eAppendix 1. Details of the candidate predictors of antidepressant prescriptions for indications other than depression

Variable	Description
Prescription-related factors	
Molecule name	Generic name of the antidepressant prescribed; categorical variable with 19 levels
Prescribed dose (mg/day)	Continuous variable
Drug prescribed on a take-as-needed basis	Binary variable (yes vs. no)
No. other drugs concurrently prescribed	Continuous variable
Patient-related factors	
Sex	Binary variable (female vs. male)
Age (years)	Continuous variable
Household income (CAD)	Area-level measure of the median household income in the patient's census tract area; continuous variable
Less than university education (%)	Area-level measure representing the percentage of adults in the patient's census tract area with less than university education; continuous variable
Unemployment rate (%)	Area-level measure representing the percentage of unemployed adults in the patient's census tract area; continuous variable
Type of drug insurance	Binary variable (public vs. private drug insurance)
Diagnostic codes in the past year	
Plausible antidepressant treatment indications	26 binary variables used to indicate whether diagnostic codes in physician billings data or hospital discharge abstracts were recorded for each of 13 plausible treatment indications for antidepressants* within 2 separate time windows: a) ±3 days around the index prescription date, and b) -4 to -365 days before the index prescription date *Depression, anxiety/stress disorders, sleeping disorders, pain, migraine, fibromyalgia, obsessive-compulsive disorder, vasomotor symptoms of menopause, nicotine dependence, attention deficit/hyperactivity disorder, sexual dysfunction, pre-menstrual dysphoric disorder, and eating disorders
Chronic conditions in the Charlson comorbidity index	17 binary variables used to indicate whether diagnostic codes were recorded for each of the chronic conditions in the Charlson comorbidity index* in the past year *Myocardial infarction, congestive heart failure, peripheral vascular disease, cerebrovascular disease, dementia, chronic pulmonary disease, rheumatic disease, peptic ulcer disease, mild liver disease, diabetes without chronic complication, diabetes with chronic complication, hemiplegia or

	paraplegia, renal disease, any malignancy, moderate or severe liver disease, metastatic solid tumor, and AIDS/HIV.
Other morbidities	86 binary variables used to represent each four-digit ICD-9 code that was recorded in physician billings data or hospital discharge abstracts for at least 1% of all antidepressant prescriptions in the past year (after excluding diagnostic codes for antidepressant treatment indications and Charlson conditions)
Health services use in the past year	
Number of outpatient visits	Continuous variable
Number of outpatient physicians seen	Continuous variable
Continuity of care with the prescribing physicians (%)	The percentage of all outpatient visits in the past year that were made to the prescribing physician; continuous variable
Previous hospitalization	Binary variable (yes vs. no)
Previous ER visit	Binary variable (yes vs. no)
Medical services	Based on billing codes recorded for the patient in physician billings data over the past year. Individual billing codes were grouped into broader 'billing code categories' using mapping tables obtained from the RAMQ. Binary variables were used to represent the presence of billing codes from any category that was recorded for at least 1% of antidepressant prescriptions in the past year (a total of 52 categories).
Drugs prescribed in the past year	Binary variables used to represent the presence of a prescription in the past year for any drug (by generic name) that had been prescribed in the past year for at least 1% of all antidepressant prescriptions (a total of 99 drugs).
Physician-related factors	
Sex	Binary variable (female vs. male)
Place of medical training	Binary variable (Canada/US vs. other)
Experience (years in practice)	Categorical variable with 3 levels: 24+ years, 15-23 years, and <15 years
Workload (average no. patients per working day)	Continuous variable
Factors affecting physician response to new information on evidence-based clinical practice	Measured using physician scores in three domains (evidence, nonconformity, and practicality) from a psychometric instrument[a] for determining how physicians would likely respond to new information about good clinical practice; 3 continuous variables

Abbreviations: ER = emergency room; RAMQ = Régie de l'assurance maladie du Québec

[a] 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 Nov;51(11):938–42.

eAppendix 2. Description of the five machine learning algorithms included in the super learner and their corresponding hyperparameters

LASSO

LASSO (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator) is a type of penalized regression model that simultaneously (i) shrinks the coefficients of a conventional regression model, and (ii) performs variable selection by shrinking some of the coefficients right to zero (1). Coefficients with higher variance are shrunk more. The amount of shrinkage applied to the coefficients increases as the value of regularization parameter lambda increases, and a lambda of zero yields the conventional (unpenalized) logistic regression model.

Decision tree

Decision trees are non-parametric learning algorithms that apply a set of rules to partition the multi-dimensional space of covariates into hypercubes within which the outcome is more homogeneous (2). Decision trees are well-suited to handle high-dimensional and sparse data, but they can produce trees that are unstable (ie, slight changes in the data can produce notably different trees) and prone to overfitting as the depth of the tree increases (2). To reduce overfitting, a stopping rule is often applied (1). In the rpart package (3), this stopping rule is controlled by the hyperparameter cp (which stands for "complexity parameter") that retains only those splits that improve the overall performance of the tree by a factor of cp. Thus, larger values of cp imply smaller, simpler trees with fewer nodes.

