Student ID: s3716113

Student Name: LiangyuNie

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*.

# Identifying Proteins Critical to Learning Ability of Down Syndrome Mouse Model

**June 9, 2020**

**Liangyu Nie**

**Bachelor of Computer Science, RMIT University**

**s3716113@student.rmit.edu.au**

## Table of Content

# Abstract

Down syndrome (DS) is a chromosomal abnormality (trisomy of human chromosome 21) associated with intellectual disability and affecting approximately one in 1000 live births worldwide. The overexpression of genes encoded by the extra copy of a normal chromosome in DS is believed to be sufficient to perturb normal pathways and normal responses to stimulation, causing learning and memory deficits. [1]

This data mining project aims to test out multiple classification techniques to find the most suitable ones for classifying mice into 8 categories based on the protein expression data. These model will enable future analysis of similar datasets to be performed in a faster and more efficient manner.This is a high-dimension data clusterization exercise due to 77 proteins used as dimensions.

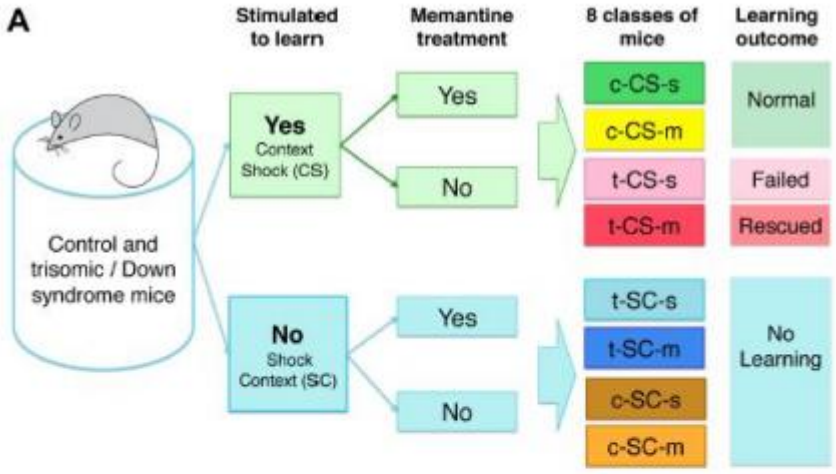
The analysis is implemented using Matplotlib, Pandas, Numpy, Scikit learn libraries in Python. The analysis is done in Jupyter Notebook.

Following cleaning the data was visualised through a variety of plots to help gain understanding and insight. The data was then modelled with a K-Nearest Neighbour Classification and a Decision Tree Model. The K-Nearest Model worked best with 5 neighbours, distance weighting and Manhattan distance (p = 1). The Decision Tree Models had an average accuracy ranging between 0.79 to 0.98, indicating extremely good predictive capability.

# Introduction

Down Syndrome (DS) has a prevalence globally of 1 in a 1000 live human births, and is the most common genetically defined cause of intellectual disabilities [1][2]. DS in humans is caused by the presence of an additional chromosome 21, referred to as trisomy [2]. Protein expression is significantly perturbed by human trisomy 21, leading to the physical and intellectual manifestations associated with DS. Due to its prevalence and health implications, a strong imperative exists to further understand and treat the condition.

The Ts65Dn mouse is the best-characterized of the DS models (Mitra, 2012). Higuera et al employed Ts65Dn and normal mice in experimental and control groups, exposing them to a range of variables (figure 1). The rodents were then euthanized and their cortex protein levels were analysed in a quantitative fashion. The resultant data was subject to an unsupervised clustering method known as Self Organising Maps (SOM) followed by labelling, in order to establish causal relationships between learning outcome, substance exposure, genetic structure and protein expression. The dataset generated by Higuera et al (_Mice Protein Expression Data Set_) is now freely available from the online, open-source repository _UCI Machine Learning Repository_ (http://archive.ics.uci.edu/ml/), and will be the focus of this analysis.




