsets (the "derivation set") and calculated the scaled Brier score in the held-out subset (the "validation set"). We repeated this process three times using a different subset as the validation set each time and then averaged the scaled Brier score across the three validation sets. Even with cross-validation, however, repeated use of the training data for model selection can lead to some overfitting of the validation sets.[39] Thus, it is for this reason that we tested the final models on a third independent dataset (the testing set).

Fitting the Multivariable Logistic Regression Model

In our previous analysis,[20] we implemented the classical model-building procedures for the multivariable logistic regression model (as described previously) on the training set using the cross-validated estimate of the scaled Brier score to guide all modeling decisions. The final logistic regression model included 40 main terms, which were comprised of three prescription-related variables (molecule name, prescribed dose, and whether the drug was prescribed on a take-as-needed basis), 36 patient-related variables (age^{-2}; less than university education; 26 indicator variables for whether diagnostic codes for 13 plausible antidepressant treatment indications were recorded within ±3 days and −4 to −365 days of the prescription date; three indicator variables for whether diagnostic codes were recorded within the past year for three conditions: diabetes without chronic complications, dementia, and unspecified nonpsychotic mental disorder following organic brain damage; number of outpatient visits in the past $year^{-0.5}$; whether the patient had a diagnostic procedure performed in the past year; and three indicator variables for whether three drugs were prescribed in the past year: trazodone, quetiapine, and furosemide), and one physician-related variable (average number of patients seen per working $day^{-0.5}$). The final logistic regression model also included one interaction term between the molecule name and the prescribed dose.

Fitting the Two Super Learners

Because the purported advantage of using more flexible machine learning algorithms is that they can automatically detect and model complex, nonlinear associations in the data, we submitted all 373 candidate predictors to each machine learning algorithm without applying any categorization or transformations to continuous variables (other than standardization).

To tune the algorithms' hyperparameters, we applied a grid search procedure that iteratively assessed the cross-validated performance of the algorithms in the training set over a range of plausible hyperparameter values (Table 2). For the LASSO model, rather than define our own subset of possible

Table 2. Machine Learning Algorithms in the Super Learner and Their Corresponding Hyperparameters

Algorithm	R Package	Hyperparameter	Description of Hyperparameter	Subset of Values Assessed in the Grid Search
LASSO	glmnet	lambda	Regularization parameter	Sequence of values automatically selected by glmnet
Decision tree	rpart	cp	Complexity parameter where splits that decrease the overall lack of fit by at least a factor of cp are retained	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001, 0.000005, 0.000001}
Random forest	randomForest	nTree	No. trees to grow	{10, 100, 1000, 1500, 2000}
		mTry	No. variables randomly sampled as candidates at each split	{10, 25, 50, 100, 150, 200}
Neural network	nnet	size	No. nodes in the hidden layer	{1, 2, 3, 4, 5}
Support vector machine	svm	C	Regularization term	{0.001, 0.01, 0.1, 1, 10}
		gamma	Parameter in the radial basis function kernel	{0.001, 0.01, 0.1, 1, 10}

lambda values, we used the sequence of values automatically generated by the glmnet package. For algorithms with multiple hyperparameters, we assessed all possible combinations of their candidate hyperparameter values. For example, for the random forest, we assessed a total of 30 unique combinations of nTree and mTry. For each algorithm, the tuned value for its respective hyperparameter (or combination of hyperparameters) was defined as the value(s) yielding the best cross-validated scaled Brier score in the training set (for R code showing how we implemented the grid search procedure for the random forest, see eAppendix 3; http://links.lww.com/EDE/B514).

After completing the grid search procedure, we used the SuperLearner package to fit two super learners on the training set. For the first super learner, we specified a library of learners that included each of the five machine learning algorithms fit using their tuned hyperparameter values identified from the grid search (the "tuned" super learner). For the second super learner, we specified a library of learners that included each algorithm fit using the default value in the SuperLearner package (the "untuned" super learner) (see eAppendix 4; http://links.lww.com/EDE/B513 for details on how these tasks were implemented). The SuperLearner package then determined the optimal weighted combination of algorithms for each super learner as follows.[14] First, it obtained the cross-validated predictions for each algorithm (i.e., the predictions in the validation set of each fold when the algorithm was fit on the derivation set). Next, it performed a constrained regression of the observed outcome on a matrix of the cross-validated predictions (one column per algorithm) to determine the optimal convex combination of regression coefficients (i.e., a vector of nonnegative coefficients summing to one), corresponding to the weights for combining the predictions from each algorithm in the super learner. Finally, the SuperLearner package refit each machine learning algorithm on the entire training set. The predictions from these fitted algorithms, combined with their corresponding weights, constituted the final super learner prediction function, developed in the training set.

