MICE PROTEIN EXPRESSION ANALYSIS TO IDENTIFY THE PROTEINS CRITICAL TO LEARNING ABILITY IN MICE USING CLASSIFICATION ALGORITHMS.

A REPORT SUBMITTED IN THE FULFILMENT OF THE REQUIREMENTS FOR THE ASSIGNMENT

OF

PRACTICAL DATA SCIENCE

WITH

PYTHON

BY

VISHAL D. PATNAIK

S3811521

IN

MASTERS

OF

DATA SCIENCE

AT

ROYAL MELBOURNE INSTITUTE OF TECHNOLOGY

MELBOURNE, AUSTRALIA

S3811521@STUDENT.RMIT.EDU.AU

2020 - 06 - 06

TABLE OF CONTENTS

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 honour code by typing "Yes": Yes.

ABSTRACT

*The mice protein expression dataset was created in - order to study the learning effect between **"Normal"** and **"Trisomic"** or Down Syndrome mice. i.e., type of the mice (Genotype). Down syndrome is the disorder where there is an extra chromosome in organisms. Which is known as chromosome 21 or trisomy 21. This effect the alteration of metabolic pathways by causing learning and memory deficiencies. The treatment for this learning deficiency is "Memantine". Here, in this assignment after cleaning the dataset we get some facts or plausibilities from the descriptive and visualised analysis then we will predict the critical proteins for the learning ability in mice when exposed to CFC. For this, 2 supervised machine learning tasks are implemented, and the data is classified initially using the classifiers. KNN and Decision tree are used to classify the proteins. The decision tree added flavours of predicting the crucial mice protein that affects the learning ability recovery in mice. Expression levels of proteins accounting to 77 among "Control" and "Trisomic" type are analysed in 2 conditions based on drug (Memantine) influence. One with the drug influence and two without. From the results we can say that KNN is more efficient in classifying data. But decision tree comes a bit forward in predicting the most crucial mice proteins. This helps in the development or identification of mice learning ability drugs. So that the deficiencies related to learning and memory can be overcome.*

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
DS	DOWN SYNDROME
CFC	CONTEXT FREE CONDITIONING
DNA	DEOXY RIBO NUCLEIC ACID
KNN	K NEAREST NEIGHBOR
CS	CONTEXT SHOCK
SC	SHOCK CONTEXT

INTRODUCTION

The human interior bodies consist of many cells, and each cell consists of different types of 46 chromosomes. The body event by encoding the amino acid sequence in proteins shall be determined by the DNA captured in chromosomes. In cells of people with DS 47 chromosomes are present, instead of 46. the additional chromosome is named the trisomy of human chromosome 21 (Hsa21) The symptoms a trisomies like Ds are the results of an over expression of proteins encoded on extra chromosome. Approximately 0.45% of human conceptions are trisomic for Ha21. Trisomy of Ha21 is related to a mild-to-moderate learning disorder, hypotonia in early infancy & craniofacial abnormalities.

To comprehend Ds, we would like to know the genome Trisomy protein content of Ha21 and to judge how the expression levels of those genes are altered by presence of 3rd copy of Ha21. for instance, trisomy Protein of a neuronal channel proteins like GIRK2 may influence learning in people with DS.

In this assignment, testing of 72 expression levels of protein modifications that are critical to successful and failed learning is done and the entire 72 mice it gathered for the protein production which produced detectable signals in nuclear fraction of cortex. there have been 38 control/normal mice and 34 trisomic mice which are trained in CFC. The process of CFC requires these mice into 2 groups, context-shock (CS) and shock- context (SC). First mice from CS group are placed during a cage, allowed to explore cage, and given a quick electric shock, whereas trisomic mice are excluded. Secondly, mice in SC group are placed during a cage and given a proper electric shock, therefore both the groups of mice failed to tell the association between cage and shock.

The assistance of trisomic mice in learning is completed by injecting memantine in before training. The effect of the injection is controlled by injecting memantine to 1/2 mice in CS and SC groups, while the other group is injected with saline. 15 measurements are being conducted for every of proteins. Therefore, there are 15×38 (570 measurements) for control/ normal mice and 15×34(510 measurements) for trisomic mice. Totally, this dataset has 1080 measurements from 72 mice in each expression level of proteins.

Finally, the goal during this project is to understand the Success and failure of mice learning which contribute the trisomy protein classes. By creating a model prediction of the 8 classes mice is completed where some mice were supported their protein expression level. We can determine significant proteins within the predictions, which might support a hypothesis of that specific Protein that affects learning in trisomic mice by creating a successful model.

