# Data423 Assignment 3

Oscar Vanhanen 34640481

## 1. Data Description

The data consists of 1069 observations of 21 variables. The target variable is “Response” a continuous numeric value.

There are missing values in all of the Reagent variables. The missing values visually appear random. 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 2.4 so these are not of great significance. There are outliers present in the Response variable visible until an IQR of 3.3.

The one nominal variable “BloodType” has a cardinality of 52, and the other, “District”, has low cardinality.

The Reagent numeric variables have a variety of scales with the means ranging from ~100 to ~500.

The other categorical variables appear to have their means centred towards 0, as they are very small values.

The predictor correlation shows four blocks of numeric variables are highly correlated (negatively). These all have similar variable names i.e. “Reagent\*”. The groups are as follows:

* Reagents I, A, C, G
* Regents B, K
* Reagents M, E
* Reagents F, H, J, D, L, N

The date variable “ObservationDate” has been converted to date format.

## 2. Strategies

### Missing Data

There are no excessively missing variables or observations to discard.

The methods that implicitly handle missing values can be tried on the raw data.

For the other methods, we shall employ knn (neighbours=5) imputation as a standard approach. Once selected, the best model was improved slightly by using bag imputation.

### Outliers

There are no significant outliers present in the data.

The only issue was the significance of residual outliers in section 7.

### Processing

Missing values (and future missing values) can be dealt with using imputation. Experimentation with the slow “bag imputation” was reserved for the best-performing models.

### Methods

In choosing methods to try, the strategy was based on covering all 4 main family groups: Neural Networks, Ordinary Least Squares, Tree-based, and Kernel methods. Methods were picked at random from each. If the method performed well then methods similar in the same vicinity or axis (on a distance plot of methods, Appendix A)

## 3. Trials

An initial set of pre-processing methods used for all methods was as follows: ("naomit", "impute\_knn", "month", "dow", "dummy"). Once the top 3 performing models were obtained, the pre-processing steps were tuned as in the next section Models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Characteristics** | **Notes** | **Reason chosen** |
| Brnn - Bayesian Regularization for Feed-Forward Neural Networks | Bayesian Model Neural Network Regularization | Failed when missing present.  1 hyperparameter | Random example of Neural Network based method (Euclidean) |
| Neural Net | Neural Network | 3 hyperparameters | Close proximity to brnn |
| Multi-Layer Perceptron | Neural Network | 1 hyperparameter | Close proximity to brnn |
| Quantile Regression Neural Network | Neural Network, L2 Regularization, Quantile Regression, Bagging, Ensemble Model, Robust Model | 3 hyperparameters | Random NN, and close proximity to brnn |
| Deep neural network (dnn) | Neural Network | 5 hyperparameters | Close proximity to brnn, also a NN method. Training not completed > 1 hour. |
| SIMPLS | Partial Least Squares, Feature Extraction, Linear Classifier, Linear Regression | hyperparameters | Close proximity to brnn, cover PLS group |
| Bagged CART (treebag) | Tree-Based Model, Ensemble Model, Bagging, Accepts Case Weights | NA | Cover tree-based method -random |
| Boosted Tree (bstTree) | Tree-Based Model, Ensemble Model, Boosting | 3 hyperparameters | Chosen as next closer to brnn, on the same y-axis. |
| Gaussian Process with Polynomial Kernel  (gaussprPoly) | Kernel Method, Gaussian Process, Polynomial Model | 2 hyperparameters | Random Kernel method |
| Partial Least Squares  (kernelpls) | Partial Least Squares, Feature Extraction, Kernel Method, Linear Classifier, Linear Regression | 1 hyperparameter | Random kernel method |
| *Going off bad results for kernelpls (657) may be best to stay within the area of brnn Y or X axis* | | | |
| svmPoly |  | 3 hyperparameters | Same x-axis as brnn, but towards the top |
| Support Vector Machine with Polynomial Kernel  (svmRadial) | Kernel Method, Support Vector Machines, Radial Basis Function, Robust Methods | 2 hyperparameters | Another svm method |
| Support Vector Machines with Linear Kernel  (svmLinear) | Kernel Method, Support Vector Machines, Linear Regression, Linear Classifier, Robust Methods | 1 hyperparameter | Chose as closer to brnn, but still within SVM group |
| Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) | Rule-Based Model | 2 hyperparameters | NN model close to brnn |

## 4. Models

The following models were successfully trained (Table 1). A visual summary of the models is shown below in Figure 1. Models that performed worse than the null model have been omitted (Appendix B).