Performance Assessment in the Testing Set

We assessed the performance of the final logistic regression model and the two super learners by applying these models to prescriptions in the independent testing set (Figure Box B). For each model, we used the scaled Brier score as our primary performance metric to assess its overall performance. As our secondary performance metric, we calculated the concordance (*c*) statistic to assess its discriminative ability.[38] We compared the performance of these models by measuring the relative efficiency (RE), which we defined as the performance of a given model relative to that of the tuned super learner. For example, the RE of the scaled Brier score for the logistic model was calculated as $RE_{logistic}$ = scaled Brier $score_{logistic}$/scaled Brier $score_{tunedSuperLearner}$, where RE > 1 indicated an efficiency gain (i.e., better performance) and RE < 1 indicated an efficiency loss (i.e., worse performance) compared to the tuned super learner.

To report the level of statistical uncertainty around our performance estimates in the testing set, we calculated 95% confidence intervals (CIs) using a two-stage cluster bootstrap[40] to account for clustering of prescriptions within patients, who in turn were clustered within physicians. The reported 95% CIs correspond to the values of the 2.5th and 97.5th percentiles of the distribution of the respective estimates across 1000 bootstrap resamples of the testing set (Figure Box C).

All analyses were performed in the R environment for statistical computing, version 3.4.1.[41] The following R packages were used: glmnet,[27] rpart,[42] randomForest,[43] nnet,[30] e1071,[44] SuperLearner,[34] and AUC.[45]

RESULTS

The analytical dataset included 73,576 antidepressant prescriptions that were written by 141 physicians for 16,262 patients (Figure). Of these, 52,019 (70.7%) antidepressant prescriptions, written by 103 physicians for 11,827 patients,

Table 3. Tuned and Default Hyperparameter Values for Each Machine Learning Algorithm

Algorithm	Hyperparameter	Tuned Value[a]	Default Value[b]
LASSO	Lambda[c]	0.001	0.001
Decision tree	cp	0.01	0.01
Random forest	nTree	1000	1000
	mTry	50	19[d]
Neural network	Size	1	2
Support vector machine	C	1	1
	Gamma	0.01	0.00256[e]

[a]Value for the hyperparameter that yielded the best cross-validated scaled Brier score for the corresponding algorithm in the grid search procedure.

[b]Default value for the hyperparameter in the algorithm's corresponding wrapper function in the SuperLearner package.

[c]The tuned value of lambda coincided with the default value because the wrapper function for LASSO regression in the SuperLearner package automatically used the value of lambda with the lowest cross-validated error.

[d]Calculated using the formula: floor(sqrt(ncol(x))), where ncol(x) = 391 in the study dataset.

[e]Calculated using the formula: 1/ncol(x), where ncol(x) = 391 in the study dataset.

Table 4. Weights for the Individual Machine Learning Algorithms in the Super Learner Functions

	Weight in the Super Learner Function	
Algorithm	Tuned Hyperparameters	Default Hyperparameters
LASSO	0.173	0.424
Decision tree	0.000	0.045
Random forest	0.526	0.424
Neural network	0.186	0.106
Support vector machine	0.114	0.000

were randomized to the training set. The remaining prescriptions were assigned to the testing set. Overall, 32,405 (44.0%) antidepressant prescriptions were written for indications other than depression, with this prevalence being similar between the training (43.0%) and testing sets (44.5%).

Grid Search Procedure

The grid search procedure revealed that for the random forest, neural network, and support vector machine, there was a better hyperparameter value (or combinations of hyperparameter values) than the default values in the SuperLearner package (Table 3). For instance, for the random forest, although the best value for nTree was the same as the default value of 1000 trees, the best value for mTry was 50 compared to the default value of 19. For the decision tree and LASSO model, the best values for their corresponding hyperparameters coincided with the default values.

