Data Analysis

(Regression Model Selection)

Name: Shi Chen
E-mail: yuejianyingmm@icloud.com

Data Description

This assignment is about numeric target variable regression. The dataset using in this assignment is Ass3Data.

In this dataset, there are 1280 observations and 21 variables. The target variable is "Y", which is numeric.

Other 20 variables can be selected as predictors. 18 of them are numeric and 2 of them are nominal. One of the nominal variables is "BloodType", which has 4 types. The other is "TreatmentDate", which has been transformed as "date" type in the pre-processing pipeline.

It should be pay attention, when using numeric method to do the prediction, these two nominal variables should be transformed to numeric in pre-processing part. This will explained in more detail in the strategy part.

By comparing the "ID" and observations, there are no duplicate observations and similar observations. The resample methods like K-fold cross validation and Boot. In this case, using "boot" as re-sampling method.

In this case, by default "IQR multiplier" = 1.5, there are 4 variables ("Alcohol", "Coffee", "exercise", "MDocVisits") do not have outliers. Other variables' outliers are obvious. However, most of them will be removed when increasing "IQR multiplier" up to 2.1. But the outliers of target variable "Y" are extremely significant and in a large amount. Even increasing the "IQR multiplier" to 2.6, a bunch of outliers still cannot be removed. It suggests that the method which robust outliers should be preferred.

As to missing values, there are no excessively missing variables. The target variable doesn't having missing value. So there is no need to delete observations. The two nominal variables also has no missing values. So imputation can come before dummy. On the other hand, there are 320 observations (25%) contain missing values. It is more sensible to impute the missing value rather than remove them directly.

From correlation matrix, there are four groups of variables are in strong linear co-relationship. Except one of these groups show slightly negative linear relation with target "Y", other variables show non-linear relationship with "Y". It suggest that non-linear regression methods may perform better than the linear regression methods.

Strategies

1. The process to find the relatively good model

First, select one basic reliable preprocessing recipe pipeline to compare model resampled performance;

Second, choose 1~2 relatively simple and easy train methods from different families (Neural networks, Ordinary Least Squares, Tree based, Kernel methods).

Third, comparing their resampled performance and find patterns and features of the good performance methods group;

Fourth, using the features found in last step to find more methods (10 - 20) and using validation test MSE to compare the resampled performance;

Fifth, studying the first two or three models and modifying the preprocessing recipe (such as scaling and PCA these kind of model tuning methods) to each method and evaluate the resampled performance;

At last, apply the choosing model to the actual test dataset and evaluate the actual model test performance.

2. Preprocessing recipe pipeline

Missing values

For some methods, missing values will cause problem. As mentioned before, in this case, there are 25% observations containing missing values. The **"knnimputate (k=5)"** is more sensible than "naomit" (part delete method) when dealing with the missing values in preprocessing part.

Nominal variables

To the numeric methods, the nominal variables value should be transformed into numeric.

For combine the information with other variables, the variable "TreatmentDate" can use "step_date()" method to convert date data into three numeric variables ("dow", "month", "year") in this case. After that, the "step_dummy" method will convert nominal variables "BloodType" into several binary type variables for each type of the blood.

3. Methods

As mentioned before, the methods which robust outliers, non-linear, feature selection and have the inner function to deal with the predictors-collinearity should be selected first.

For instance, quantile regression makes no assumptions about the distribution of the residuals. Lasso is good at shrinkage. PLS is good at solving predictors-collinearity problem. The ensemble methods like "boost" methods can improve accuracy. PCA is good for unsupervised gradient decent and LDA is good for supervised classification, both of them can deal with predictors-collinearity.

