Analysis of protein expression in mice

10th June, 2020

Shonil Dabreo, s3835204

Affiliations: Master of Data Science, RMIT University, s3835204@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	3
Introduction	3
Methodology	4
Results	8
Discussion	10
Conclusion	10
References	10

Abstract

Down Syndrome (DS) is a chromosomal disorder cause by the presence of an additional chromosome 21 referred as trisomy, which alters the normal pathways and normal responses to stimulation, causing learning and memory deficits. Expression levels of 77 proteins were measured in the cerebral cortex of 8 classes of control and Down syndrome mice which were exposed to context fear conditioning. CFC is a task used to assess associative learning. The measurements were taken with and without the injection of drug Memantine. This research attempts to understand the impacts of DS by analyzing the protein expression in mice which could have affected the stimulated ability to learn among the trisomic mice. Two classification models are implemented; K-Nearest Neighbors (KNN) Classifier and Decision Tree. It is observed that the selected feature subsets not only yield higher accuracy classification results but also are composed of protein responses which are important for the learning and memory process and the immune system. Results suggest that KNN classification approach can identify the most important proteins which may help to identify more effective drug to help learning process in people with DS.

Introduction

The mice protein expression dataset was created to study the effect of learning between normal i.e. control and trisomic mice i.e. mice with Down Syndrome (DS). 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. Humans are made up of millions of cells, and in each cell there are normally 23 pairs of chromosomes with a total of 46. A DNA contains the specific instructions that make each type of living creature unique. Genes are segments of deoxyribonucleic acid (DNA) that contain the code for a specific protein that functions in one or more types of cells in the body. Chromosomes are structures within cells that contain a person's genes. There are total 47 chromosomes in the cell of people with Down Syndrome. This additional chromosome 21 is known as trisomy of human chromosome 21 (hsa21). The characteristics of DS can be diagnosed by the observation of extra copy of chromosomes. Over expression levels of the proteins causes trisomy symptoms i.e. Down Syndrome.

The expression levels of 77 proteins obtained from normal genotype control mice and from trisomic Ts65Dn mice were examined to find out which proteins were successful and which failed in recovering the learning ability. These proteins produced detectable signals in the nuclear fraction of cortex. A total of 72 mice were gathered for analysis, out of which 38 were control mice and 34 were trisomic mice (Down syndrome). The mice were separated into two groups for determining their behavior into Context-Shock (CS) and Shock-Context (SC).

Firstly, mice are exposed to a training environment(context) after which they are habituated in a chamber where a mild foot shock is given to them. The normal/control mice would remember and able to associate between chamber and shock. Whereas, the trisomy mice is expected to fail to remember the association. Secondly, mice are habituated in a chamber for 3–5 minutes before being exposed to a training environment (context), at the end of which they receive a mild foot shock which generates a freezing response(fear). Both the control and trisomy mice would fail to remember the association.

In order to assess the effect of the drug memantine in recovering the ability to learn in trisomic mice, some mice have been injected with the memantine drug and others have not (i.e. saline) before the training. In the experiments, 15 measurements of each protein per sample/mouse were registered. Therefore, for control mice, there are 38x15, or 570 measurements, and for trisomic mice, there are 34x15, or 510 measurements. There are total of 1080 measurements per protein. Each measurement can be considered as an independent sample/mouse. The eight classes of mice are described based on features

such as genotype (i.e. Control mice and Trisomy mice), behavior (i.e. CS and SC) and treatment (i.e. Memantine and Saline). The description of classes is listed below:-

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)

The aim is to understand which trisomy protein classes that contribute to the success and the failure of mice learning. Analysis is conducted by creating a model which predicts the 8 classes of mice based on their protein expression levels. We can then decide which proteins were significant in the predictions i.e. support a hypothesis where we could say that a particular protein might affect learning in trisomic mice.

