DATA423 Assignment3

Jiangwei Wang (19364744)

Data description

This assignment 3 data consists of 1280 observations and 21 variables. The target variable is “Y”.

There are 1 date variable, 1 nominal variable and all the rest 19 variables, including the target variable, are all numeric variable.

There are missing values in all of the “Reagent” prefixed numeric variables, and they visually missing random. There are no excessively missing variables or observations.

From “Dfsummary”, numeric variables are seemed normal distributed, apart from “y”.

The numeric predict variables with “Reagent” prefix and the target variable all have uni-variable outliers, but not significant for the predict variables, the outliers disappeared when the IQR multipliers reaches 2.3. The target variable’s outliers are showing a little bit more significant as they won’t disappear until the IQR multiplier reaches to 3.

The nominal variable blood type has a low cardinality of 4.

From the boxplot we can determine the numeric data has 3 levels variety of scales. The first 4 numeric variables variate between 0 and 12, ‘Alcohol’ has the least variety to 4. The “Reagent” prefixed numeric variables variate from just over 100 to under 1100. The target variable variates the most, between over 2000 and below -4000.

Blocks of numeric predict variables are highly correlated to each other. These variables all have a similar name prefix as “Reagent”. There are 4 blocks: L, N, F, J, D and H; E and M; B and K; A, G, C and I.

The date formatted character variable format is YYYY-mm-dd. It covers from 2008-10-18 to 2019-06-28 almost 11 years’ time period.

The pair plots show some unlinear relationships between some of the predict variables and the target variable, especially ‘ReagentH’, ReagentJ’, ReagentK’, ReagentL’, ReagentM’, ReagentN’.

Strategies

Missing data

There are no excessively missing variables or observations to eliminate. There are only “Reagent” prefixed variables have missing values, most of them are missing around 5%, only “ReagentE” is missing just over 6%. These are shown in the “DfSummary”.

Because we have a raw obs/parametres ratio of 60, we can do a partial deletion if needed.

For the methods, we could use either naomit or knn with neighbour equals to 5 imputation as a default process approach. However, mode imputation has the same result as naomit after we employed, the mode imputation will reserve all the information from the data rather than delete, I would prefer to use mode imputation. Knn imputation has better result than mode imputation, once this approach selected, bag imputation can improve the result more or less in most models.

Since there are no missing target values, we don't need to eliminate these observations.

Outliers

The uni-variable outliers from predict variables are not significant, it is not necessary to delete any observations. However, we can consider using robust method to make sure catch any information outliers can offer.

Processing

Center and scaling are necessary due to the variety from the numeric predict variables are different, especially the difference between the first 4 variables and the rest. It is much the same to implement them before or after the knn imputation. Once this is selected, the best model will improve slightly either before or after in some of the models, apart from tree models or some other models.

PLS processing approach, which creates one or more new dimensions on numeric variables, work well on Tree-Based methods and some general linear methods.

To nomalize data by using YeoJohnson processing approach works well on Tree-Based method as well.

The date variable was converted to decimal, day-of-week, month, year. It has a significant impact on some models' accuracy improvement.

The correlation between some of the numeric variables can be broken by implementing PLA, ICA processing approaches. However, ICA is useful on fewer models, especially the two best models: gaussprPoly and cubist, ICA add brilliance to their present splendor.

NZV processing is useful on highly sparse and unbalanced data, which is not quite useful here.

Other processing isn't quite useful here as well, which will classify infrequently occurring data into an 'other' category.

Every single method is used dummy encoding for the treatment on low cardinality nominal variable blood type.

Mode imputation can’t be used if PCA is used at the mean time.

A static test set of 20% will be set aside to evaluate the best model using stratified sampling based upon the target.

The hyper-parameters will be tuned by using 10 fold cross-validation resampling by default. The distribution
of metrics will be plotted below.