B

| Control mice | #Mice |
|---|---|
| c-SC-s | 9 |
| c-SC-m | 10 |
| c-CS-s | 9 |
| c-CS-m | 10 |
| **Trisomic mice** | |
| t-SC-s | 9 |
| t-SC-m | 9 |
| t-CS-s | 7 |
| t-CS-m | 9 |
| **Total** | 72 |

C

| | Proteins | | | | |
|---|---|---|---|---|---|
| Mice | $P_1$ | $P_2$ | … | $P_{77}$ | Class |
| $m_1$ | 0,3 | 0,5 | … | 1,3 | |
| $m_2$ | | | | | |
| $m_3$ | | | | | |
| … | | | | | … |
| $m_n$ | | | | | |

Fig 1.[1] Classes of mice.(A) There are eight classes of mice based on genotype (control, c, and trisomy, t), stimulation to learn (Context-Shock, CS, and Shock-Context, SC) and treatment (saline, s, and memantine, m). (B) Number of mice in each class. (C) Format of protein expression data: rows are individual mice, and columns, P 1 … P77, are the measured levels of the 77 proteins.

# Methodology

This data analysis of this project consisted of 3 steps:

(1) Retrieving and Preparing the Data

(2) Data Exploration

(3) Data Modalling

**1. Retrieving and Preparing the Data**

After retrieve data,there are 1080 rows by 82 columns and both numerical and categorical data in the data set.

The first step is **selecting a subset of target** and NaN values checked once the subset is chosen, to avoid unnecessary computation.

The second step is **remove all categorical data** because the native data contains both numerical (the protein expression levels) and categorical data (Genotype, Behavior, Treatment and Class). Class has 8 different values in the data set, while Genotype, Behavior and Treatment are all Boolean. For successful model creation, this categorical data must be replaced with numerical data.Summarizes the proposed strategy for replacing the categorical data. Control, C/S and Saline will be replaced with the integer value 1, while Ts65Dn, S/C and Memantine will be replaced with the integer value 0. In the literature Class is an amalgamation of these three Boolean values top generate one of eight unique tags. We will also amalgamate the three binary integers to generate one of eight unique numerical tags.

The final step is **identify outliers in the native data**,there are lots outliers and NaN in the data, this step was given significant consideration, as it will affect all subsequent analysis and conclusions about the data.Take the native date and for each protein column determines a count, mean, standard deviation (SD) and then then calculates values +- 2 units of SD on either of the mean (a 95% confidence interval) and +- 3 units of SD on either side of the mean (a 99% confidence interval). A query is then submitted to determine all values falling outside the 99% confidence interval. Note that this calculation was done on the native data, including all values, as the exclusion of what may be outliers would only serve to narrow the range of the 99% confidence, and possibly lead to stripping out of to many data point in further steps.

**2. Data Exploration**

The data exploration was done on the data following the replacement of nulls with the means, but before it was normalized. We made series of violin plots of the expression value for each protein type as our first step of data exploration. Violin plots are useful to see full distribution of the data with their probability density.

To make this violin plot series, the mean value of each protein is first calculated and the data frame is sorted based on their mean magnitudes. This helped to organize the data for easier visual inspection.

