Student ID: s3798988

Student Name: Rohit Gupta

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*.

Title: Practical Data Science- Assignment 2

Name: Rohit Gupta

Student ID: s3798988

Contact details: s3798988@student.rmit.edu.au

Date of report – 10 June 2019

Table of Contents

Executive Summary

Research question: The aim of this report is to try and identify subsets of protein discriminant to each class.

The data set consists of the expression levels of 77 proteins/protein modifications that produced detectable signals in the nuclear fraction of cortex. There are 38 control mice and 34 trisomic mice (Down syndrome), for a total of 72 mice. In the experiments, 15 measurements were registered of each protein per sample/mouse. The dataset contains a total of 1080 measurements per protein. Each measurement was considered as an independent sample/mouse.

The eight classes of mice are described based on features such as Genotype, Behavior and Treatment. According to Dataset, Genotype can be either control or trisomic. Behavior of some mice have been 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.

INTRODUCTION

Memantine is used to treat moderate to severe Alzheimer's disease. It acts on the glutamatergic system by blocking NMDA receptors. It was first synthesized by Eli Lilly and Company in 1968 as a potential agent to treat diabetes; the NMDA activity was discovered in the 1980s.

Memantine is used to treat moderate-to-severe Alzheimer's disease, especially for people who are intolerant of or have a contraindication to AChE (acetylcholinesterase) inhibitors. Memantine has been associated with a moderate decrease in clinical deterioration with only a small positive effect on cognition, mood, behavior, and the ability to perform daily activities in moderate to severe Alzheimer's disease. There does not appear to be any benefit in mild disease.

Saline, also known as saline solution, is a mixture of sodium chloride in water and has a number of uses in medicine. Applied to the affected area it is used to clean wounds, help remove contact lenses, and help with dry eyes. By injection into a vein it is used to treat dehydration such as from gastroenteritis and diabetic ketoacidosis. It is also used to dilute other medications to be given by injection.

Classes(Target):

c-CS-s: control mice, stimulated to learn, injected with saline (9 mice)

c-CS-m: control mice, stimulated to learn, injected with memantine (10 mice)

c-SC-s: control mice, not stimulated to learn, injected with saline (9 mice)

c-SC-m: control mice, not stimulated to learn, injected with memantine (10 mice)

t-CS-s: trisomy mice, stimulated to learn, injected with saline (7 mice)

t-CS-m: trisomy mice, stimulated to learn, injected with memantine (9 mice)

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

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

Methodology:

We first import the dataset into Jupyter notebook. We prepare the data and clean it before getting it ready to fit into a model. For data preparation steps we get all the information about the dataset and all its descriptive statistical information using appropriate functions. We further check for presence of typos and extra white spaces in the dataframe as a precautionary measure to avoid any problems during modelling. Checks were introduced for unusual values, any extra type of categorical data apart from those mentioned in the dataset description. Then we check for missing values, which are present in large numbers and needs to be treated. To avoid reducing the number of observations in the datset I chose to treat those missing values using Simple Imputer of scikit learn.

Now we get to know about the data set and its descriptive features using univariate descriptive analysis and Bivariate descriptive analysis along with a plausible hypothesis. Detailed insights and results are given under Univariate Descriptive analysis and Bivariate descriptive analysis.