Methods

The following method were tried:

Method	Characteristics	Notes	Reason chosen
glmnet – glmnet	Generalized Linear Model Implicit Feature Selection L1 Regularization L2 Regularization Linear Classifier Linear Regression	Failed when missing still present. Failed when nominal still present. 2 hyperparameters	
pls – Partial Least Squares	Partial Least Squares Feature Extraction Linear Classifier Linear Regression	1 hyperparameter	
rpart – CART	Tree-Based Model Implicit Feature Selection Handle Missing Predictor Data Accepts Case Weights	1 hyperparameter	
randomGLM – Ensembles of Generalized Linear Models	Generalized Linear Model Linear Classifier Ensemble Model Bagging	1 hyperparameter Crashed missing parameter ‘length’	Search for another glm method again after glmnet performing well
bayesglm – Bayesian Generalized Linear Model	Generalized Linear Model Logistic Regression	no hyperparameter	Search for another glm method after glmnet performing well

	Linear Classifier Bayesian Model Accepts Case Weights		
glmStepAIC – Generalized Linear Model with Stepwise Feature Selection	Generalized Linear Model Feature Selection Wrapper Linear Classifier Implicit Feature Selection Two Class Only Accepts Case Weights	Failed when missing still present. No hyperparameter	Search for another glm method again after glmnet performing well
plsRglm – Partial Least Squares Generalized Linear Models	Generalized Linear Models Partial Least Squares Two Class Only	Failed when nominal still present. 2 hyperparameters Slow to train	Search for another glm method again after glmnet performing well
ANFIS – Adaptive-Network-Based Fuzzy Inference System	Rule-Based Model	Failed when missing still present. Failed when nominal still present. 2 hyperparameters Aborted More than 2 hours to train	Random example of neural network based method
qrnn – Quantile Regression Neural Network	Neural Network L2 Regularization Quantile Regression Bagging Ensemble Model Robust Model	Failed when missing still present. Failed when nominal still present. 3 hyperparameters Aborted took too long to train	Search for neural network based method due to previous failed method
brnn – Bayesian Regularized Neural Networks	Bayesian Model Neural Network Regularization	Failed when missing still present. Failed when nominal still present. 1 hyperparameter Aborted took too long to train	Search for neural network based method due to previous failed method
avNNet – Model Averaged Neural Network	Neural Network Ensemble Model Bagging	Failed when missing still present. 3 hyperparameters	Search for neural network based method

	L2 Regularization Accept case weights		due to previous failed method
knn – k-Nearest Neighbours	Prototype Model	Failed when missing still present. Failed when nominal still present. 1 hyperparameter Slow to train	Random example of simple kernel based method we most familiar with
rf – Random Forest	Random Forest Ensemble Model Bagging Implicit Feature Selection	Failed when missing still present. 1 hyperparameter Slow to train	Random example of most common tree based method
kernelpls – Partial Least Squares	Partial Least Squares Feature Extraction Kernel Method Linear Classifier Linear Regression	1 hyperparameter Fast to train	Random example of ordinary least squares based method
rlm – Robust Linear Model	Linear Regression Robust Model Accepts Case Weights	2 hyperparameters Fast to train	Random example of robust method
rqlasso – Quantile Regression with LASSO penalty	Linear Regression Quantile Regression Implicit Feature Selection L1 Regularization	Failed when nominal still present. 1 hyperparameter	Random example of Quantile Regression
cubist – Cubist	Rule-Based Model Boosting Ensemble Model Prototype Models Model Tree Linear Regression Implicit Feature Selection	2 hyperparameters	Random example of Rule-Based method
gaussprPoly – Gaussian Process with Polynomial Kernel	Gaussian Process with Polynomial Kernel	Failed when missing still present. Failed when nominal still present. 2 hyperparameters	Random example of Gaussian Process method, choose the one with ‘poly’ in terms of unlinear relationship discovered before