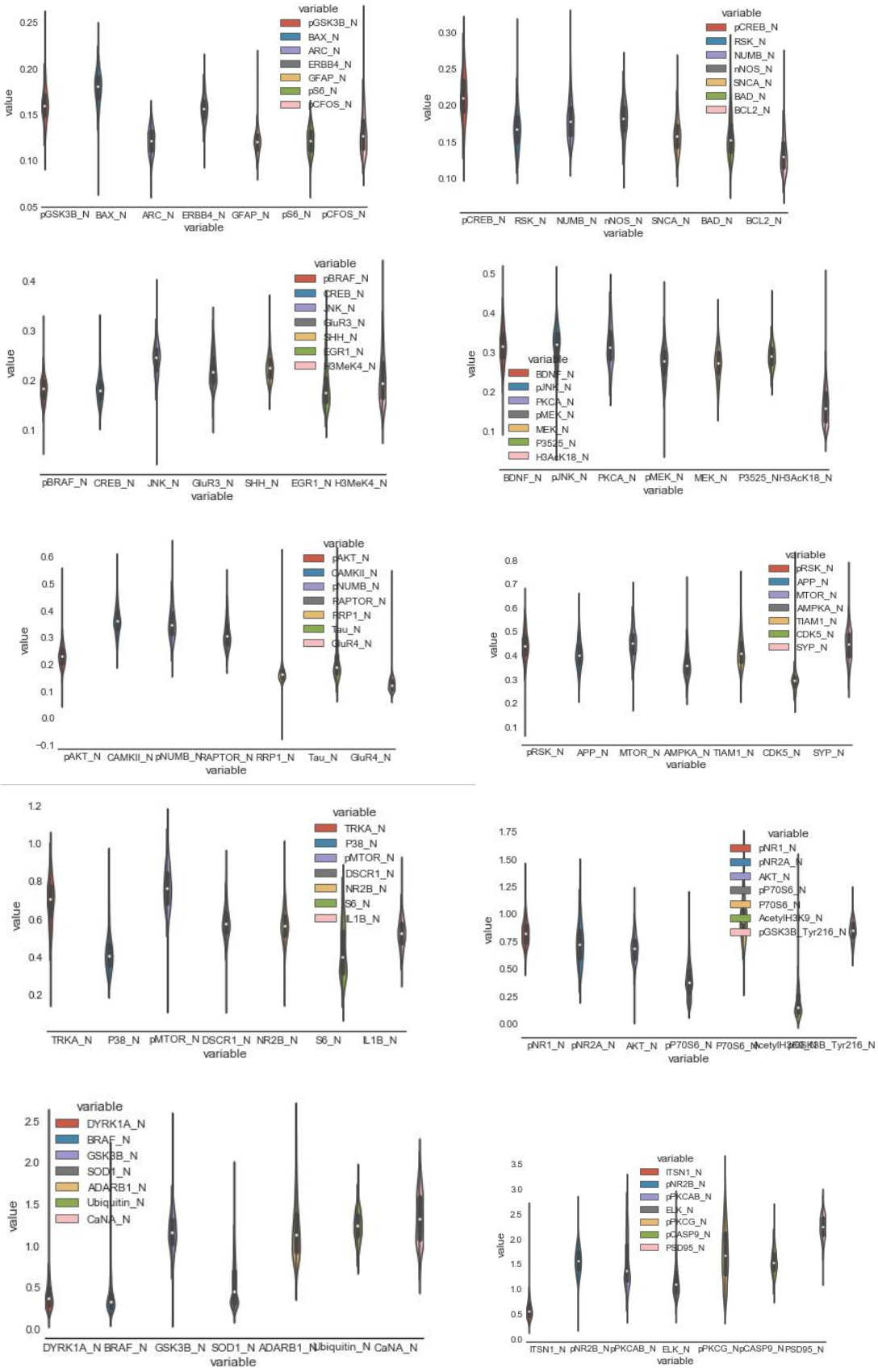



Figure 2 – Exploration of proteins based on their magnitudes

These violin plots showed how varied the magnitudes of each protein was, and shows why data standardization is needed. Also, the plots show most of the data have large ranges of outliers.

In order to find the outliers source, the 5 proteins whit the most significant range of outliers chose and examined: DYRK1A_N,AcetylH3K9_N,RRP1_N, NR2A_N, and pP70S6.

