Student ID: s3726377

Student Name: Ji Hei Kim

I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in my submission. I will show I agree to this honor code by typing "Yes": *Yes*.

Data Modelling to identify Proteins critical in learning for Down Syndrome Mice

Ji Hei Kim[1]

[1]Practical Data Science, Master of Information Technology, RMIT University, Australia
s3726377@student.rmit.edu.au

10th June 2020

Table of Contents

Abstract

The aim of this report was to observe the expression levels of proteins critical to learning in a mouse model of Down syndrome [1]. Expression levels of 77 proteins were measured in samples of mice with Down syndrome and without Down syndrome [1]. Mice protein expression data set from UCI repository was used to classify using some of the common classification models such as K- Nearest Neighbour and Decision Tree and its accuracy was compared using 10-fold validation. To achieve optimal modelling, several tuning of parameters tuning was done obtaining best scores from validation and confusion matrix. Performance of KNN was higher than Decision Tree where accuracy score of KNN was 0.99.

Introduction

Down syndrome(DS) is one of the most common genetic root causes of learning deficits [2]. Despite high number of incidences, there are no pharmacotherapies available for learning deficits in DS. Because of its complexity, it is reasonable to factor "downstream, integrated, consequences of all Hsa21 genes that are overexpressed, that is to observe perturbations in pathways that are critical to learning and memory and then to consider drugs that would correct the observed perturbations" [2]. For preclinical evaluation of drug effects, data set of expression levels of 77 proteins measured in the cerebral cortex of 8 classes of control and Down syndrome mice exposed to context fear conditioning was created [2].

To evaluate some of complex challenges, protein expression modelling has been used to help analysis of biological data [3]. Although both supervised and unsupervised learning have been knowledge to be helpful, this report focuses on supervised learning methods. KNN and Decision Tree modelling was created to predict 8 classes of mice based on protein expression levels. With success of creating the model, it can identify proteins that are outstanding in classifying and concludes that some proteins are discriminate among classes.

Methodology

1. Dataset/ Data preparation
   Dataset was downloaded from UCI repository webpage where it data contains 77 protein expression levels that produced detectable signals in the nuclear fraction of cortex from 38 control mice and 34 trisomic mice [1]. Original data was in excel format (.xls) which was converted into csv format from Microsoft Excel before importing it to Python. Dataset was first explored to identify any possible errors and missing values. It was identified multiple mice with missing values in protein expression levels. To handle missing values, the mean value of same class was used to fill in rather than eliminating the entire mouse from the data.
   There were also large number of outliers within expression levels however this has not been handled separately as it as assumed that all values are unique and meaning.
   Mouse ID was also redefined to separate between mouse number (72 unique ID numbers) and its measurement registration number (1-15).

2. Data exploration (Analytical techniques)
   Before building a data modelling, data itself was closely explored. Data was explored by analysing individual column as well as comparing relationships between features (columns addressing plausible hypothesis.
   Firstly, to explore each column, inbuilt describe function was used to obtain general statistical figures such as mean, min and max values for each columns. Boxplot was used to visualise these statistical values for each protein expression levels. To look at 77 proteins more closely, proteins with similar size of expression levels was grouped together so that they are scaled appropriately. Each boxplot set contains 7 proteins with similar range of expression levels. Boxplot is beneficial to glance look at distribution of variables, giving indication of the data's symmetry and possible skewness [10]. Also it has a high advantage of spotting outliers and allows to compare to other variables (proteins). Another key feature is class column described by based on features such as genotype, behaviour and treatment. It was observed to see the total count of 8 different classes. This has been visualised by using bar graph to show each class in a frequency distribution and overall trends.
   Secondly relationship between each protein expression level against 8 classes was observed. Relationship of 77 proteins against different classes was analysed and again boxplot was used to observe for any outliers, distributions etc. To explore further, each protein levels against genotype, behaviour and treatment was compared using boxplot. Than some of proteins that had significant features were selected, with assumption that these proteins would have significant impact on certain genotype, behaviour and treatment.

3. Data Modelling
   a. Data Selection
      Two main classification algorithms, KNN and Decision Tree, was used for classification of mice protein expressions.

K-Nearest Neighbour (KNN)
KNN algorithm under supervised learning has an advantage of being simple and is used in many areas including medical & scientific data analysis. The KNN algorithm is based on the assumption that the new sample will include the class that has the closest properties to it [12]. The KNN algorithm processes with following steps: 1. Calculate the distance between the new sample and all the samples in the training set using distance functions e.g., Euclidean or Manhattan. 2. From the training set, the closest k samples to the new sample are taken. 3. The new sample is allocated to the highest class among the nearest k neighbours.