PROBLEM STATEMENT

Mice Protein Expression analysis to identify the proteins critical to learning ability in mice using classification algorithms.

METHODOLOGY

DATA

- ***Data Set Information:***
  Expression levels of 77 proteins/protein modifications that produced detectable signals in the nuclear fraction of cortex.
  Control mice - 38
  Trisomic mice (Down syndrome) - 34.
  15 measurements of each protein per sample/mouse are registered in the dataset.
  Resulting in Control mice - 38x15 - 570 and Trisomic mice - 34x15 - 510 measurements.
  The dataset contains a total of 1080 measurements per protein.
  Genotype - Control or Trisomic.
  Behavior - Stimulated to learn (context-shock) and others have not (shock-context) and 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 drug and others have not.
- ***Classes: 8 in number***
  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)
- ***Attribute Information:***
  1 : Mouse ID
  2 – 78 : Values of expression levels of 77 proteins. For example: DYRK1A_n
  79 : Genotype: control (c) or trisomy (t)
  80 : Treatment type: memantine (m) or saline (s)
  81 : Behavior: context-shock (CS) or shock-context (SC)
  82 : Class: c-CS-s, c-CS-m, c-SC-s, c-SC-m, t-CS-s, t-CS-m, t-SC-s, t-SC-m

In this assignment we mainly undergo 3 steps:

- *Data Pre - processing*
- *Data Exploration*
- *Modelling the data.*

DATA PRE-PROCESSING

- *Initially the data is loaded and checked the data types using "info()".*
- *Then the data is checked for missing values, obvious errors, or typos.*
- *The column "MouseID" is split into "MouseNumber" and "MouseVersion" and dropped. As it has a delimiter "_" making if untidy.*
- *Then I checked for obvious or typo errors using "unique()"and changed the values of the "Behavior" column into respective names of tidy format.*
- *The "Genotype" has a mouse type "Ts65Dn" which is "Trisomy" I don't want to convert this because that is the name of the type.*
- *These missing values are then replaced by the mean values grouped by "class".*
- *This approach is better than removing the entire mice or observation or by replacing the null with the mean of all classes of mice.*
- *Then I checked for the white spaces on string variables.*
- *I converted the columns "Treatment" and "Behavior" into upper case.*
- *The data looks good and sent for further exploration.*

DATA EXPLORATION

- *Descriptive statistics is calculated on the data. The data is now checked for normalization as the classifier or the model performs more accurate on standard normalized data. The results of this visualization suggested that the data is to be normalized and transformation should be applied by scaling the data. This is done in the data preprocessing part of data modelling.*
- *The plausible hypothesis is stated based on the assumptions or findings from the Descriptive statistics and visualization.*
- *Investigation is done to prove these assumptions or facts or plausibility by selecting 10 random pairs based on the relationship between the pairs.*

DATA MODELLING

- ### MODEL BACKGROUND
  - *Two Supervised Machine learning algorithms, KNN and Decision tree are selected and build for modeling to predict the classification of mice classes or simple classes itself.*
  - *Also, to predict the important or crucial proteins of each class.*
  - *Based on input variables the model should predict the values of target variable.*
  - **KNN ALGORITHM –**
    - *The target variable is to be selected. Here the data is retrieved and Scikit-learn is imported.*
    - *Reshaping the prepared data into a matrix. We here use "pl.matshow()" to convert the data into 2-dimentional array. As KNN requires a list of values "reshape()" function is used.*
    - *Now the data is finally converted into 1 – dimensional array.*
    - *Splitting the data into test and train sets.*
    - *Selecting the KNN (K – nearest neighbour) classifier.*
    - *Data is fitted and the unseen data is predicted.*
    - *Performance report along with classification score and error rate is generated.*
    - *A confusion matrix is created.*
  - **DECISION TREE ALGORITHM –**
    - *Problem solving in the form of tree representation of decision series.*
    - *The Dataset (Root or Parent node) is split into subsets (Decision nodes) based on the value of an attribute.*
    - *If there is only one set or decision i.e., if root node is the leaf of the tree same classification or same decision is taken on the entire data. i.e., similar classification is made for the entire data. Or - else multiple classifications or decisions are made based on the factor values of that variable or attribute.*
    - *Each attribute is observed at each level and moved to next level based on the result of the observation and continues until a leaf node is met.*
- ### PRE - PROCESSING :
  - *The data is initially separated into class and levels by removing the unnecessary columns.*
  - *The labeled or categorial data is taken as "v_class" and their labels as "class_labels". All the numerical protein values are taken into "v_levels".*
  - *"MouseID" which is converted as "MouseNumber" and "MouseVersion" is dropped as we have nothing to do with that variable.*