gaussprRadial – Gaussian Process with Radial Basis Function Kernel	Kernel Method Gaussian Process Radial Basis Function	Failed when nominal still present. 1 hyperparameter Slow to train	Search for Gaussian Process method due to previous gaussprPoly method's outstanding performance
gaussprLinear – Gaussian Process	Kernel Method Gaussian Process Linear Classifier	Failed when missing still present. Failed when nominal still present. No hyperparameter	Search for Gaussian Process method due to previous gaussprPoly method's outstanding performance
svmPoly – Support Vector Machines with Polynomial Kernel	Kernel Method Support Vector Machines Polynomial Model Robust Methods	Failed when nominal still present. 3 hyperparameters	Searching for polynomial method due to previous gaussprPoly method's outstanding performance
krlsPoly – Polynomial Kernel Regularized Least Squares	Kernel Method L2 Regularization Polynomial Model	Failed when missing still present. Failed when nominal still present. 2 hyperparameters	Searching for polynomial method due to previous gaussprPoly method's outstanding performance
rvmPoly - Relevance Vector Machines with Polynomial Kernel	Kernel Method Relevance Vector Machines Polynomial Model Robust Methods	Failed when missing still present. Failed when nominal still present. Failed when heteroscedastic still present. 2 hyperparameters Slow to train	Searching for polynomial method due to previous gaussprPoly method's outstanding performance
rfRules - Random Forest Rule-Based Model	Random Forest Ensemble Model Bagging Implicit Feature Selection Rule-based Model	2 hyperparameters Aborted took too long to train	Search for tree based method to find out if it can do any better than rf and it is a Rule-Based method, in terms of cubist method's outstanding performance
qrf - Quantile Random Forest	Random Forest Ensemble Model	Failed when missing still present.	Search for tree based method again to find

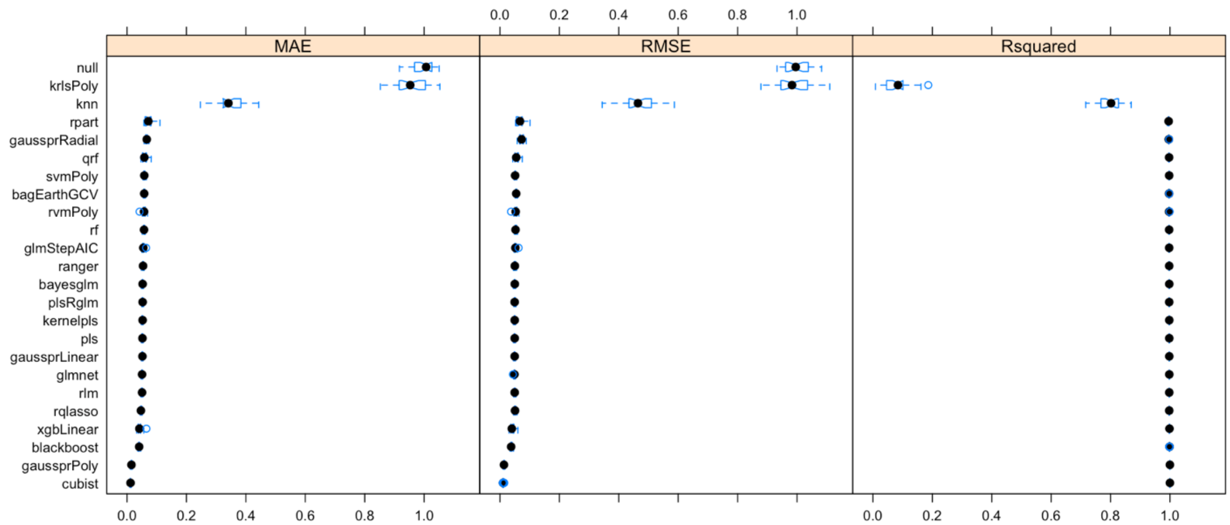
	Bagging Implicit Feature Selection Quantile Regression Robust Model	1 hyperparameter Slow to train	out if it can do any better than rf, and it is a robust method
ranger - random forest	Random Forest Ensemble Model Bagging Implicit Feature Selection Accepts Case Weights	3 hyperparameters Slow to train	Search for tree based method again to find out if it can do any better than ‘rf’, and it is an ensemble method in terms of ‘cubist’ method’s outstanding performance
M5 - Model Tree	Rule-Based Model Tree-Based Model Linear Regression Implicit Feature Selection Model Tree	3 hyperparameters Crashed Rweka library can’t find ‘JVM’	Searching for Rule-Based method due to previous cubist method’s outstanding performance
M5Rules - Model Tree	Rule-Based Model Linear Regression Implicit Feature Selection Model Tree	2 hyperparameters Crashed Rweka library can’t find ‘JVM’	Searching for Rule-Based method due to previous cubist method’s outstanding performance
HYFIS - Hybrid Neural Fuzzy Inference System	Rule-Based Model	2 hyperparameters Aborted took too long to train	Searching for Rule-Based method due to previous cubist method’s outstanding performance
xgbLinear - eXtreme Gradient Boosting	Linear Classifier Models Linear Regression Models L1 Regularization Models L2 Regularization Models Boosting Ensemble Model Implicit Feature Selection	Failed when nominal still present. 4 hyperparameters Slow to train	Searching for ensemble method with boosting due to previous cubist method’s outstanding performance