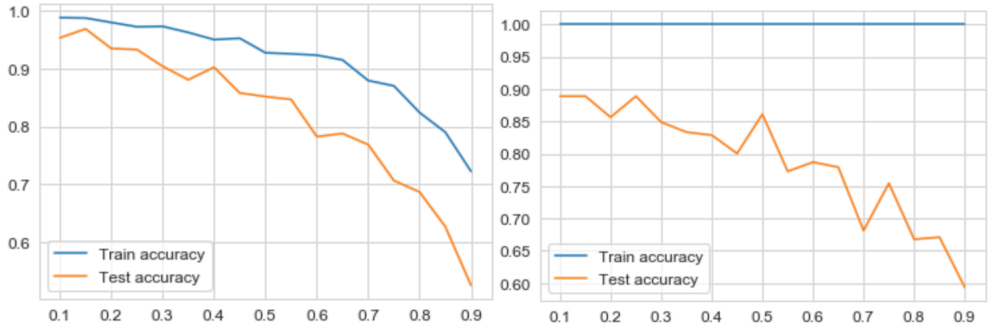
Decision Tree
Decision tree model is one of the most common algorithm that can be used for classification analytics. A decision tree is a set of conditions arranged in a flowchart-like structure [12]. Data item is categorized by following the path, which represents the classification rules, with fulfilled conditions from the root of the tree till reaching a leaf node. The leaf represents to a class label. The basic algorithm for decision tree is it split the sample into two or more homogeneous sets based on the significance of input variables [12]. It explores the structure of a set of data, while creating easy to visualize decision rules for a classification tree.

b. Feature Selection
For KNN, Hill Climbing technique was used for feature selection. The basic algorithm of this technique is that it will start from first feature from the randomized order of features. Existing feature plus 'selected' feature will be kept if it gives higher accuracy score [9]. Using this algorithm, 54 proteins have been selected.
For Decision Tree, feature selection was unnecessary as itself has the ability to split the features that are most homogeneous than others by using Gini index algorithm.

c. Determination of test size, parameter tuning
To design best performance model, tuning parameters are essential and critical [12]. Thus numerous training model was done by selecting appropriate values for each parameter in the model.

First to determine appropriate test size for each of the model train/test split score was calculated to find the most appropriate size. The highest accuracy with minimum difference from train and test data was chosen. As a result, for KNN 0.15 of data was set as test size and for Decision Tree, 0.30 of data was used for test data.

Train/Test split score for KNN vs Decision Tree



For KNN, choosing the appropriate number of neighbours, weight function used, and distance function was critical to eliminate noise. First step is to determine the value of K. When K is small, it creates restraining region of a given prediction and forcing classifier to unseen overall distribution. Smaller K value provides the most flexible fit, which means it will be less bias but high variance. On the other hand, a higher K averages more values in each prediction and hence is more resilient to outliers suppressing the effect of noise. Higher K values will have smoother classification boundaries which leads to lower variance but increased bias. In this dataset, I have used three numbers, K = 3, 5 & 10. Weight is another parameter that can be tunned either selecting 'uniform' or 'distance'. Lastly calculating distance using Euclidean distance or Manhattan distance needs to be determined. This is dependant on the data but small p value is recommended for high dimensional vectors.
From the 10-fold validation, setting K=3 and others to default had the highest average score 0.9787 which assumed over-fitting. From the data exploration it was seen that many outliers exist, thus higher number of neighbour was chosen. After the tunning, *K =10, weights='distance', p=1* was selected to have the highest average score.
Confusion matrix was performed to see how well the prediction was made. Than Hill Climbing technique was used for feature selection, i.e. to select the best possible proteins. 54 proteins were chosen to give the best possible result. 10-Fold validation gave average score of 0.9852 which resulted slightly higher than modelling with 77 proteins (0.9796).

Optimum Decision Tree is performed by tuning parameters such as depth of the tree, min number of split, minimum sample for the leaf, max number of terminal leaf and maximum features. I have demonstrated DT modelling per variation in these with keeping value for max_leaf_nodes=None, max_features = 'auto' constant. These variables are used to control overfitting. Too high min sample for a node split leads to under-fitting, similarly min samples for a terminal node generally require lower values for imbalanced class problems. Max depth of the tree is also important to control overfitting as too higher values will adopt very specific to a particular sample. Firstly, validation was done with setting no constraint on tree size which resulted in relatively high average score of 0.8639. However, this is result of overfitting. Setting max_depth too low resulted decrease in score. As the size of the data was not too big, max_depth limit was removed. After the tuning, *min_samples_split=6, min_samples_leaf=2, max_leaf_nodes=None, max_features = 'auto'* resulted in highest average score of 0.7953.

d. Validation using k-fold validation

Parameters of KNN and Decision Tree was tuned using 10-fold cross validation. Test data was divided into 10 parts and each parts was used once while using the others as a training data set. Mean value of 10 scores was calculated to obtain average score of validation. K=10 has been chosen as it is a number that is commonly used, and recommended []. Model with higher score represents that model has good predictive power and better generalization of unseen data.

