Student ID: s3778722
Student Name: Han Chien Leow

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*.

PRACTICAL DATA SCIENCE ASSIGNMENT 2

Mice Protein Expression Data Set

Han Chien Leow

s3778722 | 08 June 2020

Table of Content

1.0 Abstract

The Mice Protein Expression Data Set provided by UCI will be used for this project. The data set is about the expression levels of 77 proteins 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 [1]. The data set contains 1080 instances and 82 attributes as well as having a multivariate characteristic. It consists 77 expression levels measured in the nuclear fraction of different proteins. Also, there are two different genotypes for the total 72 mice in this data set, which are control (38 mice) and trisomy (34 mice). For your information, a genotype of trisomy may indicate that the mice is Down Syndrome. Furthermore, there might be stimulation to learn (context-shock) or no stimulation to learn (shock-context) for the mice in this experiment. The treatment types are memantine (m) or saline (s). By doing this, we can carefully study and assess the effects of various treatment types on the mice with trisomy genotype. Nevertheless, there are 8 distinctively classified classes, such as c-CS-s, c-CS-m, c-SC-s, c-SC-m, t-CS-s, t-CS-m, t-SC-s, t-SC-m based on the features from genotype, behaviour and treatment.

2.0 Introduction

Down syndrome, known as trisomy 21, is a genetic disorder caused by the additional copy of chromosome 21. This genetic disorder deeply affects the physical growth, intellectual disability, and characteristic facial features of a person [2]. There are 6000 people born with Down syndrome each year, which came with a frequency of 1 in 800 live births [3]. Therefore, we will be conducting supervised machine learning techniques with the classification approach using decision tree and kNN to identify subsets of proteins that are discriminant between the classes in order to assess associative learning. So that, we can help identify effective and beneficial drug using the result found by correctly identifying the best protein from the data set.

Our goal is set to clearly satisfy the aim of this project, which is to identify subsets of proteins that are discriminant between the classes and research the best data model possible with the approach of classification for this particular project.

Methodology

3.0 Retrieving and Preparing the Data

3.1 Data Retrieving

Firstly, the data set was retrieved and loaded to our working environment using the .read_excel() function to read the mice protein expression data set as the original data set is given in an excel format.

3.2 Check data types

The data types were checked using the .dtypes function, which shows the data types for all of the columns present in the mice protein expression data frame. Besides, the data can be checked to ensure that the data set is loaded properly and matched the source (.xls) file. The loaded data are checked thoroughly to have appropriate data types assigned. Therefore, the original data types are decided to be retained and unchanged because the data types listed are sensible and able to get the work done.

3.2 Typos

The typos in the mice protein expression data frame were checked using the .value_counts() function, which helped to check for data entry errors and outliers with a generated frequency table. It shows all the unique values along with the count. Upon checking the frequency table, it was found that there are no typos present in the data set. So, no necessary changes were made.

3.3 Extra-whitespaces

The extra-whitespaces in the mice protein expression data frame were checked using the .value_counts() function. Upon checking the frequency table, it was found that there are no extra-whitespaces present in the data set. So, no necessary changes were made.

3.4 Upper/Lower-case

The upper or lower-case in the mice protein expression data frame were checked using the .value_counts() function. Upon checking the frequency table, it was found that there are no inappropriate upper or lower-case present in the data set. So, no necessary changes were made.

3.5 Sanity checks

Sanity checks were performed for the proteins to check for impossible and illogical values. An algorithm using a simple for loop was developed to check if there are overly high values (>10) and negative values (<0) of expression levels of the 77 proteins. Upon checking, it was found that there are no values above or below the bound set. So, no necessary changes were made.

3.6 Missing values

The missing values were checked using .isna().sum() to get the total missing values count on each of the columns in the mice protein expression data frame. The missing values were quickly filled with the column's mean, so the original mean will not be affected and other statistical components will not be greatly affected after the missing values (NaN) were filled with the mean as a way to minimize the guessing error. Also, the .describe() function is used to observe the changes in common statistical details like percentile, mean, std etc for all the proteins. The missing values were rechecked using .isna().sum() and .isnull().any().any() to confirm that there are no any remaining missing values left in the data frame.

4.0 Data Exploration

4.1 Explore each column

Box plot was used to explore the values of expression levels for DYRK1A_N protein because it drafts key figures like median, lower quartile (Q1), upper quartile (Q3), bottom whisker and upper whisker as well as showing outliers. The box plot regarding the values of expression levels for DYRK1A_N showed us a median of 0.366540, Q1 of 0.288163, Q3 of 0.487574. The result indicates that protein DYRK1A_N are mostly present with a low value of the expression level compared to other proteins. Moreover, there are a lot of outliers present around the value of 0.9 and some around the range from value 1.5 to 2.5 expression levels.

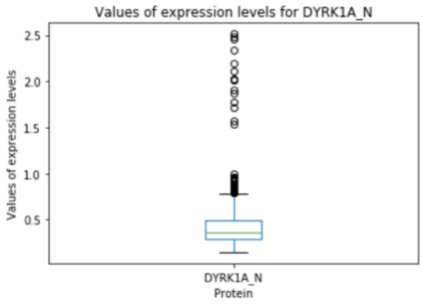


Figure 1: Box Plot for the Values of Expression Levels against DYRK1A_N protein

Also, box plot was used to explore the values of expression levels for ITSN1_N protein. The box plot regarding the values of expression levels for ITSN1_N showed us a median of 0.566365, Q1 of 0.473669, Q3 of 0.697500. The result indicates that protein ITSN1_N are mostly present with a low value of expression level compared to other proteins. Moreover, there are a lot of outliers present after the upper whisker especially around value 1.2 of the expression level.

Additionally, box plot was used to explore the values of expression levels for BDNF_N protein. The box plot regarding the values of expression levels for BDNF_N showed us a median of 0.316703, Q1 of 0.287650, Q3 of 0.348039. The result indicates that protein BDNF_N are mostly present with a low value of expression level compared to other proteins. Moreover, there are a lot of outliers present above the upper whisker and some below the bottom whisker.