Model comparison chart of top performing models

A picture containing line, diagram, rectangle, parallel

Description automatically generated

**Figure 1.** Best three trained models and the associated statistics.

**Table 1**. The final list of best-performing models and associated statistics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Model** | **Processing steps** | **Hyper-parameters** | **Resampled Performance** |
|  | brnn | naomit, impute\_bag, month, dow, dummy, nzv | Neurons = 5 | RMSE: 129.59, R2: 0.98 |
|  | svmRadial | naomit, impute\_knn, month, dow, dummy, interact, centre, scale | Sigma = 0.01  C = 16.00 | RMSE: 299.27, R2: 0.91 |
|  | Qrnn | naomit, impute\_knn, month, dow, dummy | n.hidden = 3.00  penalty = 0.10  bag= FALSE | RMSE: 325.86, R2: 0.89 |
|  | bstTree | naomit, impute\_bag, month, dow, dummy | mstop = 250.00  maxdepth = 2  nu = 0.10 | RMSE: 512.01, R2: 0.73 |

The set of good candidate models were the top 3 performing models. These were fine-tuned with the following pre-processing variations.

* Centring and scaling,
* Dimensional reduction: ica, pls,pca
* Bag imputation
* Near-Zero Variance Filter and Zero Variance Filter (Neural Networks)

From this set of good models, the best model was determined.

See Appendix C for pre-processing method trials for initial methods and the best three performing methods. If the step that was added/removed resulted in an increase of RMSE then the recipe was reverted to the original before. The final tuned recipe used for each method is shown in Table 1.

## 5. Best model

Based upon RMSE, the best model is **brnn** with an RMSE of 129.59. It is significantly better than the other top methods qrnn and svmRadial.

## 6. Performance on unseen data

The test data was predicted using the best model. It generated the predicted vs observed plot in Figure 2. There is a strong linear relationship between the predicted and observed values. There does not appear to be any significant deviations from the diagonal line. However, the test residual results show one outlier at 1.7 IQR, although not significant it foes

A picture containing text, screenshot, plot, line

Description automatically generated

**Figure 2.** Performance of the best model brnn on the test data (# observations). The diagonal line represents perfect agreement between the predicted and observed values.

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

|  |  |
| --- | --- |
| **Test Metric** | **Value** |
| RMSE | 111.076 |
| MAE | 85.124 |
| R2 | 0.987 |

## 7. Observations that do not fit the model

The model-based outliers (at IQR Multiplier of 1.7 for test, and 3.3 for train) are shown in Figure 3. There is one outlier present at 1.7 IQR in the test residuals. There are significant outliers present at 3.3 IQR in the train residuals, which are not present after 3.5 IQR. Therefore some data points cannot be explained by the model. Looking at data point tid-57344 there are missing values in Reagents B, H, and K, and a relatively low Alcohol level of 0.135, these could be the reasons why this point deviates from the predicted values.

Looking at test residual outlier tid-57646, there also appear to be three Reagents with missing values (D, F, I). There may have been too many missing values for the bag-imputation step to successfully impute these values.

A picture containing text, screenshot, diagram, receipt

Description automatically generatedA picture containing text, screenshot, diagram, receipt

Description automatically generated

**Figure 3.** Residual boxplots of best-performing model brnn.

## 8. Method description

The best model uses the method BRNN. This method seems to work well with this data because the data has groups of high correlation between predictors (complex relationships), and nonlinear associations between predictors and the target, which can be handled by BRNN [1].

BRNN Is a form of Artificial Neural Network (ANN) that mimics how the human brain functions by making connections between neurons and using weights. There is a problem with ANNs in that they can overfit the data so are not generalisable on future data. BRNN was developed to alleviate this problem using Bayesian principles to drop certain weights, to minimise the loss function and therefore the model's capacity [2].

BRNN is useful for problems with a small number of observations and a large number of predictors. BRNN works by having the input layer with input variables, which feed into the hidden layer with a certain number of nodes (here 5 nodes are determined to be the best). Weights are learned for each connection from the input node to the hidden layer node. Hidden layer nodes are finally connected to an output layer of a single node (with learned weights). It is a two-layer neural network, as can be seen in Figure 4 [3]. The Bayesian part acts to regularise (shrink the coefficients) to constrain the size of the network connection strengths [4].