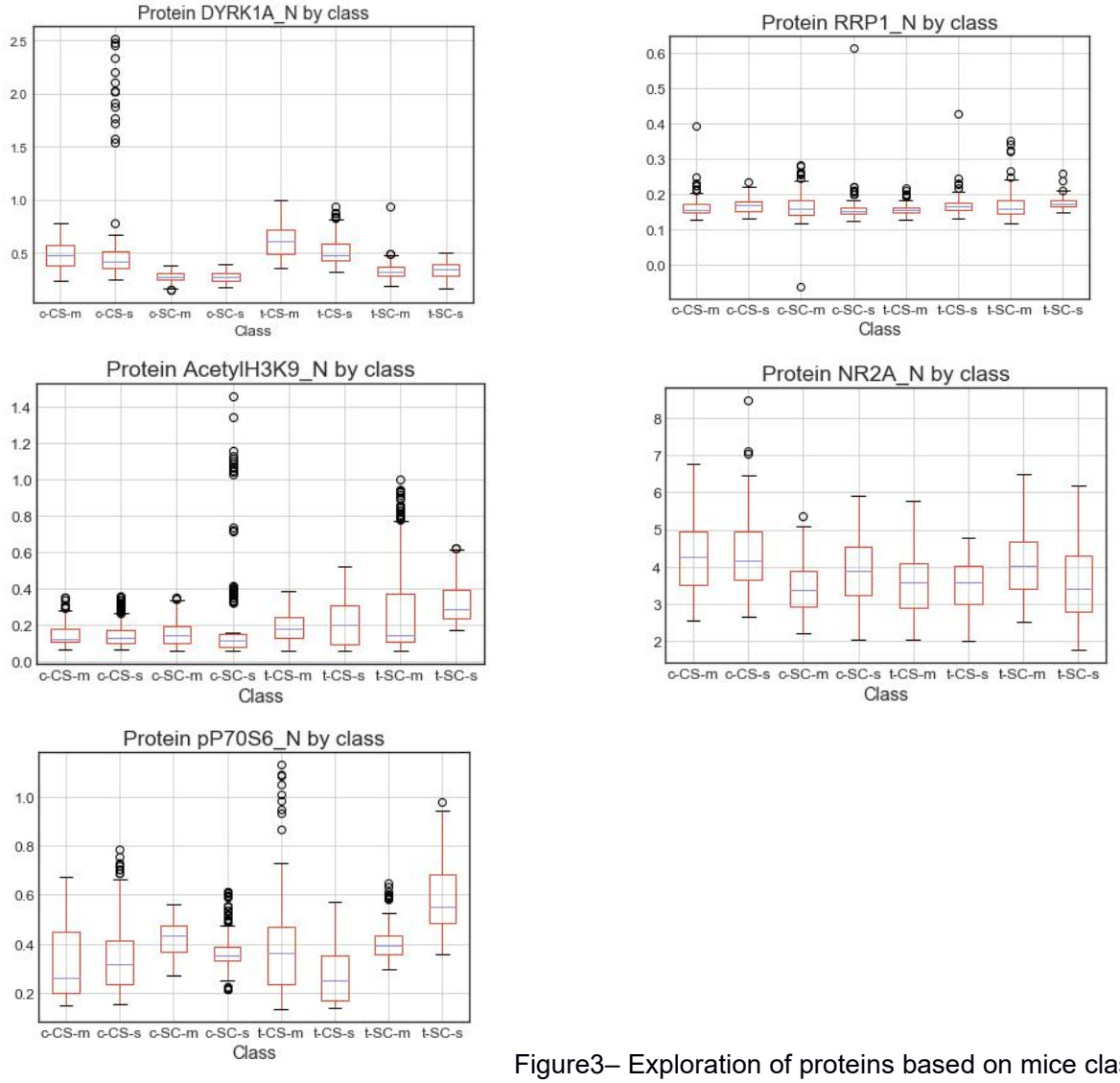



Figure3– Exploration of proteins based on mice class

These pairwise explorations show most of the big range of data magnitudes caused by outliers that usually dominated by one of the class. For example, big range of data magnitudes in protein DYRK1A_N is caused by outliers from c-CS-s class. Also, in protein pP70S6 its caused by outliers in t-CS-m class.