Additionally, box plot was used to explore the values of expression levels for NR1_N protein because it drafts key figures like median, lower quartile (Q1), upper quartile (Q3), bottom whisker and upper whisker as well as showing outliers. The box plot regarding the values of expression levels for NR1_N showed us a median of 2.297269, Q1 of 2.059152, Q3 of 2.528035. The result indicates that protein NR1_N are mostly present with a high value of expression level compared to other proteins. Moreover, there are only several outliers present outside the whiskers, indicating that the values of expression levels for NR1_N are mostly accurate with minimal measurement / experimental error.

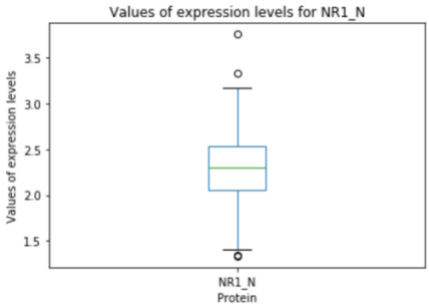


Figure 2: Box Plot for the Values of Expression Levels against NR1_N protein

Besides, box plot was used to explore the values of expression levels for NR2A_N protein. The box plot regarding the values of expression levels for NR2A_N showed us a median of 3.763306, Q1 of 3.160287, Q3 of 4.425107. The result indicates that protein NR2A_N are mostly present with a high value of expression level compared to other proteins. Moreover, there are some outliers present above the upper whisker.

Also, box plot was used to explore the values of expression levels for pAKT_N protein. The box plot regarding the values of expression levels for pAKT_N showed us a median of 0.231246, Q1 of 0.205821, Q3 of 0.257225. The result indicates that protein pAKT_N are mostly present with a very low value of expression level compared to other proteins. Moreover, there are some outliers present above the upper whisker and below the bottom whisker.

In addition, box plot was used to explore the values of expression levels for pBRAF_N protein. The box plot regarding the values of expression levels for pBRAF_N showed us a median of 0.197226, Q1 of 0.182270, Q3 of 0.197226. The result indicates that protein pBRAF_N are mostly present with a very low value of expression level compared to other proteins. Moreover, there are some outliers present above the upper whisker and below the bottom whisker.

Furthermore, box plot was used to explore the values of expression levels for pCAMKII_N protein because it drafts key figures like median, lower quartile (Q1), upper quartile (Q3), bottom whisker and upper whisker as well as showing outliers. The box plot regarding the values of expression levels for pCAMKII_N showed us a median of 3.329624, Q1 of 2.479861, Q3 of 4.480652. The result indicates that protein pCAMKII_N are mostly present with a high value of expression level compared to other proteins. Moreover, there are no outliers present outside the whiskers, indicating that the values of expression levels for pCAMKII_N are accurate with no measurement / experimental error.

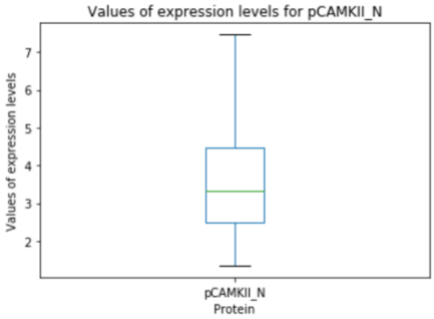


Figure 3: Box Plot for the Values of Expression Levels against pCAMKII_N protein

Besides, box plot was used to explore the values of expression levels for pCREB_N protein. The box plot regarding the values of expression levels for pCREB_N showed us a median of 0.210681, Q1 of 0.190828, Q3 of 0.234558. The result indicates that protein pCREB_N are mostly present with a very low value of expression level compared to other proteins. Moreover, there are some outliers present above the upper whisker and below the bottom whisker.

Additionally, box plot was used to explore the values of expression levels for pELK_N protein. The box plot regarding the values of expression levels for pELK_N showed us a median of 1.356368, Q1 of 1.206389, Q3 of 1.560931. The result indicates that protein pELK_N are mostly present with a low value of expression level compared to other proteins. Moreover, there are some outliers present above the upper whisker and below the bottom whisker.

4.2 Explore the relationship between pairs of attributes

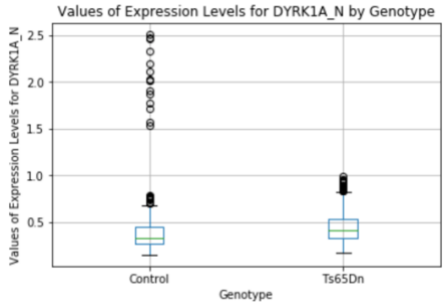


Figure 4: Side-by-side Box Plot for the Values of Expression Levels for DYRK1A_N by Genotype

The hypothesis for the values of expression levels for DYRK1A_N by genotype is Ts65Dn (trisomy) genotype is more likely to have higher expression level compared to the control genotype. Hypothesis is accepted because the Ts65Dn genotype shows higher value of expression levels for DYRK1A_N judging from the median and interquartile range on the side-by-side box plot compared to Control genotype. Hence, it proves that there is a relationship between values of expression levels for DYRK1A_N by genotype.

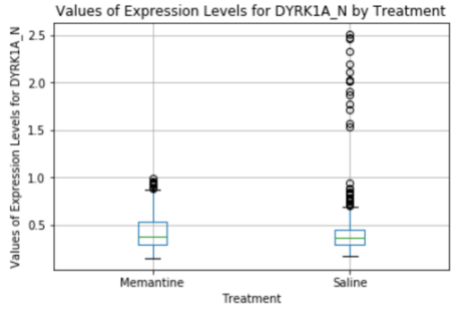


Figure 5: Side-by-side Box Plot for the Values of Expression Levels for DYRK1A_N by Treatment

The hypothesis for the values of expression levels for DYRK1A_N by treatment is Memantine treatment is more likely to have higher expression level compared to the saline treatment. Hypothesis is accepted even though both treatments show almost identical median and interquartile range on the side-by-side box plot for DYRK1A_N, but Memantine has slightly higher upper quartile and upper whisker compared to Saline. However, there is still a lack in relationship between values of expression levels for DYRK1A_N by treatment.