## A typical two layer neural network. Input layer does not count as the... | Download Scientific Diagram

**Figure 4.** Diagram of two-layer neural network.

## 9. Transparency

The best model, **brnn**, is not particularly transparent. It would not meet the criterion for being able to explain how the important variables affect the outcome of the model. This is because the Bayesian approach involves estimating a posterior distribution over the weights rather than obtaining explicit rules or coefficients.

To achieve transparency we would need to revert to the model **pls**. This was subjected to some pre-processing fine-tuning giving a final RMSE of 223.46.

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

|  |  |
| --- | --- |
| **Test Metric** | **Value** |
| RMSE | 178.140 |
| MAE | 131.235 |
| R2 | 0.967 |

## 10. References

1. Kayri, Murat. (2016). Predictive Abilities of Bayesian Regularization and Levenberg–Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data. Mathematical and Computational Applications. 21. 1-11. 10.3390/mca21020020.
2. H. Okut, ‘Bayesian Regularized Neural Networks for Small n Big p Data’, Artificial Neural Networks - Models and Applications. InTech, Oct. 19, 2016. doi: 10.5772/63256.
3. Rodriguez P., Package ‘brnn’, Bayesian Regularization for Feed-Forward Neural Networks. CRAN, 2022-05-16. <https://cran.r-project.org/web/packages/brnn/brnn.pdf>
4. Gianola, D., Okut, H., Weigel, K. A., & Rosa, G. J. (2011). Predicting complex quantitative traits with Bayesian neural networks: a case study with Jersey cows and wheat. *BMC genetics*, *12*, 87. https://doi.org/10.1186/1471-2156-12-87

## 10. Appendix

### Appendix A

A picture containing diagram, line, text, map

Description automatically generated

### Appendix B

**Table 1.** Omitted methods that were trialled with results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Model | Processing steps | Hyper-parameters | Resampled Performance |
|  | neuralnet | naomit, impute\_knn, month, dow, dummy | layer1 (#Hidden Units in Layer 1)  layer2 (#Hidden Units in Layer 2)  layer3 (#Hidden Units in Layer 3) | RMSE: 965.57 |
|  | Mlp | “ | size | RMSE: 1136 |
|  | dnn | “ |  | RMSE: 985.48 |
|  | Logreg | “ |  | Crashes error |
|  | Simpls | “ |  | RMSE: 657.35 |
|  | Treebag | “ |  | RMSE: 633.73 |
|  | gaussprPoly | “ |  | Crashes error |
|  | kernelpls | “ |  | RMSE: 657.35 |
|  | svmPoly | “ |  | Crashes error |
|  | svmLinear | “ |  | RMSE: 506.97 |
|  | DENFIS | “ |  | Too long >1hr |

### Appendix C

The ordering of steps followed the Recipe package documentation on ordering: (<https://recipes.tidymodels.org/articles/Ordering.html>).

1. **GLMNET –** starting (lowest) RMSE: 486.60

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Comment** | **Result (RMSE)** |
| -Dummy | This [page](https://parsnip.tidymodels.org/reference/details_logistic_reg_glmnet.html) mentions that Factor/categorical predictors need to be converted to numeric values (e.g., dummy or indicator variables) for this engine. | Crashes |
| +bag -knn |  | RMSE: 478.37 |
| +yeoJohnson transform |  | RMSE: 481.17 |
| +poly |  | Crashes |
| +centre +scale |  | RMSE: 518.52 |

Redoing this model with bag imputation on an 80% split resulted in higher RMSE, so the recipe was left as original: naomit, impute\_knn, month, dow, dummy.

**Final RMSE: 486.60**

1. **PLS – starting RMSE:** 657.35

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Recipe** | **Result (RMSE)** |
| +pca | naomit, impute\_knn, pca, month, dow, dummy | 722.87 |
| +pls | naomit, impute\_knn, pls, month, dow, dummy | 492.08 |
| +ica | naomit impute\_knn ica, month dow dummy | Crashes |
| +impute\_bag | naomit,impute\_bag, pls, month, ow, dummy | 521.83 |
| +interact | naomit, ipute\_knn, ps, month, do dummy, interact | 349.92 |
| +centre | naomit, impte\_knn, pls month, dow dummy, interact, centre | 349.92 |
| +scale | naomit, imput\_knn, pls, month, dow, dummy interact, centre, scale | 9772.76 |

