PRACTICAL DATA SCIENCE WITH PYTHON ASSIGNMENT 2

STUDENT NAME : APARUPA MITRA

MASTERS OF DATA SCIENCE , RMIT UNIVERSITY

STUDENT ID : S3831724

EMAIL ID : s3831724@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 honour code by typing "Yes": *Yes*.

TABLE OF CONTENT

ABSTRACT SUMMARY

With advance research on Down Syndrome, a genetic disorder in human beings , different experiments have been carried out in mice because it was found that mice have some similar sets of chromosomes with few similar features which human being have. Thus it displays many relevant features which have been identified in down syndrome patient. This is known to be mouse model of Down Syndrome which helps to study more genetic features among mice and human being.

Many data science and machine learning methods such as supervised and unsupervised learning have explored answers to many complex question in this fields of genetics and biology. Through machine learning many models has been generated to classify new unlabelled instances of data.

INTRODUCTION

Mice model data set gave us an opportunity to explore various features that was a part of mouse model DS data. The Mice Protein Expression dataset consist of 77 protein expression levels of all mice which was experimented along with their Genotype, Treatment type , Behaviour and Classes of mice . We explored these different protein levels through Data Exploration techniques of Data Science. Also we generated model which will help to identify the subsets of protein which provides a distinguishing feature between class labels. Model was built through Data Modelling process in Data Science .

Thus this assignment provides an ample opportunity to learn and apply different Data Science techniques in various fields.

METHODOLOGY

Whole process involves below three steps which has been elaborated in different sections:

- Data Preparation
- Data Exploration
- Data Modelling

Data Preparation

Data Preparation process involves two steps :

Data Retrieval -: In this step we have extracted dataset **'Data_Cortex_Nuclear.csv'** in Jupyter Notebook python from website: https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression. After file retrieval ,we have check data types for all columns.

Data Cleaning-: There are some missing values in some columns .Assuming that NAN values on some cells have been missing because no inputs has been provided at the time of mouse experiment ,so we will fill those missing cells with zero .Thus it will preserve the values of original dataset that we will be working without adding any new values in it which might alter dataset and can impact further analysis.

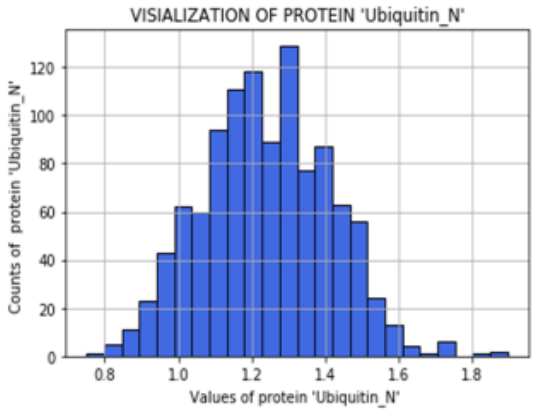
Data Exploration

PART I : Explore each column (or at least 10 columns if there are more than 10 columns), using appropriate descriptive statistics and graphs.

1. Column Exploration 1

We explored the distribution of protein type 'Ubiquitine_N' in Mice protein dataset.

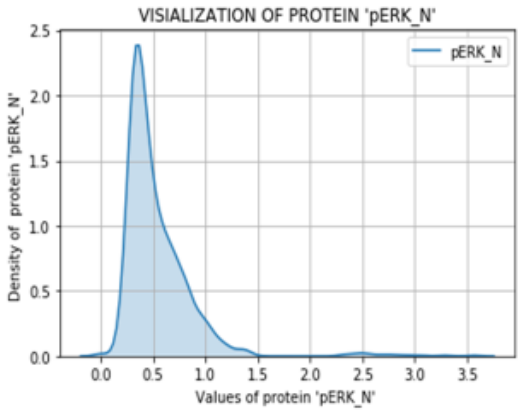
Analysis: From the histogram plotted, we can conclude that the data is almost Normally distributed with high concentration at the centre ranging between 1.0 – 1.4.



2. Column Exploration 2

We have explored the distribution protein type 'perk_N' in Mice protein dataset.