Methodology

Data analysis is performed in 4 steps:-

- Data Pre-processing
- Data Exploration
- Data Modelling
- Testing the model

- **Data Pre-processing**

Data pre-processing is extremely important because it allows improving the quality of the raw experimental data. The primary aim of preprocessing is to eliminate those small data contributions associated with the experimental error.
Firstly, the dataset and the needed packages are to be imported in the kernel environment. In this case, mice data was successfully imported. All the mice data columns/features had appropriate data types except for the Genotype, Behavior, Treatment and class. These features were changed to category datatype. There were total 1396 missing values in the mice data. The missing values of each column were replaced by the corresponding mean values for that column. This was done by iterating through all the columns. The categorical features contained 0 null values in the mice data. For columns (proteins) with a few null values, mean values were used as replacing the null values with mean value wouldn't make any difference. For columns (proteins) with more number of null values, mean values were used as replacing the null values with mean value could have most of the protein measurements with similar values, where there was variance in the measurements of the proteins.

- **Data Exploration**

The sums of the first 10 proteins were calculated and the graph displaying the sum for each protein/column was plotted. As we can see in the fig 1 below, the NR2A_N protein had the highest signal values, whereas, the pBRAF_N protein had the lowest signal values.

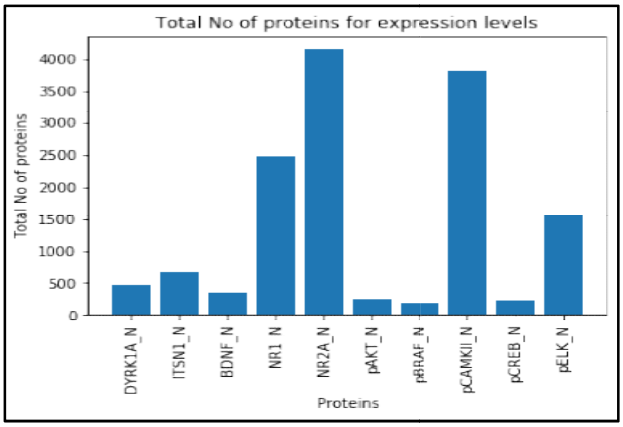

Fig 1

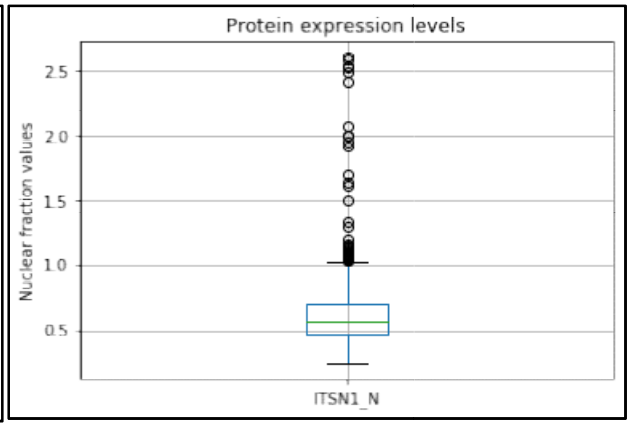

Fig 2

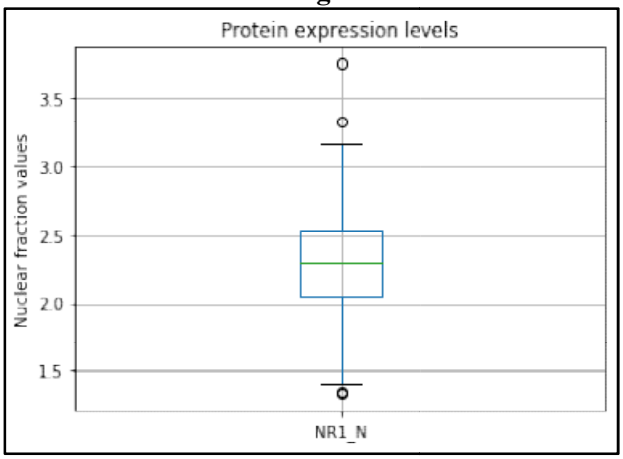