Super Learner Coefficients

Table 4 shows the weights (or coefficients) for each machine learning algorithm in the two super learners. In the tuned super learner, the random forest contributed the most with a weight of 0.526, followed by the neural network and LASSO model with similar weights of 0.186 and 0.173, and finally the support vector machine with the lowest non-zero weight of 0.114. The decision tree did not contribute at all (weight of 0). In the untuned super learner, the LASSO model and random forest contributed the most with a weight of 0.424 each, followed by the neural network and decision tree with much a lower weight of 0.106 and 0.045, respectively. This time, the support vector machine did not contribute at all (weight of 0).

Performance of the Two Super Learners in the Testing Set

In the testing set, the tuned super learner had a scaled Brier score of 0.322 (95% CI = 0.267, 0.362), corresponding to a 32% reduction of the mean squared error relative to random classification (Table 5). The tuned super learner also had good discrimination, with a *c* statistic of 0.822 (95% CI = 0.795, 0.847). In comparison, the untuned super learner using default hyperparameter values had a scaled Brier score of 0.309 (95% CI = 0.256, 0.353), corresponding to an efficiency loss of 4% relative to the tuned super learner (RE of 0.96; 95% CI = 0.93, 0.99). The *c* statistic for the untuned super learner was also slightly lower at 0.817 (95% CI = 0.791, 0.846), but the efficiency loss in model discrimination relative to the tuned super learner was not statistically significant (RE of 0.99; 95% CI = 0.99, 1.00).

In terms of the individual performance of each machine learning algorithm in the super learners, the decision tree had by far the worst performance of any algorithm in both super learners, with a scaled Brier score of 0.226 (95% CI = 0.168, 0.276) and *c* statistic of 0.746 (95% CI = 0.717, 0.779) (Table 6). In the tuned super learner, the support vector machine had the best individual performance (highest scaled Brier score and *c* statistic), although the random forest and neural network had comparable performance (Table 6). For those algorithms where the tuned hyperparameter value differed from the default value, the performance of the tuned version was always better than that of the default version, especially for the neural network (Table 6).

Performance of the Tuned Super Learner Compared to the Final Logistic Model

The final logistic regression model had a scaled Brier score of 0.307 (95% CI = 0.245, 0.360) and *c* statistic of 0.815 (95% CI = 0.787, 0.847) (Table 5). These point estimates were slightly lower (worse) than those for the tuned super learner, but the efficiency loss was not statistically significant for both the scaled Brier score (RE of 0.95; 95% CI = 0.88, 1.01) and the *c* statistic (RE of 0.99, 95% 0.98 – 1.00).

DISCUSSION

In this case study, we used an ensemble machine learning approach called super learning to predict when primary care physicians prescribed antidepressants for indications

Table 5. Performance of Super Learning (Using Tuned and Default Hyperparameters) and Classical Epidemiologic Methods (Using Logistic Regression) for Predicting When Antidepressants are Prescribed for Indications Other Than Depression

Method	Scaled Brier Score (95% CI)	$RE_{\text{scaled Brier score}}$[a] (95% CI)	*c* Statistic (95% CI)	$RE_{c\ statistic}$[a] (95% CI)
Super learning				
Tuned hyperparameters	0.322 (0.267, 0.362)	1.00 (reference)	0.822 (0.795, 0.847)	1.00 (reference)
Default hyperparameters	0.309 (0.256, 0.353)	0.96 (0.93, 0.99)	0.817 (0.791, 0.846)	0.99 (0.99, 1.00)
Logistic regression using classical epidemiologic methods	0.307 (0.245, 0.360)	0.95 (0.88, 1.01)	0.815 (0.787, 0.847)	0.99 (0.98, 1.00)

[a]For each performance measure, the RE represents the ratio of the value for the corresponding method compared to the value for the super learner using tuned hyperparameter values.

