# Predicting Emergency Roadside Assistance Car breakdown Call Volume

Purpose

This is to demonstrate how to predict the call volume with models lm, glmnet, svmRadial, knn, CART, rf, gbm, cubist and treebag in *caret*.

Included models: tree-based models (*ctree*, *rpart*), boosting models (*gbm*, *gamboost*), bagged models (*treebag*, *bagEarth*), random forest (*rf*,*cforest*, *qrf*), linear regression models (*enet*, *pcr*, *glmnet*), Radial-kernal regression (*rvmRadial*), and neural network (*nnet*, *pcaNNet*, *neuralnet*).

The list is by no means complete. For each model, I only try a grid of simple/convenient parameters. The purpose is to taste a’s many models as I could, and I am not intended to do a serious benchmark comparisons between models.

Data

Used the following data.

Integrated historical data and public road side events

Data set from NJ DOT

[http://www.state.nj.us/transportation/refdata/](%20%20%20%20%20%20%20%20http://www.state.nj.us/transportation/refdata/)

Weather Data

<http://www.wunderground.com/>

Imputed the missing data with median value of Temp and removed duplicate variables from the dataset 1). Date\_key and 2). wevents

split the datetime to workingday, is\_publicholiday,

Merged the two datasets call\_volume and weather data together.

Split the data using sample split and cv – use *caret*

Split the sample into 20% testing subsample and 80% training subsample.

set.seed(849)

sample\_idx <- createDataPartition(call\_cnts$call\_cnt, p = 0.8, list=FALSE, times=1)

data\_train <- call\_cnts[sample\_idx, ]

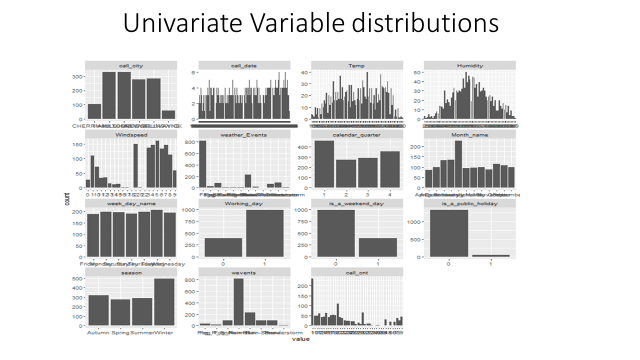
data\_test <- call\_cnts[-sample\_idx, ]

str(data\_train)

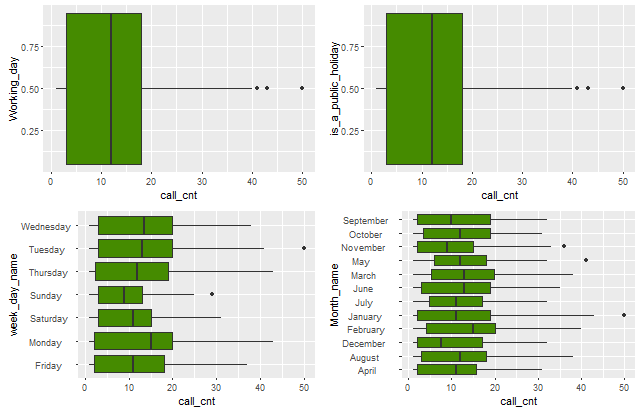
str(data\_test)

Data Exploratory

Used the Histograms, boxplot and scatterplot matrix for data distributions for all the dependent variables







As from the graphs above the distribution of the call counts is right skewed. And there is some collinearity between some weather variables like Temp, Humidity, and wind speed.

And it seems there is no linear relationship between the response variable and predicators.

Modeling with *caret*

Tried using some models from caret and here is the summary of all the models with no transformation.

Models: glmnet, CART, SVM, lm, knn

Number of resamples: 10

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 4.185 4.408 4.697 4.694 4.853 5.522 0

CART 5.520 5.685 6.354 6.482 7.120 8.158 0

SVM 4.241 4.649 4.740 4.862 4.976 6.043 0

lm 4.190 4.412 4.701 4.697 4.861 5.530 0

knn 6.692 7.026 7.129 7.308 7.767 8.044 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 0.6648 0.7111 0.7246 0.7252 0.7369 0.7713 0

CART 0.2295 0.3858 0.5162 0.4711 0.5608 0.6191 0

SVM 0.6474 0.6891 0.7229 0.7172 0.7475 0.7676 0

lm 0.6639 0.7114 0.7253 0.7249 0.7363 0.7709 0

knn 0.2221 0.3092 0.3522 0.3469 0.3811 0.4669 0

From the above results it seems that glmnet has lower RMSE values compared to other models.