Uniform Pre-Processing

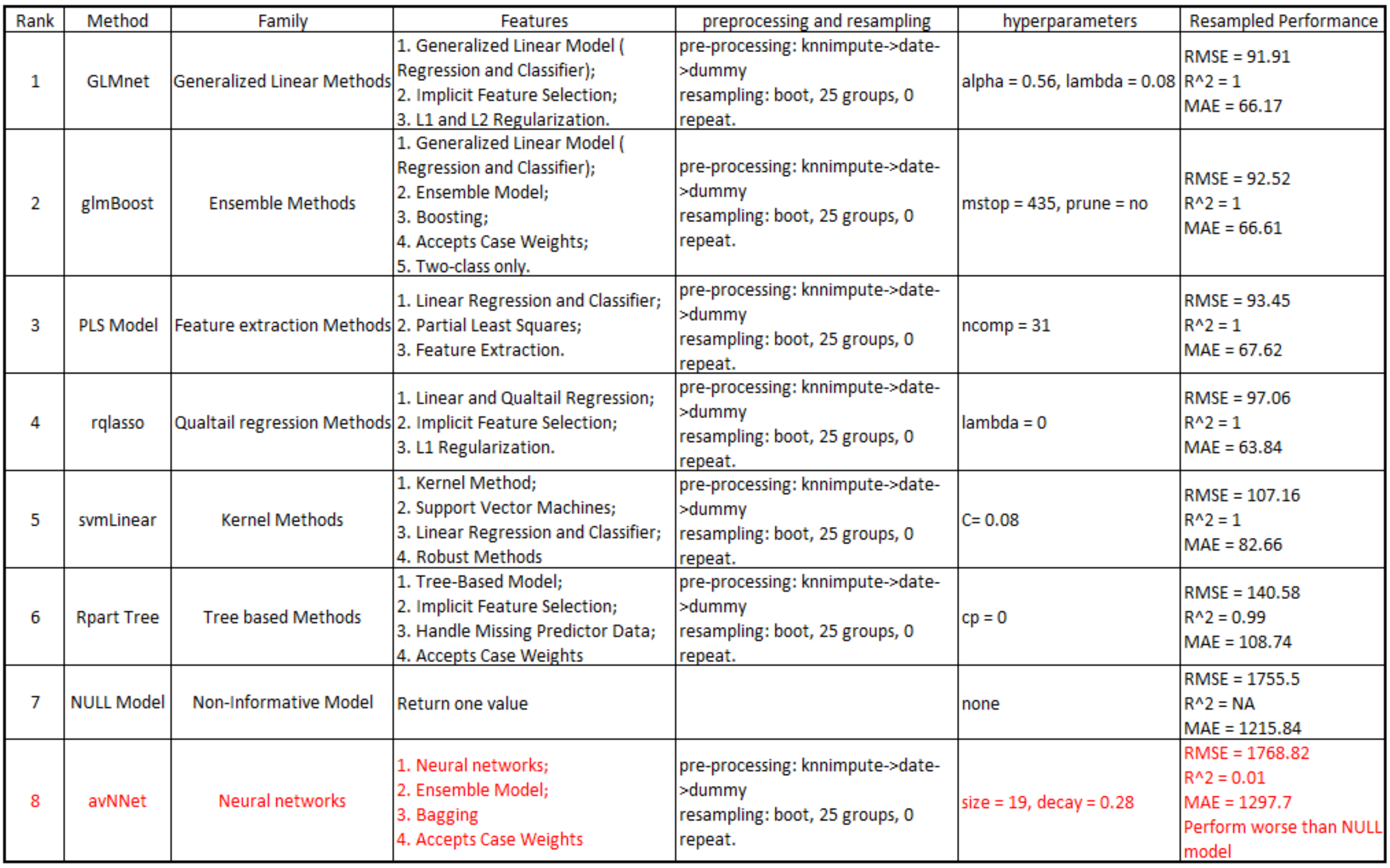
1. **Basic reliable recipe pipeline**

   "knnimputate" → "date" → "dummy"

