ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Modeling Mice Down Syndrome through Protein Expression: A Decision Tree based Approach

Kamath RS[1]* and Kamat RK[2].

[1]Department of Computer Studies, Chhatrapati Shahu Institute of Business Education and Research, University Road, Kolhapur 416004, Maharashtra, India.
[2]Department of Electronics, Shivaji University, Kolhapur – 416 004, Maharashtra, India.

ABSTRACT

We report Decision Tree (DT) modeling of expression levels of proteins critical to learning in a mouse model of Down syndrome and delivered detectable signals in the nuclear fraction of the cortex. Dataset employed in the present study comprises eight classes of mice are described based on features such as 77 protein expression levels, genotype, behavior and treatment. Present research aims at deriving DT model for classification of mice protein expressions. Decision tree model is one of the most common data mining models can be used for classification and predictive analytics. The reported investigation depicts optimum decision tree architecture achieved by tuning parameters such as Min split, Min bucket, Max depth and Complexity. DT model, thus derived is easy to understand and entails recursive partitioning approach implemented in the *"rpart"* package. Moreover the performance of the model is evaluated with reference Mean Square Error (MSE) estimate of error rate.

Keywords: Decision Trees, Protein Expression, Down syndrome, Random Forest, Classification.

Corresponding author

ISSN: 0975-8585

INTRODUCTION

Mouse models are a standard tool in the investigation of numerous human ailments, in fundamental exploration and in preclinical evaluation of potential therapeutics. Their utilization, in sub-atomic, cell and behavioral trials, can give understanding into the ordinary elements of a quality, how these are modified in ailment and how they add to a disease process, and in addition data on medication activity, adequacy and reactions. Down disorder has for some time been respected by numerous in both basic and clinical exploration as excessively complex a test for successful pharmacological mediation. In our previous paper in this journal [1] we explored classification model for mice protein expression levels by employing Random forest technique, where the classification is done by constructing a large number of decision trees at training time in which each individual tree over fitted the data and the randomness is in the selection of both observations and choice of variables for partitioning the dataset [1]. A random forest is a collection of unpruned decision trees which are used to model very large datasets. The algorithm generated multiple classification trees, and the final classification result is voted among all the trees in the forest [1]. Decision tree is a most widely deployed machine learning based data mining model builder. Its attraction lies in the simplicity of the resulting model, where a decision tree is quite easy to view and explain [8]. This algorithm works based on divide and conquer principle and thus follows recursive partitioning. It explores the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical.

Down syndrome is one of the most common genetic innate causes of learning deficits [4]. There are barely any pharmacotherapies available for learning deficits in DS. Presently, protein expression modeling is also turning into an incontestably supportive strategy in microbial cell factories [1]. Protein expression modeling has been reported by number of researchers in the literature. Centeno et al presented an introduction to comparative modeling with special emphasis on the basic concepts, opportunities and challenges of protein structure prediction [5]. Alireza has depicted collection of Neural Networks to solve class imbalance problem of prediction of secondary protein structure [6]. Benuskova et al have revealed a methodology for using computational neurogenetic modeling to bring new original insights into how genes control the dynamics of brain neural networks [7].

In the backdrop of the research endeavors portrayed above, to the best of our knowledge there are no instances in the literature regarding design of decision tree model for classifying mice protein expressions. This algorithm handles very efficiently conditional information, subdividing the space into sub-spaces that are handled individually. In the present investigation, we demonstrate the DT modeling of 77 proteins expression levels measured in the cerebral cortex of 8 classes of control and Down syndrome mice exposed to context fear conditioning, a task used to assess associative learning. The reported experiment is simulated in R and Rattle [3]. The research concludes that the derived DT model efficiently classifies inputted protein samples with very less error.

The rest of paper is structured as follows; after a brief introduction, second and third sections deals with the infusing theory of mice protein expression and decision tree respectively. The fourth section outlines our computational details of the DT model with results and discussions. The conclusion at the end discusses aptness of the DT for modelling the mice protein expression.