blackboost - Boosted Tree	Tree-Based Model Ensemble Model Boosting Accepts Case Weights	2 hyperparameters Slow to train	Searching for ensemble method and boosting with Tree-Based method rather than linear method due to previous cubist method’s outstanding performance
gbm_h2o - Gradient Boosting Machines	Tree-Based Model Boosting Ensemble Model Implicit Feature Selection	Failed when missing still present. 5 hyperparameters Crashed no active connection to an H2o cluster	Searching for ensemble method and boosting with Tree-Based method rather than linear method due to previous cubist method’s outstanding performance
bstSm - Boosted Smoothing Spline	Ensemble Model Boosting Implicit Feature Selection	Failed when missing still present. Failed when nominal still present. 2 hyperparameters Crashed 'tol' must be strictly positive and finite	Searching for ensemble method with boosting due to previous cubist method’s outstanding performance
bagEarthGCV - Bagged MARS using gCV Pruning	Multivariate Adaptive Regression Splines Ensemble Model Implicit Feature Selection Bagging Accepts Case Weights	Failed when missing still present. Failed when nominal still present. 1 hyperparameter	Searching for ensemble method with boosting due to previous cubist method’s outstanding performance

Neural network methods all seem take a long time to train, especially the ones that need to import ‘frbs’ library. They consist most of the Rule-Based methods, which are not very successful, either from package issue, or taking too long to train. I tried my best to train all the other methods after few unsuccessful attempts.

Models

The following models were successfully trained. A visual summary of the models is showing below. Models that perform worse than null model have been omitted.



Model	Processing steps	Resampled performance
cubist	bag center scale ica date dummy	RMSE 19.52 R^2 = 1.00 MAE = 14.08
gaussprPoly	bag center scale ica date dummy	RMSE = 23.87 R^2 = 1.00 MAE = 17.88
blackboost	bag date dummy	RMSE = 67.54 R^2 = 1.00 MAE = 49.45
xgbLinear	bag pls YeoJohnson date dummy	RMSE = 73.05 R^2 = 1.00 MAE = 52.26
glmnet	naomit center scale date dummy	RMSE = 85.37 R^2 = 1.00 MAE = 61.41
gaussprLinear	mode	RMSE = 86.44

	date dummy	R^2 = 1.00 MAE = 62.75
pls	mode center scale date dummy	RMSE = 86.49 R^2 = 1.00 MAE = 62.78
kernelpls	mode center scale date dummy	RMSE = 86.59 R^2 = 1.00 MAE = 62.95
bayesglm	mode date dummy	RMSE = 87.06 R^2 = 1.00 MAE = 63.41
rlm	bag center scale date dummy	RMSE = 87.49 R^2 = 1.00 MAE = 61.01
plsRglm	bag ica date dummy	RMSE = 88.03 R^2 = 1.00 MAE = 63.64
ranger	bag YeoJohnson dummy	RMSE = 88.64 R^2 = 1.00 MAE = 65.37
rf	bag YeoJohnson dummy	RMSE = 93.33 R^2 = 1.00 MAE = 69.66
rqlasso	mode center scale date dummy	RMSE = 90.06 R^2 = 1.00 MAE = 57.67
svmPloy	bag dummy	RMSE = 90.38 R^2 = 1.00 MAE = 70.98
glmStepAIC	bag pls ica	RMSE = 91.36 R^2 = 1.00 MAE = 66.71