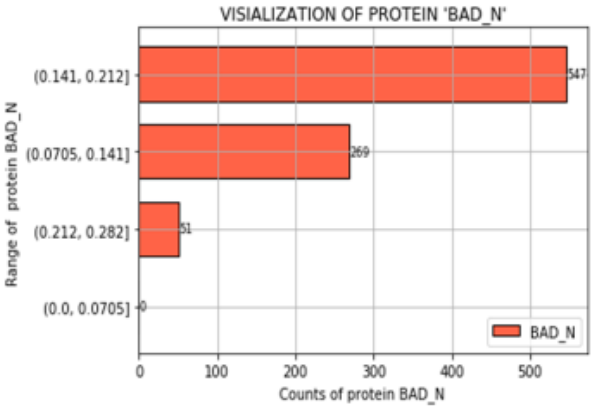
Analysis :From the density plot plotted, we can conclude that the data is almost distributed with high concentration at the left end around value 0.05.



3. Column Exploration 3

We explored the count of protein type 'BAD_N' in Mice protein dataset.

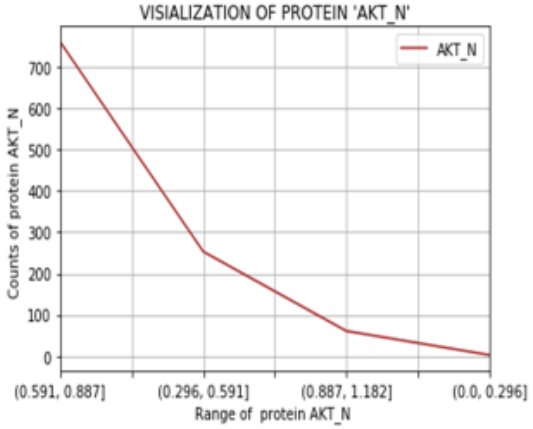
Analysis: From the annotated bar plot so plotted, we can conclude that protein level in the range between [0.141-0.212] has highest count of 547 where as lowest count of 51 is found for protein level [0.0 – 0.0705].



4. Column Exploration 4

We have explored the count of range of protein type 'AKT_N' in Mice protein dataset.

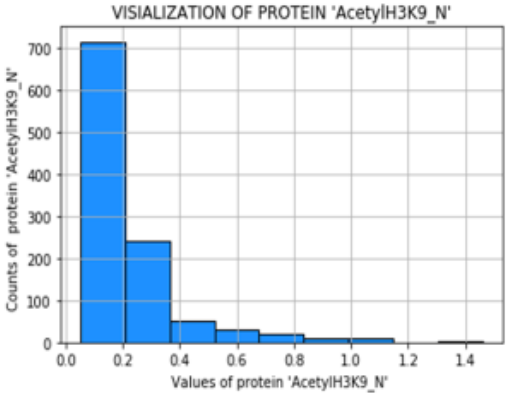
Analysis: From the line plot, we can conclude that the protein level is highest in range [0.591 -0.887]. Other ranges of protein level shows lower in counts.



5. Column Exploration 5

We explored the distribution of protein type 'AcetylH3K9_N' in Mice protein dataset.

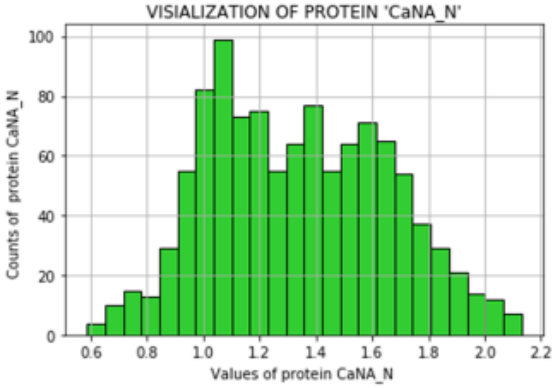
Analysis: From the histogram plotted, we can conclude that the graph is left skewed distributed and protein level is mostly concentrated at left end ranging between 0.0-0.2



6. Column Exploration 6

We have explored the count of range of protein type 'CaNA_N' in Mice protein dataset.