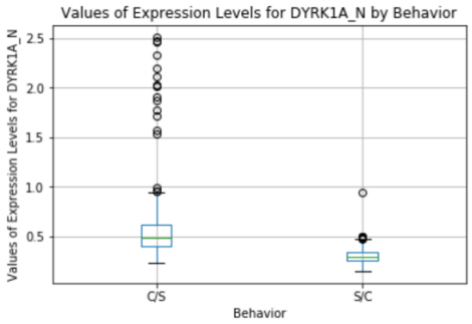


Figure 6: Side-by-side Box Plot for the Values of Expression Levels for DYRK1A_N by Behavior

The hypothesis for the values of expression levels for DYRK1A_N by behavior is context-shock (C/S) behavior is more likely to have higher expression level compared to the shock-context (S/C). Hypothesis is accepted because the context-shock behavior shows higher value of expression levels for DYRK1A_N judging from the median and interquartile range on the side-by-side box plot compared to shock-context. Hence, it proves that there is a relationship between values of expression levels for DYRK1A_N by behavior.

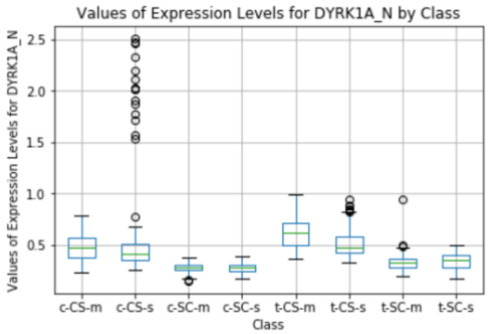


Figure 7: Side-by-side Box Plot for the Values of Expression Levels for DYRK1A_N by Class

The hypothesis for the values of expression levels for DYRK1A_N by class is trisomy mice, stimulated to learn, injected with memantine (t-CS-m) is more likely to have higher expression level compared to the other classes. Hypothesis is accepted because the t-CS-m class shows higher value of expression levels for DYRK1A_N judging from the median and interquartile range on the side-by-side box plot compared to other classes. Hence, it proves that there is a relationship between values of expression levels for DYRK1A_N by class. Moreover, It further strengthen and proves that our previous hypothesis along the justification conducted for the relationship between DYRK1A_N on genotype, treatment and behaviour are correct.

Besides, the hypothesis for the values of expression levels for GFAP_N by genotype is Ts65Dn (trisomy) genotype is more likely to have higher expression level compared to the control genotype. Hypothesis is rejected because both genotypes have almost identical values of expression levels for GFAP_N judging from the median and interquartile range on the side-by-side box plot. Hence, it proves that there is a lack in relationship between values of expression levels for GFAP_N by genotype.

The hypothesis for the values of expression levels for GFAP_N by treatment is Saline treatment is more likely to have higher expression level compared to the Memantine treatment. Hypothesis is accepted because the Saline treatment shows higher value of expression levels for GFAP_N judging from the median and interquartile range on the side-by-side box plot for GFAP_N compared to Memantine. Hence, it proves that there is a relationship between values of expression levels for GFAP_N by treatment.

The hypothesis for the values of expression levels for GFAP_N by behavior is context-shock (C/S) behavior is more likely to have higher expression level compared to the shock-context (S/C). Hypothesis is accepted because the context-shock behavior shows higher value of expression levels for GFAP_N judging from the median and interquartile range on the side-by-side box plot compared to shock-context. Hence, it proves that there is a relationship between values of expression levels for GFAP_N by behavior.

The hypothesis for the values of expression levels for GFAP_N by class is trisomy mice, stimulated to learn, injected with saline (t-SC-s) is more likely to have higher expression level compared to the other classes. Hypothesis is accepted because the t-SC-s class shows higher value of expression levels for GFAP_N judging from the median and interquartile range on the side-by-side box plot compared to other classes, although c-CS-s is just slightly below. Hence, it proves that there is a relationship between values of expression levels for GFAP_N by class. Moreover, It further strengthen and proves that our previous hypothesis along the justification conducted for the relationship between GFAP_N on genotype, treatment and behaviour are correct.

In addition, the hypothesis for the values of expression levels for NR2A_N by genotype is control genotype is more likely to have higher expression level compared to the Ts65Dn (trisomy). Hypothesis is accepted because the control genotype shows higher value of expression levels for NR2A_N judging from the median and interquartile range on the side-by-side box plot compared to trisomy genotype. Hence, it proves that there is a relationship between values of expression levels for NR2A_N by genotype.

The hypothesis for the values of expression levels for NR2A_N by behavior is context-shock (C/S) behavior is more likely to have higher expression level compared to the shock-context (S/C). Hypothesis is accepted because the context-shock behavior shows higher value of expression levels for NR2A_N judging from the median and interquartile range on the side-by-side box plot compared to shock-context. Hence, it proves that there is a relationship between values of expression levels for NR2A_N by behavior.

5.0 Data Modeling

5.1 Generating Train/Test Set

We will be generating the train and test set so that we can make them easily accessible while training in different model, under different parameters and conditions. The X target of all the protein columns and y target of the class column are set in this process. The train and test are generated as X_train, X_test, y_train and y_test using sklearn train_test_split() method, where the test size is set to 40%. We will also be checking the train and test sets carefully to ensure that they are correct and ready to be trained in various conditions. Furthermore, the dimension /shape of the train and test sets are checked as well to make sure that they are correctly split.