Let us try tuning the model with some tune parameters and see if that helps further improvement in RMSE

Used the tunelength=10 and tried the glmnet, svmRadial and rf models and here are the results

summary.resamples(object = cvValuestune)

Models: glmnet, SVM, rf

Number of resamples: 10

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 4.187 4.408 4.699 4.689 4.849 5.521 0

SVM 4.249 4.607 4.717 4.803 5.006 5.549 0

rf 4.129 4.329 4.441 4.481 4.557 5.172 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 0.6647 0.7111 0.7249 0.7259 0.7410 0.7708 0

SVM 0.6646 0.6909 0.7240 0.7175 0.7402 0.7666 0

rf 0.7080 0.7425 0.7533 0.7498 0.7600 0.7783 0

It seems there is not much improvement with just tuning the model. So let us try some BoxCox transformation on the Independent variables and build the models. And here are the results.

Models: glmnet, CART, SVM, lm, knn

Number of resamples: 10

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 4.175 4.431 4.722 4.684 4.825 5.520 0

CART 5.520 5.685 6.354 6.482 7.120 8.158 0

SVM 4.193 4.501 4.703 4.664 4.780 5.248 0

lm 4.172 4.433 4.726 4.685 4.826 5.523 0

knn 6.395 6.692 6.975 6.945 7.073 7.709 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 0.6649 0.7067 0.7278 0.7262 0.7486 0.7726 0

CART 0.2295 0.3858 0.5162 0.4711 0.5608 0.6191 0

SVM 0.6997 0.7104 0.7324 0.7334 0.7546 0.7752 0

lm 0.6646 0.7070 0.7284 0.7262 0.7489 0.7726 0

knn 0.3331 0.3677 0.4117 0.4027 0.4378 0.4498 0

Models: gbm, rf

Number of resamples: 10

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

gbm 4.065 4.279 4.415 4.480 4.623 5.142 0

rf 4.120 4.306 4.485 4.493 4.565 5.153 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

gbm 0.7092 0.7349 0.7562 0.7508 0.767 0.7801 0

rf 0.7099 0.7371 0.7530 0.7487 0.761 0.7703 0

Models: cubist, treebag

Number of resamples: 25

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

cubist 4.908 5.099 5.336 5.299 5.486 5.752 17

treebag 4.302 4.493 4.607 4.637 4.738 5.000 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

cubist 0.6181 0.6424 0.6646 0.6626 0.6868 0.6990 17

treebag 0.6878 0.7201 0.7294 0.7303 0.7432 0.7624 0

It seems there is not much improvement even after the BoxCox transformation.

Since the distribution is right skewed and the relation with some other variables is highly non-linear. Let us try with the usual log transformation to the dependent variable that will help fix the problem of variance across the range of predictions

Models: glmnet, CART, SVM, lm, knn

Number of resamples: 10

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 0.3900 0.3941 0.4085 0.4124 0.4255 0.4446 0

CART 0.4350 0.4524 0.6245 0.6494 0.8204 0.9370 0

SVM 0.4061 0.4297 0.4539 0.4476 0.4669 0.4788 0

lm 0.3912 0.3931 0.4097 0.4124 0.4256 0.4440 0

knn 6.3950 6.6920 6.9750 6.9450 7.0730 7.7090 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

glmnet 0.8468 0.8633 0.8730 0.8733 0.8890 0.8970 0

CART 0.3324 0.5106 0.6826 0.6586 0.8479 0.8651 0

SVM 0.8200 0.8360 0.8528 0.8537 0.8759 0.8822 0

lm 0.8473 0.8640 0.8719 0.8732 0.8887 0.8970 0

knn 0.3331 0.3677 0.4117 0.4027 0.4378 0.4498 0

Models: gbm, rf

Number of resamples: 10

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

gbm 0.3903 0.4294 0.4358 0.4419 0.4488 0.4891 0

rf 0.3711 0.4053 0.4258 0.4233 0.4448 0.4768 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