After go deeper into each class, that contains those outliers by examining the mouse numbers, the result shows below:

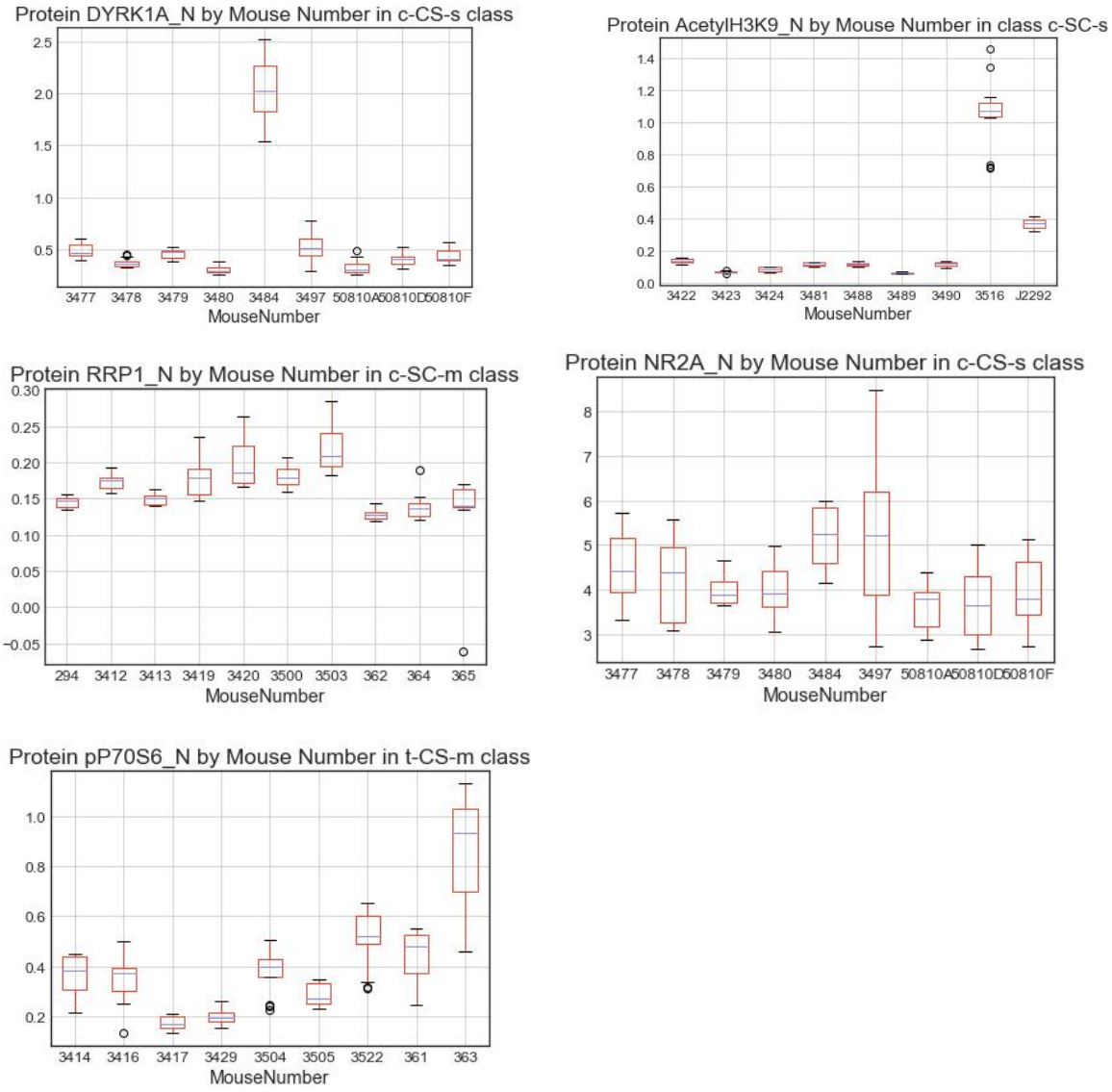



Figure 4 – Exploration of proteins based on mouse numbers in class contains dominating outliers.

In deeper analysis, we can see that most of the dominating outliers in each class comes from specific mouse numbers. For example, the obvious outliers from c-CS-s class in protein DYRK1A_N is comes from mouse number 3484. Also, the big magnitudes in protein NR2A_N in c-CS-s class is comes from mouse number 3497.