2. Resample method
   Boot with 25 groups.

First round methods selection and performance comparison

Table1: First group methods and performance comparison

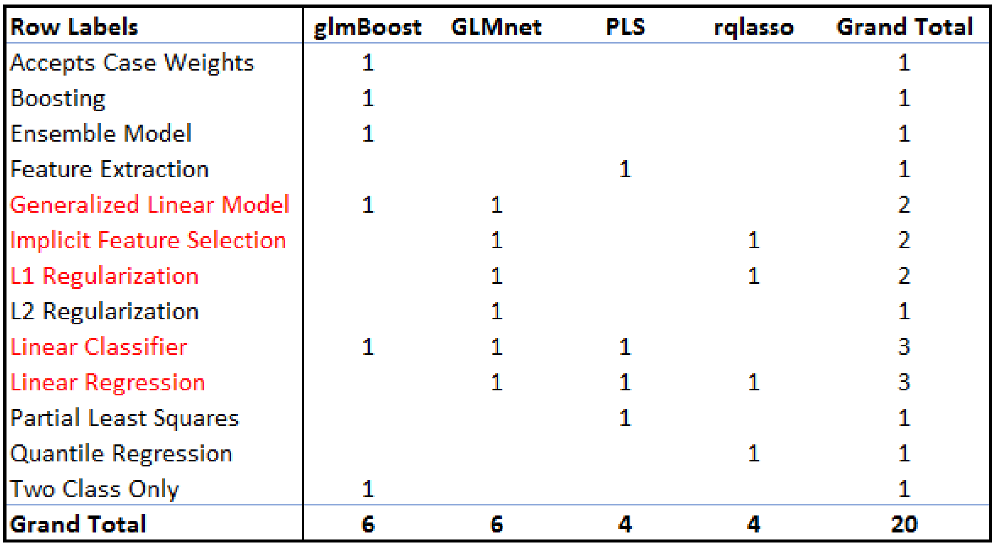


Rank	Method	Family	Features	preprocessing and resampling	hyperparameters	Resampled Performance
1	GLMnet	Generalized Linear Methods	1. Generalized Linear Model (Regression and Classifier); 2. Implicit Feature Selection; 3. L1 and L2 Regularization.	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	alpha = 0.56, lambda = 0.08	RMSE = 91.91 R^2 = 1 MAE = 66.17
2	glmBoost	Ensemble Methods	1. Generalized Linear Model (Regression and Classifier); 2. Ensemble Model; 3. Boosting; 4. Accepts Case Weights; 5. Two-class only.	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	mstop = 435, prune = no	RMSE = 92.52 R^2 = 1 MAE = 66.61
3	PLS Model	Feature extraction Methods	1. Linear Regression and Classifier; 2. Partial Least Squares; 3. Feature Extraction.	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	ncomp = 31	RMSE = 93.45 R^2 = 1 MAE = 67.62
4	rqlasso	Qualtail regression Methods	1. Linear and Qualtail Regression; 2. Implicit Feature Selection; 3. L1 Regularization.	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	lambda = 0	RMSE = 97.06 R^2 = 1 MAE = 63.84
5	svmLinear	Kernel Methods	1. Kernel Method; 2. Support Vector Machines; 3. Linear Regression and Classifier; 4. Robust Methods	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	C= 0.08	RMSE = 107.16 R^2 = 1 MAE = 82.66
6	Rpart Tree	Tree based Methods	1. Tree-Based Model; 2. Implicit Feature Selection; 3. Handle Missing Predictor Data; 4. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	cp = 0	RMSE = 140.58 R^2 = 0.99 MAE = 108.74
7	NULL Model	Non-Informative Model	Return one value		none	RMSE = 1755.5 R^2 = NA MAE = 1215.84
8	avNNet	Neural networks	1. Neural networks; 2. Ensemble Model; 3. Bagging 4. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	size = 19, decay = 0.28	RMSE = 1768.82 R^2 = 0.01 MAE = 1297.7 Perform worse than NULL model

Ps: Rank by resampled performance

Analysis the common features in good performance models

Table 2: First group good performance model feature analysis



Row Labels	glmBoost	GLMnet	PLS	rqlasso	Grand Total
Accepts Case Weights	1				1
Boosting	1				1
Ensemble Model	1				1
Feature Extraction			1		1
Generalized Linear Model	1	1			2
Implicit Feature Selection		1		1	2
L1 Regularization		1		1	2
L2 Regularization		1			1
Linear Classifier	1	1	1		3
Linear Regression		1	1	1	3
Partial Least Squares			1		1
Quantile Regression				1	1
Two Class Only	1				1
Grand Total	**6**	**6**	**4**	**4**	**20**

Based on the information given by table 2, extending the candidate methods, then comparing performance under the same pre-processing and resampling conditions.

Second round methods selection.