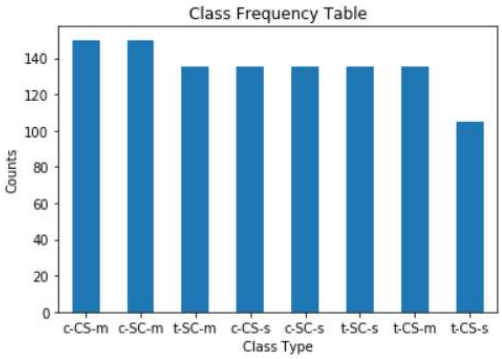
Description of the Dataset

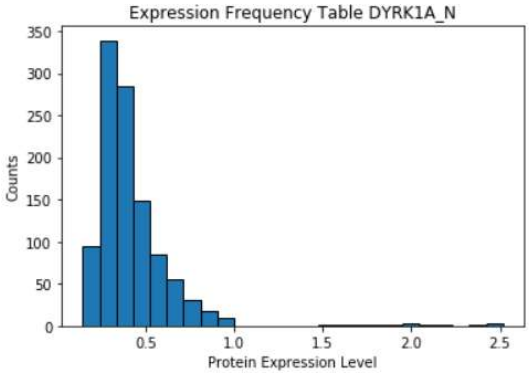
Size: (1080,82)

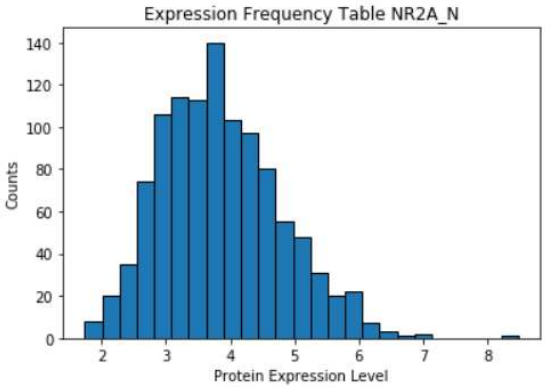
Univariate Descriptive Analysis

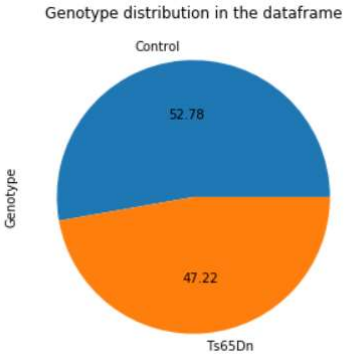
To understand and get more insight of the dataset, I tried to plot 10 important features using histograms, seaborn violin plots, pie chart, bar chart.

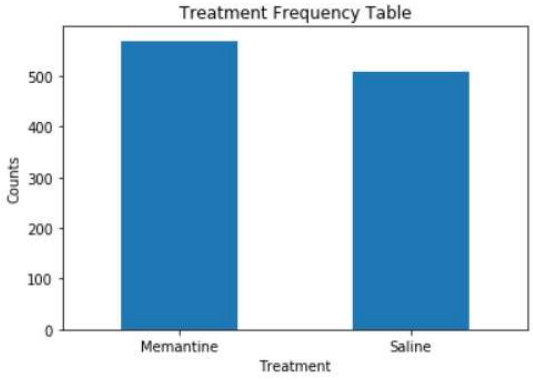
It was observed with the help of a bar plot that in the class variable, c-CS-m and c-SC-m have around 150 instances, t-SC-m, c-CS-s, c-SC-s, t-SC-s, t-CS-m have 135 instances respectively and t-CS-s has 105 instances. The distribution of protein DYRK1A_N lie within 1. Protein pCAMKII_N follows a normal distribution curve and its values ranges from 0 to 7. The Pie chart of Genotype shows that 'control' Genotype is about 53% and Trisomic Genotype is about 47% in this dataset. The number of instances of Memantine is 570 as compared to 510 instances of Saline. The density distribution plot of protein SOD1_N depicts the majority of its distribution well below its median values.The histogram plot of protein pCAMKII_N depicts that the distribution is skewed towards right. The distribution plots of protein NUMB_N and Bcatenin_N are more concentrated around its median values.