The result of exploration of protein based on mouse version shows below:

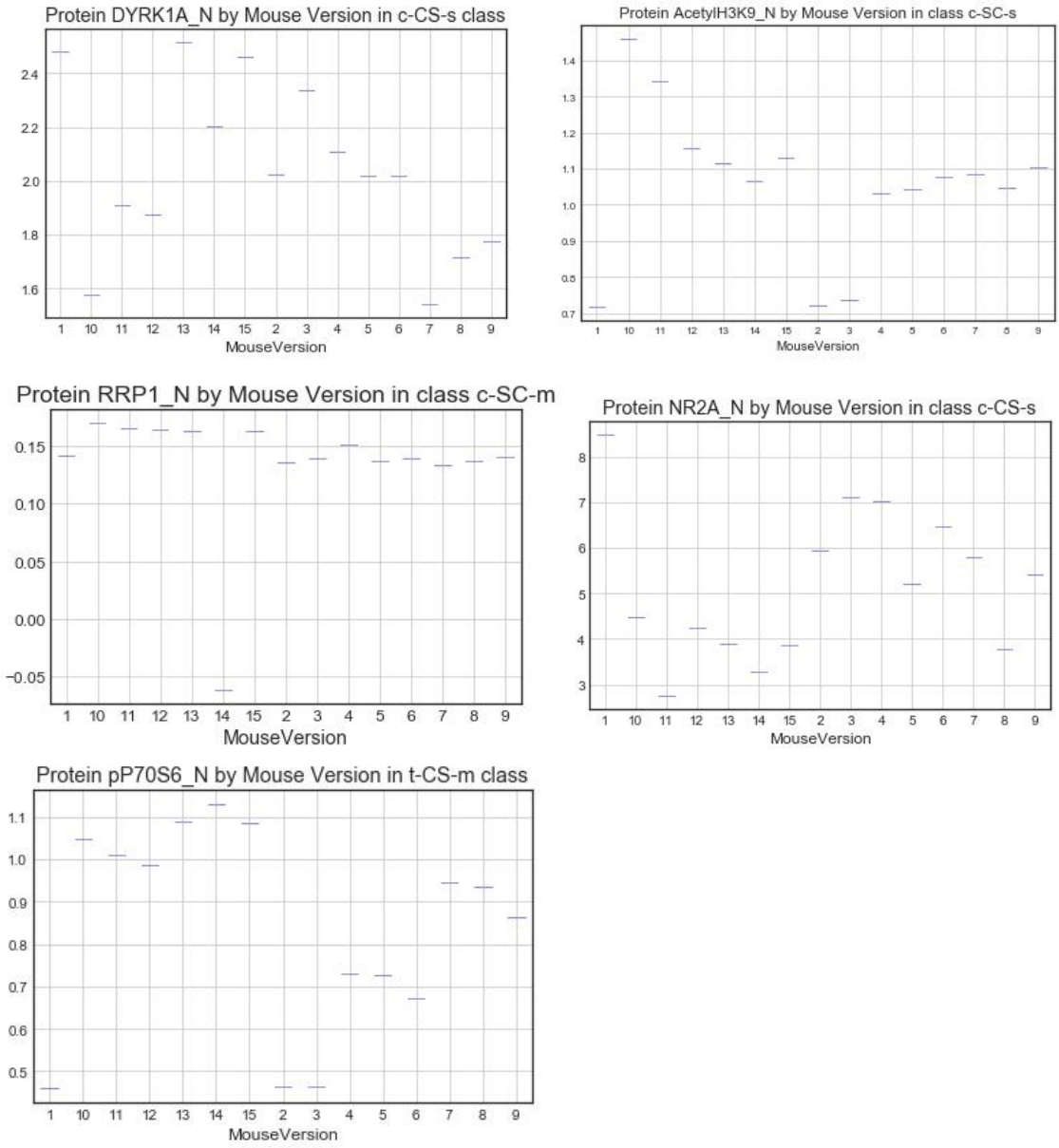



Figure 5– Exploration of proteins based on mouse numbers in class contains dominating outliers.