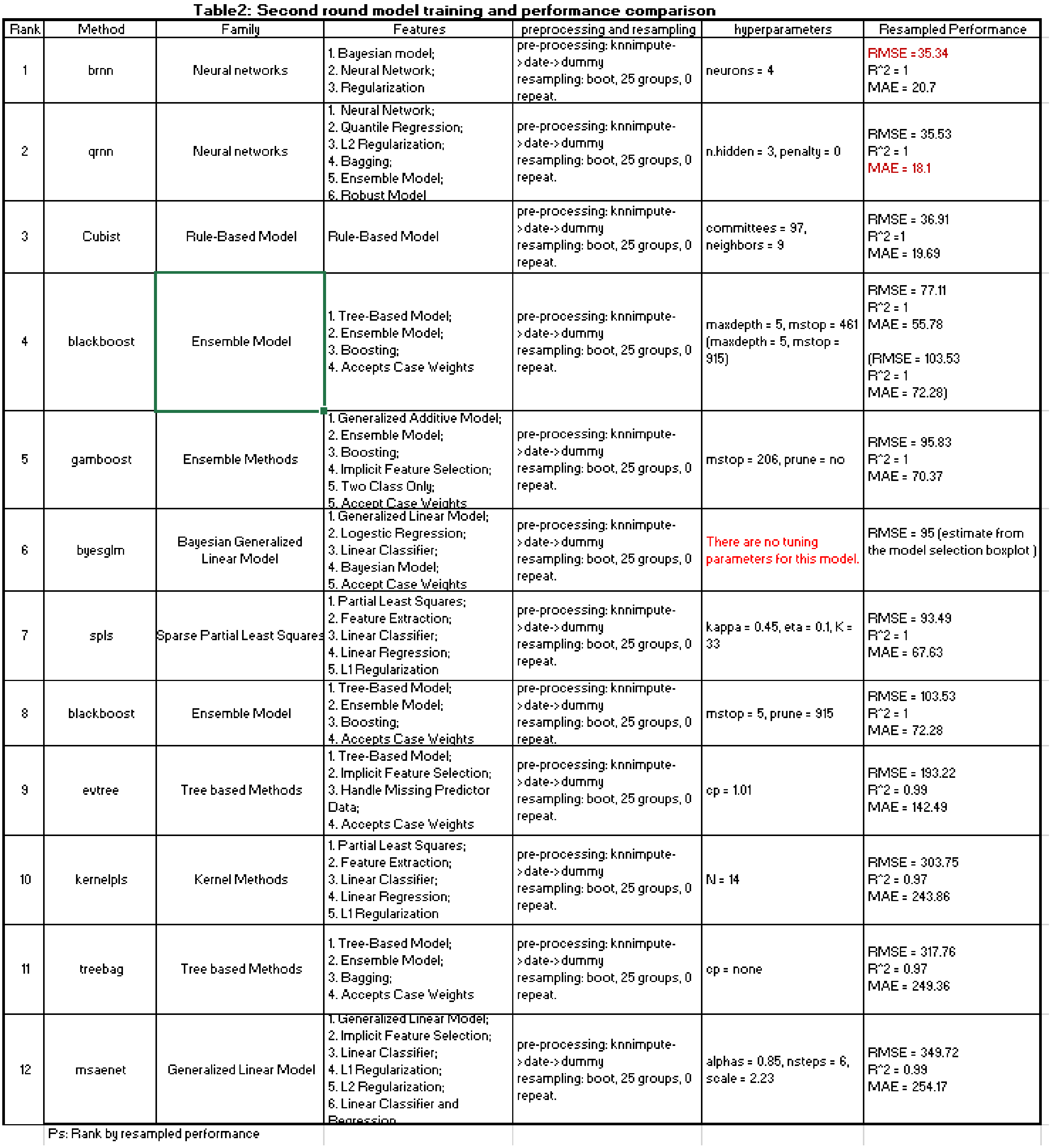
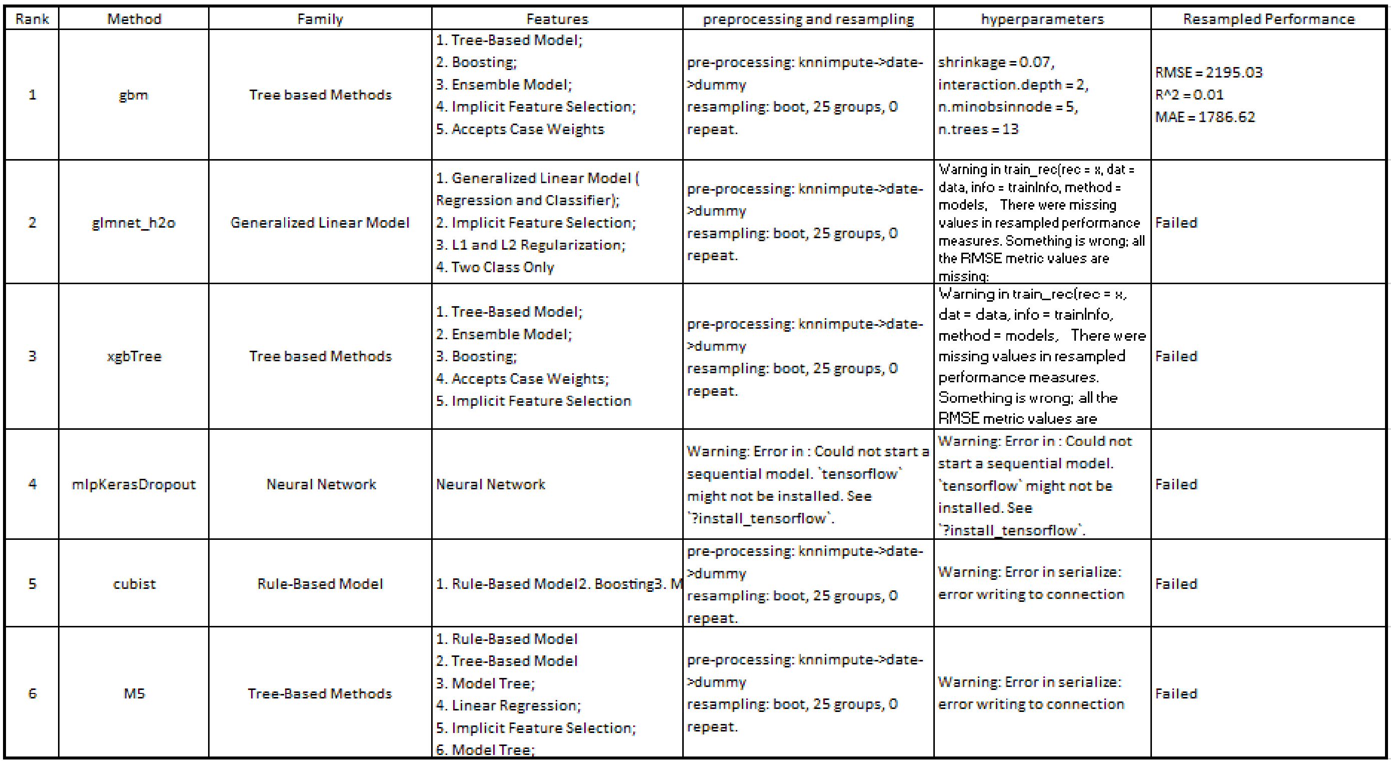