Results

A. Explore single column

First, each protein expression level was explored to understand their characters. The protein that had highest expression level was 'NR2A_N' with value of 8.482553. The lowest protein level for this protein was 1.737540. 'pCAMKII_N' had second highest expression level of 7.46407 closely followed by 'pELK_N' which had expression level of 6.113347.

TABLE II.
7 PROTEINS WITH HIGHEST EXPRESSION LEVEL

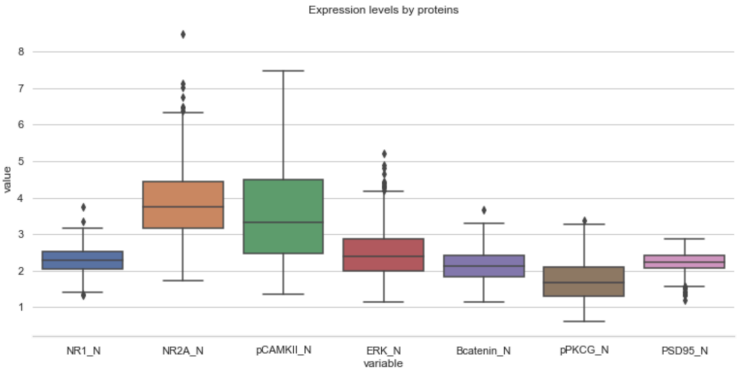
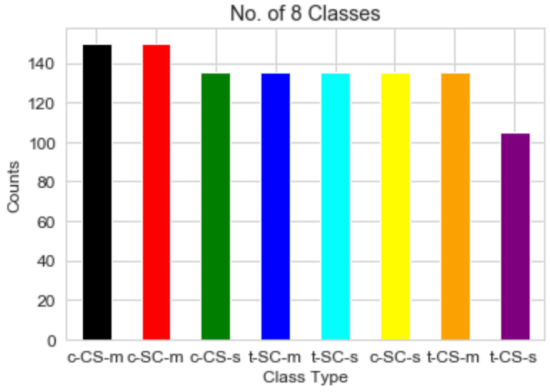


Table I shows 7 proteins grouped together with highest range of expression level. In contrast, protein 'RRP_N' had the lowest expression level of -0.062008. From the observation this protein was the only protein which had negative expression level. The highest expression level for this protein was 0.612377. Second lowest expression level was from protein 'JNK_N' with a value of 0.046298. Below boxplot shows 7 proteins with smallest range of expression level. Lastly the average value of expression levels for each protein was explored. Protein 'NR2A_N' which had the highest expression level had the highest mean value of 3.843934. 'pCAMKII_N' ranked second highest mean value of expression level with its value of 3.537109.

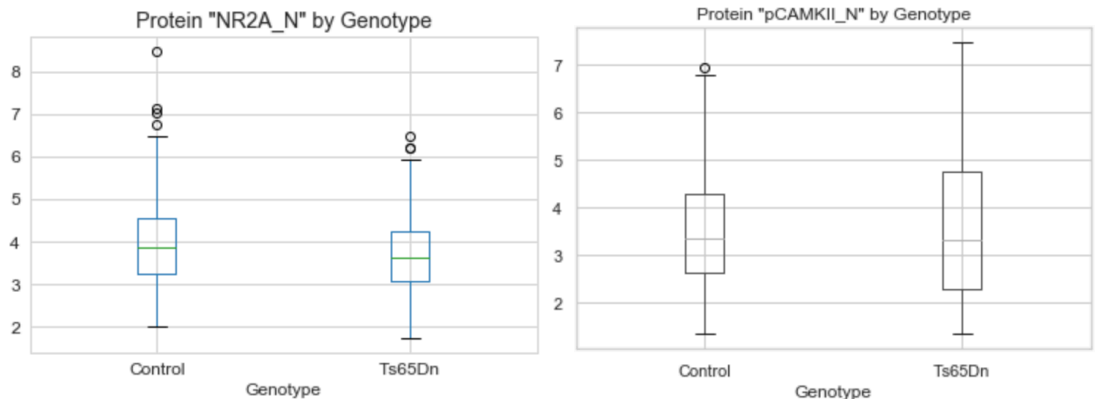
TABLE II.
EIGHT CLASSES OF MICE



Column ‘class’ was also analysed to see the number of classes among dataset. The highest count is class ‘c-SC-m’ and ‘c-CS-m’ with 150 each. The least was class ‘t-CS-s’ which had 105 out of all the classes.