gbm 0.8114 0.8477 0.8597 0.8564 0.8715 0.8910 0

rf 0.8233 0.8506 0.8643 0.8665 0.8858 0.8994 0

Models: cubist, treebag

Number of resamples: 25

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

cubist 0.4368 0.4377 0.4386 0.4386 0.4394 0.4403 23

treebag 0.4170 0.4267 0.4402 0.4395 0.4493 0.4673 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

cubist 0.8547 0.8558 0.8570 0.8570 0.8582 0.8594 23

treebag 0.8300 0.8470 0.8534 0.8546 0.8644 0.8725 0

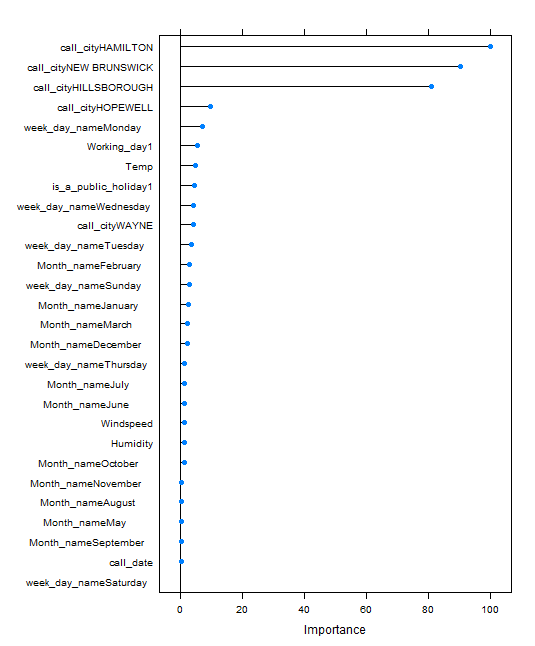
It seems there is good improvement in the RMSE scores and RSquared value with the log transformation.

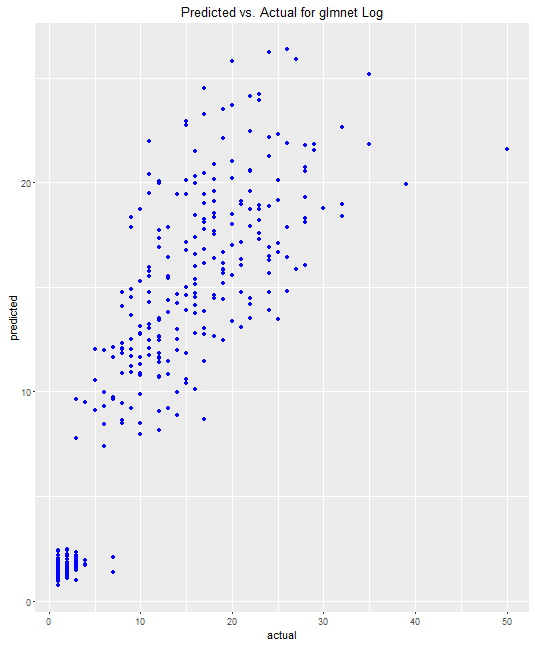
Let us mow evaluate the accuracy of the Mdel by predicting the models that we build so far.

|  |  |  |
| --- | --- | --- |
| Predicted Values on the test dataset | | |
|  | RMSE | Rsquared |
| Linear regression (lm) | 4.43 | 0.75 |
| glmnet with log | 4.34 | 0.77 |
| lmtransx | 4.43 | 0.75 |
| glmnettune | 12.6 | 0.7 |

The RMSE seems to be pretty close the training model call count and the predicted call count of the test data set.

Here is the variable importance and the Predicted Vs Actual values of the final model.





Summary

*This is good exercise for me to understand and demonstrate the end to end phase of machine learning process starting from reading data into R to predict the Response variable. And trying different models using caret helped to experiment and understand how the model predicts different algorithms and the paramters that are using the models.*

*The benefit for the client by predicting the call volume and having identified the values let the business be*

* *Be prepared for inclement weather conditions and respond faster to customer for assistance*
* *Reduce the wait time and response time*
* *CSAT scores can be improved and customer acquisition can be improved.*
* *“Expect the best. Plan for the worst” – customer loyalty*