	date dummy	
rvmPoly	bag YeoJohnson center scale dummy	RMSE = 93.28 $R^2 = 1$ MAE = 70.28
bagEarthGCV	naomit dummy	RMSE = 96.08 $R^2 = 1.00$ MAE = 70.74
qrf	knn YeoJohnson pls dummy	RMSE = 99.06 $R^2 = 1.00$ MAE = 73.31
rpart	bag pls date dummy	RMSE = 118.93 $R^2 = 0.99$ MAE = 89.20
gaussprRadial	bag center scale ica dummy	RMSE = 126.69 $R^2 = 1.00$ MAE = 83.69
knn	bag center scale ica dummy	RMSE = 827.60 $R^2 = 0.80$ MAE = 427.79
krlsPoly	knn pls dummy	RMSE = 1737.66 $R^2 = 0.08$ MAE = 1162.38
avNNet	knn center scale dummy	RMSE = 1768.55 $R^2 = 0.66$ MAE = 1297.49

Bag imputation process approach takes a long time, but it worth the waiting, it boosts the resampled performance better than the knn imputation pre-process in most models. Especially on ‘rvmPoly’ model, bag imputation process took a huge impact on its resampled performance. It also even pushed the RMSE of ‘cubist’ model to the top against ‘gaussprPoly’ model, which cubist was my second-best model originally, before bag imputation pre-preocess kicked in.

Bag imputation, date and dummy is my outstanding pre-processing combination. It depends on the method, if it is a tree-based method, we won't apply center and scale pre-processing approaches, cos they won't work. However, my best pre-process combination is bag imputation, center, scale, ica, date and dummy. Center, scale and ica combination seem work well in some methods when they are appropriate. Especially with my top two performed models, which are shown above, this combination boosted the resampled performance much better than the third model, which center, scale and ica won't take an effect.

Avnnet model has been omitted since its resampled performance is worse than null model

KrlsPoly and knn models' resampled performance are both fall behind other models. Their metrics boxplots all have long tails and notches.

Rpart, gaussprRadial and qrf models' RMSE and MAE metrics boxplots all have tails, but they are much better than krlsPoly and knn models. GaussprRadial's R squared metrics has a little variation shown, which is the same as bagEarthGCV, rvmPoly and blackboost models.

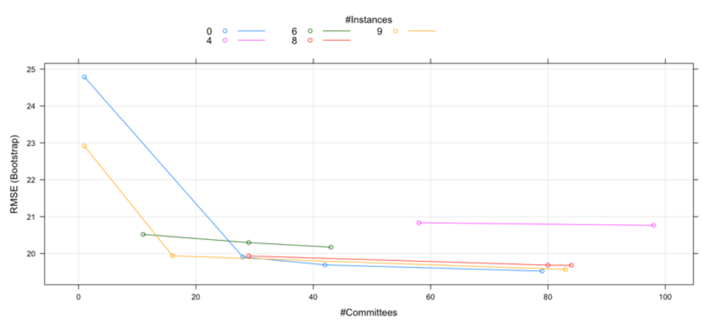
RvmPoly and glmnet models' MAE and RMSE metrics boxplots have an outlier towards the optimistic side. However, glmStepAIC and xgbLinear metrics boxplots have outliers towards the opposite side.

Best Model

Cubist model resampled performance plot

Resampled performance:

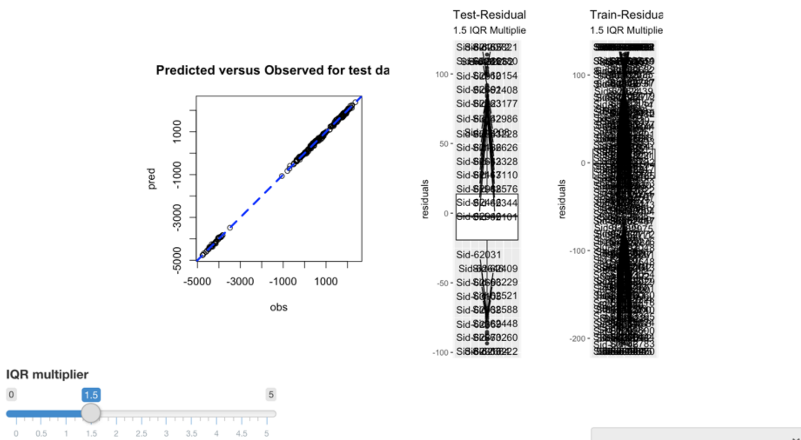
committees	neighbors	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
79	0	19.52	1.00	14.08	2.22	0.00	1.71



