PRACTICAL DATA SCIENCE WITH PYTHON

Assignment 2

Author: Kushagra Malik(s3788406)

Email: s3788406@student.rmit.edu.au

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*.

Table of Contents

Abstract

The following document is a report submitted as part of the specifications of the subject COSC2670 Practical Data Science with Python(RMIT university) for assignment 2. The report attempts to visualize mice protein expression levels while also describing the two classification models developed here in this report. The dataset was obtained from the UCI machine learning repository webpage and is permitted for public use. As part of the report, many data visualizations were generated to explore the potential effects of memantine on both control and trisomic mice. Furthermore, two machine learning models were developed to classify the classes of the mice. The models generated were based on the K nearest neighbor(KNN) and decision tree algorithms. The models generated accuracy scores of 100%(KNN) and 87% (Decision Tree).

Introduction

The study involved sampling the expression levels of 77 proteins within the cortex of a mouse brain. A total of 72 mice were used of which 38 were control mice and 34 were mice that expressed the down syndrome mutation. Down syndrome is a condition were there are abnormalities in the DS chromosome leading to impaired cognition (Higuera, Gardiner, & Cios, 2015).The 72 proteins that were being investigated were all measured 15 times per each mouse, totaling 1080 measurements per protein. The dataset contains eight classes of mice that were distinguishable based on features such as genotype, behavior and treatment. For the genotype feature, a mouse can be control (c) or trisomic (t). As for the behavior feature, mice have either been stimulated to learn (context shock) (CS) or have not (SC). Lastly, mice were either injected with saline(s) or memantine(m). The purpose of this study was to explore the effect of memantine in recovering the ability to learn in trisomic mice.

Below is a breakdown of the classes of mice taken from the UCI website:

Classes:
c-CS-s: control mice, stimulated to learn, injected with saline (9 mice)
c-CS-m: control mice, stimulated to learn, injected with memantine (10 mice)
c-SC-s: control mice, not stimulated to learn, injected with saline (9 mice)
c-SC-m: control mice, not stimulated to learn, injected with memantine (10 mice)

t-CS-s: trisomy mice, stimulated to learn, injected with saline (7 mice)
t-CS-m: trisomy mice, stimulated to learn, injected with memantine (9 mice)
t-SC-s: trisomy mice, not stimulated to learn, injected with saline (9 mice)
t-SC-m: trisomy mice, not stimulated to learn, injected with memantine (9 mice

Methodology

Upon downloading the dataset from the UCI Machine Learning repository, the excel file must first be opened and then subsequently saved as a csv file. The link for the dataset has been listed in the reference section of the report (Gardiner). After completing this initial step, the following packages needs to be imported into Jupyter notebook: numpy, pandas, matplotlib.pyplot and seaborn. The pd.read_csv function is then used to import the dataset and the shape method is used to check if it has successfully imported all of the observations. Furthermore, the types of data included in the dataset are also checked to familiarize ourselves with the column types.

Once the initial stage of importing of the file has been completed, the next step is to preprocess the data. This step primarily involves analyzing the dataset for the missing values in order to have a clear understanding of how much data is missing. A missing values table has been generated to help keep track of them. The table itself counts the number of missing observations per column and provides a percentage of what the missing data constitutes. This table is saved as missing_df. To identify missing values in a dataset, a new variable called missing_values is generated, and this points to the rows with missing values. When missing_values is printed, it can be noted that along these rows there are many missing values. It becomes obvious that these rows do not provide any useful information and are subsequently dropped. The missing values table is subsequently updated showing the remaining missing values. To identify the rows of the missing values, the missing _values data frame is called using a condition. The table is called to show only proteins that have more than zero missing values in their rows. Once this is completed, it becomes obvious that there are only nine proteins with missing values left. To rectify this, the missing values of each protein are imputed with the mean of the corresponding column. Upon completing this step, there are no more missing values and the dataset is ready to be used to construct visualisations and machine learning models.

As part of the data visualisation component of the assignment, 10 one-variable plots were constructed using the following features: Genotype, Treatment, Treatment, DYRK1A_N, CAMKII_N, RAPTOR_N, H3AcK18_N, H3MeK4_N and class. Every plot was given appropriate x and y labels as well as titles and all plots were set to the same aspect ratio of (12,8).

Next Dual variable plots were constructed and many of them examine the different protein concentrations of specific proteins amongst different classes. Again, all these graphs were given appropriate x and y labels and were set to the same aspect ratios as each other. This marks the end of the data visualisation component of the report.