- *The data is now normalized as there is high variance in measurements. i.e., the values of some proteins are explored or observed as "0 to 1" whereas some other have "0 to 8".*
- *If not normalized the proteins with higher values will influence the result of the classification or the classifier making if to give false or misleading reports.*
- *After handling null and obvious errors, the protein values are standardized with zero means and unit variance.*
- ***STANDARDIZATION :***
  - *It is the basic requirement in scikit-learn for implementing machine learning estimators. These estimators fail if the features do not be completely or tend to be standard normally distributed data or tend to be normal (Gaussian with zero mean and unit variance).*
  - *Which in practice transformation is applied to decrease the skewness of the distribution by removing the feature mean value and then scaled by dividing features by standard deviation*
  - *The estimator works assuming the values of the features are centered around 0 with a small same ordered variance (Normalized). If there is more variance the estimator fails to learn from the other features. So, the features are to be scaled to make the estimator learn quickly and efficiently.*
- *For this I imported preprocessing from sklearn and used "StandardScaler()". This scaler now transforms the protein values or "v_levels" data into standard values by each column.*
- *These standard values are then normalized using "Normalizer()". Here each sample is rescaled by each row.*
- *The negative values obtained by standardizing the data are now converted to positive or relative protein values with unit variance for better learning by the estimator using "MinMaxScaler()".*

- ***IMPLEMENTATION :***
  - *Now the data is prepared model should be built and implemented.*
  - *Initially, the models are created using sklearn.*
  - *Instead of using the pre - defined models or functions I created my own user – defined function "run_classifier()".*
  - *Classifier along with a single class_label is sent to the function.*
  - *Train and Test data are now sampled using "make_random_indices()".*
  - *This function randomly sets the indices with ¾ for train and ¼ for test data.*
  - *The training data is now fitted with the classifier model.*
  - *Then the model is run on the test data to predict the model.*
  - *The "performance report" is given by the calculating the "confusing matrix" and the "classification error rate" by calling a user defined function "performance_report()".*
  - *The metrics and methods used in this function helps in deciding which classifier is effective on this data.*
    - ***CONFUSION MATRIX***
    - ***CLASSIFICATION ERROR RATE***
    - ***CLASSIFICATION SCORE***
    - ***PRECISION***
    - ***RECALL***
    - ***F1-SCORE***

RESULTS AND REPORTS

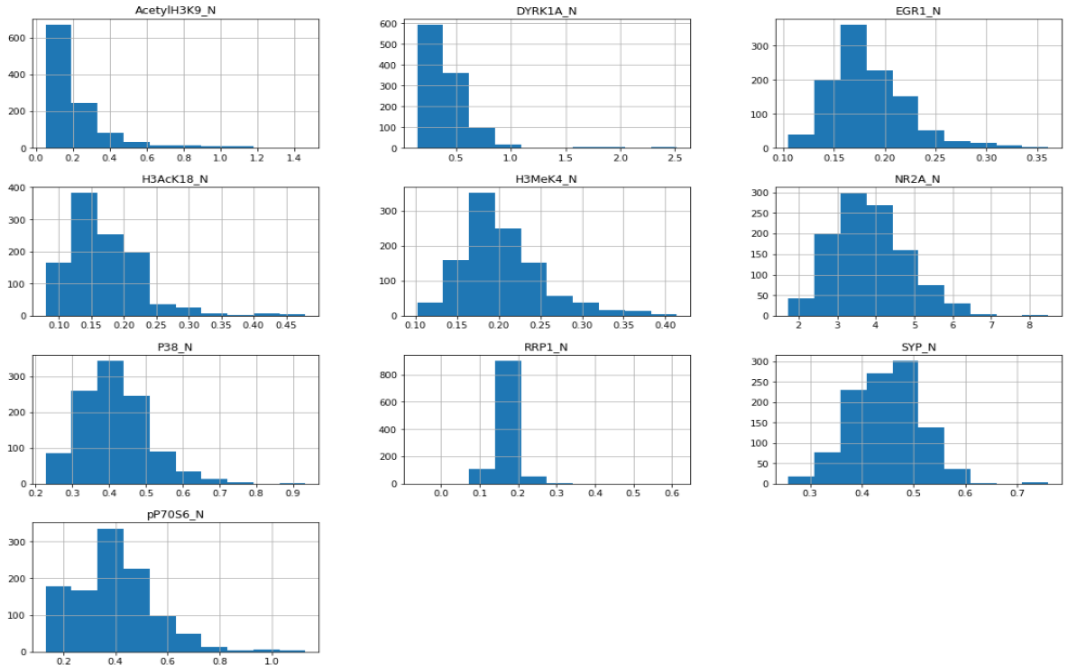
DATA EXPLORATION :