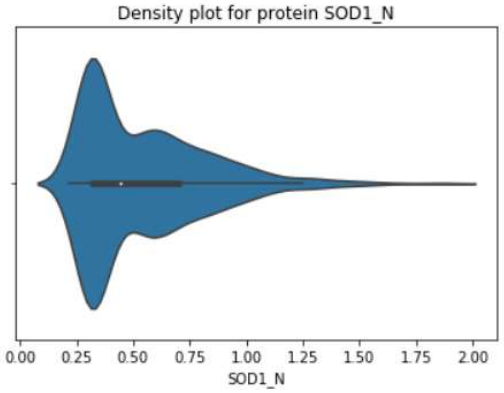

Class Frequency Table
Counts
Class Type
c-CS-m
c-SC-m
t-SC-m
c-CS-s
c-SC-s
t-SC-s
t-CS-m
t-CS-s

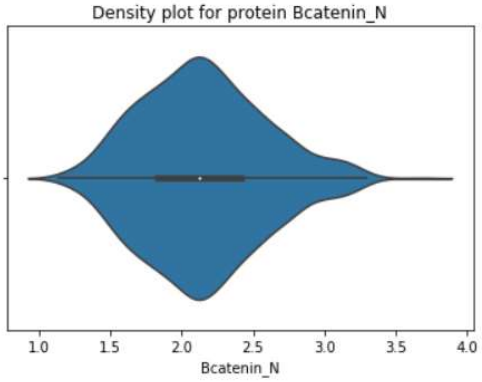

Expression Frequency Table DYRK1A_N
Counts
Protein Expression Level

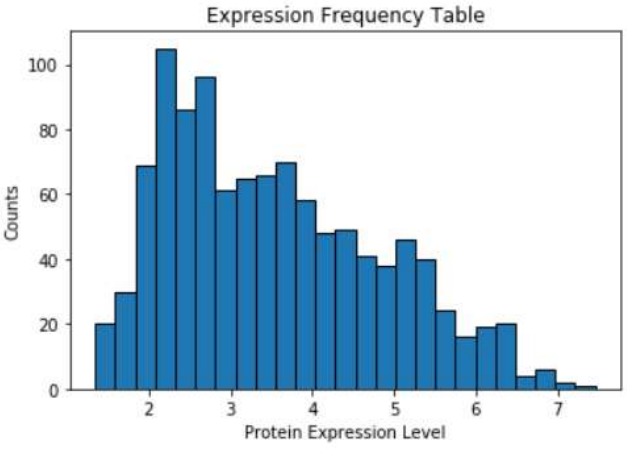

Expression Frequency Table NR2A_N
Counts
Protein Expression Level

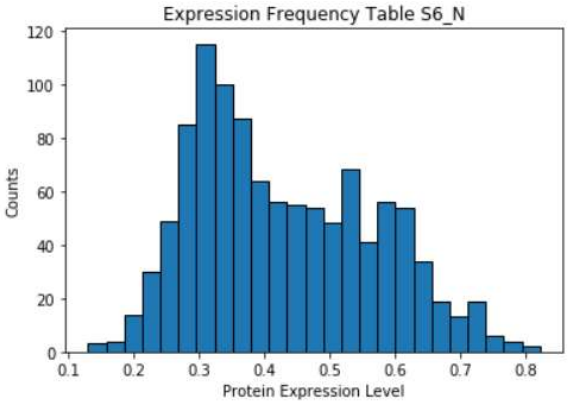

Genotype distribution in the dataframe
Control
52.78
47.22
Ts65Dn
Genotype

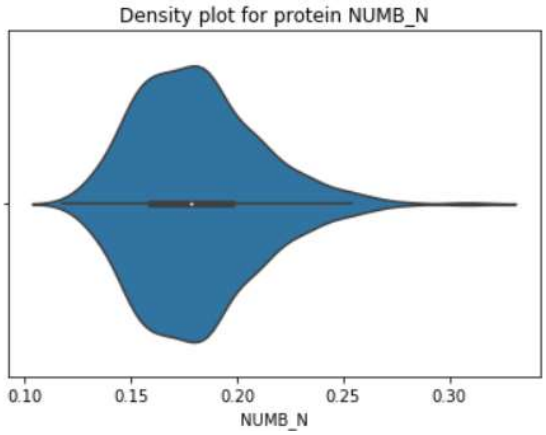

Treatment Frequency Table
Counts
Memantine
Saline
Treatment


Density plot for protein SOD1_N
SOD1_N


Density plot for protein Bcatenin_N
Bcatenin_N

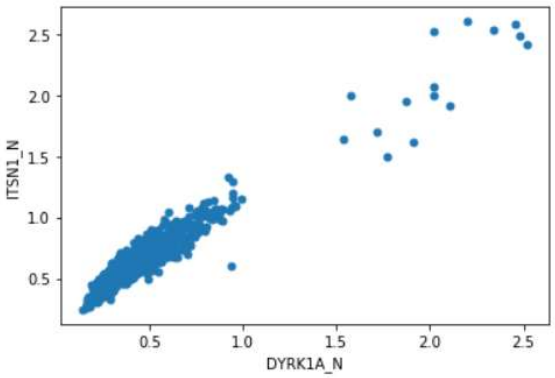

Expression Frequency Table
Counts
Protein Expression Level





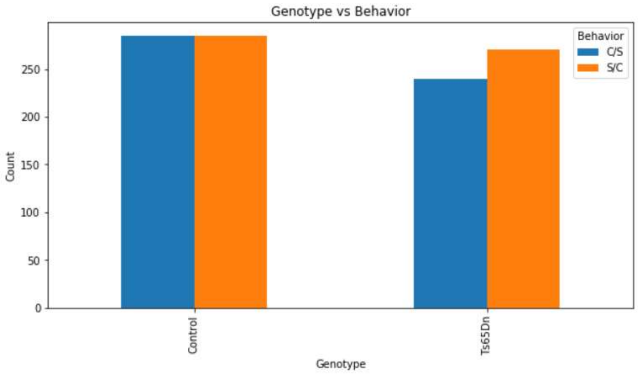
Multivariate Descriptive Analysis

For further investigation and more meaningful insights I tried to explore relationship between variables using Boxplot and Seaborn Violin plots to explore and compare the density relation and other descriptive features like median comparison, density distribution, etc