The next topic discussed was the data modelling. Before any models could be generated, the 'MouseID' column had to be dropped. Furthermore, the label encoder from the sklearn package had to be imported and subsequently, the 'class' feature was label encoded.

The next step was to construct the training and target datasets. The training dataset was assigned to the variable x an contained all observations but excluded the class, genotype, treatment and behaviour columns. The target dataset(class column) was assigned to the variable y.

Next, the train_test_split method is imported and used to split the dataset into a 75% training group and a 25% testing group. The random state was set to 999 and stratification was enabled.

Next two functions that defined Euclidean and Manhattan scores were developed and an array that included the numbers from 1-20 was created and saved under k_range. k_range will be used to identify the optimal k parameter in the KNN Algorithm via the use of for loops. Each for loop appended the values acquired from the Euclidean and Manhattan score functions to the corresponding lists: knn_euclidean_score and knn_manhattan_score. Once these lists have been populated, a data frame called 'knn_Al' was created and depicted the k values tested for as well as Euclidean and Manhattan scores. Lastly the .max() function was used on the 'knn_Al' table to identify the best scores.

Next, the decision tree model is constructed. It must first be exported from the sklearn package and an array containing the numbers from 1- 20 was created and assigned to the variable depth. Subsequently a gini and entropy function was defined. As what was done with the k range variable, a similar approach was used with the depth variable to generate gini and entropy scores with each score being appended to its corresponding list (dt_gini_score and dt_entropy_score). A data frame containing the depth values, entropy score and gini score was created. The max() function was used to identify the best entropy and gini scores.

Results

The figure below depicts the spread of control and trisomic observations recorded in the study.

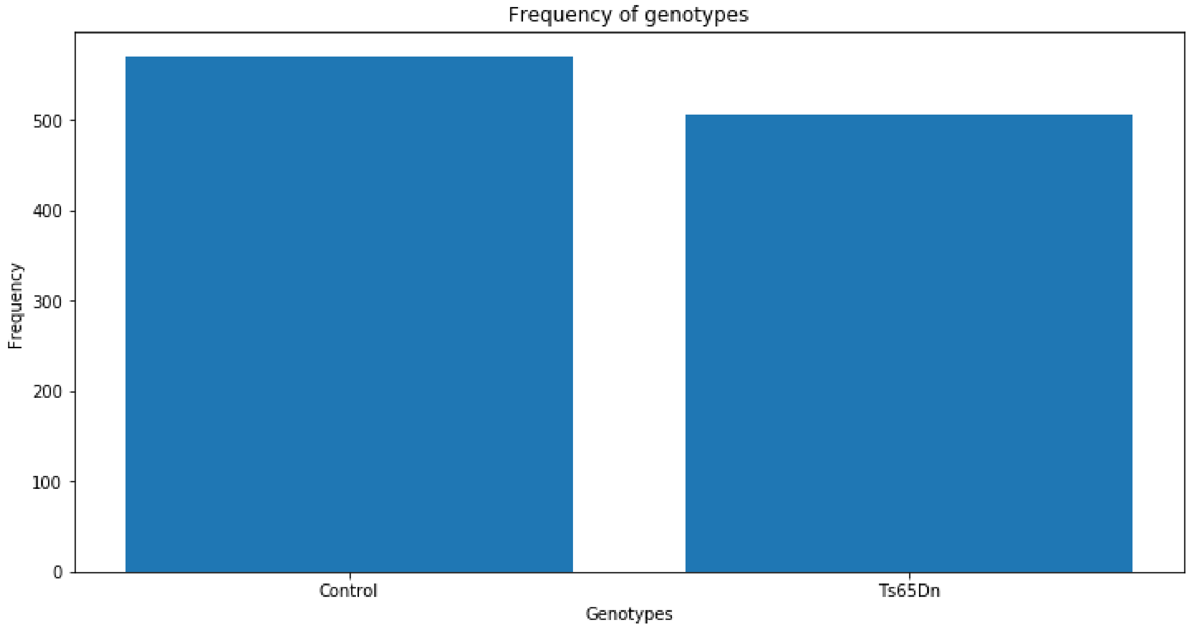

Figure 1: Graph depicting the relative number of control and trisomic observations recorded in the study.

Figure 2 displays the number of observations that underwent a context shocked learning stimulus in comparison to those that didn't.