Mice Protein Expression: Theoretical Considerations

The dataset for reported modeling contains a total of 1080 measurements per protein is taken from UCI data repository [2]. It consists of the expression levels of 77 proteins modifications that produced detectable signals in the nuclear fraction of the cortex. The eight classes of mice are described based on features such as genotype, behavior and treatment. Table 1 lists set of mice classes and corresponding number of observations in the dataset.

ISSN: 0975-8585

Table 1: Mice protein class details

Mice Protein Class	No. of Observations
c-CS-s	135
c-CS-m	150
c-SC-s	135
c-SC-m	150
t-CS-s	105
t-CS-m	135
t-SC-s	135
t-SC-m	135

Decision Tree: Theoretical considerations

A decision tree is a set of conditions arranged in a hierarchical structure [3]. This is a classification/predictive model in which a data item is categorized by following the path of fulfilled conditions from the root of the tree till reaching a leaf. The leaf corresponds to a class label. A set of classification rules can be easily derived from decision tree.

The basic algorithm for decision tree is the greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner [8]. It explores the structure of a set of data, while developing easy to visualize decision rules for a classification tree. Algorithm given in fig. 1 explains construction of decision tree for classification.

Fig. 2 shows decision tree derived in the present investigation represents supervised learning model for mice protein expressions. The root node of decision tree tests "*behavior*" has a value *"less than 1.6"* continues down to the left side of the tree, otherwise right side of the tree. The next test down this right side of the tree is *"treatment"* value. Thus it proceeds as explained in the aforesaid algorithm. Table 2 gives details of tuning parameters varied in Rattle to obtain optimized decision tree model for mice protein expressions. Column 3 gives corresponding values derived in the present investigation.

Step 1: A subset of data taken as input and evaluate all possible splits

Step 2: The single variable is found which best splits the data into two groups. The best split decision, i.e. the split with the highest information gain, is chosen to partition the data in two subsets

Step 3: The data is separated, and then this process is applied separately to each sub-group

Step 4: The method (Step 1 to 3) is called recursively until the subgroups either reach a minimum size or until no improvement can be made.

Figure 1: Decision tree algorithm for classification

Table 2: Rattle tuning parameters for decision tree modelling

Tuning parameter	Description	Value for DT model
Min split	Minimum number of observations that must exist in a node resulting from a split before a spilt will be performed	25
Min Bucket	This is the minimum number of observations allowed in any leaf node of the decision tree	8
Max Depth	This is the maximum depth of any node of the final tree	30
Complexity	This parameter is used to control the size of the decision tree and to select optimal tree size	0.01

ISSN: 0975-8585

The performance of the model is calculated by using Mean Square Error between expected output and estimated output, given in equation (1). The Y_i represents the observed value of the i^{th} observation, where, i=1, 2,…….n and $\hat{Y}_i$ denote the predicted value of the i^{th} observation. The difference $(Y_i - \hat{Y}_i)$ is termed as an error.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \quad (1)$$

