Technical Reference for Applying Machine Learning in Predicting Medication Treatment Outcomesfor Opioid Use Disorder

Raymond R. Balise, PhD with Kyle Grealis, MS and Gabriel Odom, PhD

2025-08-27

This is a highly abridged technical reference for the modeling done in "Applying Machine Learning in Predicting Medication Treatment Outcomes for Opioid Use Disorder" which is under review. For more details please see: https://CTN-0094.github.io/ml_paper_2025.

Inclusion Criteria

Of all the individuals who were randomized in the three trials (N = 2,492), 99.4%, or all except 14 people, had *any* drug use information either self-reported use or via urine drug screen (UDS). Thus, the analysis cohort consisted of the 2,478 people where there was any information on their drug use during treatment.

Variables/Features

Table 1: Features used to predict treatment failure

Feature	Details
Age	Numeric
Ethnicity (is Hispanic)	Yes, No, Unknown
Race	Black, White, Other
Unemployed	Yes, No, Unknown
Stable Housing	Yes, No, Unknown
Education	Missing, Less than HS, HS or GED, More than HS
Marital Status	Unknown, Never married, Married or Partnered, Separated/Divorced/Widowed

Sex (is Male)	Yes, No, Unknown
Smoking History	Yes, No, Unknown
Fagerstrom Test for Nicotine Dependence	Numeric
IV Drug use History	Yes, No, Unknown
Pain Closest to Enrollment	None, Very mild to moderate, Severe
Schizophrenia	Yes, No, Unknown
Depression	Yes, No, Unknown
Anxiety	Yes, No, Unknown
Bipolar	Yes, No, Unknown
Neurological Damage	Yes, No, Unknown
Epilepsy	Yes, No, Unknown
Alcohol	Yes, No, Unknown
Amphetamines	Yes, No, Unknown
Cannabis	Yes, No, Unknown
Cocaine	Yes, No, Unknown
Study Site	Clinic Number
Clinic Type	Inpatient, Outpatient
Medication	Inpatient BUP, Inpatient NR-NTX, Methadone, Outpatient BUP, Outpatient BUP + Enhanced Medical Management, Outpatient BUP + Standard Medical Management
Number of Distinct Substances	Numeric
Number of Days with Any Use	Numeric

Training Testing Split and Validation

The analysis data was initially split using a stratification algorithm that assured the same percentage of people experienced treatment success in the training dataset (3/4 of the data) and the testing dataset (1/4 the data). For model tuning, five-fold cross validation was used.

Final Recipe

The preprocessing recipe followed the steps listed below. Algorithm details are covered in the documentation for the R `recipes` package (Version 1.3.1).

1. For all predictors, remove any variables with zero or near zero variance. See `step_nzv()` documentation.
2. For all nominal predictors, any string variables are converted to categorical factors. See the `step_string2factor()` documentation.

3. For all predictors, all missing values are imputed using a k = 5 nearest neighbors algorithm. See the `step_impute_knn()` documentation.
4. For all nominal predictors, dummy code the variables. See the `step_dummy()` documentation.
5. For all nominal predictors, pool infrequently occurring values (less than 5% of the data) into another category. See the `step_other()` documentation.
6. For all numeric predictors, recursively remove variables that have absolute correlations > 0.9 (beginning with the highest correlation). See the `step_corr()` documentation.
7. For all numeric predictors, normalize values to have a mean of zero and a standard deviation of one. See the `step_normalize()` documentation.

The original 46 variables are converted to features. The details for this conversion, if the recipe is applied to the full training data, are shown in the table below. Please note that the results may differ subtly across the five-fold resamples because each fold's recipe is fitted independently on that fold's analysis set. This means preprocessing parameters (like normalization estimates) and feature selection decisions (from steps like `step_corr()`, `step_nzv()`, or `step_other()`) may vary across folds, as they depend on the specific data characteristics within each fold.

Table 2: Feature conversion process during recipe step application

Step	N	Variables_2
Original variables	46	trial, medication, in_out, used_iv, age, race, is_hispanic, job, is_living_stable, education, marital, is_male, is_smoker, per_day, ftnd, pain, any_schiz, any_dep, any_anx, has_bipolar, has_brain_damage, has_epilepsy, has_alcol_dx, has_amphetamines_dx, has_cannabis_dx, has_cocaine_dx, has_sedatives_dx, is_homeless, did_use_cocaine, did_use_heroin, did_use_speedball, did_use_opioid, did_use_speed, days_cocaine, days_heroin, days_speedball, days_opioid, days_speed, days_iv_use, shared, tlfb_days_of_use_n, tlfb_what_used_n, withdrawal, detox_days, site_masked, did_relapse

Step 1: step_nzv()	43	Variables REMOVED: is_living_stable, has_epilepsy, days_speedball
Step 2: step_string2factor()	43	NO CHANGES
Step 3: step_impute_knn()	43	NO CHANGES