Different from previous result, now most of the obvious outliers is not coming from single mouse version, but from multiple mouse versions.

Therefore, the conclusion of this exploration is most of the magnitude data in proteins is vary because the outliers. These outliers usually comes from one specific mouse class and one specific mouse number in those class. However, from those specific mouse number comes multiple mouse versions that produced these outliers. Standardization of these magnitudes is clearly needed for the next step, data modelling.

### 3. Data Modalling

#### K Nearest Neighbors Classifier

One classification analysis was undertaken using K Neighbors Classifier. Several different classification tasks were undertaken to explore the data and the impact that different input variables has on the classification result.
The first analysis run used the default settings for weights (uniform), metrics and p (2). Behavior was picked by classified using inputs for SOD1, genotype and treatment. Initially we varied the number of neighbours to see the impact this would have on the error rate. This was repeated using the protein ISTN1 to investigate how different input variables impacted by classification. ISTN1 was then used to test the varying of other input parametres. Weighting the importance of the nearest 5 or 8 neighbours as well as using Manhattan (p=1) distance rather than Euclidean (p=2) were investigated.

Classification was also undertaken to predict the class (combination of genotype, treatment and behavior) of the mouse based on the expression of all 11 proteins. The same approach was followed as previously described varying the number of neighbours, weighting of neighbours and distance metric.

The analysis was undertaken by subset of Protein11 data were split into training/ testing datasets using sklearn.cross_validation.train_test_split. 60% of the data was used to train the model while the remaining 40% was used to test its predictive power. This breakdown was chosen to ensure the model was fitted with ample data to perform well when predictions were made. The model was trained using sklearn.neighbors.KNeighborsClassifier. Confusion matrixes were used to assess the error rate of the classification analysis. They were constructed manually using sklearn.metrics.confusion_matrix and tabulated using Microsoft Excel.

#### Decision Tree

Decision tree model is easier we don't need to consider parametres, because the experimental methodology employed by Higuera et al and the format of the data lend themselves to the Decision Tree Classification Model. This is due to the population of mice being broken up into control or Ts65Dn, followed by memantine or saline treatment, followed by CS or SC conditioning, so that we can just use all 11 proteins with all 8 classes to create a predictive model.

# Result

The K neighbours classification performed as part of the project highlighted how different input parameters or subsets of data can have a profound impact on the results. Initial analysis was performed using a subset of the data to explore the impact of different input parameters. SOD1 was chosen for this analysis as the scatter plots indicated that there was a substantial difference in the expression of this protein. This was reflected in the results of the first set of analysis . The results of the ITSN1 analysis in comparison had higher error rates. Looking at the scatter plot , it was also expected that the classification wouldn't be as clear cut. The higher error rates confirmed this hypothesis. Further analysis was undertaken using the ITSN1 data, to see if changing the input parameters would improve the predictive ability of the fitted model. There was no point doing this with the SOD1 data as the initial results were already very accurate.

Weighting the classification based on proximity did decrease the error rate but the largest impact was changing the distance measure. The average error rate drops from 0.3 or 30% incorrectly classified  to 0.01 or 1% . This is a huge difference that is solely due to the distance measure. It

is reasonable to assume that a highly dimensioned dataset such as this one would be impacted upon by the changing of the distance measured from Euclidean to Manhattan. The resulting analysis of the classification of behavior using the ITSN1 data highlighted how important it is to properly train a model and to explore a variety of input parameters.

The more interesting classification is using the 11 proteins to predict which class the mouse is from. Some intriguing patterns in the data were uncovered. The mice that were classified with the lowest error rate were the c-SC-s mice. This class of mice is the control for all 3 variables. The mice with the highest error rates are c-SC-m & t-SC-m. The only difference between c-SC-s and c-SC-m is the treatment with either the drug memantine or saline. This difference in only one of the 3 variables (genotype, behavior and treatment) is enough to change the mice from being easily classified to being the most difficult to classify. The other mice that are difficult to classify are those with Down syndrome and have been treated with memantine but not exposed to shock.