Table 6. Performance of the Individual Machine Learning Algorithms in the Two Super Learners

Algorithm	Scaled Brier Score (95% CI)		*c* Statistic (95% CI)	
	Tuned hyperparameters	Default hyperparameters	Tuned hyperparameters	Default hyperparameters
LASSO	0.287 (0.225, 0.339)	Same as tuned	0.805 (0.777, 0.838)	Same as tuned
Decision tree	0.226 (0.168, 0.276)	Same as tuned	0.746 (0.717, 0.779)	Same as tuned
Random forest	0.301 (0.251, 0.341)	0.294 (0.284, 0.329)	0.813 (0.787, 0.840)	0.817 (0.791, 0.843)
Neural network	0.299 (0.239, 0.345)	0.239 (0.177, 0.289)	0.812 (0.786, 0.839)	0.787 (0.759, 0.817)
Support vector machine	0.310 (0.251, 0.356)	0.300 (0.246, 0.345)	0.817 (0.793, 0.843)	0.812 (0.787, 0.839)

other than depression. We applied an iterative grid search procedure to tune the hyperparameter values of the five machine learning algorithms in the super learner and found that, compared to using the default values, the super learner using tuned hyperparameter values had slightly better overall performance. When we compared the performance of the tuned super learner to that of a carefully constructed logistic regression model derived using classical epidemiologic techniques, we found no differences in performance.

A growing number of researchers are using super learning to predict clinical outcomes[7,8,12–14] and improve confounding control when estimating causal effects.[6,10,15] However, researchers may oftentimes not tune the hyperparameter values of the machine learning algorithms in their super learners. Indeed, in many studies, hyperparameters are not mentioned at all,[7–13,15] or if they are mentioned, their reported values often correspond to the default values in the statistical packages used to implement super learning.[6,14] Our findings from this case study suggest that if investigators tune the hyperparameter values of their machine learning algorithms, their super learners may achieve slightly better performance than if default values were used. These gains in performance–even if small–may be practically meaningful to avoid "losing" any benefits of undertaking the extra effort to use the super learning methodology instead of classical prediction methods.

There are several reasons why researchers may often not perform hyperparameter tuning when fitting a super learner. First, the SuperLearner package is a "black box" that allows investigators to easily run complex machine learning algorithms without requiring much knowledge about the algorithms themselves. However, to tune an algorithm's hyperparameters, one must first understand the algorithm's architecture, know the main hyperparameters that influence its performance, and identify a plausible range of hyperparameter values to test. Second, users may find it daunting to modify an algorithm's hyperparameter values within this "black box." We found that the *create.Learner* function in the SuperLearner package was very helpful for doing this task as long as the hyperparameter of interest was a modifiable parameter in the algorithm's original wrapper function in the SuperLearner package. When this requirement was not met (in our case, for the support vector machine), we had to create our own custom wrapper for the algorithm, which required extra programming and a deeper understanding of the SuperLearner code. Third, the process of manually searching over a grid of possible hyperparameter values to identify the one with the best cross-validated performance requires advanced programming skills and can be computationally expensive, especially for algorithms like neural networks and support vector machines that can have long training times. To address these barriers, we suggest taking a heuristic approach to building a super learner whereby investigators include a smaller, yet still diverse collection of algorithms and take extra care to ensure each algorithm's hyperparameters are carefully tuned.

As an alternative to hyperparameter tuning, some researchers[46] have suggested including multiple versions

of an algorithm in a super learner library–each using different hyperparameter values–and then letting the super learning methodology choose the best variant or combination of variants to use. Given that some algorithms have multiple hyperparameters that must be tuned simultaneously (yielding a multidimensional matrix of possible hyperparameter values rather than a vector) or hyperparameters with a wide range of possible values, it may not suffice to include only several variants of a given algorithm. However, alternatively including a large number of variants representing a more thorough subset of possible hyperparameter values for each algorithm would likely yield a super learner that is not only computationally prohibitive but also cumbersome to present and interpret. In contrast, performing hyperparameter tuning before implementing super learning–as done in this study–is advantageous because it yields a more parsimonious super learner and allows investigators to better allocate their often-limited computing power to include a greater variety of algorithms in their super learners rather than multiple variants of the same one. Furthermore, as a byproduct of assessing the individual performance of each algorithm (i.e., outside the super learner) during the tuning process, investigators may be able to better interpret the super learner weights. For example, in this study, the support vector machine received a very low weight of only 0.114 and 0.000 in the tuned and untuned super learner, respectively. Based on these weights alone, one might conclude that the support vector machine performed poorly. However, Table 6 shows that the support vector machine in fact had the highest (best) scaled Brier score among all algorithms in both super learners, suggesting that its low weight was instead likely due to its high correlation with the predictions from other algorithms in the super learner.