Table2: Second round model training and performance comparison

Rank	Method	Family	Features	preprocessing and resampling	hyperparameters	Resampled Performance
1	brnn	Neural networks	1. Bayesian model; 2. Neural Network; 3. Regularization	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	neurons = 4	RMSE =35.34 R^2 = 1 MAE = 20.7
2	qrnn	Neural networks	1. Neural Network; 2. Quantile Regression; 3. L2 Regularization; 4. Bagging; 5. Ensemble Model; 6. Robust Model	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	n.hidden = 3, penalty = 0	RMSE = 35.53 R^2 = 1 MAE = 18.1
3	Cubist	Rule-Based Model	Rule-Based Model	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	committees = 97, neighbors = 9	RMSE = 36.91 R^2 =1 MAE = 19.69
4	blackboost	Ensemble Model	1. Tree-Based Model; 2. Ensemble Model; 3. Boosting; 4. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	maxdepth = 5, mstop = 461 (maxdepth = 5, mstop = 915)	RMSE = 77.11 R^2 = 1 MAE = 55.78 (RMSE = 103.53 R^2 = 1 MAE = 72.28)
5	gamboost	Ensemble Methods	1. Generalized Additive Model; 2. Ensemble Model; 3. Boosting; 4. Implicit Feature Selection; 5. Two Class Only; 5. Accept Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	mstop = 206, prune = no	RMSE = 95.83 R^2 = 1 MAE = 70.37
6	byesglm	Bayesian Generalized Linear Model	1. Generalized Linear Model; 2. Logestic Regression; 3. Linear Classifier; 4. Bayesian Model; 5. Accept Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	There are no tuning parameters for this model.	RMSE = 95 (estimate from the model selection boxplot)
7	spls	Sparse Partial Least Squares	1. Partial Least Squares; 2. Feature Extraction; 3. Linear Classifier; 4. Linear Regression; 5. L1 Regularization	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	kappa = 0.45, eta = 0.1, K = 33	RMSE = 93.49 R^2 = 1 MAE = 67.63
8	blackboost	Ensemble Model	1. Tree-Based Model; 2. Ensemble Model; 3. Boosting; 4. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	mstop = 5, prune = 915	RMSE = 103.53 R^2 = 1 MAE = 72.28
9	evtree	Tree based Methods	1. Tree-Based Model; 2. Implicit Feature Selection; 3. Handle Missing Predictor Data; 4. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	cp = 1.01	RMSE = 193.22 R^2 = 0.99 MAE = 142.49
10	kernelpls	Kernel Methods	1. Partial Least Squares; 2. Feature Extraction; 3. Linear Classifier; 4. Linear Regression; 5. L1 Regularization	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	N = 14	RMSE = 303.75 R^2 = 0.97 MAE = 243.86
11	treebag	Tree based Methods	1. Tree-Based Model; 2. Ensemble Model; 3. Bagging; 4. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	cp = none	RMSE = 317.76 R^2 = 0.97 MAE = 249.36
12	msaenet	Generalized Linear Model	1. Generalized Linear Model; 2. Implicit Feature Selection; 3. Linear Classifier; 4. L1 Regularization; 5. L2 Regularization; 6. Linear Classifier and Regression	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	alphas = 0.85, nsteps = 6, scale = 2.23	RMSE = 349.72 R^2 = 0.99 MAE = 254.17

Ps: Rank by resampled performance

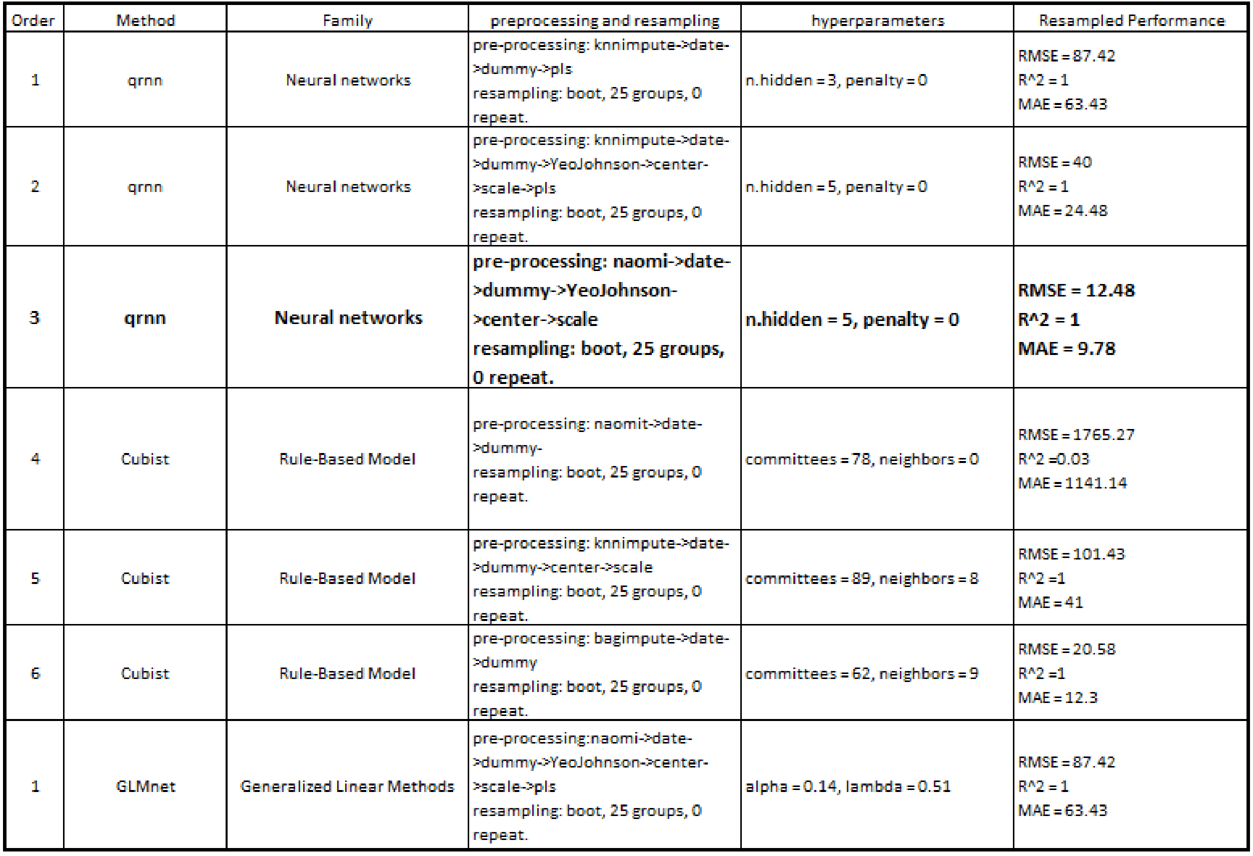
Table3: Fail to train or bad than NULL model