5.2 Feature Selection

Feature selection is performed to select the "best" proteins based on the importance retrieved using the Hill Climbing Technique. Feature engineering and selection is necessary to reduce overfitting, enhancing the efficiency of the Machine Learning algorithm, improving accuracy for most of the model as it reduces the complexity and redundancy of data for the model to interpret. Hill climbing technique is used because it is a powerful yet simple algorithm to feature select the best attributes for our model as it examines the neighboring features one by one and selects the first neighboring feature which optimizes the current cost as next feature.

From the results shown on the feature selection using Hill Climbing Technique, we found that there are 37 features to be selected as our important proteins with their indexes provided by the algorithm, which show the best score. After that, 60% training and 40% test set are generated using only the important proteins as the X target with the train_test_split() function, so that they can be used for training a model easily when needed. Moreover, the generated train and test set were carefully checked to ensure that they are correct in sizes after splitting as well as to make sure it is the correct feature selected proteins from the index list provided from the Hill Climbing Technique.

5.3 Model execution

5.3.1 K-Nearest Neighbors Classification

In this section, the kNN classifier with default value of 5 n_neighbors using all proteins and kNN classifier with default value of 5 n_neighbors using only the important proteins were executed and trained, in order to have a clear comparison of the results between training the model using all proteins and using only important proteins on default values. The default value is used for the parameters first because it will help us to indicate the base results.

```
[[57  0  0  0  1  0  0  0]
 [ 7 42  0  0  0  4  0  0]
 [ 0  0 59  2  0  0  1  0]
 [ 0  0  0 54  0  0  4  0]
 [ 8  1  0  0 48  1  0  1]
 [ 4  1  0  0  2 36  0  0]
 [ 0  0  5  0  0  0 41  0]
 [ 0  0  5  1  0  0  1 46]]
```

	precision	recall	f1-score	support
c-CS-m	0.75	0.98	0.85	58
c-CS-s	0.95	0.79	0.87	53
c-SC-m	0.86	0.95	0.90	62
c-SC-s	0.95	0.93	0.94	58
t-CS-m	0.94	0.81	0.87	59
t-CS-s	0.88	0.84	0.86	43
t-SC-m	0.87	0.89	0.88	46
t-SC-s	0.98	0.87	0.92	53
accuracy			0.89	432
macro avg	0.90	0.88	0.89	432
weighted avg	0.90	0.89	0.89	432

```
Accuracy: 0.8865740740740741
```

Figure 8: Confusion Matrix, classification report, accuracy score for kNN classification trained using all proteins

```
[[57  1  0  0  0  0  0  0]
 [ 2 49  0  0  0  2  0  0]
 [ 0  0 60  2  0  0  0  0]
 [ 0  0  0 58  0  0  0  0]
 [ 0  0  0  0 51  8  0  0]
 [ 4  2  0  0  1 36  0  0]
 [ 0  0  4  1  0  0 41  0]
 [ 1  0  0  2  0  0  0 50]]
```

	precision	recall	f1-score	support
c-CS-m	0.89	0.98	0.93	58
c-CS-s	0.94	0.92	0.93	53
c-SC-m	0.94	0.97	0.95	62
c-SC-s	0.92	1.00	0.96	58
t-CS-m	0.98	0.86	0.92	59
t-CS-s	0.78	0.84	0.81	43
t-SC-m	1.00	0.89	0.94	46
t-SC-s	1.00	0.94	0.97	53
accuracy			0.93	432
macro avg	0.93	0.93	0.93	432
weighted avg	0.93	0.93	0.93	432

```
Accuracy: 0.9305555555555556
```

Figure 9: Confusion Matrix, classification report, accuracy score for kNN classification trained using only important proteins

The kNN classification with default value of 5 n_neighbors along training using only the important proteins shows an overall better result compared to the kNN classification with default value of 5 n_neighbors along training using all proteins. The confusion matrix for the feature selected proteins shows us higher value in the diagonal line compared to the confusion matrix for the non-feature selected proteins. This shows that the classification error rate is lower on the feature selected model. The kNN classification trained using only the important proteins shows better precision of 0.93, recall of 0.93 and f1-score of 0.93 compared to the kNN classification trained using all proteins with a precision of 0.90, recall of 0.89 and f1-score of 0.89. The kNN classification trained using only the important proteins has a higher accuracy at 0.93 compared to 0.89 for the kNN classification trained using all proteins.

Comparison between training using all protein and feature selected important protein on the kNN classifier using default value gives us a better picture on how the model will perform under feature or non-feature selected protein as well as providing us crucial information and results. Also, the comparison gives us valuable insight of the classification error rate from the confusion matrix, precision which is the fraction of correctly predicted instances, recall which is the fraction of relevant instances that are successfully predicted and F1-score which is the harmonic mean of precision and recall, from the classification report.

5.3.2 Parameter Tuning

The parameter tuning was commenced to improve the results and accuracy of a model. Firstly, we will be tuning the weights parameter to 'distance', which is the weight points by the inverse of their distance, where closer neighbors of a query point will have a greater influence than neighbors which are further away, instead of the neighbors being weighted in uniform or equally. This parameter is chosen because it is expected to see an increase for the accuracy and improved results by tuning the weights parameter to 'distance'.

```
Confusion Matrix:

[[58  0  0  0  0  0  0  0]
 [ 6 44  0  0  0  3  0  0]
 [ 0  0 59  2  0  0  1  0]
 [ 0  0  0 56  0  0  2  0]
 [ 6  0  0  0 52  0  0  1]
 [ 1  0  0  0  0 42  0  0]
 [ 0  0  3  0  0  0 43  0]
 [ 0  0  1  0  0  0  0 52]]

Classification Report:
```