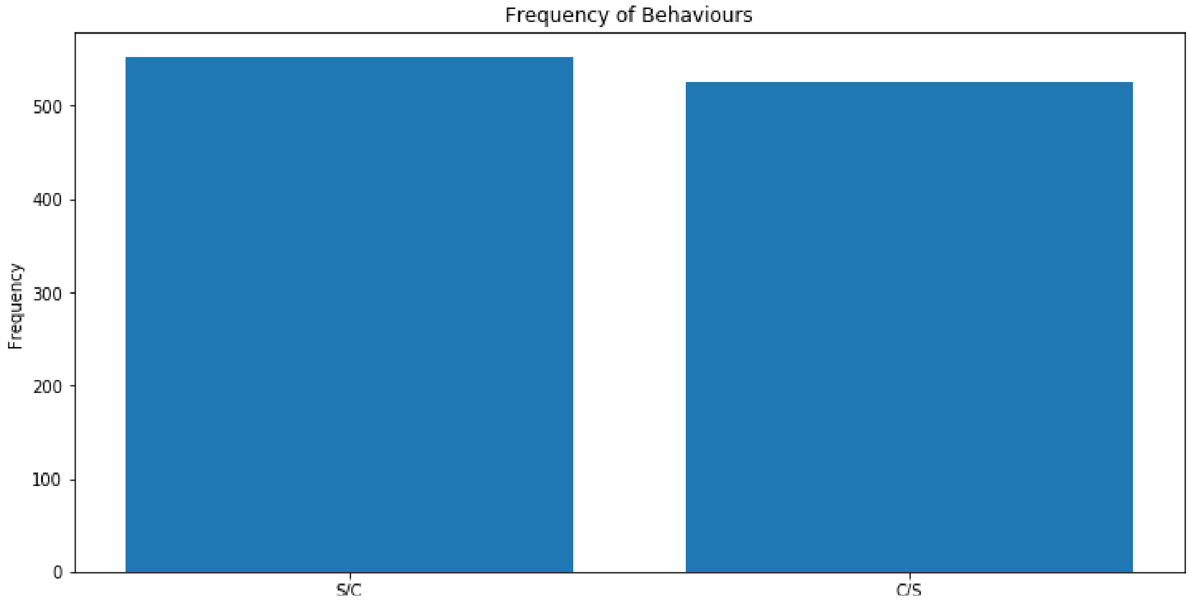

Figure 2: Distribution frequency of observations that underwent context shock versus those that didn't.

The figure below depicts the frequency distribution of the classes used in the study.

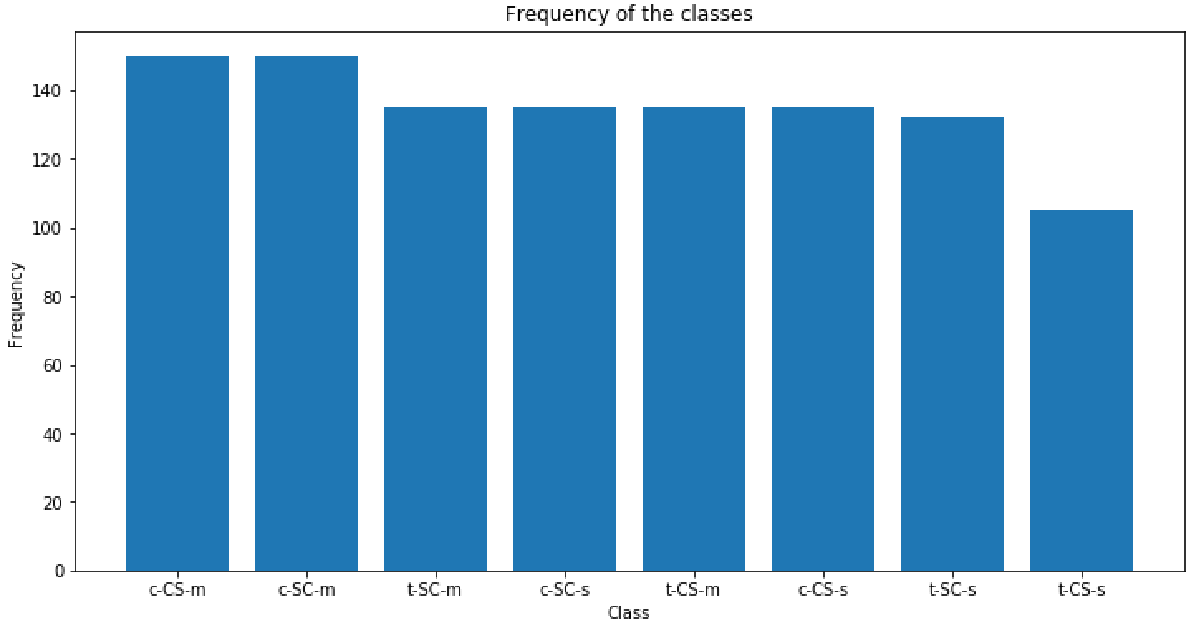

Figure 3: Frequency distribution of the classes used in the study.