Fig 3

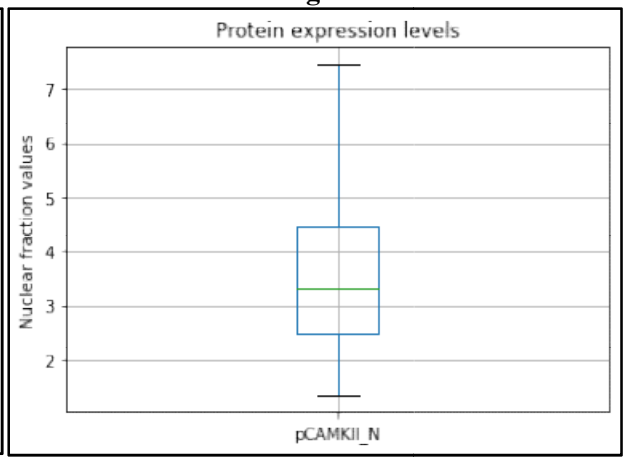

Fig 4

A series of Box plots for those columns were plotted separately. Fig 2, 3, 4 are the box plots for different proteins. The plotted points are the outliers which are visible in the Fig 2 and Fig 3 i.e. for proteins ITSN1_N and NR1_N. However, there seems to be no outliers in the Fig 4 i.e. for protein pCAMKII_N. The Fig 3 i.e. NR1_N protein contains three outliers and the Fig 2 i.e. ITSN1_N protein has a lot of outliers. We can see that the ITSN1_N (Fig 2) protein has similarity of the values that are closer to the mean value (i.e. Horizontal line between the box) and the NR1_N and pCAMKII_N proteins have variation of the values. The box plot is not centered between the upper and lower fence which means that the data is not normally distributed.

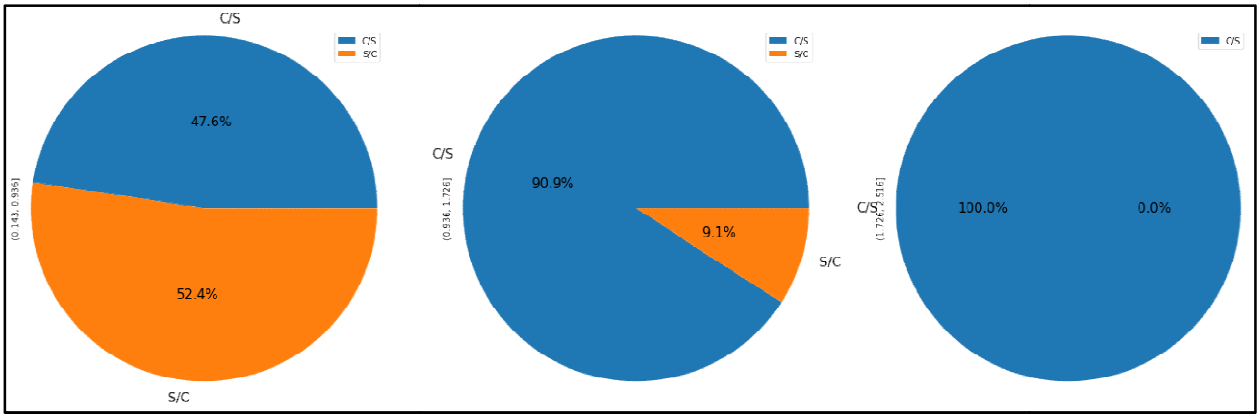

Fig 2.1

As we can see in the Fig 2.1, a DYRK1A_N protein was considered to determine the behavior of the mice for both the groups. The DYRK1A_N column was converted in categorical range using cut bins to better visualize the relation through the pie chart as the data had unique variables. The first pie chart represents the protein expression level values between 0.0 – 0.9, 0.9 – 1.7 for second pie chart and 1.7 – 2.5 range of values for third pie chart. The hypothesis is that, as the no of mice in CS group increases, the signal value of the protein will increase as well. The above Fig 2.1 illustrates that the percentage of CS group (47.6%) initially is lower than the SC group (52.4%). But in the 2nd pie chart, we can see that the no of CS group increases and there are no mice from SC group in the 3rd chart i.e. No of mice SC groups is 100%. Therefore, the hypothesis is statistically significant.