Cubist model performance on unseen data

Unseen data results for chosen model: cubist

```
      RMSE   Rsquared        MAE
42.7604518  0.9995004 30.3514532
```

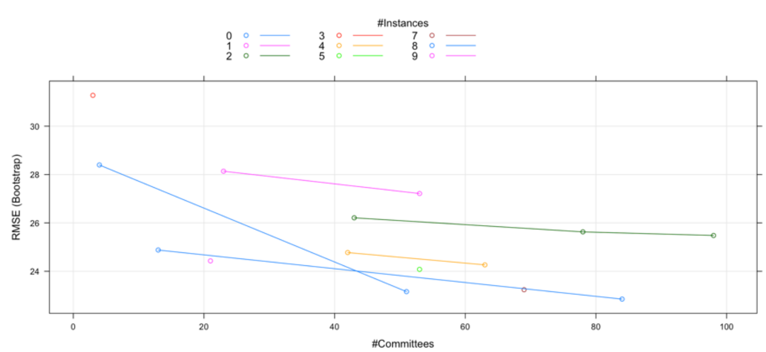


My best model supposes be cubist upon RMSE. However, its RMSE metrics boxplot is showing a variation with a wider body than the second-best model, gaussprPoly, which is performing slightly worse than cubist model based upon RMSE. When I took a look on cubist's performance on unseen data, the performance dropped a lot down to about 42.76 on RMSE. Then I realized there are massive outliers on the training residual already when the IQR multiplier is 1.5. Thus, this caused significant outliers on the test residual. This must due to the training and test split from the very beginning. Hence, I re-split the dataset as 70% training with 30% test, and 80% training with 20% testing again to make sure.

Cubist model resampled performance plot with 70% training and 30% testing split

Resampled performance:

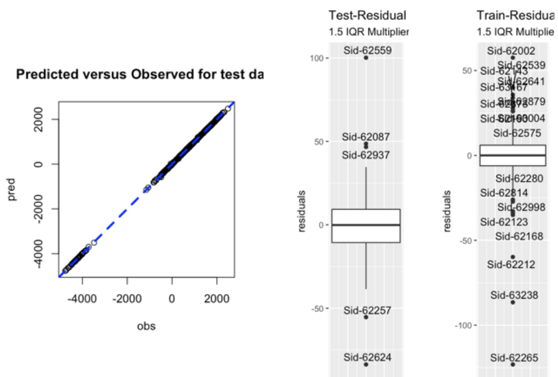
committees	neighbors	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
84	8	22.85	1.00	16.03	3.08	0.00	2.22

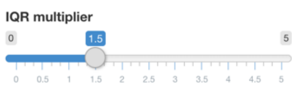


Cubist model performance on unseen data with 70% training and 30% testing split

Unseen data results for chosen model: cubist

```
      RMSE   Rsquared        MAE
15.7157569  0.9999295 11.6995584
```



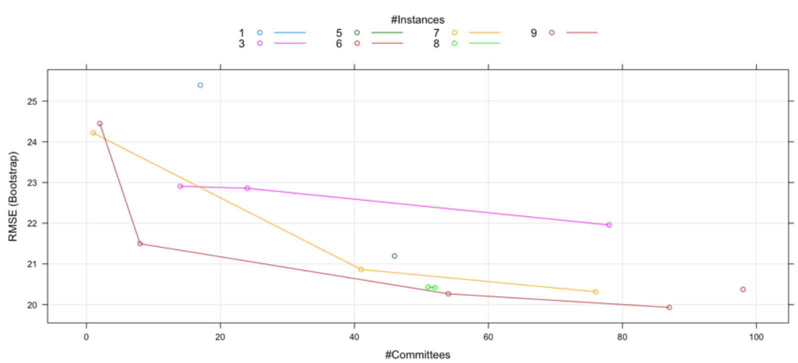


Now we can see the RMSE value is stable on both resampled performance and unseen data performance, the training residual largely reduced that helped reducing the test residual outliers a lot, there are only 5 outliers on test residual boxplot when IQR multiplier is 1.5 as well.