	precision	recall	f1-score	support
c-CS-m	0.82	1.00	0.90	58
c-CS-s	1.00	0.83	0.91	53
c-SC-m	0.94	0.95	0.94	62
c-SC-s	0.97	0.97	0.97	58
t-CS-m	1.00	0.88	0.94	59
t-CS-s	0.93	0.98	0.95	43
t-SC-m	0.93	0.93	0.93	46
t-SC-s	0.98	0.98	0.98	53
accuracy			0.94	432
macro avg	0.95	0.94	0.94	432
weighted avg	0.95	0.94	0.94	432

```
Accuracy: 0.9398148148148148
```

```
Confusion Matrix:

[[56  0  0  0  0  2  0  0]
 [ 2 48  0  0  0  3  0  0]
 [ 0  0 62  0  0  0  0  0]
 [ 0  0  0 58  0  0  0  0]
 [ 0  0  0  0 52  7  0  0]
 [ 1  0  0  0  0 42  0  0]
 [ 0  0  0  0  0  0 46  0]
 [ 0  0  0  1  0  0  0 52]]

Classification Report:
```

	precision	recall	f1-score	support
c-CS-m	0.95	0.97	0.96	58
c-CS-s	1.00	0.91	0.95	53
c-SC-m	1.00	1.00	1.00	62
c-SC-s	0.98	1.00	0.99	58
t-CS-m	1.00	0.88	0.94	59
t-CS-s	0.78	0.98	0.87	43
t-SC-m	1.00	1.00	1.00	46
t-SC-s	1.00	0.98	0.99	53
accuracy			0.96	432
macro avg	0.96	0.96	0.96	432
weighted avg	0.97	0.96	0.96	432

```
Accuracy: 0.9629629629629629
```

Figure 10: Confusion Matrix, classification report, accuracy score for kNN classification trained using all protein(left) and only important proteins (right) with weights parameter set to 'distance'.

As expected, the results had improved for both models from training using all proteins and only the important proteins, by tuning the parameter for weights as 'distance'. Each of the confusion matrix from both models shows higher value on the diagonal line, thus lower classification error rate after tuning the parameter. Both of the models also show improved results of precision, recall and f1-score as well as the accuracy shown from the classification report. In this comparison, the kNN model trained using only the important proteins with the weights set as 'distance' has better result and accuracy compared to the model using all proteins.

Secondly, we will be tuning the p value to 1, which is the power parameter for the Minkowski metric, where it will be equivalent to using manhattan_distance (l1) in Taxicab geometry. This parameter is chosen because it is expected to see an increase for the accuracy and improved results by tuning the p value to 1 and preserving the previous tuned parameter.

Confusion Matrix:

```
[[58  0  0  0  0  0  0  0]
 [ 2 49  0  0  0  2  0  0]
 [ 0  0 60  1  0  0  1  0]
 [ 0  0  0 58  0  0  0  0]
 [ 6  0  0  0 52  0  0  1]
 [ 1  0  0  0  0 42  0  0]
 [ 0  0  1  0  0  0 45  0]
 [ 0  0  1  1  0  0  0 51]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.87	1.00	0.93	58
c-CS-s	1.00	0.92	0.96	53
c-SC-m	0.97	0.97	0.97	62
c-SC-s	0.97	1.00	0.98	58
t-CS-m	1.00	0.88	0.94	59
t-CS-s	0.95	0.98	0.97	43
t-SC-m	0.98	0.98	0.98	46
t-SC-s	0.98	0.96	0.97	53
accuracy			0.96	432
macro avg	0.96	0.96	0.96	432
weighted avg	0.96	0.96	0.96	432

Accuracy: 0.9606481481481481

Confusion Matrix:

```
[[58  0  0  0  0  0  0  0]
 [ 0 53  0  0  0  0  0  0]
 [ 0  0 62  0  0  0  0  0]
 [ 0  0  0 58  0  0  0  0]
 [ 4  2  0  0 53  0  0  0]
 [ 0  0  0  0  0 43  0  0]
 [ 0  0  0  0  0  0 46  0]
 [ 0  0  0  1  0  0  0 52]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.94	1.00	0.97	58
c-CS-s	0.96	1.00	0.98	53
c-SC-m	1.00	1.00	1.00	62
c-SC-s	0.98	1.00	0.99	58
t-CS-m	1.00	0.90	0.95	59
t-CS-s	1.00	1.00	1.00	43
t-SC-m	1.00	1.00	1.00	46
t-SC-s	1.00	0.98	0.99	53
accuracy			0.98	432
macro avg	0.99	0.98	0.98	432
weighted avg	0.98	0.98	0.98	432

Accuracy: 0.9837962962962963

Figure 11: Confusion Matrix, classification report, accuracy score for kNN classification trained using all protein(left) and only important proteins (right) with weights parameter set to 'distance' and p value to 1.

As expected, the results had improved for both models from training using all proteins and only the important proteins, by tuning the parameter for weights as 'distance' and p value to 1. Each of the confusion matrix from both models shows higher value on the diagonal line, thus lower classification error rate after tuning the parameters. Both of the models also show improved results of precision, recall and f1-score as well as the accuracy shown from the classification report. In this comparison, the kNN model trained using only the important proteins with the weights set as 'distance' and p value to 1 has better result and accuracy compared to the model using all proteins.

After that, we will be tuning the n_neighbors to 3, which is number of neighbors, where lower value makes the classification boundaries more distinct. Thus, it will make the boundaries to be more concise and able to improve the result in correct condition. This parameter is chosen because it is expected to see an increase for the accuracy and improved results by tuning the n_neighbors to 3 and preserving the previous tuned parameters.