between two protiens, scatter plot to determine correlation between two proteins. Below are 10 comparisons with plausible hypothesis, graphs and plots to justify and support the hypothesis.



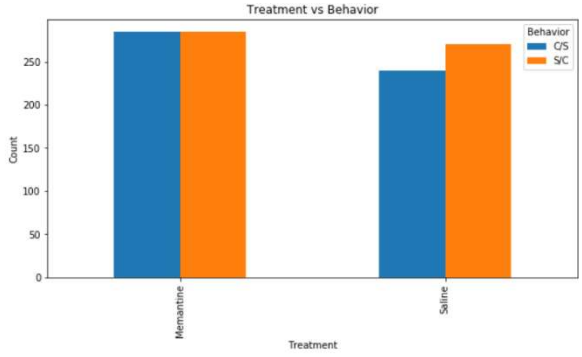
Hypothesis-1: Investigation to determine that protein DYRK1A_N and protein ITSN1_N have a linear relationship between them.

Observation: The above graph shows a positive linear relationship between the two proteins that supports my hypothesis



Hypothesis-2: Investigation to determine that Ts65Dn category mouse show more C/S type behavior as compared to S/C behavior irrespective of the type of drug Memantine or saline.

Observation: My hypothesis is proved to be wrong as the graph above depicts that Ts65Dn category mouse show more S/C type behavior as compared to C/S behavior irrespective of the type of drug Memantine or saline.

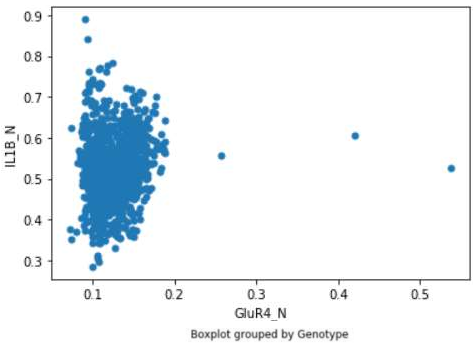


Hypothesis-3: Investigation to determine that all mouse injected with Memantine show more C/S type behavior as compared to S/C behavior irrespective of the type of mouse it is injected to.