- *The descriptive statistics of all proteins,*

	DYRK1A_N	ITSN1_N	BDNF_N	NR1_N	NR2A_N	pAKT_N	pBRAF_N	pCAMKII_N	pCREB_N	pELK_N	...	SHH_N	10
count	1080.000000	1080.000000	1080.000000	1080.000000	1080.000000	1080.000000	1080.000000	1080.000000	1080.000000	1080.000000	...	1080.000000	
mean	0.425565	0.616913	0.319106	2.297134	3.843159	0.233206	0.181856	3.538885	0.212614	1.428060	...	0.226676	
std	0.249058	0.251316	0.049316	0.346819	0.931918	0.041583	0.027005	1.293806	0.032551	0.466403	...	0.028989	
min	0.145327	0.245359	0.115181	1.330831	1.737540	0.063236	0.064043	1.343998	0.112812	0.429032	...	0.155869	
25%	0.288163	0.473669	0.287650	2.059152	3.160287	0.205821	0.164619	2.479861	0.190828	1.204546	...	0.206395	
50%	0.366125	0.565494	0.316703	2.295648	3.738908	0.231246	0.182472	3.329624	0.210681	1.355423	...	0.224000	
75%	0.487574	0.697500	0.348039	2.528035	4.425107	0.257225	0.197226	4.480652	0.234558	1.560931	...	0.241655	
max	2.516367	2.602662	0.497160	3.757641	8.482553	0.539050	0.317066	7.464070	0.306247	6.113347	...	0.358289	