Step 4: step_dummy()	104	age, days_cocaine, days_heroin, days_opioid, days_speed, days_iv_use, tlfb_days_of_use_n, tlfb_what_used_n, detox_days, did_relapse, trial_CTN.0030, trial_CTN.0051, medication_Methadone, medication_Naltrexone, in_out_Outpatient, used_iv_Yes, race_Other, race_Refused.missing, race_White, is_hispanic_Yes, job_Other, job_Part.Time, job_Student, job_Unemployed, education_Less.than.HS, education_More.than.HS, marital_Never.married, marital_Separated.Divorced.Widowed, is_male_Yes, is_smoker_Yes, per_day_1, per_day_2, per_day_3, per_day_4, ftnd_X1, ftnd_X2, ftnd_X3, ftnd_X4, ftnd_X5, ftnd_X6, ftnd_X7, ftnd_X8, ftnd_X9, ftnd_X10, pain_No.Pain, pain_Severe.Pain, pain_Very.mild.to.Moderate.Pain, any_schiz_Unknown, any_schiz_Yes, any_dep_Unknown, any_dep_Yes, any_anx_Unknown, any_anx_Yes, has_bipolar_Yes, has_brain_damage_Yes, has_alcol_dx_Yes, has_amphetamines_dx_Yes, has_cannabis_dx_Yes, has_cocaine_dx_Yes, has_sedatives_dx_Yes, is_homeless_Yes, did_use_cocaine_Yes, did_use_heroin_Yes, did_use_speedball_Yes, did_use_opioid_Yes, did_use_speed_Yes, shared_Yes, withdrawal_X1, withdrawal_X2, withdrawal_X3, site_masked_X270002, site_masked_X270003, site_masked_X270004, site_masked_X270005, site_masked_X270006, site_masked_X270007, site_masked_X270008, site_masked_X270009,

Step 5: step_other()	104	NO CHANGES
Step 6: step_corr()	103	Variables REMOVED: trial_CTN.0051
Step 7: step_normalize()	103	NO CHANGES

Models

For models that tune many hyperparameters, values were selected using a space-filling parameter grid instantiated using the `dials::grid_latin_hypercube()` function.

Logistic Regression

A standard logistic model was fit using the default `glm.fit` method in `stats::glm()`.

Logistic Regression Via Lasso

A logistic model, allowing for the same resampling estimates for all other models, was fit using `glmnet` engine configured to run a lasso model (`mixture = 1`) but with a minuscule 10^{-10} penalty.

LASSO

A LASSO model was fit using the `glmnet` engine (`mixture = 1`). Preliminary tuning was run across 30 samples between 10^{-10} to 1. After examining the ROC estimates, the model was revised to use 30 equally spaced values across a penalty range of 10^{-3} to 1.

KNN

A KNN model was fit with using the `kknn` engine. The model was trained with a space-filling parameter grid with 50 combinations across 1 to 50 neighbors, nine weight functions (i.e., 'rectangular', 'triangular', 'epanechnikov', 'biweight', 'triweight', 'cos', 'inv', 'gaussian', and 'rank') and Minkowski Distance Order (range: [1, 2]).

MARS

A MARS model was fit with `earth` engine tuned across five levels of the degree of interaction from one to five using a backwards pruning method.

CART

A CART model was fit with the `rpart` engine. The model was trained with a space-filling parameter grid with 50 combinations across tree depth (range: [1, 15]), minimal node size (range: [2, 40]), and cost complexity (range: [10^{-10}, 10^{-1})])

Random Forest

A Random Forest model was fit with the `randomForest` engine. The model was trained with a space-filling parameter grid with 50 combinations across minimal node size (range: [2, 40]) and number of randomly selected predictors (range: 1 to an estimated finalized during training).

XGBoost

A boosted tree model was fit using the XGBoost algorithm with the `xgboost` engine. The model was trained with a space-filling parameter grid with 50 combinations across tree depth (range: [1 to 15]), minimal node size (range: [2 to 40]), minimal loss reduction (range: [10^{-10}, $10^{1.5}$]), sample size (range: [0.1, 1]), randomly selected predictors (range: 1 to an estimated finalized during training), and learn rate (range: [10^{-10}, 10^{-1}])

BART

A Bayesian additive regression tree model was fit using the `dbarts` engine. The model was trained with a space-filling parameter grid with 50 combinations across trees (range: [1, 2000]), terminal node prior coefficient (range: (0, 1])), terminal node prior exponent (range: (1, 3]), and the prior for outcome range (range: (0, 5]).

Support Vector Machine

A support vector machine model was fit using the `kernlab` engine. The model was trained using parameters from a regular grid with 10 values across cost (range: [10^{-10}, 10^{5}]) and polynomial degree (range: [1, 3]).

Neural Network

A single layer neural network was fit using the `brulee` engine. The model was trained with a space-filling parameter grid with 30 combinations across hidden units (range: [10, 100]), amount of regularization (range: [10^{-5}, 1]), and learning rate (range: [10^{-10}, 10^{-1}]) with 100 epochs.