The best combination of input variables to fit the k nearest neighbours classification is:

- 5 neighbours
- Distance weighting
- Manhattan distance (p=1)

This gives an average error rate of 10%, ranging from 0% (c-SC-S) to 32% (c-SC-m). Of the mice that were misclassified the most often (c-SC-m & t-SC-m), they were misclassified as c-CS-m. This bias of misclassification should be taken into account when performed further analysis on this data.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.92 | 0.94 | 52 |
| 1 | 0.90 | 0.96 | 0.93 | 48 |
| 2 | 0.96 | 0.79 | 0.87 | 57 |
| 3 | 0.87 | 0.98 | 0.92 | 41 |
| 4 | 0.96 | 0.96 | 0.96 | 68 |
| 5 | 0.95 | 0.94 | 0.95 | 65 |
| 6 | 0.77 | 0.90 | 0.83 | 48 |
| 7 | 0.80 | 0.74 | 0.77 | 47 |
| accuracy | | | 0.90 | 426 |
| macro avg | 0.90 | 0.90 | 0.89 | 426 |
| weighted avg | 0.90 | 0.90 | 0.90 | 426 |

Figure 6 - 5 neightours, distance weighting, manhattan distance (p=1)

# Discussion

A further avenue of analysis would be to test if using only one of the highly correlated proteins had an impact on the results. Using highly correlated variables can lead to model overfitting and may be less able to extrapolate to conditions where the correlation may have changed (such as in a different model animal or treatment with a different drug). The lack of correlation between the other proteins would indicate that they are operating/ performing separately in the body. It is most likely that other correlated proteins were eliminated when we reduced the number of proteins from 77 to 11. Results from this study could be compared to results from classification using all 77 proteins to clarify if there are any differences. One of the benefits of using 11 instead of 77 proteins is that the data does not have as many dimensions. As previously mentioned, high dimensionality can impact on the performance of k nearest-neighbours.

The data cleaning process undertaken as part of this project highlighted some missing values for which were updated to equal averages. Based on the output from ‘Protein11.describe()’ we can see proteins BRAF_N, pERK_N, pNUMB_N, DYRK1A, ITSN1_N, SOD1_N contained NaN values. We filtered to find the NaN values using code line X, and identified identified the mouse

ID as 3426_n. The first 12 values were then filtered for the protein of interest and the average was determined using code line Y. These values were then populated into the NaN using code line Z.

For mouse ID 3426_n, only 12 out of the 15 measurements were recorded for We created the subset of data Protein11, and output ('Protein11.describe()') showed NaN values. We first filled the NaN from the original data import, 'AllProtein', and then re-created the subset Protein11.

# Conclusion

This analysis has clearly shown why it is important to properly train a model. The altering of input parameters while not changing the data used can greatly influence the accuracy of the classification. It has also highlighted that averaging of data can reduce the strength of data patterns as seen when the data was averaged to only include one value per mouse. It is important to fully explore your data before you start to undertake data modelling.

This assignment built on previous learning (e.g., data importation, exploration, cleaning, processing) in the data science process. During this task data from an online, open-source repository ("Mice Protein Expression Data Set" ref) was imported into an interactive shell or equivalent environment (e.g., Jupyter), explored, cleaned and processed, followed by the next steps in the data science process – data modelling, presentation and automation.

As the data set was explored and analysed, observations were recorded and used in the final steps of constructing classification models, k-means and a decision tree. Through the completion of these tasks, proficiency with the use of the interactive shell IPython, the libraries pandas, matplotlib, numpy and sklearn has be increased, as was proficiency with data modelling and classification. The final steps of the data science process were be formally introduced and engaged as the assignment was completed.

Undertaking an entire data science analysis has greatly increased our knowledge of data modelling and improved our python scripting skills.

# References

[1]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)

[2]Costa, A., Scott-McKean, J., & Stasko, M. (2008). Acute injections of the NMDA receptor antagonist memantine rescue performance deficits of the Ts65Dn mouse model of Down syndrome on a fear conditioning test. Neuropsychopharmacology, 33(7), 1624-1632.

[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)