Random forest

Random forests are ensemble learners that extend the decision tree framework in an attempt to address the issues of overfitting and high variability (4). Rather than a single tree, random forests contain many trees (typically hundreds) – each grown on a separate bootstrap re-sample of the training data where at each split, the tree chooses among a randomly selected subset of candidate predictors to help de-correlate the trees in the forest (1). The key tuning parameters for random forests are the number of trees grown (ntree) and the number of predictors randomly selected for consideration at each node (mtry) (2).

Neural network

Neural networks are non-linear statistical models that attempt to emulate the complex structure of the human brain (5). Neural networks consist of an input layer (ie, the variables offered to the network), one or more "hidden" layers, and an output layer that yields the final predicted probabilities from the network. Each layer in the network can contain any number of units or "nodes" that are connected to nodes in the subsequent layer by "connection weights" that act similarly to the beta coefficients in a regression model (6). Each node takes a weighted linear combination of its inputs (ie, the sum of its inputs multiplied by their connection weights) and passes this result through an "activation function" (usually the logistic or sigmoid function), which then becomes input to the node(s) in the next layer to which it is connected via another connection weight. These hidden nodes and their connection weights are what allow neural networks to automatically model more complex, non-linear relationships compared to traditional regression models (6). Fitting a network with two or more hidden layers is often referred to as “deep learning”. In this study, we could only fit a neural network with one hidden layer because the nnet package only allowed for one hidden layer.

Supper vector machine

Support vector machines (SVMs) are algorithms that classify observations by finding the optimal separating hyperplane between training observations from different outcomes classes in the multi-dimensional covariate space. The optimal hyperplane is defined as the hyperplane that separates observations from different outcome classes with the maximum margin (ie, the largest Euclidean distance between the separating hyperplane and the nearest data points from different outcomes classes on either side of it, called the “support vectors”) (1). In practice, it may be challenging to find a hyperplane that perfectly separates observations from different classes. Thus, SVMs allow for a “soft margin” whereby a fraction of the data points can be misclassified (ie, on the wrong side of the hyperplane). The regularization parameter C determines the penalty for misclassifying observations and thus controls the trade-off between minimizing the number of misclassified examples versus maximizing the margin (7). A smaller C corresponds to a smaller penalty for misclassifying data points and thus usually favors a larger margin (ie, smoother decision surface). SVMs also use kernel functions to increase the dimensionality of the input space, which often allows a hyperplane to better separate data points from different outcome classes and generally translates into more complex decision boundaries in the original covariate space (1). In the study, we used a radial basis function (RBF) kernel – one of the most commonly used kernels for SVMs (8,9), and optimized the gamma parameter of the RBF kernel. The gamma parameter controls the influence of a single observation on determining the decision boundary, with higher values generally resulting in more complex decision boundaries.

References

1. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. New York: Springer-Verlag; 2009.

2. 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):S5-11.

3. Therneau T, Atkinson B, Ripley B. rpart: Recursive Partitioning and Regresson Trees. R package version 4.1-11. [Internet]. 2017. Available from: https://CRAN.R-project.org/parkage=rpart

4. Rose S. A Machine Learning Framework for Plan Payment Risk Adjustment. Health Serv Res. 2016 Dec;51(6):2358–74.

5. Hinton GE. How neural networks learn from experience. Sci Am. 1992 Sep;267(3):144–51.

6. Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. J Clin Epidemiol. 1996 Nov;49(11):1225–31.

7. Yu W, Liu T, Valdez R, Gwinn M, Khoury MJ. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. BMC Med Inform Decis Mak. 2010 Mar 22;10:16.

8. Zanaty EA. Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification. Egypt Inform J. 2012 Nov 1;13(3):177–83.

9. Christmann A, Steinwart I. Support Vector Machines. New York, NY: Springer Science+Business Media, LLC; 2008. (Information Science and Statistics).

eAppendix 4. Details of how the machine learning algorithms were fit using their tuned and default hyperparameter values in the SuperLearner package

To fit the machine learning algorithms using their default hyperparameter values, we used their original wrapper functions in the SuperLearner package (eg, SL.randomForest for the random forest). To fit the machine learning algorithms using their tuned hyperparameter values, we used the *create.Learner* function in the SuperLearner package. This user-friendly function generates a custom wrapper for any given algorithm, which fits the algorithm using the user-specified values for the corresponding hyperparameters. However, the *create.Learner* function can only customize the value of hyperparameters that are included as modifiable parameters in the corresponding algorithm's original wrapper function in the SuperLearner package. For the support vector machine, because the gamma parameter was not a modifiable parameter in SL.svm, we could not use the *create.Learner* function to change the value of gamma. Thus, we had to create our own custom wrapper by copying the code of SL.svm, modifying the value of the gamma directly in the wrapper code, and then creating a new name for the wrapper function (eg, SL.myTunedSVM).

For the LASSO model and the decision tree, we did not have to customize their wrapper functions because the tuned hyperparameter values coincided with the default values. This result occurred by chance for the decision tree, but for the LASSO model, the glmnet package automatically identified the lambda value with the lowest cross-validated error in the training data and used this lambda value to predict on new data.

In the eAppendix 3, we have included example R code showing how we implemented the grid search procedure for the random forest to tune its hyperparameters and then fit a super learner using these tuned values.