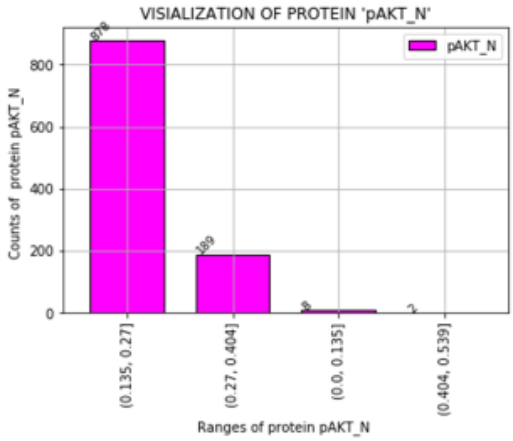
Analysis: From the histogram, we can conclude that data distribution is approximately Normal distribution. Peak concentration of protein lies within 1.0 -1.6.



7. Column Exploration 7

We explored the counts of range of protein type 'pAKT_N' in Mice protein dataset.

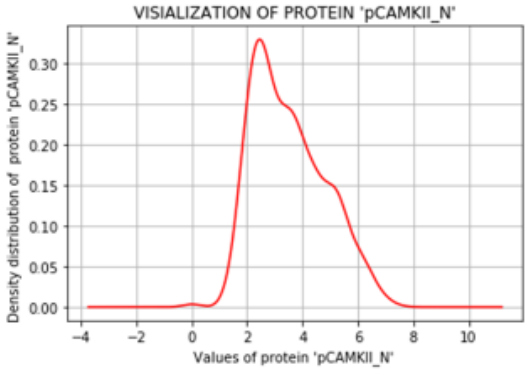
Analysis: From the annotated bar chart we conclude that the ranges of protein level between[0.135-0.271] has the highest count for the dataset.



8. Column Exploration 8

We have explored the distribution of protein type 'pCAMKII_N' in Mice protein dataset.

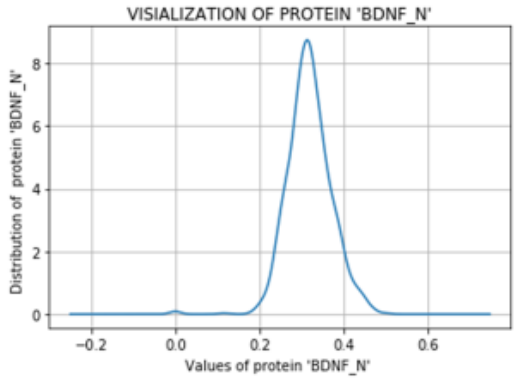
Analysis: From the density plot we can see that the density of the protein level is more concentrated at middle of the plot values between 2 to 4.The graph shows approximately Normal distribution.



9. Column Exploration 9

We explored the counts of range of protein type 'BDNF_N' in Mice protein dataset.

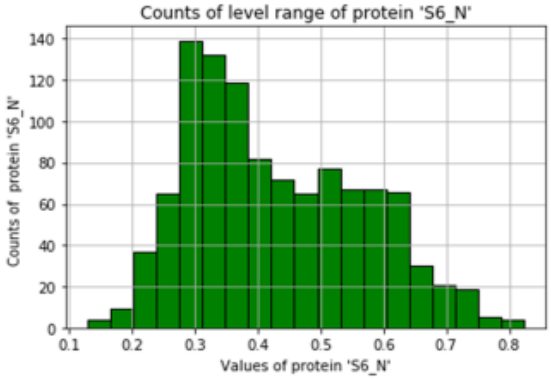
Analysis: From the density plot we conclude that the protein level has peak values for protein level between 0.2-0.4.



10. Column Exploration 10

We have explored the distribution of protein type 'S6_N' in Mice protein dataset.

Analysis: Histogram shows the graph is partly left skewed as more concentration(peak counts of protein) of protein have level between 0.2-0.4 .

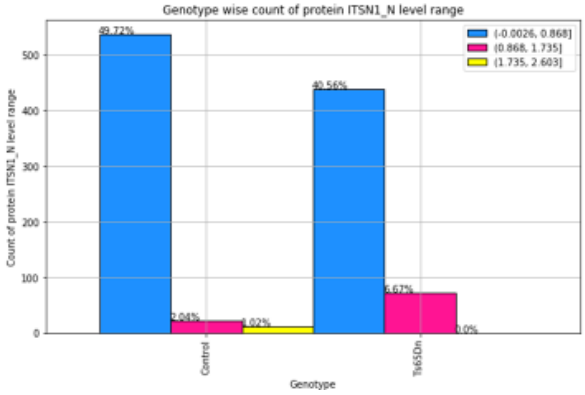


PART II - Explore the relationship between all pairs of attributes (or at least 10 pairs of attributes, if there are more in the data), and show the relationship in an appropriate graphs.

