DATA423 Assignment 3

Alexander Jefferies (85974993)

# Data description

The data consists of 690 observations of 21 variables. The target variable is “Response”. There are missing values in a most of the numeric variables, however, they appear to be random on visual inspection. There are no excessively missing variables or observations. The numeric predictor data has some uni-variable outliers, but these disappear when the IQR multiplier reaches 2.3 so these are not of great significance. The only nominal variable in the dataset is “BloodType” and it has a cardinality of 4. There is one Date variable “Observation Date”. The format is unambiguously YYYY-mm-dd. These have been converted to date variables. The numeric data visually appear to be on multiple scales. “Alcohol”, “Coffee”, “Exercise” and “Treatments” values are on the same scale and all the “reagent\_\*” variables form a second scale. “Response” seems to be on a scale of its own. Standardization may have an impact on model results. The means range from 2 to 700 across the variables and the standard deviations vary from 1.5 up to 118. The predictor correlation shows that there are two blocks of numeric variables are highly correlated. Reagents A, G, I and C for a block, Reagents J, N, D, H, L, F form a block. “Exercise”, “Coffee”, and “Alcohol” are highly correlated with "Response”.

# Strategies

## Missing Data

There are no excessively missing variables or observations to discard.

There is not a compelling case for partial deletion for two reasons. Firstly, there is no evidence of “Missing Completely at Random” missing values. Therefore, partial deletion could generate bias. Secondly, there are only a small number of observations (<690) to train on so partial deletion may reduce the effectiveness of our parametric models even further.

For most methods, we shall employ KNN (neighbors=5) imputation. However, some methods such as **rpart** can tolerate missing values, so will be tried without the imputation.

Note that there are no missing values in the Target values, so we do not need to remove these observations.

## Outliers

The uni-variable outliers are not concerning so we shall retain all the observations. In the model selection process, there were some robust methods were trialed. More pressing, the significance of residual outliers once the models had generated predictions. We shall discuss this in the Model section of the report.

## Pre-Processing

Missing values were dealt with in most situations (except for rpart) via imputation.

Centering and scaling are suggested due to the variety in means and variances as well as the multiple scales that exist within the numeric data. Furthermore, they help with interpretability for some models that return coefficients.

As a standard, Date Variables were omitted because this seemed to either cause many of the models to crash or added too much model complexity for the amount of training data present. Once select methods were found, if possible, the date variables were either converted to decimal, day-of-week, month, or a combination of the above for method tuning.

All the nominal variables were treated using dummy encoding since they all had low cardinality. This included the dummy variables for when nominal Date variables were included for the more finely tuned models.

The hyper-parameters were tuned by default using 25 bootstrap re-samples of the training data. The stipulated number of parameter combinations (tuneLength) for each model was adjusted on a case-by-case basis.

A Principal Component Analysis (PCA) step was included to reduce the feature space.

A static 20% of the data became our hold out data and was set aside to measure model generalizability.

# Method Selection

In selecting models, the strategy involved trying a reasonable number of diverse methods. This involved statistically finding 30 methods that were maximally dissimilar to reduce our set of candidate models.

The Candidate models were categorized by “Neural Networks”, “Ordinary Least Squares”, “Tree Based” and “Kernel Methods”.

A few Kernel and Neural Network type methods were incorporated to ensure that there were similar numbers of methods assessed from each category.

In an attempt to equally assess each method, it was essential to consider a preprocessing chain that was suitable for the method. Inappropriate pre-processing steps and orderings generally caused the training to fail.

## Trials

The following methods were explored:

|  |  |  |
| --- | --- | --- |
| Method | Characteristics | Notes |
| qrnn – Quantile Regression Neural Network | Neural Network  L2 Regression  Quantile Regression  Bagging  Ensemble Model  Robust Model | Issues while training. For each resample the message below produced:  “Warning: model fit failed for Resample\_: n.hidden=5, penalty=1e-01, bag=FALSE Error in qrnn::qrnn.fit(as.matrix(x), matrix(y), n.hidden = param$n.hidden, :  nlm optimization failed” |
| foba - Ridge regression with Variable Selection | Linear Regression  Ridge Regression  L2 Regularization  Feature Selection | package ‘foba’ is not available for this version of R.  Unable to download using CRAN Repository |
| ANFIS - Adaptive-Network-Based Fuzzy Inference System | Rule-Based Model | Too slow to Train. |
| rvmLinear- Relevance Vector Machines with Linear Kernel | Kernel Method  Relevance Vector Machines  Linear Regression  Robust Methods | Issues while training. Message produced for each resample: “Warning: model fit failed for Resample\_\_: parameter=none Error in chol.default(crossprod(Kr)/var + diag(1/thetatmp)) :  the leading minor of order 229 is not positive definite” |
| glmnet\_h2o-glmnet | Generalized Linear Model  Implicit Feature Selection  L1 Regularization  L2 Regularization  Linear Classifier  Linear Regression  Two Class Only | Issues while training. Message produced “Error in h2o.getConnection() :  No active connection to an H2O cluster. Did you run `h2o.init()`” |
| GLMnet-glmnet | Generalized Linear Model  Implicit Feature Selection  L1 Regularization  L2 Regularization  Linear Classifier  Linear Regression | 2 hyper-parameters |
| Pls-Partial Least Squares | Partial Least Squares  Feature Extraction  Linear Classifier  Linear Regression | 1 hyper-parameter  Slow to train |
| Rpart-Classification and Regression Trees | Tree-Based Model  Implicit Feature Selection  Handle Missing Predictor Data  Accepts Case Weights | 1 hyper-parameter |
| Cubist | Rule-Based Model  Boosting  Ensemble Model  Prototype Models  Model Tree  Linear Regression  Implicit Feature Selection | 2 hyper-parameters |
| Brnn-Bayesian Regularized Neural Network | Bayesian Model  Neural Network  Regularization | 1 hyper-parameter  Slow to train |
| bagEarth-Bagged MARS | Multivariate Adaptive Regression Splines  Ensemble Model  Implicit Feature Selection  Bagging  Accepts Case Weights | 2 hyper-parameters |
| krisPoly-Polynomial Kernel Regularized Least Squares | Kernel Method  L2 Regularization  Polynomial Model | 2 hyper-parameters |
| pcaNNet - Neural Networks with Feature Extraction | Neural Network  Feature Extraction  L2 Regularization  Accepts Case Weights | 2 hyper-parameters |
| Glmboost- Boosted Generalized Linear Model | Generalized Linear Model  Ensemble Model  Boosting  Linear Classifier  Two Class Only  Accepts Case Weights | 2 hyper-parameters |
| Bam - Generalized Additive Model using Splines | Generalized Linear Model  Generalized Additive Model | 2 hyper-parameters |
| plsRglm - Partial Least Squares Generalized Linear Models | Generalized Linear Models  Partial Least Squares  Two Class Only | 2 hyper-parameters  Slow to train |
| Rlm – Roust Linear Model | Linear Regression  Robust Model  Accepts Case Weights | 2 hyper-parameters |
| Pre -Prediction Rule Ensembles | Rule-Based Model  Regularization | 7 hyper-parameters  Slow to train |
| Ppr- Projection Pursuit Regression | Feature Extraction  Accepts Case Weights | 1 hyper-parameter |
| Dnn-Stacked AutoEncoder Deep Neural Network | Neural Network | 5 hyper-parameters  Slow to train |
| Spls-Stacked AutoEncoder Deep Neural Network | Partial Least Squares  Feature Extraction  Linear Classifier  Linear Regression  L1 Regularization | 3 hyper-parameters  Slow to train |
| svmLinear-Support Vector Machines with Linear Kernel | Kernel Method  Support Vector Machines  Linear Regression  Linear Classifier  Robust Methods | 1 hyper-parameter |

## Models

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

Chart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Pre-Processing | Hyper-parameters | Resampled performance |
| GLMnet | Impute\_knn ,month,scale,center,dummy | alpha = 1  lambda = 5.79 | RMSE: 285.07  R2: 0.91  MAE: 196.63 |
| pls | Impute\_knn,dow,scale,center,dummy | Ncomp = 12 | RMSE: 264.95  R2: 0.92  MAE: 185.36 |
| rpart | Impute\_knn,dow,month,scale,center | cp = 0.00 | RMSE: 524.78  R2: 0.69  MAE: 403.99 |
| Cubist | Naomit, month, dummy | Committess =20  Neighbors = 0 | RMSE: 287.86  R2:0.91  MAE: 196.12 |
| brnn | Impute\_knn,dow,month,scale,center,pca,dummy | Neurons = 3 | RMSE:116.58  R2: 0.98  MAE: 87.22 |
| bagEarth | Impute\_knn,dow,month,interact,scale,center,pca,dummy | Nprune=18  Degree=1 | RMSE: 272.84  R2: 0.92  MAE: 195.37 |
| krisPoly | Impute\_knn, scale, center, pca, dummy | Lambda = NA  Degree=1 | RMSE: 916  R2: 0. 18  MAE: 723.53 |
| pcaNNet | Impute\_knn,scale,center,pca,dummy | Size=1  Decay=0 | RMSE:2231  R2:NA  MAE:2035.44 |
| Glmboost | Impute\_knn,interact,month,dow,scale,center ,dummy | Mstop=650  Prune=no | RMSE:266.49  R2: 0.92  MAE: 191.25 |
| bam | Impute\_knn\*\* | Select=TRUE  Method=GCV.Cp | RMSE: 340.53  R2: 0.87  MAE: 328.88 |
| plsRglm | Impute\_knn, dow,interact,scale,center,pca,dummy | Nt= 7  alpha.pvals.expli = 1 | RMSE: 264  R2: 0.92  MAE: 190.83 |
| rlm | Impute\_knn,scale,center ,dummy | INTERCEPT=true  PSE=psi.bisquare | RMSE: 282.51  R2:0.91  MAE: 194.86 |
| pre | Impute\_knn,scale,center,pca,dummy | Sampfrac=0.5  Maxdepth=1  Learnrate=0.01  Mtry=inf  Use.grad=TRUE  Penalty.par.val=lambda.min | RMSE: 311.99  R2: 0.89  MAE: 221.67 |
| Ppr | Impute\_knn,scale,center,pca,dummy | Nterms=1 | RMSE: 298.2  R2: 0.9  MAE: 206.42 |
| Dnn | Impute\_knn,scale,center,pca,dummy | layer1 =2  layer2 =0  layer3 = 0  hidden\_dropout =0  visible\_dropout=0 | RMSE:839.32  R2:0.22  MAE:662.05 |
| Spls | Impute\_knn,Interact,scale,center,pca,dummy | K=6  Eta=0.81  Kappa=0.5 | RMSE: 284.19  R2:0.91  MAE: 202.08 |
| svmLinear | Impute\_knn,poly,month,dow,scale,center,pca,dummy | C=1 | RMSE:271.97  R2: 0.92  MAE: 189.9 |