Confusion Matrix:

```
[[58  0  0  0  0  0  0  0]
 [ 1 48  0  0  1  3  0  0]
 [ 0  0 61  0  0  0  1  0]
 [ 0  0  0 58  0  0  0  0]
 [ 6  0  0  0 51  1  0  1]
 [ 1  0  0  0  0 42  0  0]
 [ 0  0  0  0  0  0 46  0]
 [ 0  0  1  0  0  0  0 52]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.88	1.00	0.94	58
c-CS-s	1.00	0.91	0.95	53
c-SC-m	0.98	0.98	0.98	62
c-SC-s	1.00	1.00	1.00	58
t-CS-m	0.98	0.86	0.92	59
t-CS-s	0.91	0.98	0.94	43
t-SC-m	0.98	1.00	0.99	46
t-SC-s	0.98	0.98	0.98	53
accuracy			0.96	432
macro avg	0.96	0.96	0.96	432
weighted avg	0.97	0.96	0.96	432

Accuracy: 0.9629629629629629

Confusion Matrix:

```
[[58  0  0  0  0  0  0  0]
 [ 0 52  0  0  0  1  0  0]
 [ 0  0 62  0  0  0  0  0]
 [ 0  0  0 58  0  0  0  0]
 [ 3  2  0  0 54  0  0  0]
 [ 0  0  0  0  0 43  0  0]
 [ 0  0  0  0  0  0 46  0]
 [ 0  0  0  0  0  0  0 53]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.95	1.00	0.97	58
c-CS-s	0.96	0.98	0.97	53
c-SC-m	1.00	1.00	1.00	62
c-SC-s	1.00	1.00	1.00	58
t-CS-m	1.00	0.92	0.96	59
t-CS-s	0.98	1.00	0.99	43
t-SC-m	1.00	1.00	1.00	46
t-SC-s	1.00	1.00	1.00	53
accuracy			0.99	432
macro avg	0.99	0.99	0.99	432
weighted avg	0.99	0.99	0.99	432

Accuracy: 0.9861111111111112

Figure 12: Confusion Matrix, classification report, accuracy score for kNN classification trained using all protein(left) and only important proteins (right) with weights parameter set to 'distance', p value to 1 and n_neighbors to 3

As expected, the results had improved for both models from training using all proteins and only the important proteins, by tuning the parameters for n_neighbors to 3, weights as 'distance' and p value to 1. Each of the confusion matrix from both models shows higher value on the diagonal line, thus lower classification error rate

after tuning the parameters. The model trained using only the important proteins also show improved results of precision, recall and f1-score as well as the accuracy shown from the classification report, while the model trained using all proteins show improved results of precision and accuracy. In this comparison, the kNN model trained using only the important proteins with the n_neighbors set to 3, weights as 'distance' and p value to 1 has better result and accuracy compared to the model using all proteins.

Summary / Recommendation:
It is proven from the results that training a model using only the feature selected important proteins always provide better results and accuracy. From the parameter tuning, we found that training a kNN classification using important proteins with the parameters of n_neighbors set as 3, weights as 'distance', ‘p’ value as 1, produces the best results and accuracy. Therefore, it is recommended to use the parameters of n_neighbors set as 3, weights as 'distance', ‘p’ value as 1 using feature selected proteins for kNN classification.

5.4.1 Decision Tree Classification

In this section, the Decision Tree classifier with default values using all proteins and Decision Tree classifier with default values using only the important proteins will be executed and trained, in order to have a clear comparison of the results between training the model using all proteins and using only important proteins on default values. We will start by using the default values because it helps us to indicate the base results.

```
[[46  7  0  0  5  0  0  0]
 [ 5 37  0  0  4  7  0  0]
 [ 4  0 46  5  1  0  5  1]
 [ 0  0  3 50  0  0  2  3]
 [ 1 11  0  0 44  3  0  0]
 [ 0  1  0  0  3 39  0  0]
 [ 0  1  7  4  0  0 34  0]
 [ 0  0  2  0  0  0  3 48]]
```

	precision	recall	f1-score	support
c-CS-m	0.82	0.79	0.81	58
c-CS-s	0.65	0.70	0.67	53
c-SC-m	0.79	0.74	0.77	62
c-SC-s	0.85	0.86	0.85	58
t-CS-m	0.77	0.75	0.76	59
t-CS-s	0.80	0.91	0.85	43
t-SC-m	0.77	0.74	0.76	46
t-SC-s	0.92	0.91	0.91	53
accuracy			0.80	432
macro avg	0.80	0.80	0.80	432
weighted avg	0.80	0.80	0.80	432

```
Accuracy: 0.7962962962962963
```

Figure 13: Confusion Matrix, classification report, accuracy score for decision tree classification trained using all proteins

```
Confusion Matrix:

[[45  8  0  0  5  0  0  0]
 [ 6 38  0  0  6  3  0  0]
 [ 3  0 49  3  1  0  5  1]
 [ 0  0  3 53  0  0  2  0]
 [ 6  7  0  0 41  3  0  2]
 [ 2  1  0  0  1 39  0  0]
 [ 1  0  5  4  0  0 35  1]
 [ 1  0  0  0  0  0  1 51]]

Classification Report:
```

	precision	recall	f1-score	support
c-CS-m	0.70	0.78	0.74	58
c-CS-s	0.70	0.72	0.71	53
c-SC-m	0.86	0.79	0.82	62
c-SC-s	0.88	0.91	0.90	58
t-CS-m	0.76	0.69	0.73	59
t-CS-s	0.87	0.91	0.89	43
t-SC-m	0.81	0.76	0.79	46
t-SC-s	0.93	0.96	0.94	53
accuracy			0.81	432
macro avg	0.81	0.82	0.81	432
weighted avg	0.81	0.81	0.81	432

```
Accuracy: 0.8125
```

Figure 14: Confusion Matrix, classification report, accuracy score for decision tree classification trained using only important proteins