1. Relationship Exploration 1

Genotype vs count of protein ITSN1_N We explored based on types of genome what is the count of protein ITSN1 for each geno type.

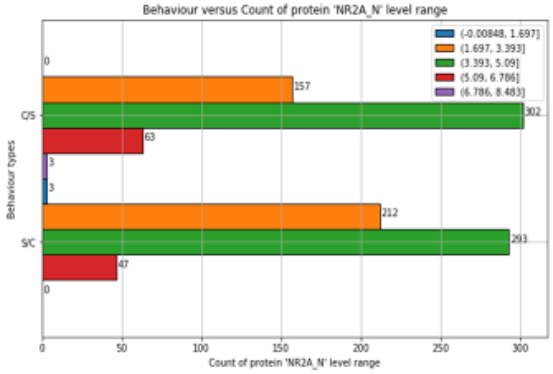
Analysis: From the annotated barplot we can see that the percentage of protein level ranging between -0.0026 to 0.868 is higher in both Control and Ts65Dn genome but slightly more in Control type. Highest protein level range is missing in Ts65Dn type.



2. Relationship Exploration 2

Behaviour type vs Count of protein 'NR2A_N' level as we want to explore counts of this protein based on behaviour type.

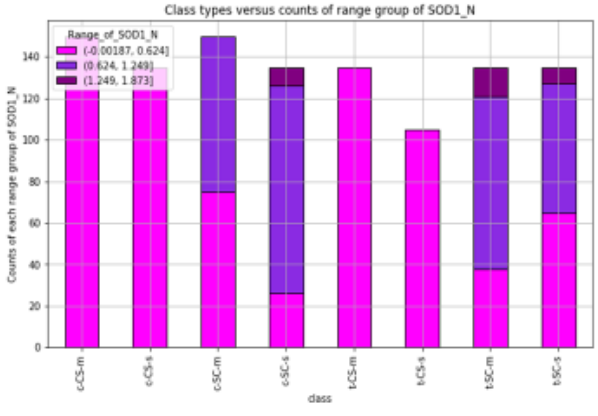
Analysis: From the annotated horizontal barplot, we can conclude that count of protein level with range 3.393-5.09 is highest for both behaviour type. Some of the range of protein level is missing in S/C behaviour type.



3. Relationship Exploration 3

Class types vs counts of range group of SOD1_N as we want to explore count of this protein based in different class type.

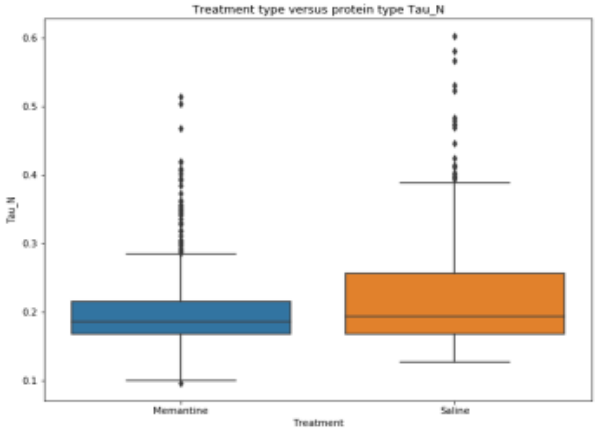
Analysis: From the stacked barplot we can see that protein level range between -0.00187-0.624 is highest among all class type. Also observed that protein level with highest range[1.294-1.873] is missing from some of class types.



4. Relationship Exploration 4

Treatment type versus protein type Tau_N as we want to explore concentration of protein level based on treatment type.

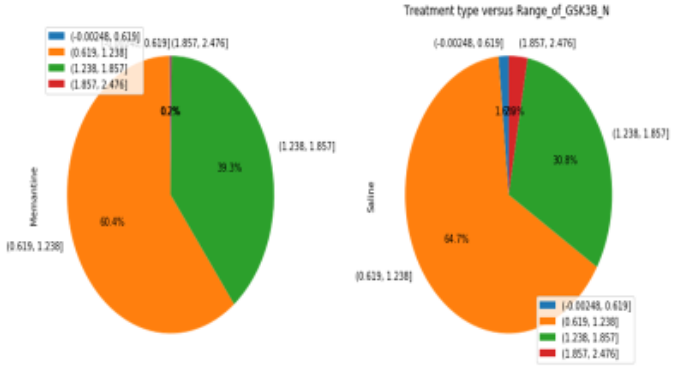
Analysis: From the box plot we can conclude that this protein level is more in Saline type of treatment as its range ,Interquartile range and mean is more than Memantine type.