Rank	Method	Family	Features	preprocessing and resampling	hyperparameters	Resampled Performance
1	gbm	Tree based Methods	1. Tree-Based Model; 2. Boosting; 3. Ensemble Model; 4. Implicit Feature Selection; 5. Accepts Case Weights	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	shrinkage = 0.07, interaction.depth = 2, n.minobsinnode = 5, n.trees = 13	RMSE = 2195.03 R^2 = 0.01 MAE = 1786.62
2	glmnet_h2o	Generalized Linear Model	1. Generalized Linear Model (Regression and Classifier); 2. Implicit Feature Selection; 3. L1 and L2 Regularization; 4. Two Class Only	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	Warning in train_rec(rec = x, dat = data, info = trainInfo, method = models, There were missing values in resampled performance measures. Something is wrong; all the RMSE metric values are missing:	Failed
3	xgbTree	Tree based Methods	1. Tree-Based Model; 2. Ensemble Model; 3. Boosting; 4. Accepts Case Weights; 5. Implicit Feature Selection	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	Warning in train_rec(rec = x, dat = data, info = trainInfo, method = models, There were missing values in resampled performance measures. Something is wrong; all the RMSE metric values are	Failed
4	mlpKerasDropout	Neural Network	Neural Network	Warning: Error in : Could not start a sequential model. `tensorflow` might not be installed. See `?install_tensorflow`.	Warning: Error in : Could not start a sequential model. `tensorflow` might not be installed. See `?install_tensorflow`.	Failed
5	cubist	Rule-Based Model	1. Rule-Based Model2. Boosting3. M	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	Warning: Error in serialize: error writing to connection	Failed
6	M5	Tree-Based Methods	1. Rule-Based Model 2. Tree-Based Model 3. Model Tree; 4. Linear Regression; 5. Implicit Feature Selection; 6. Model Tree;	pre-processing: knnimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	Warning: Error in serialize: error writing to connection	Failed

Candidate Model Optimization

Table4: Third round preprocessing selecting



Order	Method	Family	preprocessing and resampling	hyperparameters	Resampled Performance
1	qrnn	Neural networks	pre-processing: knnimpute->date->dummy->pls resampling: boot, 25 groups, 0 repeat.	n.hidden = 3, penalty = 0	RMSE = 87.42 R^2 = 1 MAE = 63.43
2	qrnn	Neural networks	pre-processing: knnimpute->date->dummy->YeoJohnson->center->scale->pls resampling: boot, 25 groups, 0 repeat.	n.hidden = 5, penalty = 0	RMSE = 40 R^2 = 1 MAE = 24.48
3	**qrnn**	**Neural networks**	**pre-processing: naomi->date->dummy->YeoJohnson->center->scale** **resampling: boot, 25 groups, 0 repeat.**	**n.hidden = 5, penalty = 0**	**RMSE = 12.48** **R^2 = 1** **MAE = 9.78**
4	Cubist	Rule-Based Model	pre-processing: naomit->date->dummy- resampling: boot, 25 groups, 0 repeat.	committees = 78, neighbors = 0	RMSE = 1765.27 R^2 =0.03 MAE = 1141.14
5	Cubist	Rule-Based Model	pre-processing: knnimpute->date->dummy->center->scale resampling: boot, 25 groups, 0 repeat.	committees = 89, neighbors = 8	RMSE = 101.43 R^2 =1 MAE = 41
6	Cubist	Rule-Based Model	pre-processing: bagimpute->date->dummy resampling: boot, 25 groups, 0 repeat.	committees = 62, neighbors = 9	RMSE = 20.58 R^2 =1 MAE = 12.3
1	GLMnet	Generalized Linear Methods	pre-processing:naomi->date->dummy->YeoJohnson->center->scale->pls resampling: boot, 25 groups, 0 repeat.	alpha = 0.14, lambda = 0.51	RMSE = 87.42 R^2 = 1 MAE = 63.43

(1) Candidate Models

Here we choose QRNN and Cubist to do the optimization. BRNN is too slow to train and crush many times, so we drop it in this step.

(2) Pre-processing

Centering and scaling is good for some methods but not essential for all of them. It is useful to improve the QRNN performance, but not good for cubist.

PLA can deal with multi-collinearity but not good for QRNN and Cubist in this case.

Naomit is better than knnimpute for QRNN, but not suitable for Cubist. Bagimpute is good for Cubist.

Complex preprocessing will make model easy overfitting or conflict with the built-in related functions of the method. For some complex model like neural networks which contain the function can deal with small sample, feature selection and vector map functions itself. The same function pre-processing method should not added in the training processing.

Different preprocessing has a very powerful influence on model performance. And the optimal pre-processing pipeline is case by case. QRNN can perform very bad in some particular pre-processing situation.