\*\*this particular model did not train unless the preprocessing contained few steps

The top 10 models were then fine-tuned with:

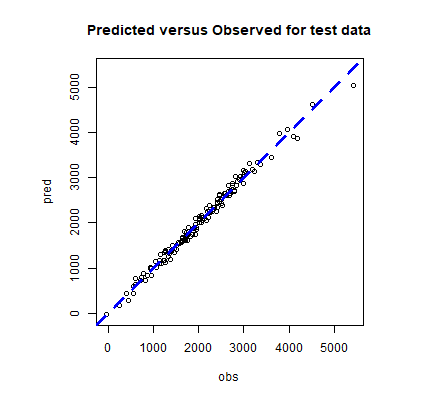
* Added dateDecimal, DOW, and Month variables
* Dimensionality reduction technique pls (no improvements made with this one)
* An interaction step amongst numeric predictors
* A polynomial transform of numeric predictors of degree 2

# Best Model

Based on its RMSE value, the best model is **brnn**. Out of the models that were trained it performed significantly better than any other model with the next best model having a RMSE value 100 greater than it.

## Performance on unseen data

We generated the following model by making predictions using the test data.



The expected performance on unseen data for the best model, **brnn**, is

|  |  |
| --- | --- |
| Metric | Value |
| RMSE | 100.185 |
| RSquared | 0.988 |
| MAE | 79.727 |

## Observations that do not fit the model

The model-based outliers at an IQR of 1.6

Chart, box and whisker chart

Description automatically generated

**Discussion of Model Outliers**

For some of these residuals it is possible that the missing values are the cause as in the case of observations tid-57408, and tid-57476 which each have 3 missing values. This indicates that it is our imputation method that is not performing very well in these cases.

However, it is less clear for the remaining model outliers. It is possible that a high value for “Chemo Treatments” is the cause of these but would need further investigation.

## Method Description

The best model uses the method **brnn.** This is quite perplexing as with 120 parameters to estimate, one might think that we do not have sufficient data to allow proper. No doubt the method indeed performed very well on the test data as with the training data. Perhaps the sheer effectiveness of Bayesian Regularization prevents the model coupled with the PCA dimensionality reduction from overfitting despite the small set of training data.

This method is a Neural Network with a penalty function that leverages Bayesian Statistics to regularize the model. It uses a Maximum A Posteriori derived cost function to instigate weight decay. It has one parameter n-number of neurons.

See documentation [here](https://www.rdocumentation.org/packages/brnn/versions/0.9/topics/brnn)

**brnn characteristics**

* Does not tolerate missing values
* Bayesian Model
* Neural Network
* Regularization
* Regression or Classification
* Eager Prediction Speed.
* Slow to train
* Good for Non-Linear Problems

## Transparency Considerations

As a Neural Network, **brnn** is not very interpretable. We are not able to easily draw conclusions about how the important variables affect the outcome of the model. This is mainly because neural networks are highly complex and contain hundreds to thousands of mathematical operations to transform inputs into outputs.

In order, to achieve transparency, we regress to **glmboost**, though any of the 8 prior candidate models would be valid (provided the models themselves were interpretable) as they are not statistically different. Although the PCA step increases model effectiveness, the model becomes harder to interpret. So, for transparency, the PCA step has been removed from the preprocessing pipeline. Note the RMSE of 266 with the PCA step and 288 without.

The expected performance on unseen data for the best transparent model, glmboost, is:

|  |  |
| --- | --- |
| Metric | Value |
| RMSE | 261.597 |
| RSquared | 0.915 |
| MAE | 195.551 |