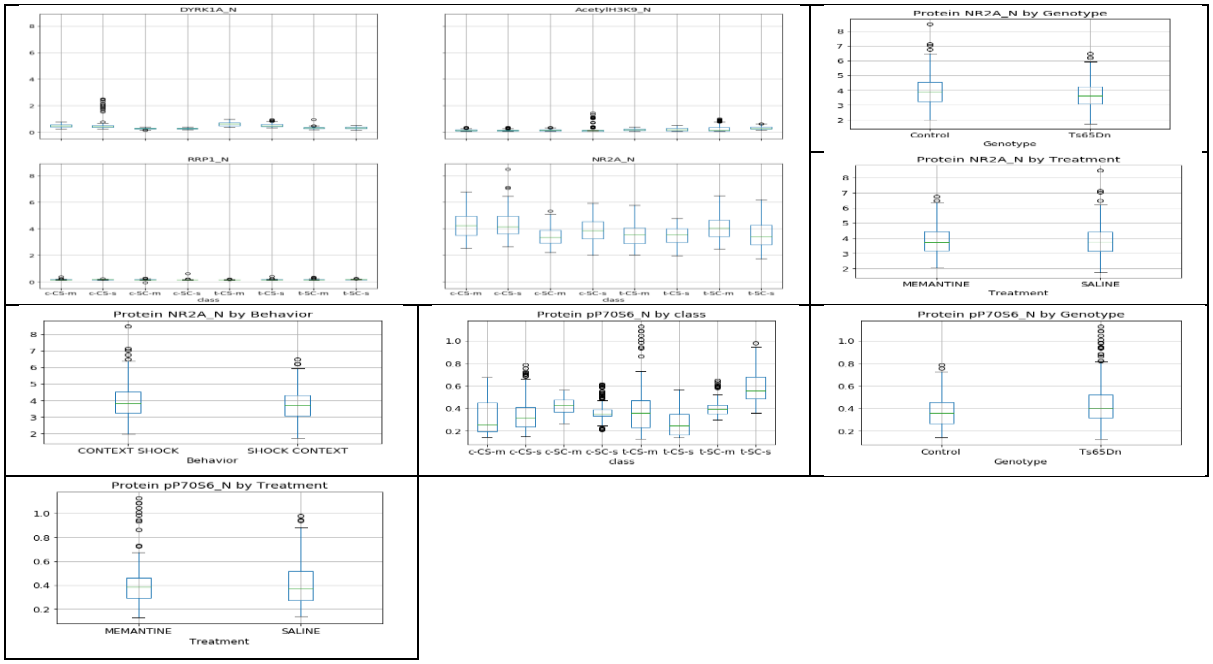
8 rows × 77 columns

- *The visualization is made on 10 different proteins as shown below,*



- *The visualizations suggest that the data has more skewness and are mostly "Right - Skewed".*
- *As we came with some facts on 10 different proteins. Now consider those variables with some plausibilities or facts.*
- ***PLAUSIBILITIES:***
  - ***THE DATA IS NOT NORMALLY DISTRIBUTED.***
  - ***THE DATA HAS LARGE VARIANCE.***
  - ***THE VARIANCE RANGE OF ONE PROTEIN ON ALL BINARY VARIABLES WILL BE APPROXIAMATLY SAME.***
  - ***THE DATA IS NOT ON A STANDARD SCALE.***
  - ***THE VALUES OF SOME PROTEINS ARE OBSERVED FROM "0 TO 1" WHEREAS SOME OTHER HAVE "0 TO 8".***
  - ***THE DATA HAS A WIDE RANGE OF OUTLIERS.***
- *These plausibilities are now proved by taking 10 pairs of variables – 1 protein and 1 binary predictor.*
- ***PAIR 1 – "DYRK1A_N" & "CLASS"***
- ***PAIR 2 – "ACETYLH3K9_N" & "CLASS"***
- ***PAIR 3 – "RRP1_N" & "CLASS"***
- ***PAIR 4 – "NR2A_N" & "CLASS"***
- ***PAIR 5 – "NR2A_N" & "GENOTYPE"***
- ***PAIR 6 – "NR2A_N" & "TREATMENT"***

- ***PAIR 7 – "NR2A_N" & "BEHAVIOR"***
- ***PAIR 8 – "PP70S6_N" & "CLASS"***
- ***PAIR 9 – "PP70S6_N" & "GENOTYPE"***
- ***PAIR 10 – "PP70S6_N" & "TREATMENT"***



PLAUSIBILITY 1 *– Boxplot is not symmetrical as the mean and median are not at the centre.*

PLAUSIBILITY 2 *– 1st 4 pairs clearly proves this.*

PLAUSIBILITY 3 *– Pairs 4, 5, 6, 7 clearly proves this.*

PLAUSIBILITY 4 *– All pairs suggest this due to normality issue.*

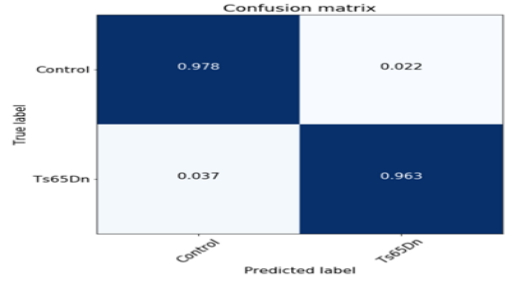
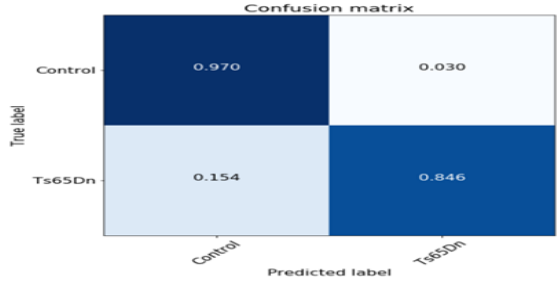
PLAUSIBILITY 5 *– Some have 0 to 1 and some plots have 0 to 8.*

PLAUSIBILITY 6 *– All plots have outliers.*

DATA MODELLING :

- *In this assignment 2 models are built,*
  - *KNN model with all factor or binary variables and reported individually.*
  - *Decision tree model same process as KNN. Also, collective feature importance and ranking for all the 4 binary predictor variables is given to find the crucial protein.*

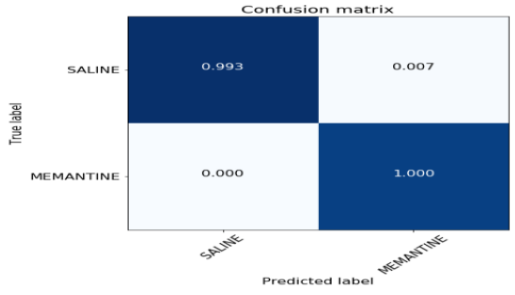
- ***GENOTYPE***

KNN	***DECISION TREE***
Genotype Classification Error Rate 0.02962962962962963 Classification Report precision recall f1-score support Control 0.96 0.98 0.97 134 Ts65Dn 0.98 0.96 0.97 136 accuracy 0.97 270 macro avg 0.97 0.97 0.97 270 weighted avg 0.97 0.97 0.97 270 Classification Score 0.9703703703703703 <Figure size 432x288 with 0 Axes>  	Genotype Classification Error Rate 0.4925925925925926 Classification Report precision recall f1-score support Control 0.86 0.97 0.91 134 Ts65Dn 0.97 0.85 0.90 136 accuracy 0.91 270 macro avg 0.91 0.91 0.91 270 weighted avg 0.91 0.91 0.91 270 Classification Score 0.9074074074074074 <Figure size 432x288 with 0 Axes> 