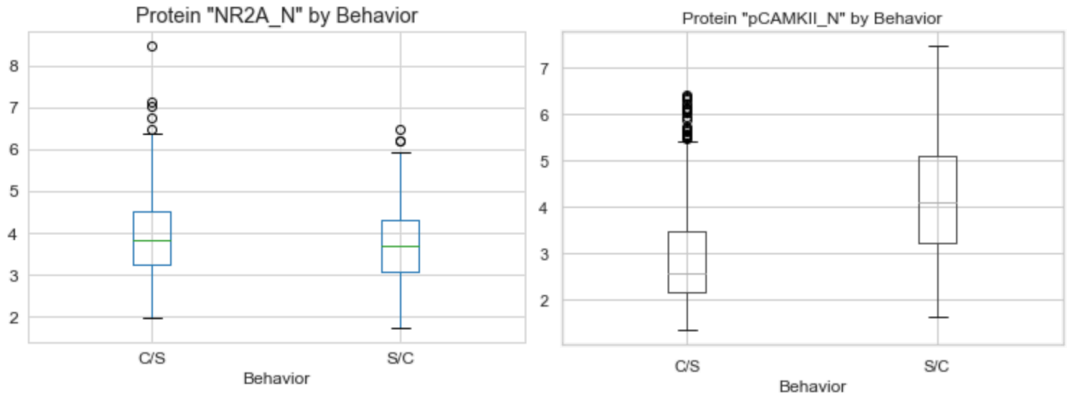
B. Explore the relationship between pairs of attributes (*Please note that pair of attributes shares common hypothesis)*
Hypothesis: Proteins with high expression level will present significance with mice that are controlled
First the relationship between protein ‘NR2A_N’ & ‘pCAMKII_N’ that has the highest expression level and genotype has been observed. From the graph it can be found that the highest expression level is present in control mice. The average expression level of the ‘NR2A_N’ did not differ too much between control (3.9847) and trisomy mice (3.6857). Likewise, protein ‘pCAMKII_N’ presented not a major different in whether it was control (3.5052) or trisomy mice (3.5730).

TABLE III.
PROTEINS WITH HIGHEST EXP LEVEL BY GENOTYPE



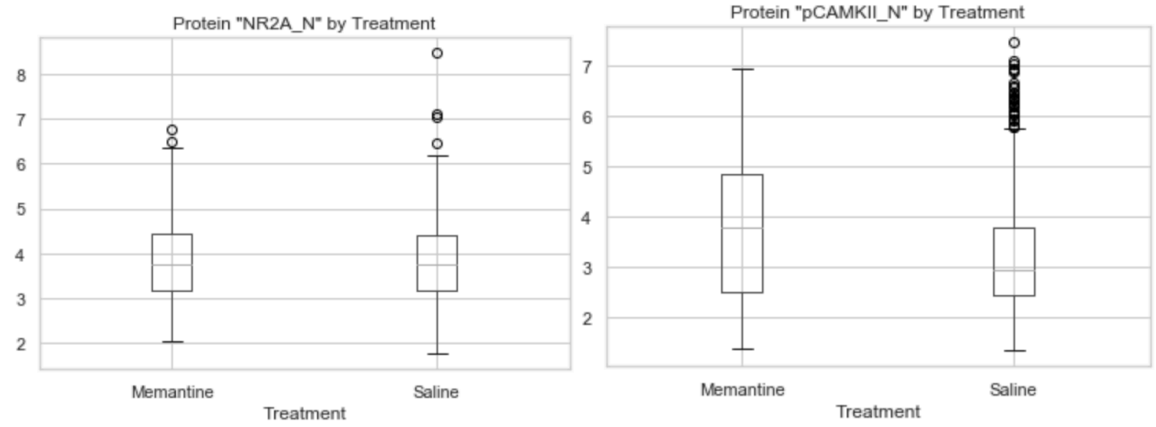
Hypothesis: Proteins with high expression level is distinguished in mice that have been stimulated to learn (context-shock)
Next, its relationship with different behaviour of mice was explored. Result shows that ‘NR2A_N’ highest outlier value of expression level is from C/S mice. Despite this, the mean value between C/S (3.9515) and S/C (3.7416) mice has minimal difference. For protein ‘pCAMKII_N’ it shows significantly higher value with mice who had not been stimulated to learn (S/C) with its mean value of 4.1539. Compared to 2.8886 for C/S mice.