(3) Test set

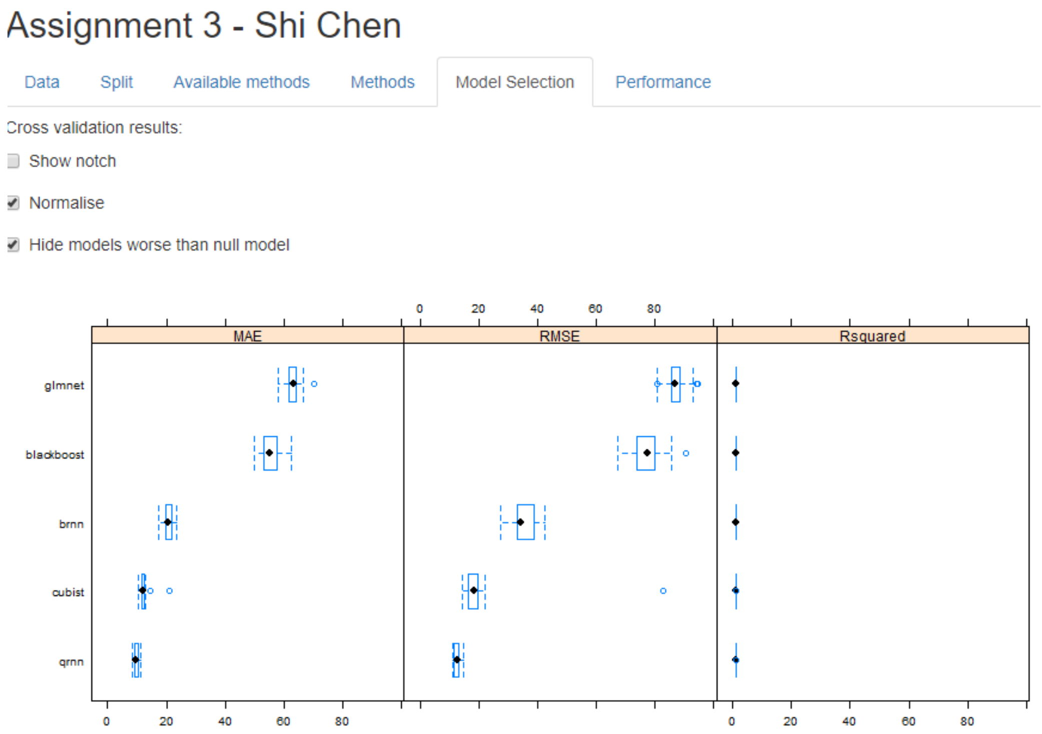
In this case the train test split is 8:2.

(4) Model Tuning

In this case, boot method is used to training the hyper-parameters, the default group is 25, without repeat.

By chance, I found that the method “blackboost” can perform better when I interrupt the training process. So overfitting can also happen in the model tuning process.

Model performance on validation dataset



Assignment 3 - Shi Chen

Cross validation results:

☐ Show notch

☑ Normalise

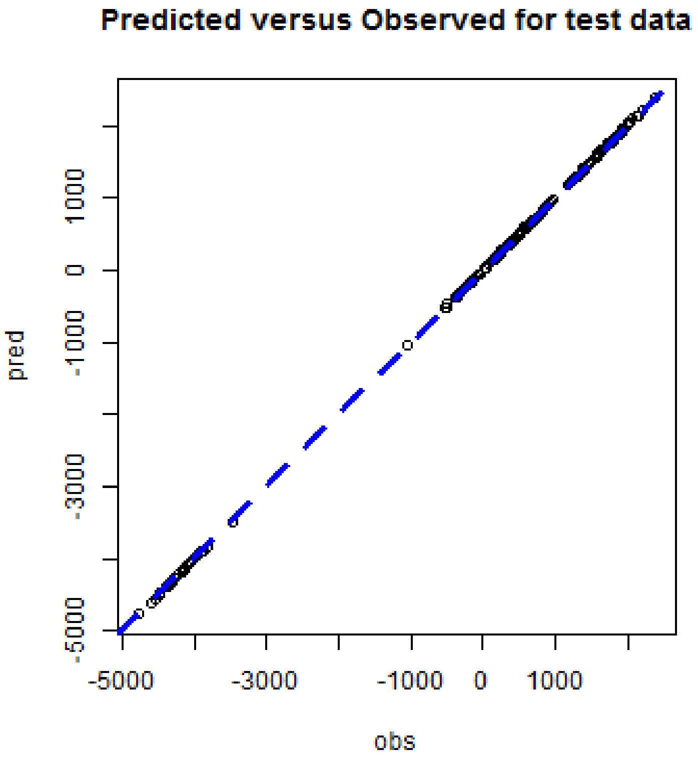
☑ Hide models worse than null model

Best (minRMSE) Model

Based on the RMSE, the best model is QRNN with "naomi->date->dummy->YeoJohnson->center->scale" pre-processing employed. If there is another model have the similar RMSE, ensemble them together might perform slightly better.