5. Relationship Exploration 5

Treatment type versus Range_of_GSK3B_N as we want to explore percentage of this protein based in different treatment type.

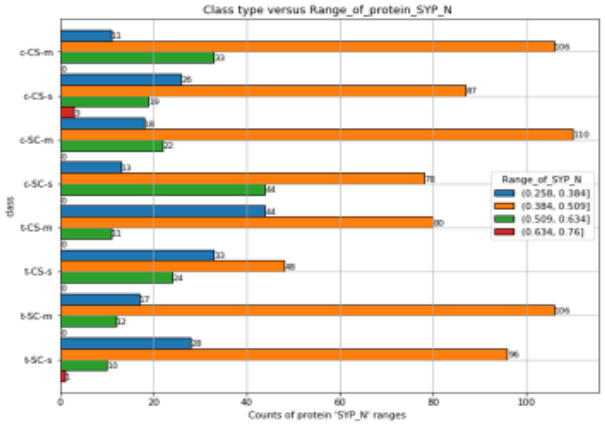
Analysis: From the piechart we can see that protein level range between [0.619-1.238]have highest percentage in both treatment type. Two range of protein level is missing in Memantine treatment.



6. Relationship Exploration 6

Class type versus Range_of_protein_SYP_N as we want to explore count of protein level based on class type.

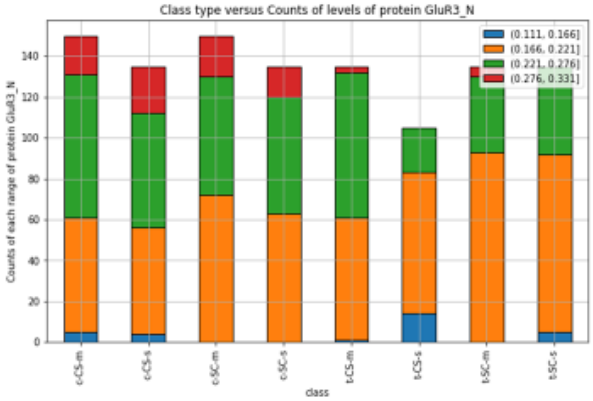
Analysis: From the annotated bar plot, we can conclude that protein level [0.384-0.509] is found to be highest among all class types .



7. Relationship Exploration 7

Class type versus Range_of_GluR3_N as we want to explore count of this protein for different class types

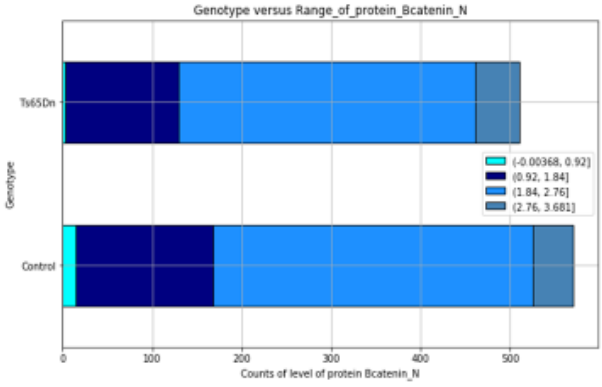
Analysis: From the stacked bar chart, we conclude that two protein levels between (0.166-0.21 and 0.221-0.276) are almost equally present in all class types .The other two ranges of protein levels have a little trace in few class types.



8. Relationship Exploration 8

Genotype type versus count_of_protein_Bcatenin_N as we want to explore count of protein level on genotype basis.