Figure 4 demonstrates that c-CS-m mice display a higher median CAMKII_N protein concentration than that of t-CS-m.

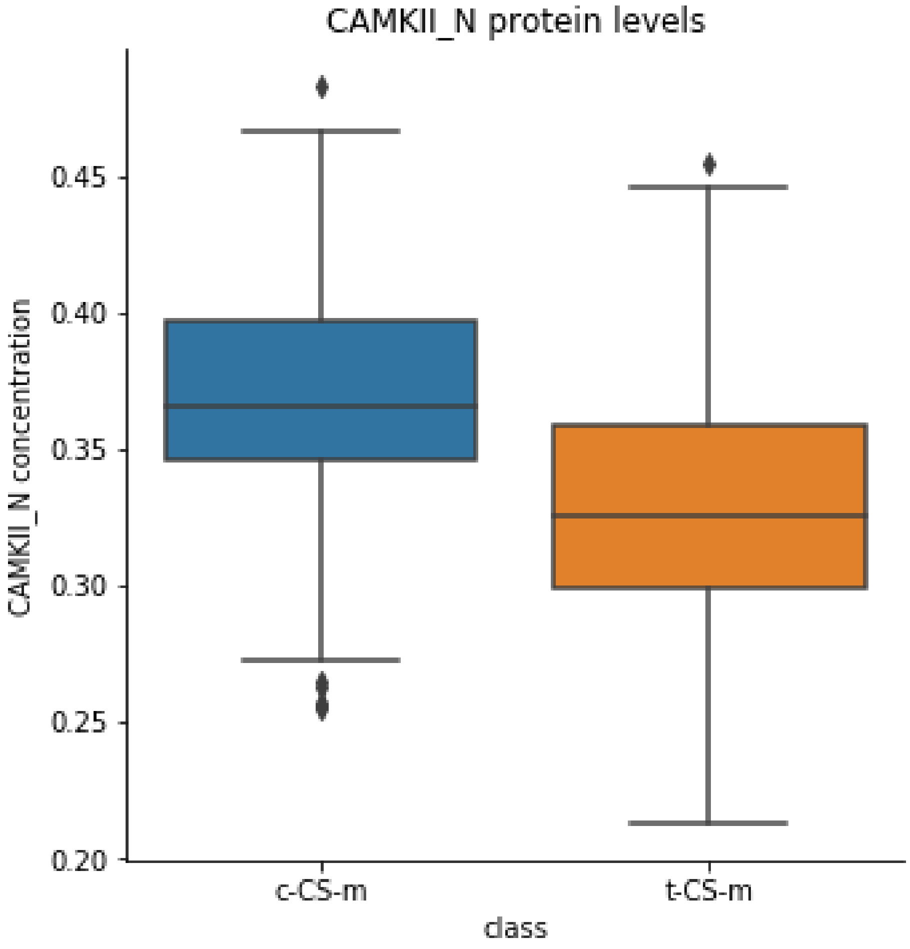

Figure 4: c-CS-m have greater a median CAMKII_N concentration.

Figure 5 displays all classes involved in the study and depicts that context shocked mice always had elevated median CaNA_N protein levels than that of non-context shocked mice.