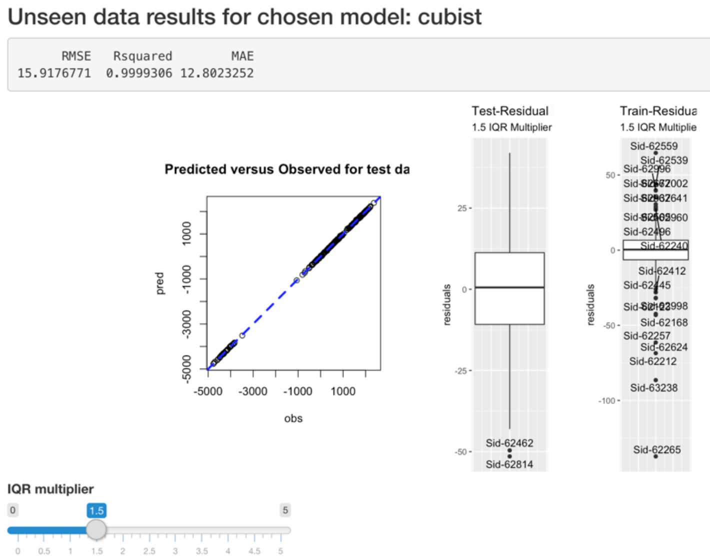
Cubist model resampled performance plot with 80% training and 20% testing split

Resampled performance:

committees	neighbors	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
87	9	19.93	1.00	14.42	2.54	0.00	2.01

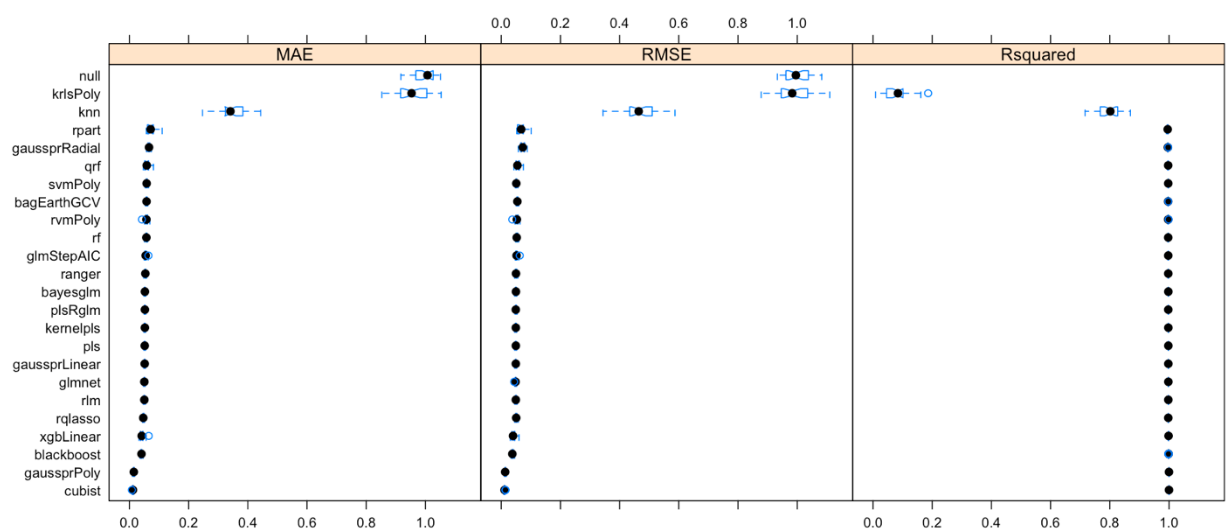


Cubist model performance on unseen data with 80% training and 20% testing split



Again, after re-split with 80% and 20%, it shows slightly better RMSE on resampled performance and there are only two outliers in test residual boxplot, where the training residual is controlled. The test residual outliers are disappeared when we increase the IQR multiplier to 1.9, so they are not significant. But still, it is a sign that these two observations do not fit our model when the IQR multiplier is 1.5, we need to go back and check the raw data's quality to reassure our model's quality.