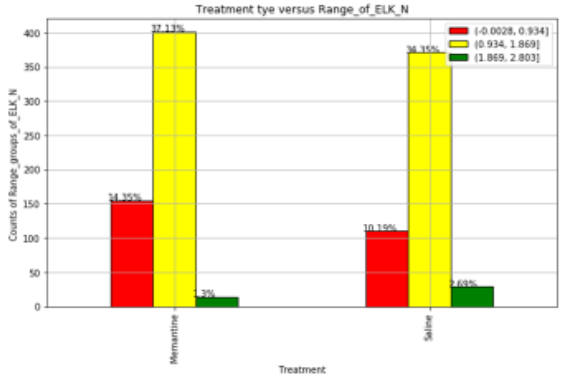
Analysis: From horizontal stacked bar plot, we can conclude for both genotype this protein holds approximately equal counts. Thus there is a ***lack of relationship*** between different genotype when compared with counts of protein type 'Bcatenin_N'.



9. Relationship Exploration 9

Treatment type versus Range_of_ELK_N as we want to explore count of this protein for different treatment types.

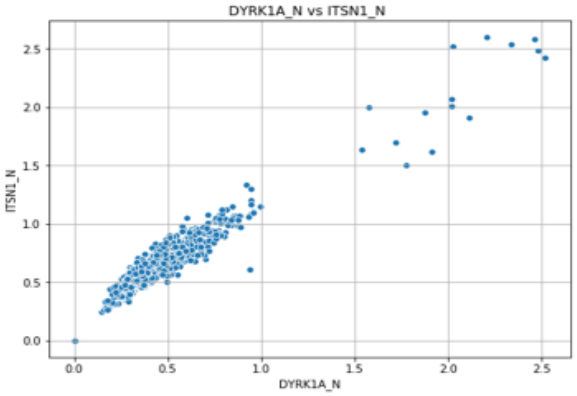
Analysis: From annotated bar plot, we can conclude for both treatment type this protein holds approximately equal percentage. Thus there is a ***lack of relationship*** between different genotype when compared with counts of protein type 'ELK_N'.



10. Relationship Exploration 10

Relationship between protein DYRK1A_N vs protein ITSN1_N

Analysis: From scatterplot , we can infer that a strong positive correlation exists between these two protein level. With the increase of protein DYRK1A_N , protein ITSN1_N also increases.



Data Modelling

Data modelling involves building model to predict class label. It mainly carried into 4 steps :**a)** Feature engineering and model selection **b)** Training the model **c)** Model validation and selection **d)** Applying the trained model to unseen data.

Our aim is to identify subsets of proteins levels provided in Mice Protein Expression dataset that are discriminant between the classes .In other words which protein levels provides a distinguishing feature that determines class of mice.

We will proceed with Classification technique .As we already have class label for different types of mice ,through classification, we can check which subsets of protein determines class types. In this process two algorithms has been applied to build model. First we tested with **KNN classifier** and then with **Decision Tree classifier**. Based on result of both, we compared the two model to decide with which model we shall proceed.

We will perform both algorithm in model with each algorithm twice on our dataset with below steps repeated for both algorithms

- Apply Classifier algorithm on full dataset taking all features ,**without** selecting specific /best feature and check its scores of model.
- Perform **Hill Climbing** search technique to **get the best features**.
- Again apply Classifier algorithm **with selected best features** and then again check scores of model.

KNN ALGORITHM

This Algorithm calculates the distance between and query instances and sample to find the Kth minimum distance. After that it gathers the category of its nearest neighbour and votes the most frequent classification. We will perform this algorithm twice.

WITHOUT SELECTION OF BEST FEATURES APPLY KNN ALGORITHM ON DATASET WITH ALL FEATURES AND CHECK THE SCORE

Feature selection : Initially we first we have selected all 77 protein level as feature and 'class' label as target

knn_feature = mice_dataset.iloc[:,1:78]

knn_target = mice_dataset.iloc[:,-1]

Then we applied KNN classifier with initial default value with n_neighbors =5 ,p=2.

In later steps, we have carried out parameter tuning .

```
KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=5,
p=2, weights='uniform')
```

Training and fitting model

Once features have been selected ,we have trained the model by taking 75% data as training and 25% data as test data.

Validation of model : Using K-Fold cross validation method ,we validated our model. The best accuracy score obtained is 0.96 approx. with all features.