TABLE IV.
PROTEINS WITH HIGHEST EXP LEVEL BY BEHAVIOR



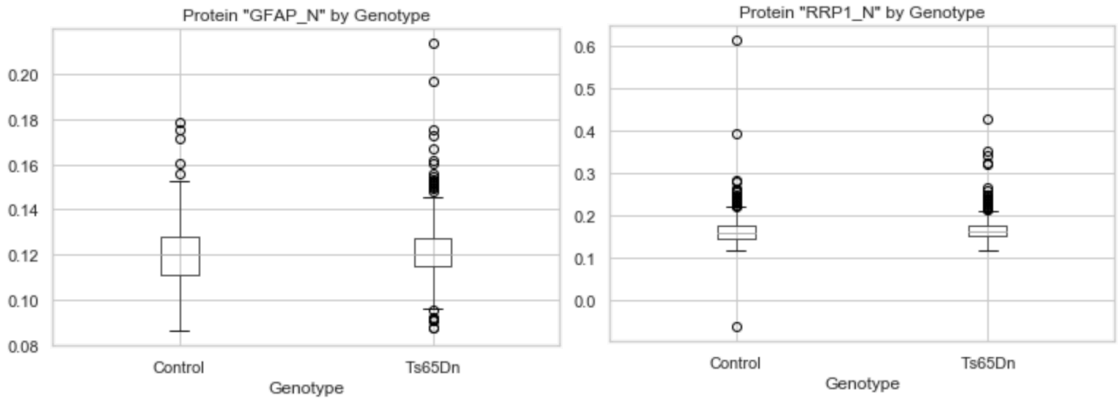
Hypothesis: Proteins with high expression level is distinguished for mice injected with Memantine.
Finally, difference in the treatment and the protein expression level was compared. The level of ‘NR2A_N’ protein expression is similar in both Memantine (3.8494) or Saline (3.8377) treated mice, where the outlier is present on the saline treated mice. For ‘pCAMKII_N’ protein, its mean expression level is higher in Memantine mice and shows large range of outliers in Saline treated mice.

TABLE V.
PROTEINS WITH HIGHEST EXP LEVEL BY TREATMENT



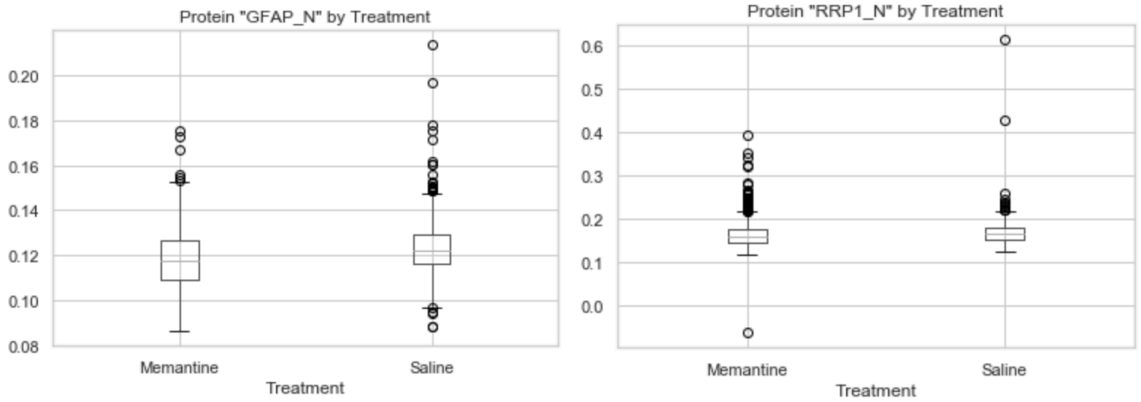
Hypothesis: Proteins with lower expression level will present no difference between genotype.
Protein 'GFAP_N' which had the lowest range of expression level and 'RRP1_N' which had the negative expression level was observed. The average expression level of 'GFAP_N' was nearly the same, for control mice value was 0.1202 and for trisomy mice 0.1216. it was also apparent that trisomy mice had large number of outliers. There is also no significant difference in average expression value of 'RRP1_N' despite the negative value present in control mice.