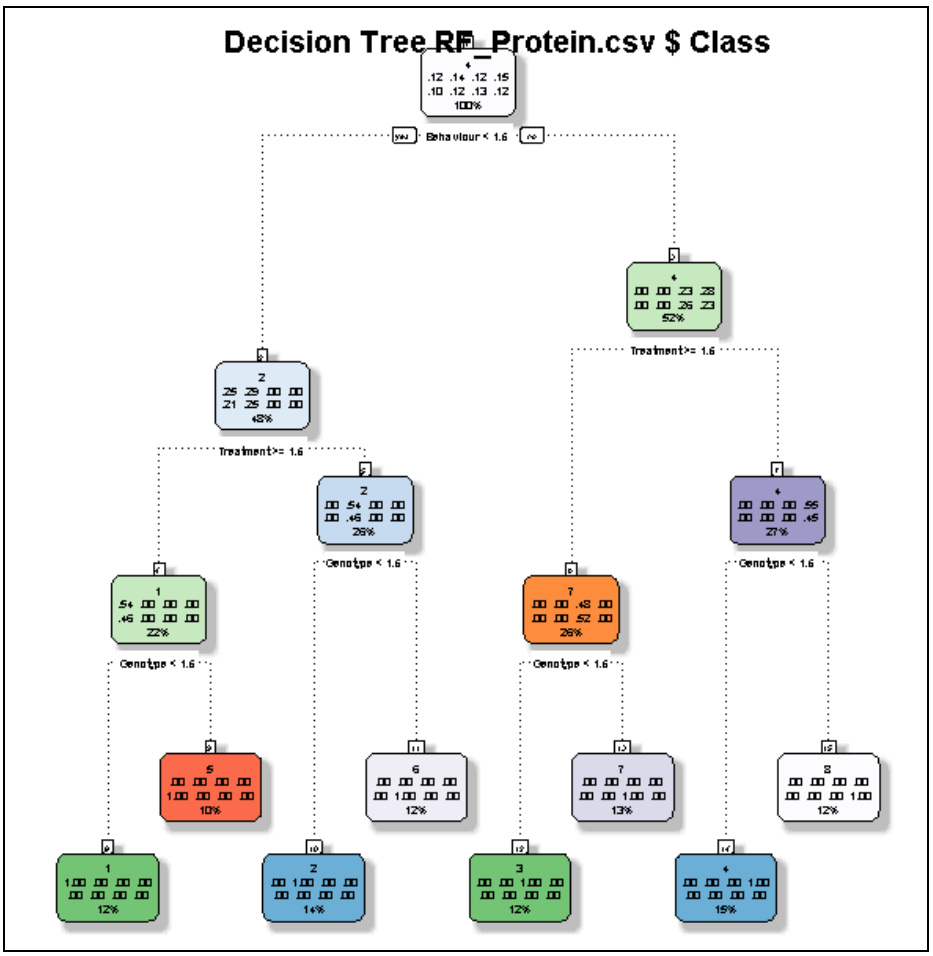

Figure 2: Decision tree model for mice protein expressions

Computational details, results and discussions

This section explores details of experiment conducted for the modeling of mice protein expressions using decision tree based approach. R and Rattle are used to analyze model structure, configuring tuning parameters to obtain optimized DT model. The model is conceived as Multi-Input Single Output. It works basically with 80 inputs viz. 77 proteins expression levels, genotype, behavior and treatment. Class to which mouse belongs to is considered as output variable. The dataset for reported modelling is taken from UCI data repository contains a total of 1080 measurements per protein is [2].

In the present investigation, model is tuned with parameters such as Min split, Min bucket, Max depth and Complexity to get optimized forest architecture. We have demonstrated DT modeling per variation in these parameters and the same is summarized in table 3. We have varied value for Min split, Min bucket, Max depth by keeping value for complexity is 0.01 constant. Performance of decision trees per variation in tuning parameters plotted and shown in fig. 3. Experiment reveals that minimum time taken for the construction of optimized decision tree is 1.83 seconds.

ISSN: 0975-8585

Table 3: Performance evaluation for *"time taken"* of DT configurations

Min Split	Min Bucket	Max Depth	Time taken (sec)
15	5	10	2.17
15	5	20	1.95
15	5	30	2.09
20	7	10	2.06
20	7	20	2.19
20	7	30	1.84
25	8	10	1.86
25	8	20	1.94
25	8	30	1.83
30	10	10	2.55
30	10	20	2.09
30	10	30	1.91
35	15	10	1.95
35	15	20	1.92
35	15	30	2.11