➢ TREATMENT

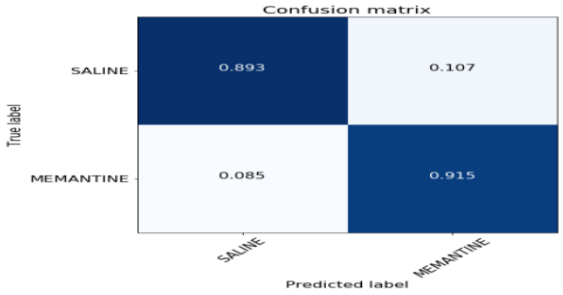
KNN

```
Treatment

Classification Error Rate
0.003703703703703704

Classification Report
              precision    recall  f1-score   support

    MEMANTINE       1.00      0.99      1.00       140
       SALINE       0.99      1.00      1.00       130

     accuracy                           1.00       270
    macro avg       1.00      1.00      1.00       270
 weighted avg       1.00      1.00      1.00       270

Classification Score
0.9962962962962963

<Figure size 432x288 with 0 Axes>
```



DECISION TREE

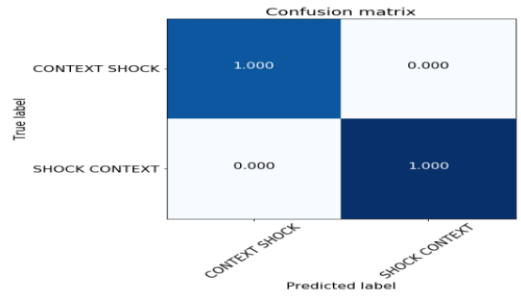
```
Treatment

Classification Error Rate
0.0962962962962963

Classification Report
              precision    recall  f1-score   support

    MEMANTINE       0.92      0.89      0.91       140
       SALINE       0.89      0.92      0.90       130

     accuracy                           0.90       270
    macro avg       0.90      0.90      0.90       270
 weighted avg       0.90      0.90      0.90       270

Classification Score
0.9037037037037037

<Figure size 432x288 with 0 Axes>
```



➢ BEHAVIOR

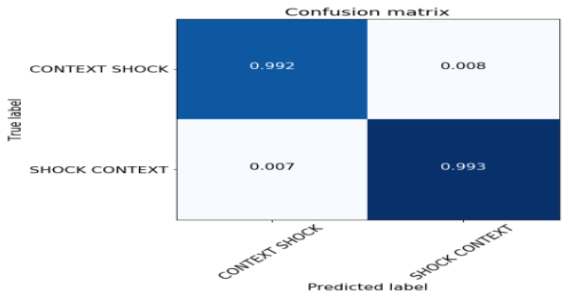
KNN

```
Behavior

Classification Error Rate
0.0

Classification Report
              precision    recall  f1-score   support

CONTEXT SHOCK       1.00      1.00      1.00       124
SHOCK CONTEXT       1.00      1.00      1.00       146

     accuracy                           1.00       270
    macro avg       1.00      1.00      1.00       270
 weighted avg       1.00      1.00      1.00       270

Classification Score
1.0

<Figure size 432x288 with 0 Axes>
```



DECISION TREE

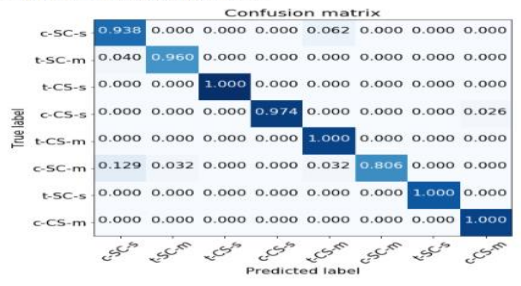
```
Behavior

Classification Error Rate
0.007407407407407408

Classification Report
              precision    recall  f1-score   support

CONTEXT SHOCK       0.99      0.99      0.99       124
SHOCK CONTEXT       0.99      0.99      0.99       146

     accuracy                           0.99       270
    macro avg       0.99      0.99      0.99       270
 weighted avg       0.99      0.99      0.99       270

Classification Score
0.9925925925925926

<Figure size 432x288 with 0 Axes>
```