Evaluation model and applying on unseen data

Finally we fit and evaluated out model. It gave below score with accuracy score of .92 or 92% which was quite good .

	precision	recall	f1-score	support
c-CS-m	0.74	1.00	0.85	29
c-CS-s	0.90	0.88	0.89	32
c-SC-m	0.89	0.94	0.91	33
c-SC-s	0.98	1.00	0.99	40
t-CS-m	1.00	0.80	0.89	41
t-CS-s	0.93	0.90	0.92	31
t-SC-m	0.96	0.90	0.93	30
t-SC-s	0.97	0.94	0.96	34
accuracy			0.92	270
macro avg	0.92	0.92	0.92	270
weighted avg	0.93	0.92	0.92	270

However instead of taking all features,now we continue to search for best features in next step.

NOW APPLY HILL CLIMBING SEARCH TECHNIQUE TO FIND BEST FEATURES

In this step with the help of **Hill climbing search** , we will select the best feature that will give us the best score of model.

We found total 48 feature out of 77 protein level as best feature given by search program.

```
There are 48 features selected: [0, 2, 5, 6, 10, 11, 12, 13, 15, 18,
19, 22, 26, 29, 30, 32, 33, 34, 36, 37, 39, 40, 42, 43, 45, 46, 49,
50, 51, 52, 53, 54, 55, 56, 58, 59, 61, 63, 64, 66, 67, 68, 70, 71,
72, 73, 74, 75]
```

AGAIN APPLY KNN ALGORITHM WITH BEST FEATURES ONLY AND AGAIN CHECK SCORE

Again we will perform the KNN algorithm on model **with these best 48 feature** that we got from above search technique to get model score and finally will compare with previous KNN model(with all features together). Here we will also perform **parameter tuning** such changing p values or k values.

Feature Selection: With this **48 features**, we again applied KNN algorithm to our model.

Here before fitting and training the model ,we did **parameter tuning** .Here we randomly selected the value of n_neighbour = 3 ,weights='distance' and value of p=1 . All three values was selected randomly to experiment the performance of model.

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=1,
                     weights='distance')
```

Training and fitting model : We have divided the set by taking 75% data as training and 25% data as test data. Finally we fitted and trained our model with KNN classifier with modified parameters

Validation of model : Using K-Fold cross validation method ,we validated our model. The best accuracy score obtained is 0.99 approx. with best features.

Evaluating model and testing on unseen data : On evaluating model ,it gave a very high score of 0.959.

```
              precision    recall  f1-score   support

      c-CS-m       0.94      1.00      0.97        29
      c-CS-s       0.97      0.94      0.95        32
      c-SC-m       0.91      0.94      0.93        33
      c-SC-s       0.95      1.00      0.98        40
      t-CS-m       1.00      0.95      0.97        41
      t-CS-s       0.97      0.97      0.97        31
      t-SC-m       0.96      0.90      0.93        30
      t-SC-s       0.97      0.97      0.97        34

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

0.9592592592592593
```

DECISION TREE ALGORITHM

This algorithm selects the best feature among all given features by segregating based on target to give best homogeneous group that can easily predict the label.

WITHOUT SELECTION OF BEST FEATURES APPLY DECISION TREE ALGORITHM ON DATASET WITH ALL FEATURES AND CHECK THE SCORE

-First we will select all 77 features and apply Decision Tree classifier will check the scores of model

Feature Selection: Here we have selected all 77 protein level as features and 'class' as target label. We have selected Decision tree algorithm with default parameters.

decision_feature = mice_dataset.iloc[:,1:78]
decision_target = mice_dataset.iloc[:,-1]

```
DecisionTreeClassifier(class_weight=None, criterion='gini',
max_depth=None, max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort=False,
random_state=None, splitter='best')
```

Training and fitting model: We have divided train- test set into 80% training and 20% test set .Then we fitted and trained our model with Decision tree classifier.

Validation of model : Our Model was validated using K-Fold cross validation method. However the best accuracy score obtained is 0.89 in one of the fold.

Evaluating model and testing on unseen data : Overall model have a score of 0.85 or 85% accuracy which is moderate.

	precision	recall	f1-score	support
c-CS-m	0.67	0.67	0.67	24
c-CS-s	0.65	0.81	0.72	27
c-SC-m	1.00	0.84	0.91	25
c-SC-s	1.00	0.97	0.98	31
t-CS-m	0.74	0.78	0.76	32
t-CS-s	0.95	0.78	0.86	27
t-SC-m	0.89	1.00	0.94	24
t-SC-s	1.00	0.92	0.96	26
accuracy			0.85	216
macro avg	0.86	0.85	0.85	216
weighted avg	0.86	0.85	0.85	216

Now we applied Hill Climbing search technique to find the best feature and apply Decision tree algorithm again on those feature.

APPLY HILL CLIMBING SEARCH TECHNIQUE TO FIND BEST FEATURES

-.Then with the help of Hill climbing search , we will select randomly generated any set of best features that will give us the best score of model.