TABLE VI.
PROTEINS WITH LOWEST EXP LEVEL BY GENOTYPE



Hypothesis: Proteins with lower expression level is is same between different treatment used.

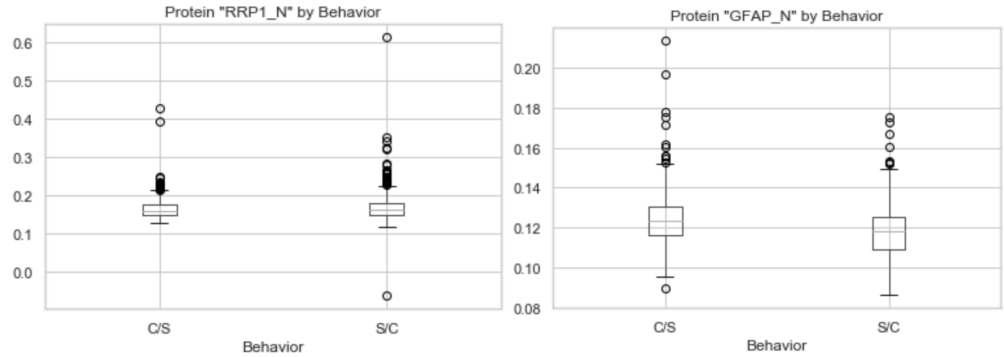
TABLE VII.
PROTEINS WITH LOWEST EXP LEVEL BY TREATMENT



From the above graph, it is visible that there is large range of outliers present in Saline treated mice in protein 'GFAP_MN'. The mean value of Memantine treated mice is 0.1646 and for Saline its mean value is 0.1689. For protein 'RRP1_N' the negative value exists in Memantine treated mice. The average expression level does not present significant difference. Average expression level for Memantine mice is 0.1645 and 0.1689 for Saline treated mice.

Hypothesis: Proteins with lower expression level shows no contrast between different behaviour of mice.
After analysis, it is found that 'RRP1_N' shares similar values between C/S and S/C mice.

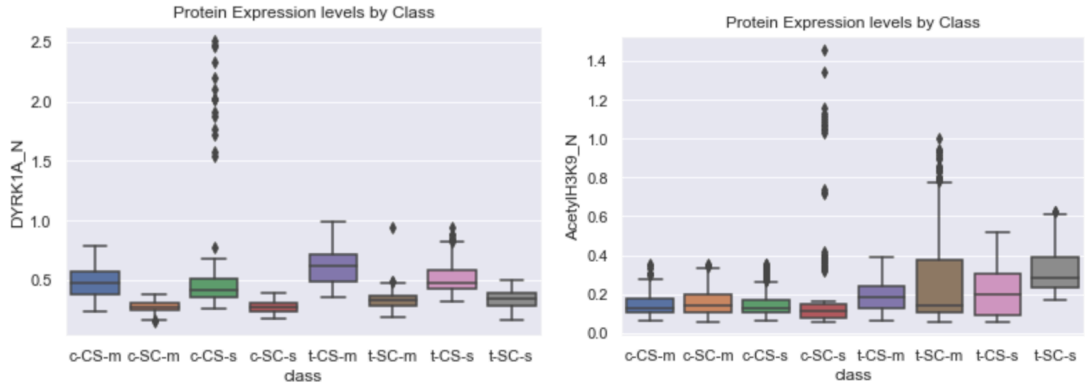
TABLE VIII.
PROTEINS WITH LOWEST EXP LEVEL BY BEHAVIOR



However, it is observed that S/C mice presents two significant outliers in two extreme ranges. Regardless of these two outliers, the average expression level for C/S (0.1653) and S/C (0.1679) is somewhat similar. Protein 'GRAP_N' shows outliers in C/S mice and has slightly higher mean value than S/C mice. Average expression level for C/S mice is 0.1242 and expression level of S/C mice is 0.1177.