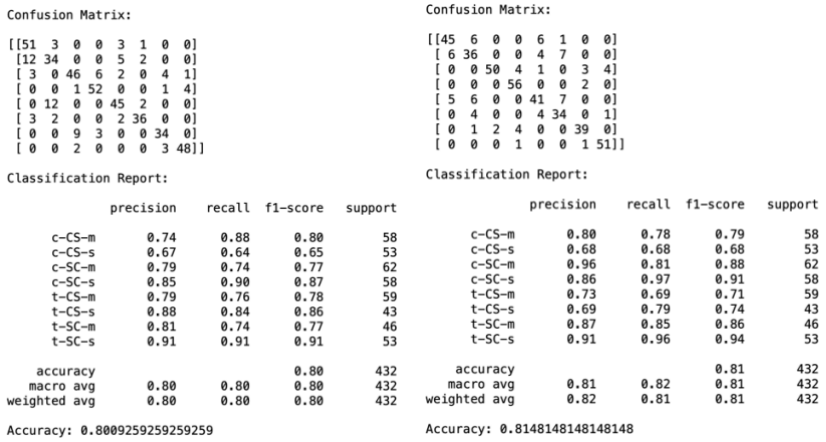
The decision tree classification with default values along training using only the important proteins shows an overall better result compared to the decision tree classification with default values along training using all proteins. The confusion matrix for the feature selected proteins shows us higher value in the diagonal line compared to the confusion matrix for the non-feature selected proteins. This shows that the classification error rate is lower on the feature selected model. The decision tree classification trained using only the important proteins shows better precision of 0.81, recall of 0.81 and f1-score of 0.81 compared to the decision tree classification trained using all proteins with a precision of 0.80, recall of 0.80 and f1-score of 0.80. The decision tree classification trained using only the important proteins has a higher accuracy at 0.8125 compared to 0.7963 for the decision tree classification trained using all proteins.
Justifications:

Comparison between training using all protein and feature selected important protein on the decision tree classifier using default values gives us a better picture on how the model will perform under feature or non-feature selected protein as well as providing us crucial information and results. Also, the comparison gives us valuable insight of the classification error rate from the confusion matrix, precision which is the fraction of correctly predicted instances, recall which is the fraction of relevant instances that are successfully predicted, F1-score which is the harmonic mean of precision and recall from the classification report.

5.4.2 Parameter Tuning

The parameter tuning was commenced to improve the results and accuracy of a model. Firstly, we will be tuning the class_weight parameter from None to 'balanced', which automatically adjust weights inversely

proportional to class frequencies in the input data, where all classes are weighted equally. This parameter is chosen because it is expected to see an increase for the accuracy and improved results by tuning the class_weight to 'balanced'. Additionally, other parameters tested provide worse results for our decision tree classifications.



Confusion Matrix:

```
[[51  3  0  0  3  1  0  0]
 [12 34  0  0  5  2  0  0]
 [ 3  0 46  6  2  0  4  1]
 [ 0  0  1 52  0  0  1  4]
 [ 0 12  0  0 45  2  0  0]
 [ 3  2  0  0  2 36  0  0]
 [ 0  0  9  3  0  0 34  0]
 [ 0  0  2  0  0  0  3 48]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.74	0.88	0.80	58
c-CS-s	0.67	0.64	0.65	53
c-SC-m	0.79	0.74	0.77	62
c-SC-s	0.85	0.90	0.87	58
t-CS-m	0.79	0.76	0.78	59
t-CS-s	0.88	0.84	0.86	43
t-SC-m	0.81	0.74	0.77	46
t-SC-s	0.91	0.91	0.91	53
accuracy			0.80	432
macro avg	0.80	0.80	0.80	432
weighted avg	0.80	0.80	0.80	432

Accuracy: 0.8009259259259259

Confusion Matrix:

```
[[45  6  0  0  6  1  0  0]
 [ 6 36  0  0  4  7  0  0]
 [ 0  0 50  4  1  0  3  4]
 [ 0  0  0 56  0  0  2  0]
 [ 5  6  0  0 41  7  0  0]
 [ 0  4  0  0  4 34  0  1]
 [ 0  1  2  4  0  0 39  0]
 [ 0  0  0  1  0  0  1 51]]
```

Classification Report:

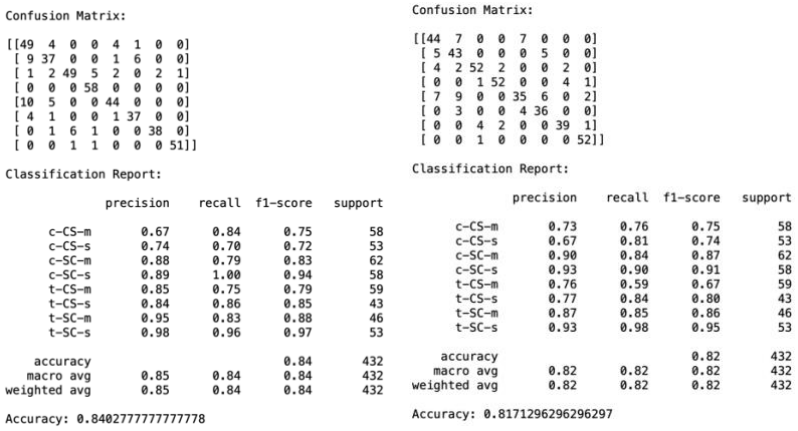
	precision	recall	f1-score	support
c-CS-m	0.80	0.78	0.79	58
c-CS-s	0.68	0.68	0.68	53
c-SC-m	0.96	0.81	0.88	62
c-SC-s	0.86	0.97	0.91	58
t-CS-m	0.73	0.69	0.71	59
t-CS-s	0.69	0.79	0.74	43
t-SC-m	0.87	0.85	0.86	46
t-SC-s	0.91	0.96	0.94	53
accuracy			0.81	432
macro avg	0.81	0.82	0.81	432
weighted avg	0.82	0.81	0.81	432

Accuracy: 0.8148148148148148

Figure 15: Confusion Matrix, classification report, accuracy score for decision tree classification trained using all proteins (left) only important proteins (right) with class_weight parameter set to 'balanced'

The results had improved slightly for both models from training using all proteins and only the important proteins, by tuning the parameter for class_weight from None to 'balanced'. Each of the confusion matrix from both models shows higher value on the diagonal line, thus lower classification error rate after tuning the parameters. The decision tree model trained using all proteins shows the same precision, recall and f1-score but with accuracy improved from 0.7963 to 0.8009, whiile the model trained using important proteins shows higher precision as well as acuracy from 0.8125 to 0.8148. In this comparison, the decision tree model trained using only the important proteins with the class_weight set to 'balanced' has better result and accuracy compared to the model using all proteins.

After that, we will be tuning the criterion from 'gini' to 'entropy', which the function to measure the quality of a split, where 'entropy' is used for the information gain. This parameter is chosen because it is expected to see an increase for the accuracy and improved results by tuning the criterion from 'gini' to 'entropy'. Additionally, other parameters tested provide worse results for our decision tree classifications.



Confusion Matrix:

```
[[49  4  0  0  4  1  0  0]
 [ 9 37  0  0  1  6  0  0]
 [ 1  2 49  5  2  0  2  1]
 [ 0  0  0 58  0  0  0  0]
 [10  5  0  0 44  0  0  0]
 [ 4  1  0  0  1 37  0  0]
 [ 0  1  6  1  0  0 38  0]
 [ 0  0  1  1  0  0  0 51]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.67	0.84	0.75	58
c-CS-s	0.74	0.70	0.72	53
c-SC-m	0.88	0.79	0.83	62
c-SC-s	0.89	1.00	0.94	58
t-CS-m	0.85	0.75	0.79	59
t-CS-s	0.84	0.86	0.85	43
t-SC-m	0.95	0.83	0.88	46
t-SC-s	0.98	0.96	0.97	53
accuracy			0.84	432
macro avg	0.85	0.84	0.84	432
weighted avg	0.85	0.84	0.84	432

Accuracy: 0.8402777777777778

Confusion Matrix:

```
[[44  7  0  0  7  0  0  0]
 [ 5 43  0  0  0  5  0  0]
 [ 4  2 52  2  0  0  2  0]
 [ 0  0  1 52  0  0  4  1]
 [ 7  9  0  0 35  6  0  2]
 [ 0  3  0  0  4 36  0  0]
 [ 0  0  4  2  0  0 39  1]
 [ 0  0  1  0  0  0  0 52]]
```

Classification Report:

	precision	recall	f1-score	support
c-CS-m	0.73	0.76	0.75	58
c-CS-s	0.67	0.81	0.74	53
c-SC-m	0.90	0.84	0.87	62
c-SC-s	0.93	0.90	0.91	58
t-CS-m	0.76	0.59	0.67	59
t-CS-s	0.77	0.84	0.80	43
t-SC-m	0.87	0.85	0.86	46
t-SC-s	0.93	0.98	0.95	53
accuracy			0.82	432
macro avg	0.82	0.82	0.82	432
weighted avg	0.82	0.82	0.82	432

Accuracy: 0.8171296296296297

Figure 15: Confusion Matrix, classification report, accuracy score for decision tree classification trained using all proteins (left) only important proteins (right) with class_weight parameter set to 'balanced' and gini to ‘entropy’

The results had improved significantly for the decision tree model from training using all proteins and slightly for the model from training using only the important proteins, by tuning the parameter for class_weight from None to 'balanced' and criterion from 'gini' to 'entropy'. Each of the confusion matrix from both models shows higher value on the diagonal line, thus lower classification error rate after tuning the parameters. The decision tree model trained using all proteins shows higher precision, recall and f1-score as well as accuracy improved from 0.8009 to 0.8403, whiile the model trained using important proteins shows higher recall and f1-score as well as acuracy from 0.8148 to 0.8171. In this comparison, the result is surprising as the decision tree model trained using all proteins with the class_weight set as 'balanced' and criterion set as 'entropy' has better result and accuracy compared to the model using important proteins.

Summary / Recommendation:

It is proven from the results that training a decision tree classification using only the feature selected important proteins and using all proteins provide different results and accuracy under different conditions. Thus, it is inappropriate to determine that using the important proteins always provide better results compared to all

proteins. From the parameter tuning, we found that training a decision tree classification using all proteins with the parameters of class_weight set as 'balanced' and criterion as 'entropy', produces the best results and accuracy. Therefore, it is recommended to use the parameters of class_weight set as 'balanced' and criterion as 'entropy' without using the feature selected proteins for decision tree classification.

6.0 Conclusion and comparison

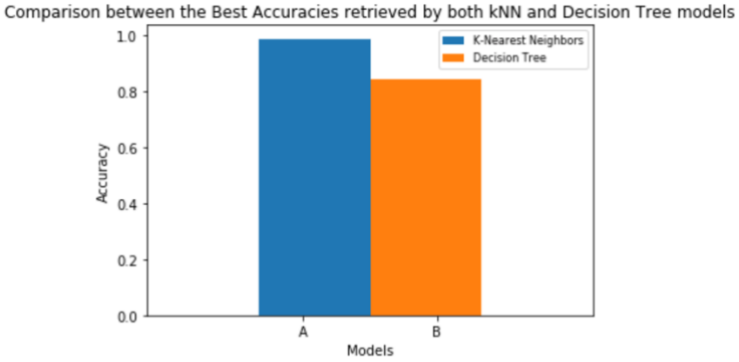


Figure 15: Bar graph for the comparison between the Best Accuracies retrieved by both kNN and Decision Tree models

From all the results retrieved by both kNN and Decision Tree models, the kNN model with the parameters of n_neighbors set as 3, weights as 'distance', 'p' value as 1 trained using only important proteins, shows the highest accuracy and best result of precision, recall and f1 score. It presented an accuracy of 0.99, which is almost 100% accurate, compared to the decision tree model with parameters of class_weight set as 'balanced' and criterion as 'entropy' trained using all proteins presented an accuracy of 0.84. Therefore, the kNN model will be my recommendation to identify subsets of proteins that are discriminant between the classes.

References:

https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression
https://en.wikipedia.org/wiki/Down_syndrome
https://www.medscape.com/answers/943216-181104/what-is-the-prevalence-of-down-syndrome
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2657943/
https://rmit.instructure.com/courses/67430/files/11273016?module_item_id=2268981
https://rmit.instructure.com/courses/67430/files/11552799?module_item_id=2314376
https://rmit.instructure.com/courses/67430/files/11273015?module_item_id=2268982