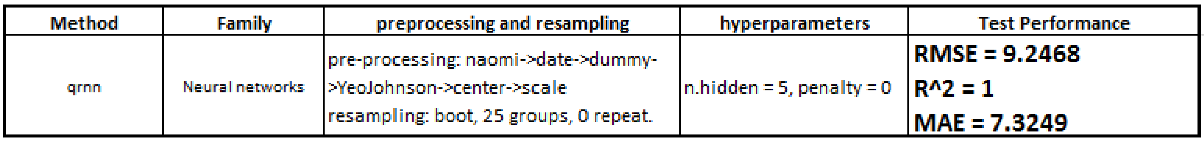
QRNN Performance on unseen data

The predicted versus is shown below.



The metrics for the model is shown below.

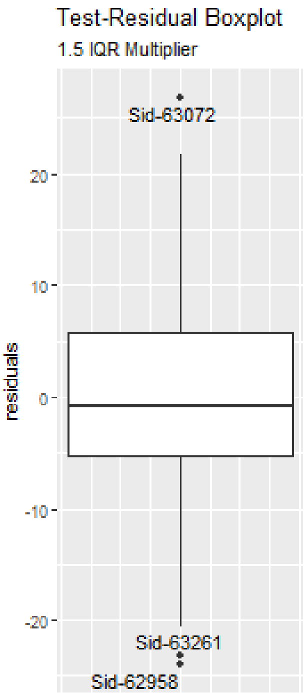
Table5: QRNN performance on unseen data

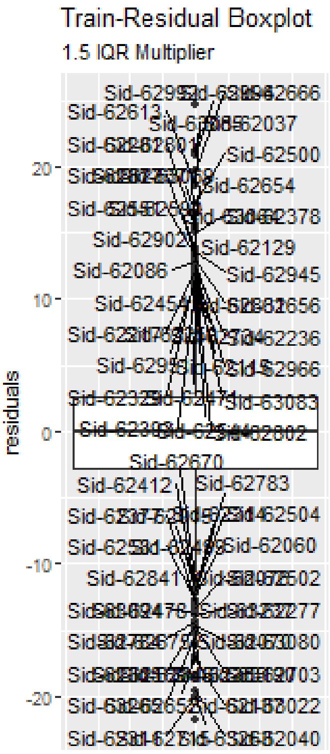


Method	Family	preprocessing and resampling	hyperparameters	Test Performance
qrnn	Neural networks	pre-processing: naomi->date->dummy->YeoJohnson->center->scale resampling: boot, 25 groups, 0 repeat.	n.hidden = 5, penalty = 0	**RMSE = 9.2468** **R^2 = 1** **MAE = 7.3249**

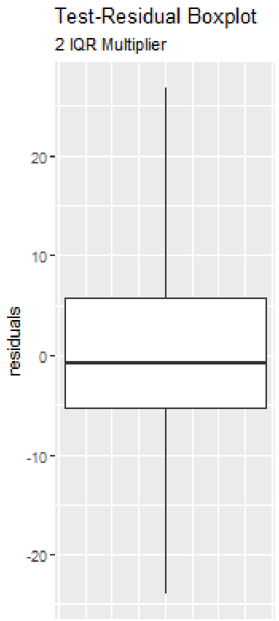
Model-based Outliers

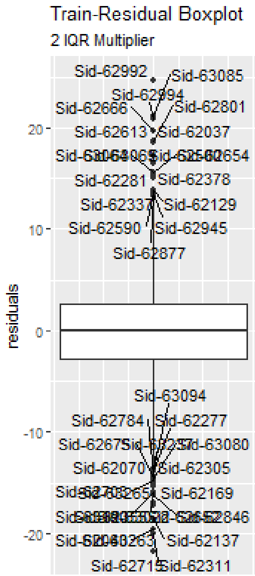
The model-based outliers (at IQR Multiplier of 1.5) are shown below.





When filtering IQR Multiplier to 2, there is no outliers in test dataset, which suggest that the model fits the pattern of the entire test dataset. It is good. Due to the large number of training dataset, there are still many outliers. It suggests, the model may not overfitting in this case.





Method description

According some article and my personal understanding, Quantile Regression Neural Network (QRNN) is an evolution of General regression neural network (GRNN), it hence models quantiles instead of mean in regression process. It selects the optimal solution based on the gradient optimization algorithm. And it introduces weight penalty and bootstrap aggregation methods to avoid overfitting. At the same time, the quantile-based probability density has better interpretability for the quantile part. Overall, it is a nonparametrix, nonlinear model which suitable for mixed discrete-continuous variables prediction, for instance weather changes, medicinal effect, power consumption and so on.

In this case, the dataset is not too big and there are multi-collinearity between predictors. It also has outliers problem and un-uniform value distribution issues. This data set may not suit OLS this kind of simple linear model. There is no **particular hypothesize** of QRNN. The QRNN is good for nonlinear regression and feature selection, and also outliers. So it is suitable for this dataset. As mentioned before, it is good at accuracy and control overfitting. As we know, complex methods like kernel and neural networks is easy to overfitting in small dataset. But QRNN can deal with it very well, that is why, naomit can be used in improving model performance in the pre-processing pipeline.

If the transparency is very important, then QRNN is not a good choice because it is hard to understand and explain how variables been chosen and how they act to output. Linear regression methods, Logistic regression and some tree methods are the best performing transparent method. In this case, GLMnet would be the best performance transparent model.