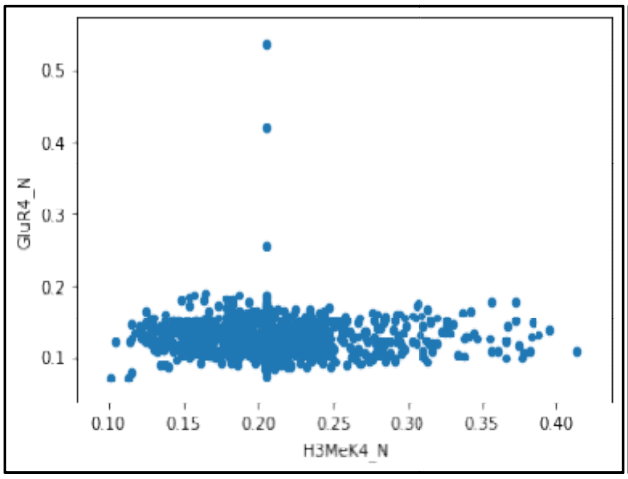

Fig 2.2

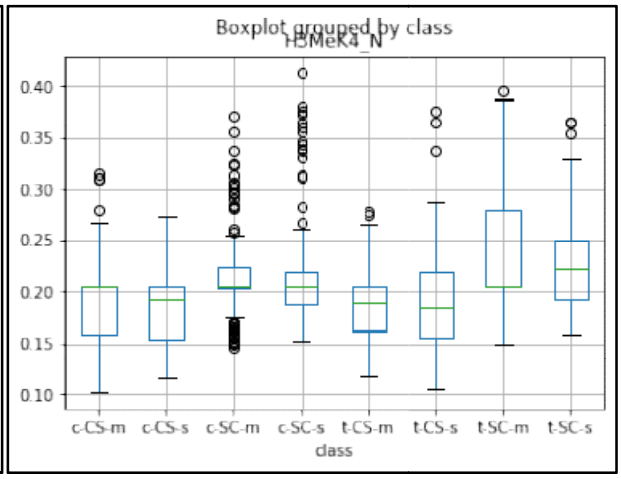

Fig 2.3

As we can see in Fig 2.2, a scatter plot between H3MeK4_N and GluR4_N proteins is plotted. This was done to check if multiple different proteins could affect the ability to learn in trisomic mice provided with the drugs injected. The hypothesis is that, as the signal value of H3MeK4_N protein increases the GluR4_N protein value decreases. In the given Fig 2.2, it is observed that as the signal value of H3MeK4_N protein increases the GluR4_N protein value remains stable i.e. between 0.1 to 0.2. Hence, there wasn't sufficient evidence to support the hypothesis. Moreover, this also means that GluR4_N protein is not effective for the trisomy mice in recovering to learn.

The Fig 2.3 shows the values of H3MeK4_N protein for 8 different classes. A box plot was used to plot the relation between a protein and the class. The graph with the values of H3MeK4_N protein was displayed by grouping the class. Total 8 Different box plots of classes are plotted for analysis. All the box plots have outliers except for the box plot of c-CS-m class. The hypothesis is that the trisomy mice with Shock-Context behavior with the treatment given by memantine drug injection (i.e. t-SC-m) will have higher effect of H3MeK4_N protein as compare to other classes. As we can see in the Fig 2.3, the box plot of t-SC-m for H3MeK4_N protein has more than 0.25 signal values which are higher than box plots of other classes. The H3MeK4_N protein is statistically significant for t-SC-m class category; hence, we can say that H3MeK4_N protein was effective for recovering the ability to learn in trisomy mice.

In addition to, the magnitude data in the proteins vary due to the outliers. A mouse number has multiple mouse versions in the mouse class which produces these outliers.

- **Data Modelling**

Classification model

Decision tree and K-Nearest Neighbors were used to build a model and predict the mice class and determine which proteins were critical for each class. These are both examples of supervised machine learning methods with the goal of creating a model that can be used predicts the class or value of the target variable based on several input variables.

Decision tree algorithm tries to solve the problem by using tree representation of a series of decisions. In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new node. The primary challenge in the decision tree implementation is to identify which attributes we need to consider as the root node and each level.

KNN is a non-parametric, lazy learning algorithm. IN KNN, the model structure is determined from the data without making any assumptions on underlying distribution. The training phase in KNN is very minimal as it doesn't use the training points for generalization. KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. The output is a class membership (i.e. predicts a class). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. The only issue is that stores almost all of the training data which indeed requires high computational power and memory and which could slow down the prediction process.