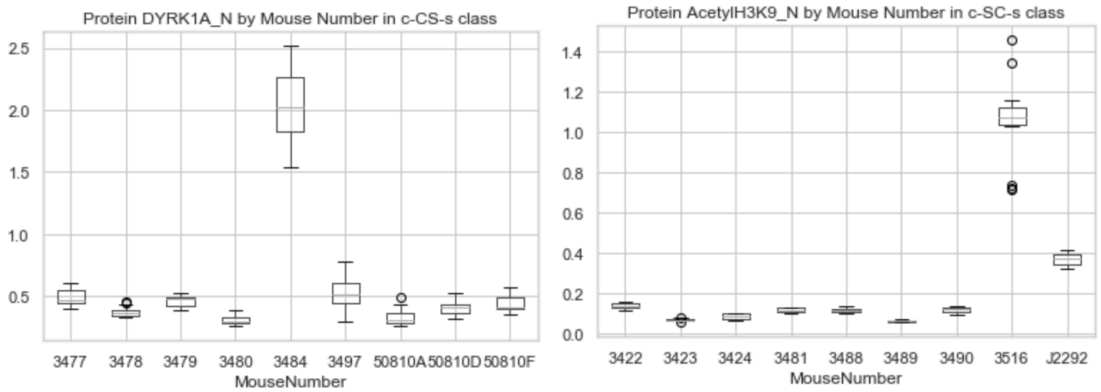
Hypothesis: Proteins with large range of outliers will be discriminant between the classes. Two different proteins has been selected to address the same hypothesis.
It was explored to see whether proteins with significantly high range of outliers are distinguished between classes. Protein 'DYRK1_N' which observed to have large outliers was explored by classes. Below graph shows that the outlier is mostly visible to class 'c-CS-s' mice.

TABLE IX.
PROTEINS WITH LARGE OUTLIER EXP LEVEL & CLASSES



Hypothesis: Proteins with large range of outliers will be visible to all mice in the same class.

TABLE X.
PROTEINS WITH LARGE OUTLIER EXP LEVEL BY MICE



It was further explored to see which mouse out of the same class had the outliers and if it was significant to particular mouse or distributed against mice. Out of 9 mice within the same class, it shows that mouse 3484 contains the most outlier expression level of 'DYRK1_N'and other 8 mice shares a similar range of expression levels. Same pattern is shown in protein 'AcetyH3K9_N' where the outlier is from mice 3516 where it shows distinctive range of values among same class of mice.

C. Modelling result

KNN

Confusion matrix below shows prediction score of 0.99 for KNN with 77 proteins and KNN with 54 proteins selected from Hill Climbing. Recall score for class ‘c-CS-S’ is 0.95, there for f1-score is 0.97. Class ‘t-CS-S’ had precision score of 0.93 and f1-score to be 0.96.

TABLE XI.
KNN CONFUSION MATRIX WITH 77 PROTEINS VS 54 PROTEINS

```
[[ 9  0  0  0  0  0  0  0]
 [ 0 18  0  0  0  1  0  0]
 [ 0  0 21  0  0  0  0  0]
 [ 0  0  0 11  0  0  0  0]
 [ 0  0  0  0 10  0  0  0]
 [ 0  0  0  0  0 13  0  0]
 [ 0  0  0  0  0  0 17  0]
 [ 0  0  0  0  0  0  0  8]]
              precision    recall  f1-score   support

      c-CS-m       1.00      1.00      1.00         9
      c-CS-s       1.00      0.95      0.97        19
      c-SC-m       1.00      1.00      1.00        21
      c-SC-s       1.00      1.00      1.00        11
      t-CS-m       1.00      1.00      1.00        10
      t-CS-s       0.93      1.00      0.96        13
      t-SC-m       1.00      1.00      1.00        17
      t-SC-s       1.00      1.00      1.00         8

    accuracy                           0.99       108
   macro avg       0.99      0.99      0.99       108
weighted avg       0.99      0.99      0.99       108
```

```
[[ 9  0  0  0  0  0  0  0]
 [ 0 18  0  0  0  1  0  0]
 [ 0  0 21  0  0  0  0  0]
 [ 0  0  0 11  0  0  0  0]
 [ 0  0  0  0 10  0  0  0]
 [ 0  0  0  0  0 13  0  0]
 [ 0  0  0  0  0  0 17  0]
 [ 0  0  0  0  0  0  0  8]]
              precision    recall  f1-score   support

      c-CS-m       1.00      1.00      1.00         9
      c-CS-s       1.00      0.95      0.97        19
      c-SC-m       1.00      1.00      1.00        21
      c-SC-s       1.00      1.00      1.00        11
      t-CS-m       1.00      1.00      1.00        10
      t-CS-s       0.93      1.00      0.96        13
      t-SC-m       1.00      1.00      1.00        17
      t-SC-s       1.00      1.00      1.00         8

    accuracy                           0.99       108
   macro avg       0.99      0.99      0.99       108
weighted avg       0.99      0.99      0.99       108
```

Deicision Tree

Confusion matrix below shows macro average score of precision 0.88 and weighted average as 0.87. Recall score is both 0.86. When classifying class ‘t-CS-m’ it had the lowest recall score of 0.69, with f1-score of 0.78. Lowest precision score was for ‘c-CS-m’ with 0.73 and f01 score to be 0.80.