```
There are 23 features selected: [10, 11, 16, 19, 20, 21, 30,
31, 32, 38, 40, 42, 45, 46, 50, 53, 54, 55, 56, 57, 58, 64,
71]
```

APPLY AGAIN DECISION TREE ALGORITHM WITH BEST FEATURES AND AGAIN CHECK SCORE

Again we will perform the Decision tree algorithm on model with these randomly selected set of best feature to get model score and finally will compare with previous model(with all features together).

Feature selection : The above 23 best feature selected randomly by search algorithm has been taken as new features.
We tried to perform parameter tuning by setting some parameters like `max_depth and max_features`. However the output score turned out to be very low. Hence we did not proceed further with parameter tuning in Decision Tree classifier. We continued with default parameters.

Training and fitting model: We have divided train- test set into 80% training and 20% test set .Then we fitted and trained our model with Decision tree classifier.

Validation of model: Model was validated using K-Fold cross validation method. However the best accuracy score obtained is 0.84 approx.

Evaluating model and testing on unseen data : Overall model from randomly generated best feature set gave approximate score of 0.80 or 80% accuracy score.

	precision	recall	f1-score	support
c-CS-m	0.65	0.62	0.64	24
c-CS-s	0.71	0.74	0.73	27
c-SC-m	0.91	0.84	0.87	25
c-SC-s	0.76	0.84	0.80	31
t-CS-m	0.78	0.88	0.82	32
t-CS-s	0.83	0.70	0.76	27
t-SC-m	0.87	0.83	0.85	24
t-SC-s	0.88	0.88	0.88	26
accuracy			0.80	216
macro avg	0.80	0.79	0.79	216
weighted avg	0.80	0.80	0.80	216

RESULT

After performing KNN algorithm twice (before and after selecting best feature) ,we found our tuned parameterized model with best 48 feature have overall very high accuracy score of 0.959 which is more than previous score with all features(which gave a score of 0.92).

After performing Decision tree algorithm twice (before and after selecting best feature) ,we found our model with best random 23 feature selected by algorithm have overall approximate low accuracy score of 0.80 which is less than previous Decision tree score with all features(which gave 0.85).

DISCUSSION

Comparing both types of model we found that model with KNN algorithm with selected best feature gave an accuracy score of 0.95 or 95% whereas model with Decision tree algorithm with selected best feature gave a score of 0.80 or 80% which is less than model with KNN algorithm. Therefore we shall proceed with our model with KNN algorithm which has very high score and it will help to identify subsets of proteins with distinguishing features to identify the class label of mice.

CONCLUSION

- Below are the final list of subsets of total 48 protein which all together determines class types of mice and when applied it to model gave high score of 0.95 or 95% approximately along with tuning of parameters.

```
['DYRK1A_N', 'ITSN1_N', 'BDNF_N', 'NR1_N', 'NR2A_N', 'pAKT_N',
'pBRAF_N', 'pCAMKII_N', 'pCREB_N', 'pELK_N', 'pERK_N', 'pJNK_N',
'PKCA_N', 'pMEK_N', 'pNR1_N', 'pNR2A_N', 'pNR2B_N', 'pPKCAB_N',
'pRSK_N', 'AKT_N', 'BRAF_N', 'CAMKII_N', 'CREB_N', 'ELK_N', 'ERK_N',
'GSK3B_N', 'JNK_N', 'MEK_N', 'TRKA_N', 'RSK_N', 'APP_N',
'Bcatenin_N', 'SOD1_N', 'MTOR_N', 'P38_N', 'pMTOR_N', 'DSCR1_N',
'AMPKA_N', 'NR2B_N', 'pNUMB_N', 'RAPTOR_N', 'TIAM1_N', 'pP70S6_N',
'NUMB_N', 'P70S6_N', 'pGSK3B_N', 'pPKCG_N', 'CDK5_N']
```

- Also model validation for these protein subsets showed a high score which ensured our model is good to identify class labels for unseen dataset.

REFERENCE:

https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression

https://towardsdatascience.com/