Model Implementation

Feature selection:
As we need to select a label data, therefore we would be using the Class column from the mice dataset as a predictor and the rest of the columns which includes 77 protein expression levels and Genotype, Behavior and Treatment would be taken based on which class would be predicted. The class was selected as target variable because it contains all the information of Genotype, Treatment and Behavior of the mice.

Model selection:
The dataset is divided into 70% train data and 30% test data to get a better indication of the models performance on unseen data.
Also, the cross validation score of 10 folds of both the models were calculated to tune the parameters accordingly. As the K value increases the accuracy score also increases. However, it is said that a small k results in predictions with high variance and low bias and K=1 could result in 100% accuracy or overfitting. So, k=5 would be the tuned value for K-Nearest Neighbors.
Likewise, changing different parameter values of Decision Tree resulted in an average score of 76%.
Therefore, the Decision Tree model was build with the default value for each parameter. The results were then interpreted.

Results

1. **TN / True Negative:** when a case was negative and predicted negative
2. **TP / True Positive:** when a case was positive and predicted positive
3. **FN / False Negative:** when a case was positive but predicted negative
4. **FP / False Positive:** when a case was negative but predicted positive

Precision TP/ (TP + FP): Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

Recall TP/ (TP + FN): Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

F1 Score (2*(Recall * Precision) / (Recall + Precision)): The F_1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. The F_1 scores are lower than accuracy measures as they embed precision and recall into their computation.

Classification-error rate: The percentage of observations in the test data that the model mislabeled.

Confusion matrix is used to gauge the accuracy of the model. The numbers on the diagonal of the confusion matrix correspond to correct predictions and the other values as total no of errors i.e. mislabeled.

	precision	recall	f1-score	support
c-CS-m	0.73	0.97	0.83	33
c-CS-s	0.94	0.80	0.87	41
c-SC-m	0.90	0.90	0.90	52
c-SC-s	0.98	1.00	0.99	47
t-CS-m	0.96	0.74	0.84	35
t-CS-s	0.87	0.87	0.87	31
t-SC-m	0.86	0.93	0.89	41
t-SC-s	0.93	0.91	0.92	44
accuracy			0.90	324
macro avg	0.90	0.89	0.89	324
weighted avg	0.90	0.90	0.90	324

Fig 3.1 Classification report

```
array([[32,  1,  0,  0,  0,  0,  0,  0],
       [ 5, 33,  0,  0,  0,  3,  0,  0],
       [ 0,  0, 47,  0,  0,  0,  5,  0],
       [ 0,  0,  0, 47,  0,  0,  0,  0],
       [ 5,  0,  0,  0, 26,  1,  0,  3],
       [ 2,  1,  0,  0,  1, 27,  0,  0],
       [ 0,  0,  2,  1,  0,  0, 38,  0],
       [ 0,  0,  3,  0,  0,  0,  1, 40]], dtype=int64)
```

Fig 3.2 confusion matrix

For KNN Algorithm, we got 90% accuracy score. As we can see in Fig 3.2, there are total 34 mislabeled errors for the KNN model. The highest precision i.e. corrected predicted class was c-SC-s.

	precision	recall	f1-score	support
c-CS-m	0.75	0.91	0.82	33
c-CS-s	0.79	0.73	0.76	41
c-SC-m	0.86	0.83	0.84	52
c-SC-s	0.93	0.79	0.85	47
t-CS-m	0.81	0.71	0.76	35
t-CS-s	0.81	0.84	0.83	31
t-SC-m	0.81	0.93	0.86	41
t-SC-s	0.89	0.93	0.91	44
accuracy			0.83	324
macro avg	0.83	0.83	0.83	324
weighted avg	0.84	0.83	0.83	324

Fig 4.1 classification report

```
array([[30,  2,  0,  0,  1,  0,  0,  0],
       [ 7, 30,  0,  0,  2,  2,  0,  0],
       [ 1,  0, 43,  1,  0,  0,  6,  1],
       [ 0,  0,  4, 37,  0,  0,  3,  3],
       [ 2,  4,  0,  0, 25,  4,  0,  0],
       [ 0,  2,  0,  0,  3, 26,  0,  0],
       [ 0,  0,  2,  0,  0,  0, 38,  1],
       [ 0,  0,  1,  2,  0,  0,  0, 41]], dtype=int64)
```

Fig 4.2 confusion matrix

For Decision Tree, the accuracy score is 83%. After tuning the parameters, it was observed that parameters with default value give the best accuracy score. The confusion matrix had a total of 47 mislabeled errors. The highest precision score was of the t-SC-s class.