TABLE XII.
DT CONFUSION MATRIX

```
[[ 8  1  0  0  0  0  0  0]
 [ 2 16  0  0  1  0  0  0]
 [ 0  0 21  0  0  0  0  0]
 [ 0  0  1 10  0  0  0  0]
 [ 0  1  0  0  8  1  0  0]
 [ 1  2  0  0  1  9  0  0]
 [ 0  0  4  0  0  0 13  0]
 [ 0  0  0  0  0  0  0  8]]
              precision    recall  f1-score   support

      c-CS-m       0.73      0.89      0.80         9
      c-CS-s       0.80      0.84      0.82        19
      c-SC-m       0.81      1.00      0.89        21
      c-SC-s       1.00      0.91      0.95        11
      t-CS-m       0.80      0.80      0.80        10
      t-CS-s       0.90      0.69      0.78        13
      t-SC-m       1.00      0.76      0.87        17
      t-SC-s       1.00      1.00      1.00         8

    accuracy                           0.86       108
   macro avg       0.88      0.86      0.86       108
weighted avg       0.87      0.86      0.86       108
```

Discussion

From the result it shows that the classification model helps to observe and identify expression levels of proteins critical to learning in Down syndrome/Control mice. Overall, the performance of KNN was more accurate than Decision Tree Using the test data of 15%, the confusion matrix showed Precision, Recall and F-1 Score to be 0.99. f1-score is important when it comes to uneven class distribution. It considers both scenario of false positives and false negatives.

However, for Decision Tree using 30% of test data, Recall & f1 score was 0.86 and weighted precision score was 0.87. Relatively high precision score over low recall score shows that there was missing a lot of positive examples but those predicted as positive were positive. It was interesting to see the ratio of correctly recognizing ‘t-SC-m’ class and ‘t-SC-S’ class was only 0.69 and 0.76 (Recall score). Which means the percentage of correctly predicted positive observations to the all observations in actual class is lower than the other classes. Based on this result, it is recommended to use KNN model

when classifying mice class based on protein expression levels. KNN having higher Accuracy, precision, recall & f1-score means the performance of this model can predict/classify better and accurately.

From KNN, using Hill Climbing algorithm 54 proteins was selected as distinctive to classify classes. For Decision Tree, Hill Climbing was not required as itself was able to features most homogeneous than others by using Gini index algorithm. Top 5 proteins: BRAF_N, pMTOR_N, pCREB_N, SOD1_N, pPKCG_N had the highest gini score, i.e. high homogeneity. Proteins BRAF_N, pMTOR_N & pPKCG were commonly selected by two different models.

Conclusion

In conclusion, K-Nearest Neighbour model gives better accuracy to classify discriminant proteins than Decision Tree. The accuracy has been validated by using 10-fold cross validation. From the Hill Climbing techniques, both KNN and Decision Tree identifies similar proteins as most distinctive proteins for learning process of mice.

Reference

1. https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression
2. Higuera C, Gardiner KJ, Cios KJ (2015) 'Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome.' PLoS ONE 10(6): e0129126. [Web Link] journal.pone.0129126
3. Ahmed MM, Dhanasekaran AR, Block A, Tong S, Costa ACS, Stasko M, et al. (2015) 'Protein Dynamics Associated with Failed and Rescued Learning in the Ts65Dn Mouse Model of Down Syndrome.' PLoS ONE 10(3): e0119491.
4. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.boxplot.html
5. https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51
6. https://machinelearningmastery.com/k-fold-cross-validation/
7. https://seaborn.pydata.org/generated/seaborn.boxplot.html
8. Tutorial04.pdf, Practical Data Science Tute/Lab 04, COSC2670, RMIT University
9. Tutorial05.pdf, Practical Data Science Tute/Lab 05, COSC2670, RMIT University
10. Tutorial06.pdf, Practical Data Science Tute/Lab 06, COSC2670, RMIT University
11. Ren, Y., 'Data Summarization: Descriptive Statistics and Visualization', Lecture notes, COSC2670, RMIT University
12. Ren, Y., 'Week1: Introduction/ What is Data Science?', Lecture notes, COSC2670, RMIT University
13. Ren, Y., 'Practical Data Science: Classification', Lecture notes, COSC2670, RMIT University
14. COSC2670 Assignment 2 Discussion panel
15. The Python Standard Library, https://docs.python.org/3/library/codecs.html
16. https://scikit-learn.org/stable/modules/cross_validation.html
17. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
18. https://scikit-learn.org/stable/modules/tree.html