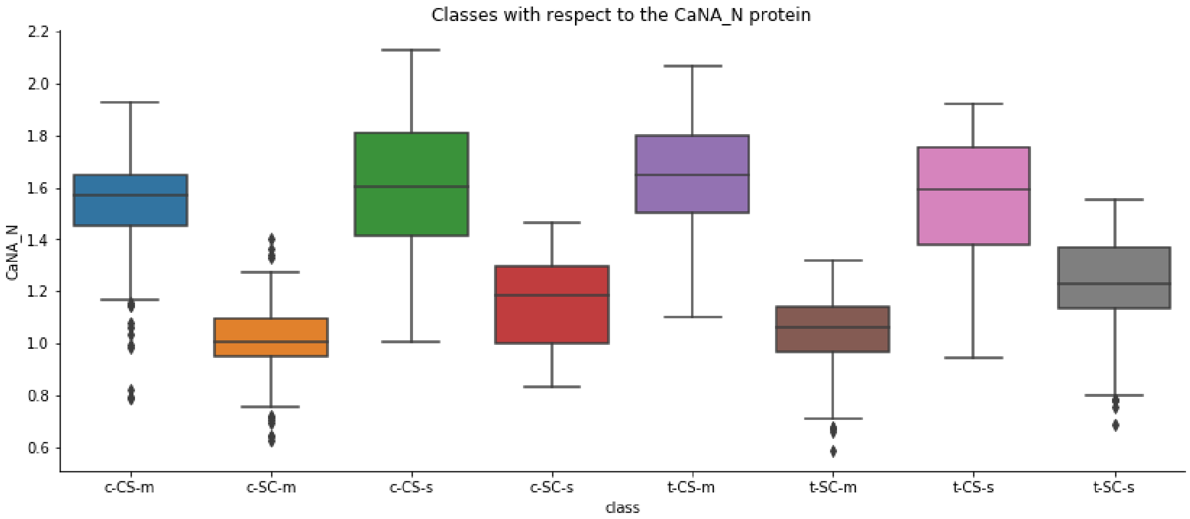

Figure 5: Figure depicting the CaNA_N protein levels of all the classes involved.

Figure 6 represents the relative difference in pCFOS_N concentration levels amongst c-SC-m and t-SC-m mice.

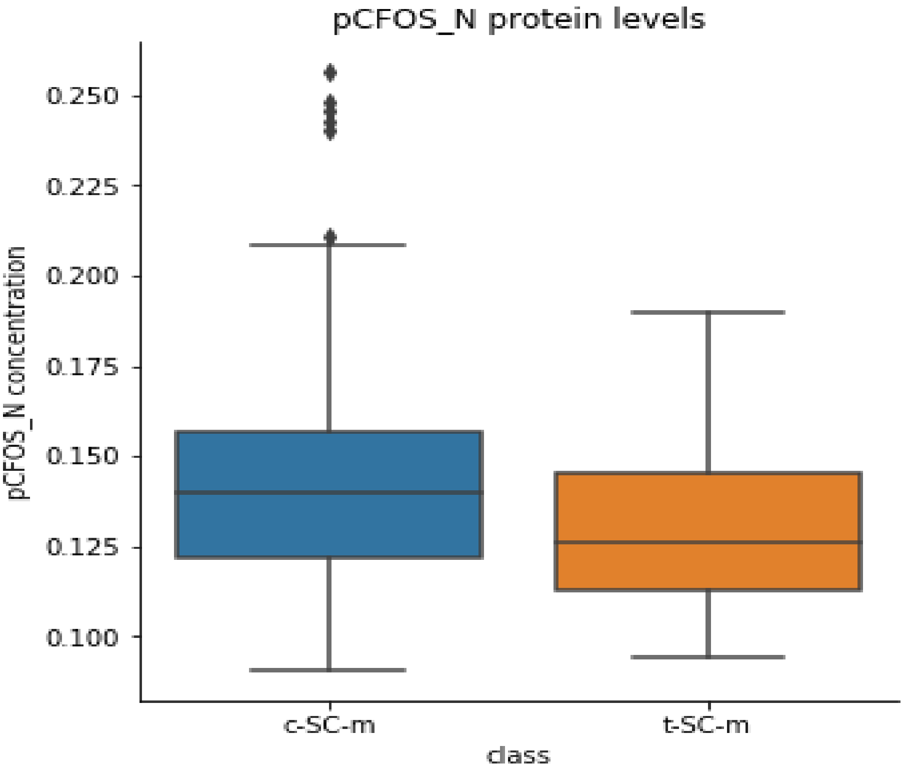

Figure 6: Difference in pCFOS_N concentration levels between c-SC-m and t-SC-m.

In regard to the two classification models, they were constructed using the KNN and Decision tree algorithms. The KNN model achieved an accuracy score of 100% with an optimal k parameter of 1 using the Manhattan score. The Decision tree model produced an accuracy score of 87% with a tree depth of 12 and using the Gini score.