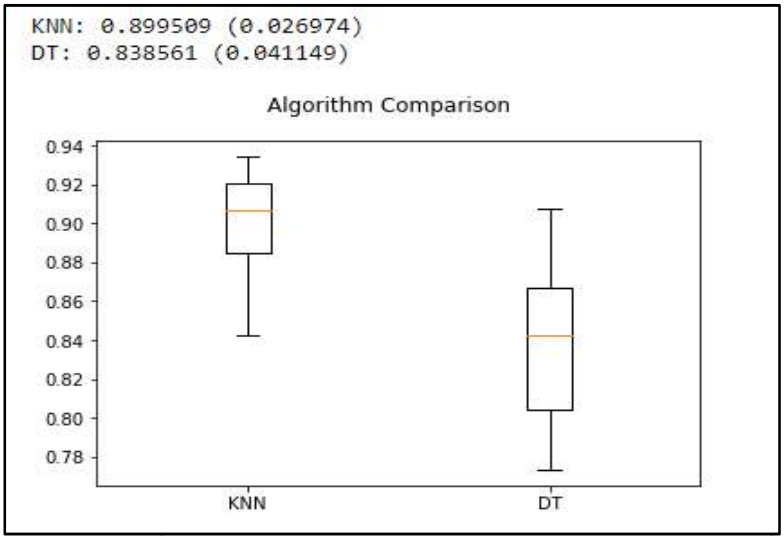

Fig 5 Algorithm comparison

The Fig 5 shows the comparison of the two models using box plot. As we can see, the KNN model has higher accuracy as compared to the accuracy of the Decision Tree model. For this dataset, KNN model was effective in yielding better scores.

Discussion

Two classifiers; K-Nearest Neighbor and Decision Tree were used for analyzing. We can see the accuracy and the confusion matrix from the above figures.
It can be seen clearly that KNN model worked efficiently better with an accuracy of 90% compared to the 83% accuracy of Decision Tree model. Therefore, we should implement KNN for this kind of dataset where features are numeric as KNN is a distance metric algorithm.

Conclusion

- To sum up, KNN model could predict the unlabelled data but couldn't classify which protein contributes to success or failed learning. The KNN algorithm doesn't have a feature importance method to classify the data. Although, Decision Tree had feature importance method but didn't had a high accuracy score.
- We should implement KNN model when the features are numeric to find the similar examples. Whereas, Decision Tree model should be implemented when we need to classify a particular class variable where the features contains binary data.

References

- Science direct. Data Pre-processing. Available at
  < https://www.sciencedirect.com/topics/engineering/data-preprocessing >
  [Accessed 9 June 2020].

- Saringat, M. Z., Mustapha, A. and Andeswari, R. (2018), Comparative Analysis of Mice Protein Expression: Clustering and Classification Approach, International Journal of Integrated Engineering, 10(6). Available at:
  < https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/2779 >
  [Accessed 10 June 2020].

- Kulan H, Dag T (2019), in silico identification of critical proteins associated with learning process and immune system for Down syndrome, PLoS ONE 14(1): e0210954. Available at
  < https://doi.org/10.1371/journal.pone.0210954 >
  [Accessed 10 June 2020].

- Bronshtein, A (2017) A Quick Introduction to K-Nearest Neighbors Algorithm, Noteworthy – The Journal Blog 12(5). Available at
  < https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7 >
  [Accessed 10 June 2020].

- Chauhan, N. S (2019) Decision Tree Algorithm – Explained, towards data science 12(24) Available at < https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4 > [Accessed 10 June 2020].