➢ CLASS

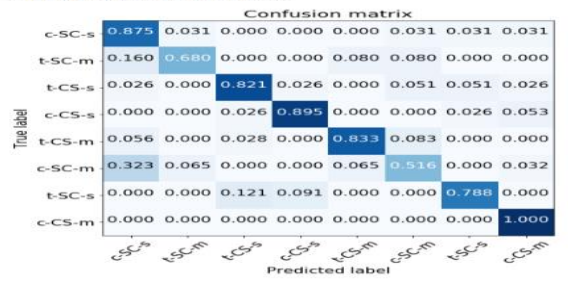
KNN

```
class

Classification Error Rate
0.037037037037037035

Classification Report
              precision    recall  f1-score   support

       c-CS-m       0.86      0.94      0.90        32
       c-CS-s       0.96      0.96      0.96        25
       c-SC-m       1.00      1.00      1.00        39
       c-SC-s       1.00      0.97      0.99        38
       t-CS-m       0.92      1.00      0.96        36
       t-CS-s       1.00      0.81      0.89        31
       t-SC-m       1.00      1.00      1.00        33
       t-SC-s       0.97      1.00      0.99        36

     accuracy                           0.96       270
    macro avg       0.96      0.96      0.96       270
 weighted avg       0.97      0.96      0.96       270

Classification Score
0.9629629629629629

<Figure size 432x288 with 0 Axes>
```



DECISION TREE

```
class

Classification Error Rate
0.18888888888888888

Classification Report
              precision    recall  f1-score   support

       c-CS-m       0.62      0.88      0.73        32
       c-CS-s       0.85      0.68      0.76        25
       c-SC-m       0.84      0.82      0.83        39
       c-SC-s       0.89      0.89      0.89        38
       t-CS-m       0.88      0.83      0.86        36
       t-CS-s       0.67      0.52      0.58        31
       t-SC-m       0.87      0.79      0.83        33
       t-SC-s       0.88      1.00      0.94        36

     accuracy                           0.81       270
    macro avg       0.81      0.80      0.80       270
 weighted avg       0.82      0.81      0.81       270

Classification Score
0.8111111111111111

<Figure size 432x288 with 0 Axes>
```



❖ *The above results suggest that KNN is better than Decision tree.*

- *The "Classification Error Rate" of all 4 class labels is more in Decision tree compared to KNN. Suggesting that KNN is more accurate.*
- *The "Classification score" is more in KNN than Decision tree leaving the high performance to fall under KNN.*
- *The "F1 – score" from the "Classification report" is more compared to Decision tree. In all cases it is 1 or nearly 1.*
- *The "Trace" of the confusion matrix of KNN is greater than that of decision tree.*
- *As we know that "if the trace of the matrix is very high compared to the sum of all matrix entries, then the predictions for majority of the parts are correct". Here this is seen in KNN than in decision tree.*
- *These all evidences prove that KNN is more accurate and predict correctly than decision tree.*
- *KNN predicted the classification of classes "t-CS-s", "t-CS-m", "t-SC-s" and "c-CS-m" accurately but struggled a little bit in the classification of classes "c-SC-m", "c-CS-s", "c-SC-s" and "t-SC-m".*
- *The behavorial classification is more accurate as the value is equal to 1. 1 indicates 100% classification and 0 implies 0% classification.*
- *Memantine has 100% classification but Saline has a slight lower one off 0.993 which implies 99.3% classification.*
- *Coming to Genotype "Control" got more classification result of nearly 0.978 accounting to 97.8% accuracy and "Ts65DN" -> 0.963 – 96.3%.*
- ***I got confusion matrix values in the range of 0 to 1 because I set the scale to 0 to 1 using StandardScaler – Normalizer – MinMaxScaler. For better understanding of data values.***

FEATURE IMPORTANCE

Decision Tree can extract the importance of each feature which helps to attain our goal by identifying crucial proteins that are critical in the learning of the metabolic pathways. Feature importance and its ranking for each binary predictor variable is shown in the pic below.