There were at least three advantages of using the scaled Brier score as the primary performance metric in our analysis. First, for the multivariable logistic regression model, the scaled Brier score directly assessed the predictive value of adding a candidate variable to the model during the forward stepwise variable selection procedure, unlike *P*-values (commonly used in epidemiology for variable selection) that can only assess a variable's statistical significance and are sensitive to large sample sizes, collinearity between variables, and multiple hypothesis testing.[47] Second, we could calculate the scaled Brier score for all the machine learning algorithms because we only needed to obtain the probability estimates from the algorithms and the observed outcomes (unlike other goodness-of-fit measures like the Akaike's Information Criterion that require additional information and cannot be calculated for nonparametric algorithms). Finally, by using the scaled Brier score to quantify model performance, our performance scores had a more meaningful interpretation compared to simply reporting the mean squared error (i.e., unscaled Brier score).

To our knowledge, this study is the first to derive a tuned super learner and compare its performance to that of a carefully constructed regression model. Acion et al[48] recently compared the performance of a super learner with logistic regression and found that their super learner outperformed three different configurations of logistic regression. In contrast, we did not find evidence to suggest performance gains of a tuned super learner over a well-specified logistic regression model. However, there are notable differences between our studies. First, none of their three logistic models simultaneously employed variable selection, tests for nonlinear associations, and tests for interactions, whereas our final logistic model was derived using all these model-building techniques. Second, all algorithms in their super learner used the default hyperparameter values, whereas all our algorithms used tuned hyperparameter values. Finally, our dataset contained many more predictors (373 versus their dataset of 28 predictors).

In the previous study,[20] adding more predictors to a "baseline" logistic regression model containing only variables based on diagnostic codes for plausible antidepressant treatment indications drastically improved its performance, increasing the scaled Brier score from 0.076 (95% CI = −0.007, 0.131) for the baseline model to 0.307 (95% CI = 0.245, 0.360) for the final model. In comparison, the gains in performance of the tuned super learner from this study over the final logistic model from the previous study were far smaller and imprecisely estimated. These observations highlight the notion that the quality of predictors often plays a far more important role in achieving good predictive performance than the type of predictive machinery used.

Our study has several considerations. First, although the grid search procedure we used is one of the most common approaches for performing hyperparameter tuning, the manual and iterative nature of this process makes it labor-intensive and requires advanced programming skills to implement.[17] Researchers may therefore want to consider using newer methods that are being developed to automatically and more efficiently select hyperparameter values.[17] Second, because the performance of a super learner depends upon the collection of algorithms in it, it is possible that our findings could have been different had we chosen a different set of algorithms. However, to decrease the likelihood of this possibility, we chose a set of algorithms that employed a diverse range of approaches to prediction and have been found to perform well in other applications. Finally, when interpreting the findings from this case study, readers should keep in mind the properties of our analytical dataset (e.g., number of training samples, number of variables, distribution of variable types), as the relative performance of different machine learning algorithms and the effect of hyperparameter tuning could differ in a dataset with different properties.[49]

In conclusion, based on this case study, we found that a super learner fit using tuned hyperparameter values performed slightly better than a super learner fit using default values. When we compared the performance of this tuned super learner to that of a multivariable logistic regression model derived using classical model-building techniques, the

difference in performance was small and imprecise. Should investigators choose to use super learning, they may want to consider first tuning the hyperparameters of their individual machine learning algorithms before applying the super learning methodology to achieve optimal predictive performance.