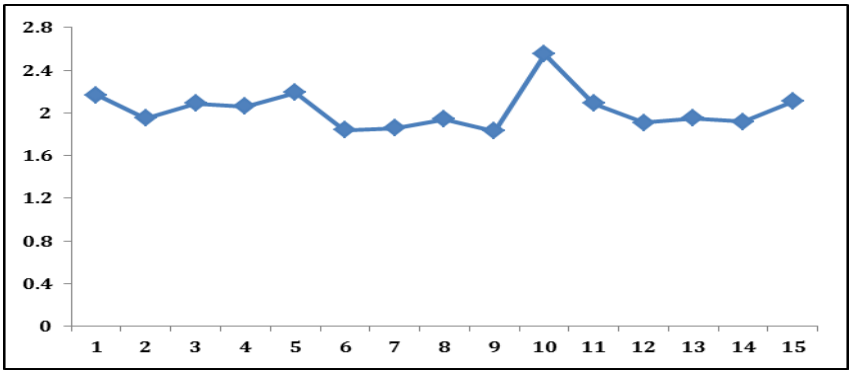

Figure 3: Performance of decision trees per variation in tuning parameters

Fig. 4 summarizes the decision tree for mice protein expressions modeling. It is the text view of resultant decision tree and also highlights the key interface widgets that need to deal with to build a tree. The tree has built to classify mice protein expressions' variable *"Class"* based on the remainder of the variables in the dataset supplied. Variables actually used in tree construction are "*Behaviour", "Treatment"* and "*Genotype"*. The decision tree shown in fig 2 translates to the rules, where each rule corresponds to one pathway through the decision tree, starting at the root node and terminating at a leaf node.

```
 1) root 756 646 4 (0.12 0.14 0.12 0.15 0.1 0.12 0.13 0.12)
   2) Behaviour< 1.5 362 256 2 (0.25 0.29 0 0 0.21 0.25 0 0)
     4) Treatment>=1.5 167  76 1 (0.54 0 0 0 0.46 0 0 0)
       8) Genotype< 1.5 91   0 1 (1 0 0 0 0 0 0 0) *
       9) Genotype>=1.5 76   0 5 (0 0 0 0 1 0 0 0) *
     5) Treatment< 1.5 195  89 2 (0 0.54 0 0 0 0.46 0 0)
      10) Genotype< 1.5 106   0 2 (0 1 0 0 0 0 0 0) *
      11) Genotype>=1.5 89   0 6 (0 0 0 0 0 1 0 0) *
   3) Behaviour>=1.5 394 284 4 (0 0 0.23 0.28 0 0 0.26 0.23)
     6) Treatment>=1.5 193  92 7 (0 0 0.48 0 0 0 0.52 0)
      12) Genotype< 1.5 92   0 3 (0 0 1 0 0 0 0 0) *
      13) Genotype>=1.5 101   0 7 (0 0 0 0 0 0 1 0) *
     7) Treatment< 1.5 201  91 4 (0 0 0 0.55 0 0 0 0.45)
      14) Genotype< 1.5 110   0 4 (0 0 0 1 0 0 0 0) *
      15) Genotype>=1.5 91   0 8 (0 0 0 0 0 0 0 1) *

Classification tree:
rpart(formula = Class ~ ., data = crs$dataset[crs$train, c(crs$input,
    crs$target)], method = "class", parms = list(split = "information"),
    control = rpart.control(minsplit = 25, minbucket = 8, cp = 0.00001,
        usesurrogate = 0, maxsurrogate = 0))

Variables actually used in tree construction:
[1] Behaviour Genotype  Treatment

Root node error: 646/756 = 0.8545

n= 756
```

Figure 4: Summary of the Decision Tree model for mice protein expressions

ISSN: 0975-8585

Thus derived optimized decision tree entails values for tuning parameters such as Min split, Min bucket, Max depth and complexity are 25, 8, 30 and 0.01 respectively. Performance evaluation of the model is summarized in Table 4. This complexity table explains iterations and associated change in the accuracy of the model as new levels are added to the tree. We are most likely interested in the cross-validated error, which is the *"xerror"* column of the table. The CP (complexity parameter) value reveals that as the tree splits into more nodes, the complexity parameter is reduced. But we also note that the cross validation error starts to increase as we further split the decision tree. This tells the algorithm to stop partitioning, as the error rate is not improving.