Final recipe: "naomit", "impute\_knn", "pls", "month", "dow", "dummy", "interact"

**Final RMSE: 22346**

1. **Rpart –** starting RMSE**:** 774.57

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Recipe** | **Result (RMSE)** |
| +bag | naomit, imute\_bag, moth, dow, dummy | 844.20 |
| +poly | naomit, impute\_knn, poly month, dow, dummy | 849.85 |
| +pca | naomit, impute\_knn, pca month, dow, dummy | 923.04 |
| +centre | naomit, impute\_knn month, dow dummy, centre | 844.99 |
| +scale | naomit, impute\_knn month,o w, dummy, centre, scale | 844.99 |

No improvements to the model. Removing dow, month, and dummy increased the error by ~0.1. Original recipe retained: "naomit", "impute\_knn", "month", "dow", "dummy"

**Final RMSE: 774.57**

1. **Brnn** – starting RMSE: 174.87

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Recipe** | **Result (RMSE)** |
| +pca | naomit, impute\_knn, pca, month, dow, dummy | 348.77 |
| +pls | naomit, impute\_knn, pls, month, dow, dummy | 317.19 |
| +ica | naomit, impute\_knn, ica month, dow, dummy | Crashes |
| +Bag -knn | naomit, impute\_bag, month, dow, dummy | 128.63 |
| +nzv (no rec. position for filters) | naomit, impute\_bag, month, dow, dummy, nzv | **127.98** |
| +zv | naomit, impute\_bag, month, dow, dummy, nzv, zv | 128.01 |
| +centre | naomit, impute\_bag, month, dow, dummy, nzv, center | 129.59 |
| +scale | naomit, impute\_bag, month, dow, dummy, nzv, center, scale | 129.60 |
| +interact | naomit, impute\_bag, month, dow, dummy, interact, nzv, center, scale | 132.09 |

Tuned final recipe: naomit, impute\_bag, month, dow, dummy, nzv.

Zero variance, scale and centre increased the RMSE (although slightly), this is in contrary to other research where scaling and normalisation was used.

*Sensor data reconstruction using bidirectional recurrent neural network with application to bridge monitoring,*

*Advanced Engineering Informatics, Volume 42, 2019.*

**Final RMSE: 127.98**

1. **Qrnn** –starting RMSE: 325.86

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Recipe** | **Result (RMSE)** |
| +impute\_bag | naomit, impute\_bag, month, dow, dummy | 433.02 |
| +pca | naomit, impute\_bag, pca, month, dow, dummy | 554.86 |

NOTE: Qrnn modelling took too long to model (>1 hour) so only some recipe steps were tried.

Tuned final recipe: initial recipe (no improvements)

**Final RMSE: 325.86**

1. **BstTree** – starting RMSE: 513.62

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Recipe** | **Result (RMSE)** |
| +pca | naomit, impute\_knn, pca, month, dow, dummy | 703.28 |
| +pls | naomit, impute\_knn, pls, month, dow, dummy | 539.08 |
| +ica | naomit, impute\_knn, ica, month, dow, dummy | crashes |
| +impute\_bag | naomit, impute\_bag, month, dow, dummy | 511.83 |
| +interact | naomit, impute\_knn, month, dow, dummy, interact | 512.95 |
| +centre +scale | naomit, impute\_knn, month, dow, dummy, centre, scale | 512.01 |

NOTE: BstTree modelling took too long to model using impute\_bag (>1 hour) so only some further recipe steps were tried.

Tuned final recipe: naomit , impute\_bag , month , dow , dummy

**Final RMSE: 511.83**

1. **SvmRadial** – starting RMSE: 442.38

|  |  |  |
| --- | --- | --- |
| **Step added(+)/removed(-)** | **Recipe** | **Result (RMSE)** |
| +pca | naomit, impute\_knn, pca, month, dow, dummy | 636.48 |
| +pls | naomit, impute\_knn, pls, month, dow, dummy | 643.41 |
| +ica | naomit, impute\_knn, ica, month, dow, dummy | crashes |
| +impute\_bag | naomit, impute\_bag, pls, month, dow, dummy | 449.67 |
| +interact | naomit, impute\_knn, month dow, dummy, interact | 907.89 |
| +centre +scale | naomit, impute\_knn, month dow, dummy, interact, centre, scale | 299.41 |

Tuned final recipe: naomit, impute\_knn, month, dow, dummy, interact, centre, scale

**Final RMSE:** **299.27**