REFERENCES

1. Steyerberg EW. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. New York, NY: Springer-Verlag; 2009.
2. Kruse CS, Goswamy R, Raval Y, Marawi S. Challenges and opportunities of big data in health care: a systematic review. *JMIR Med Inform*. 2016;4:e38. Available at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5138448/.
3. Kruppa J, Liu Y, Biau G, et al. Probability estimation with machine learning methods for dichotomous and multicategory outcome: theory. *Biom J*. 2014;56:534–563.
4. Wiens J, Shenoy ES. Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology. *Clin Infect Dis*. 2018;66:149–153.
5. Kreatsoulas C, Subramanian SV. Machine learning in social epidemiology: learning from experience. *SSM Popul Health*. 2018;4:347–349. Available at: http://www.sciencedirect.com/science/article/pii/S2352827318300405.
6. Karim ME, Platt RW; BeAMS study group. Estimating inverse probability weights using super learner when weight-model specification is unknown in a marginal structural Cox model context. *Stat Med*. 2017;36:2032–2047.
7. Petersen ML, LeDell E, Schwab J, et al. Super learner analysis of electronic adherence data improves viral prediction and may provide strategies for selective HIV RNA monitoring. *J Acquir Immune Defic Syndr*. 2015;69:109–118.
8. Pirracchio R, Petersen ML, Carone M, Rigon MR, Chevret S, van der LAAN MJ. Mortality prediction in the ICU: can we do better? Results from the Super ICU Learner Algorithm (SICULA) project, a population-based study. *Lancet Respir Med*. 2015;3:42–52.
9. Rose S. A machine learning framework for plan payment risk adjustment. *Health Serv Res*. 2016;51:2358–2374.
10. Wyss R, Schneeweiss S, van der Laan M, Lendle SD, Ju C, Franklin JM. Using super learner prediction modeling to improve high-dimensional propensity score estimation. *Epidemiology*. 2018;29:96–106.
11. Park SK, Zhao Z, Mukherjee B. Construction of environmental risk score beyond standard linear models using machine learning methods: application to metal mixtures, oxidative stress and cardiovascular disease in NHANES. *Environ Health Glob Access Sci Source*. 2017;16:102.
12. Rosellini AJ, Dussaillant F, Zubizarreta JR, Kessler RC, Rose S. Predicting posttraumatic stress disorder following a natural disaster. *J Psychiatr Res*. 2018;96:15–22.
13. Hubbard A, Munoz ID, Decker A, et al; PROMMTT Study Group. Time-dependent prediction and evaluation of variable importance using super-learning in high-dimensional clinical data. *J Trauma Acute Care Surg*. 2013;75(1 suppl 1):S53–S60.
14. Rose S. Mortality risk score prediction in an elderly population using machine learning. *Am J Epidemiol*. 2013;177:443–452.
15. Neugebauer R, Fireman B, Roy JA, Raebel MA, Nichols GA, O'Connor PJ. Super learning to hedge against incorrect inference from arbitrary parametric assumptions in marginal structural modeling. *J Clin Epidemiol*. 2013;66(8 suppl):S99–S109.
16. van der Laan MJ, Polley EC, Hubbard AE. Super learner. *Stat Appl Genet Mol Biol*. 2007;6:Article25.
17. Luo G. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Netw Model Anal Health Inform Bioinforma*. 2016;5:18.
18. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York, NY: Springer-Verlag; 2009.
19. Weng SF, Reps J, Kai J, Garibaldi JM, Qureshi N. Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PLoS One*. 2017;12:e0174944.
20. Wong J, Abrahamowicz M, Buckeridge DL, Tamblyn R. Derivation and validation of a multivariable model to predict when primary care physicians prescribe antidepressants for indications other than depression. *Clin Epidemiol*. 2018;10:457–474.
21. Wong J, Motulsky A, Eguale T, Buckeridge DL, Abrahamowicz M, Tamblyn R. Treatment indications for antidepressants prescribed in primary care in Quebec, Canada, 2006–2015. *JAMA*. 2016;315:2230–2232.
22. Wong J, Motulsky A, Abrahamowicz M, Eguale T, Buckeridge DL, Tamblyn R. Off-label indications for antidepressants in primary care: descriptive study of prescriptions from an indication based electronic prescribing system. *BMJ*. 2017;356:j603.
23. Tamblyn R, Huang A, Kawasumi Y, et al. The development and evaluation of an integrated electronic prescribing and drug management system for primary care. *J Am Med Inform Assoc*. 2006;13:148–159.
24. Eguale T, Winslade N, Hanley JA, Buckeridge DL, Tamblyn R. Enhancing pharmacosurveillance with systematic collection of treatment indication in electronic prescribing: a validation study in Canada. *Drug Saf*. 2010;33:559–567.
25. Green LA, Gorenflo DW, Wyszewianski L; Michigan Consortium for Family Practice Research. Validating an instrument for selecting interventions to change physician practice patterns: a Michigan Consortium for Family Practice Research study. *J Fam Pract*. 2002;51:938–942.
26. Sauerbrei W, Meier-Hirmer C, Benner A, Royston P. Multivariable regression model building by using fractional polynomials: description of SAS, STATA and R programs. *Comput Stat Data Anal*. 2006;50:3464–3485.
27. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33:1–22.
28. Breiman L, Friedman J, Stone CJ, Olshen RA. *Classification and Regression Trees*. Boca Raton, FL: Chapman & Hall/CRC; 1984.
29. Breiman L. Random forests. *Mach Learn*. 2001;45:5–32.
30. Venables W, Ripley B. *Modern Applied Statistics with S*. 4th ed. New York, NY: Springer; 2002.
31. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol*. 2011;2:Article 27.
32. Dasgupta A, Sun YV, König IR, Bailey-Wilson JE, Malley JD. Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience. *Genet Epidemiol*. 2011;35(suppl 1):S5–S11.
33. Foster KR, Koprowski R, Skufca JD. Machine learning, medical diagnosis, and biomedical engineering research—commentary. *Biomed Eng Online*. 2014;13:94.
34. Polley E, LeDell E, Kennedy C, Lendle S, Laan Mvd. SuperLearner: Super Learner Prediction. R package version 2.0–21 [Internet]. 2016. Available at: https://CRAN.R-project.org/package=SuperLearner. Accessed 16 July 2017.
35. Zanaty EA. Support vector machines (SVMs) versus multilayer perception (MLP) in data classification. *Egypt Inform J*. 2012;13:177–183.
36. Christmann A, Steinwart I. *Support Vector Machines: Information Science and Statistics*. New York, NY: Springer Science + Business Media, LLC; 2008.
37. Brier GW. Verification of forecasts expressed in terms of probability. *Mon Weather Rev*. 1950;78:1–3.
38. Steyerberg EW, Vickers AJ, Cook NR, Gerds T, Gonen M, Obuchowski N, et al. Assessing the performance of prediction models: a framework for some traditional and novel measures. *Epidemiol Camb Mass*. 2010;21:128–38.
39. Bishop CM. *Neural Networks for Pattern Recognition*. New York, NY: Oxford University Press, Inc.; 1995.
40. Xiao Y, Abrahamowicz M. Bootstrap-based methods for estimating standard errors in Cox's regression analyses of clustered event times. *Stat Med*. 2010;29:915–923.
41. R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing [Internet]. Vienna, Austria; 2017. Available at: https://www.R-project.org/. Accessed 7 August 2017.
42. Therneau T, Atkinson B, Ripley B. rpart: Recursive Partitioning and Regresson Trees. R package version 4.1–11 [Internet]. 2017. Available at: https://CRAN.R-project.org/parkage=rpart. Accessed 9 August 2017.
43. Liaw A, Wiener M. Classification and regression by randomForest. *R News*. 2002; 2:18–22.

44. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6–8. 2017. Available at: https://www.rdocumentation.org/packages/e1071/versions/1.6-8. Accessed 9 August 2017.
45. Ballings M, Van den Poel D. AUC: Threshold independent performance measures for probabilistic classifiers. R package version 0.3.0. 2013. Available at: https://CRAN.R-project.org/package=AUC. Accessed 9 August 2017.
46. Polley EC, Rose S, van der Laan MJ. Super learning. In: *Targeted Learning: Causal Inference for Observational and Experimental Data: Springer Series in Statistics [Internet]*. New York, NY: Springer-Verlag; 2011:43–66. Available at: //www.springer.com/us/book/9781441997814. Accessed 9 November 2018.
47. Lu M, Ishwaran H. A prediction-based alternative to P values in regression models. *J Thorac Cardiovasc Surg*. 2018;155:1130–1136.e4.
48. Acion L, Kelmansky D, van der Laan M, Sahker E, Jones D, Arndt S. Use of a machine learning framework to predict substance use disorder treatment success. *PLoS One*. 2017;12:e0175383.
49. Khondoker M, Dobson R, Skirrow C, Simmons A, Stahl D. A comparison of machine learning methods for classification using simulation with multiple real data examples from mental health studies. *Stat Methods Med Res*. 2016;25:1804–1823.