Successfully trained models with updated cubist model

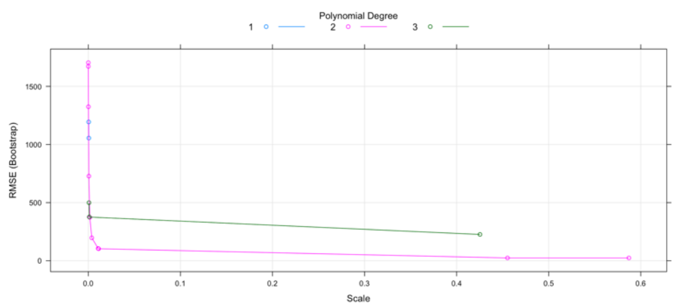


The MAE metric boxplot of cubist model has a outlier towards the optimistic side, but the outlier from RMSE metric boxplot is opposite, but still not significant at all.

GaussprPoly model resampled performance plot

Resampled performance:

degree	scale	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
2	0.59	23.87	1.00	17.88	2.73	0.00	2.41

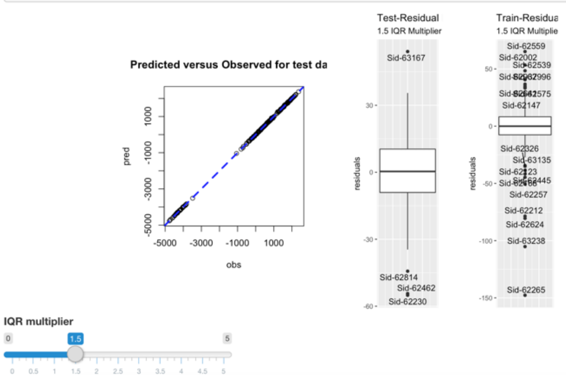


The best hyperparameter here is when it takes polynomial degree as 2 and 0.59 on scale.

GaussprPoly model performance on unseen data

Unseen data results for chosen model: gaussprPoly

RMSE	Rsquared	MAE
16.4236931	0.9999262	12.7319409



The 4 test residual outliers disappeared when IQR multiplier is 2.4, which is not significant.

Our best model is still cubist model, it is slightly statistically better than gaussprPoly model. We can try an ensemble of both of them might perform slightly better than any of them.

Method description

The best model uses method cubist. Cubist is a rule–based model that is an extension of Quinlan's M5 model tree. A tree is grown where the terminal leaves contain linear regression models. These models are based on the predictors used in previous splits. Also, there are intermediate linear models at each step of the tree. A prediction is made using the linear regression model at the terminal node of the tree, but is "smoothed" by taking into account the prediction from the linear model in the previous node of the tree (which also occurs recursively up the tree). The tree is reduced to a set of rules, which initially are paths from the top of the tree to the bottom. Rules are eliminated via pruning and/or combined for simplification.

This method seems work well with this data because the data has a linear relationship with the predictors, it ensembled both linear method and tree-based method's advantages and captured most of the underline true between the predictors and the outcomes. The second-best model, gaussprPoly, utilised the non-linear relationships and captured the underline truth pretty well too. If we ensemble these two, the results will be better in most chances. However, cubist method has much better transparency than gaussprPoly method, from the resampled performance detail, cubist model with a good explanation of how this model is trained, with a lot of detail. Especially the blood type seems has a great control on the training flow. This nominal variable is utilised by tree-based method was another reason why this model has the best statistical performance. Cubist with a tree-based method would give us a variable importance detail, which can't be reached than gaussprPoly model.

In conclusion, cubist model is statistically the best as well as the best transparency model.