Discussion

The dataset used in this study had a lot of missing values and so before it can be properly utilized for creating models, it first had to be preprocessed. Missing values had to be dealt with. Initially the dataset was checked for missing values in the first column. It became clear that for rows 987,988 and 989 all had missing values. Furthermore, the missing values for these rows stretched across all columns. With this in mind, it would be easier to delete these three observations. When the missing value table was checked for the 2nd time, nine proteins had missing values ranging from 4 to 285 instances. The optimal way to deal with this issue was to replace the missing values with the mean score of each protein. This was deemed the most appropriate way to deal with the missing values without impacting much of the data.

The first data visualisation (Fig 1) depicts the spread between the control and trisomic mice. It becomes apparent that there is a similar amount in each group which is ideal for training models. The same can be said about figure 2 and figure 3 as they all had a relatively equal distribution amongst the groups displayed in the graphs. If for example one group had far more observations than the other, the models generated from this data will not provide an accurate prediction. It essentially provides the models with enough training and testing observations.

Figure 4 explores the effect of the gene mutation in trisomic mice on CAMKII_N concentration levels. To elucidate the effects of the gene mutation, both mice were placed under the same conditions. That is, both mice were context shocked to stimulate learning as well as both receiving an injection of the drug memantine. Essentially any difference observed will be due to the genetic differences of the mice. The hypothesis was that c-CS-m will have a higher median concentration than that of t-CS-m. The graph depicts that control context shocked mice injected with memantine demonstrated greater median concentrations than that of trisomic context shocked mice injected with the drug. Therefore, the null hypothesis is rejected. These results could potentially mean that CAMKII_N production in trisomic mice is impacted by the mutation.

Figure 5 attempts to explore the effect that context shocking has on the relative concentration levels of the CaNA_N protein in mice irrespective of their genotype. The Hypothesis is that regardless of genotype, context shocked mice will display a greater median concentration of the protein CaNA_N compared to non-context shocked mice. The graph does indeed depict elevated CaNA_N protein levels amidst context shocked groups and therefore the null hypothesis is rejected. From this graph it can be noted that the context shocking has a drastic effect on the CaNA_N protein expression. However the graph does not do a good job at distinguishing the effects of being injected with memantine or saline amongst control context shocked groups and trisomic context shocked groups and may need further studies to confirm the effect of the drug on CaNA_N protein expression.

Figure 6 aims to distinguish between the median protein concentration levels of PCFOS_N in the groups that had no context shock-based learning and were injected with memantine however only differed based on their genotype. A plausible hypothesis would be to test if c-SC-m would have a greater median concentration than that of t-SC-m. For this experiment, the null hypothesis is rejected. The graph does indeed depict a greater median concentration for the c-SC-m class in comparison to t-SC-m. Since both of these groups have not underwent context shocking and have been injected with memantine, a possible explanation for these differences could be that in trisomic mice, basal levels of PCFO_N proteins could be significantly less than that of the control genotype.

Two classification models were generated as part of the assignment, a KNN and decision tree model. When determining the optimal hyperparameters for K and tree depth, arrays from 1 to 20 were used. If the optimal K value or tree depth was 20, only then would we increment the values to be tested for. If the optimal hyper parameter fell within this range, there would be no need to test for further values, thus saving computation power and time. These tests would be performed by using for loops. The best model produced was the KNN model with a score of 100% however there could be potential overfitting in these results. This means that the model performed exceptionally well on the training dataset but will most likely fail most of the time when used on new data. Hence using the decision tree model may prove to be a better choice in making more consistent predictions.

Conclusion

From the graphs, it can be said that by stimulating mice to learn by context shocking them, this would correlate to an increased production in protein levels. These results also appeared to independent of the condition of mice being injected with saline or memantine. When examining PCFOS_N protein levels of mice that were not context shocked and given memantine, control groups exhibited higher basal levels of the protein than the trisomic group thus potentially pointing towards impaired protein production. This report only examined a few of these proteins within these classes and it is important to note that some proteins maybe expressed differently with in all these classes. The best machine learning model produced was based on the KNN algorithm that produced an accuracy of 100% with optimal hyperparameters being K:1 and p:1. But, this model appears to be overfitted. Hence, the decision tree model would be better as it makes more consistent predictions.

References

Data from:

Gardiner, K. J. Mice Protein Expression Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression

Higuera, C., Gardiner, K. J., & Cios, K. J. (2015). Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome. *PloS one, 10*(6), e0129126-e0129126. doi:10.1371/journal.pone.0129126