Table 4: Complexity Table for DT model

Level	CP	nsplit	rel error	xerror	xstd
1	0.16409	0	1.000000	1.02941	0 .013850
2	0.15635	1	0.83591	0.95975	0.016348
3	0.14241	2	0.67957	0.67183	0.021046
4	0.14087	3	0.53715	0.49845	0.021046
5	0.13777	5	0.25542	0.39319	0.020104
6	0.13777	6	0.11765	0.11765	0.020104
7	0.13777	7	0.00000	0.00000	0.00000

Thus derived DT model efficiently classifies new protein samples with very less error. We have tested model with known protein samples. Fig. 5 shows the result obtained in terms of error matrix by applying the test dataset on the derived DT model. Result concludes that DT modeling is a suitable approach since the resulting analysis is much more accurate and precise.

```
Error matrix for the Decision Tree model on RF_Protein_testing.csv (counts):

      Predicted
Actual 1 2 3 4 5 6 7 8
     1 5 0 0 0 0 0 0 0
     2 0 5 0 0 0 0 0 0
     3 0 0 5 0 0 0 0 0
     4 0 0 0 5 0 0 0 0
     5 0 0 0 0 5 0 0 0
     6 0 0 0 0 0 5 0 0
     7 0 0 0 0 0 0 5 0
     8 0 0 0 0 0 0 0 5

Error matrix for the Decision Tree model on RF_Protein_testing.csv (proportions):

      Predicted
Actual    1    2    3    4    5    6    7    8 Error
     1 0.12 0.00 0.00 0.00 0.00 0.00 0.00 0.00     0
     2 0.00 0.12 0.00 0.00 0.00 0.00 0.00 0.00     0
     3 0.00 0.00 0.12 0.00 0.00 0.00 0.00 0.00     0
     4 0.00 0.00 0.00 0.12 0.00 0.00 0.00 0.00     0
     5 0.00 0.00 0.00 0.00 0.12 0.00 0.00 0.00     0
     6 0.00 0.00 0.00 0.00 0.00 0.12 0.00 0.00     0
     7 0.00 0.00 0.00 0.00 0.00 0.00 0.12 0.00     0
     8 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.12     0
```

Figure 5: Execution result of DT model on test dataset

CONCLUSION

In the present paper, we have reported decision tree (DT) modelling of mice protein expressions. Dataset employed in the present study comprises eight classes of mice are described based on features such as 77 protein expression levels, genotype, behavior and treatment. Present research concludes by deriving DT model for classification of mice protein expressions. The reported investigation depicts optimum decision tree architecture achieved by varying tuning parameters. DT model, thus derived is easy to understand and entails recursive partitioning approach implemented in the "*rpart*" package. Result concludes that DT prediction is a suitable approach since the resulting analysis is much more accurate and precise.

ISSN: 0975-8585

REFERENCES

[1] Kamath RS, Dongale TD, Pawar P, Kamat RK. Research journal of Pharmaceutical, Biological and Chemical Sciences 2016; 7(4): 830-836.
[2] Higuera C, Gardiner KJ, Cios KJ. PLoS ONE 2015; 10(6): e0129126.
[3] Kamath R, Kamat R. Educational Data Mining with R and Rattle, River Publishers, Netherland, 2016, pp. 55-58.
[4] Irving C, Basu A, Richmond S, Burn J, Wren C. Eur J Hum Genet 2008; 16(11): 1336–40.
[5] Centeno N, Planas-Iglesias J, Oliva B. Microb Cell Fact 2005; 4(1): 20.
[6] Alirezaee M. Int. J. Artificial Intelligence & Applications 2012; 3(6): 9-20.
[7] Benuskova L, Jain V, Wysoski S, Kasabov N. Int. J. Neur. Syst. 2006; 16(03): 215-226.
[8] Graham W. Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery. Springer, UK, 2011, pp. 205-244.