	Feature	class_Importance	class_rank
0	SOD1_N	0.145943	1.0
1	pPKCG_N	0.121262	2.0
2	APP_N	0.103374	3.0
3	pCAMKII_N	0.073070	4.0
4	IL1B_N	0.052426	5.0
...	...	...	...
72	SHH_N	0.000000	60.0
73	BAD_N	0.000000	60.0
74	BCL2_N	0.000000	60.0
75	pCFOS_N	0.000000	60.0
76	H3MeK4_N	0.000000	60.0

77 rows × 3 columns

	Feature	Behavior_Importance	Behavior_rank
0	SOD1_N	0.878369	1.0
1	pERK_N	0.088939	2.0
2	CaNA_N	0.013038	3.0
3	ITSN1_N	0.009828	4.0
4	BRAF_N	0.009826	5.0
...	...	...	...
72	pCFOS_N	0.000000	41.5
73	SYP_N	0.000000	41.5
74	H3AcK18_N	0.000000	41.5
75	EGR1_N	0.000000	41.5
76	H3MeK4_N	0.000000	41.5

77 rows × 3 columns

	Feature	Treatment_Importance	Treatment_rank
0	pPKCG_N	0.115171	1.0
1	BRAF_N	0.095951	2.0
2	Ubiquitin_N	0.094347	3.0
3	pNR2A_N	0.066961	4.0
4	pJNK_N	0.048286	5.0
...	...	...	...
72	PSD95_N	0.000000	55.5
73	SHH_N	0.000000	55.5
74	BAD_N	0.000000	55.5
75	pCFOS_N	0.000000	55.5
76	EGR1_N	0.000000	55.5

77 rows × 3 columns

	Feature	Genotype_Importance	Genotype_rank
0	APP_N	0.307995	1.0
1	pMTOR_N	0.128092	2.0
2	H3MeK4_N	0.089694	3.0
3	Tau_N	0.076104	4.0
4	GluR3_N	0.054880	5.0
...	...	...	...
72	BAD_N	0.000000	54.0
73	BCL2_N	0.000000	54.0
74	pS6_N	0.000000	54.0
75	pCFOS_N	0.000000	54.0
76	EGR1_N	0.000000	54.0

77 rows × 3 columns

DISCUSSION

From the above results and reports we can find that knn is more accurate than decision tree in modelling our data. Coming back to KNN, it has PRECISION, RECALL and F1-SCORE values almost equal to 1 and in most of the cases it is 1 itself which implies its perfectness. Whereas coming to decision tree these values fall below the values of KNN. They recorded low values, 0.52 for RECALL and 0.62 for precision which shows the models weak performance. The classification score, error rate and including the trace of the matrix suggest that decision tree performed weak in obtaining results. The same data is given for the 2 models there is a large variation in their predictions. 3/4th of data is selected randomly for training and 1/4th of the data is selected for testing on a random basis. Despite its low accuracy decision tree helped us in identifying the crucial proteins which are critical in learning of mice. Each feature importance lies in the range 0 to 1. 1 implies the feature is used by the model completely, 0 implies feature is not involved in modelling. This affects the performance of the model. More the number of features having 0 or low importance, lesser the performance. This might be due to the overfitting of the data with the decision tree. Only the top 7 features contribute 50% of the model decisions. Similarly, 90% decisions are made by top 27 features. The rankings of binary variables such as "Behavior, Treatment and class" are same with different importance values. Leaving genotype different from these. Identifying the crucial proteins strengthened the facts of model weakness.

The top 5 proteins for each classification are as follows,

Genotype	Treatment	Behavior	Class
APP_N	pPKCG_N	SOD1_N	SOD1_N
pMTOR_N	BRAF_N	pERK_N	pPKCG_N
H3MeK4_N	Ubiquitin_N	CaNA_N	APP_N
Tau_N	pNR2A_N	ITSN1_N	pCAMKII_N
GluR3_N	pJNK_N	BRAF_N	IL1B_N

CONCLUSION

From the observations we can conclude that KNN classifier is more accurate than decision tree in predicting the crucial proteins of mice. Finally, the feature importance table from the decision tree classifier gives the rank of each protein among based on each group ['Genotype', 'Treatment', 'Behavior', 'class'].

FUTURE ENHANCEMENT

- ✓ *Random forest will give better results in this case as they are good at over fitting. This model gives the importance and predicts the classification more accurately.*
- ✓ *The prediction of crucial proteins helps in the development or identification of mice learning ability drugs. So that the deficiencies related to learning and memory can be overcome.*

REFERENCES

- ✓ https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression
- ✓ https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2657943/
- ✓ KNN - ASSIGNMENT 1 PART 2
- ✓ DECISION TREES
  https://www.r-bloggers.com/why-do-decision-trees-work/
- ✓ PERFORMANCE METRICS OR METHODS
  https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b#:~:text=We%20can%20use%20classification%20performance,primarily%20used%20by%20search%20engines.