Observation: My hypothesis is not exactly true as the graph above depicts that all mouse injected with Memantine show equal C/S type behavior and S/C behavior irrespective of the type of mouse it is injected to.

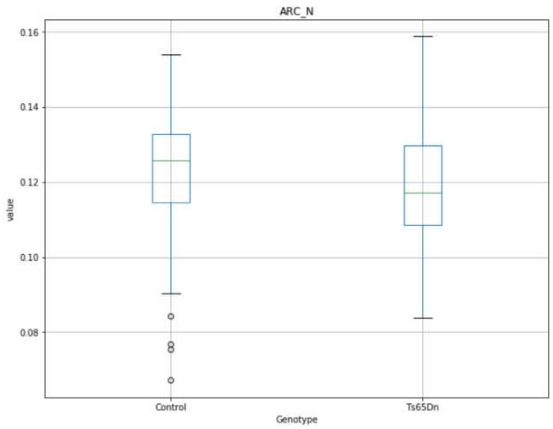
Hypotheis-4: Investigation to determine that protein GluR4_N and protein IL1B_N have a linear relationship between them

Observation: The above scatter plot shows that there is no specific kind of relation between protein type GluR4_N and protein IL1B_N as the points are well scattered in the plot.



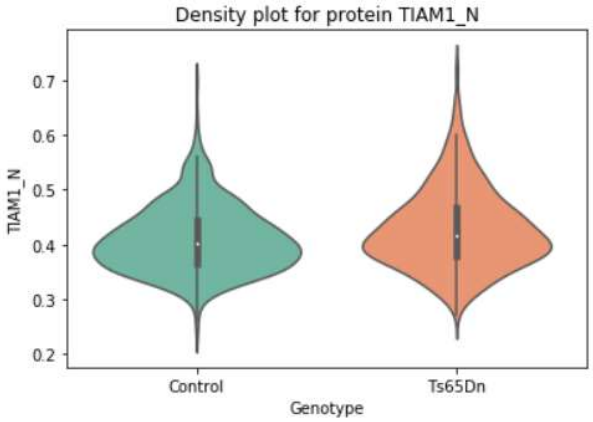
Hypothesis-5: Investigation to determine the equal distributon of protein ARC_N across both Genotypes

Observation: The above box plots shows that the distribution protein ARC_N is more varied in case of trisomic type mouse than that of control mouse.



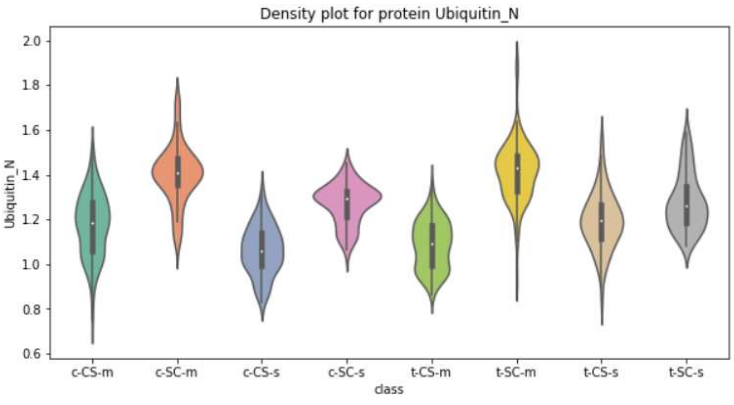
Hypothesis 6: Investigation to determine the similar distribution of protein TIAM1_N across both Genotype

Observation: The above graph supports my hypothesis that their distribution is similar and is more concentrated around their median.



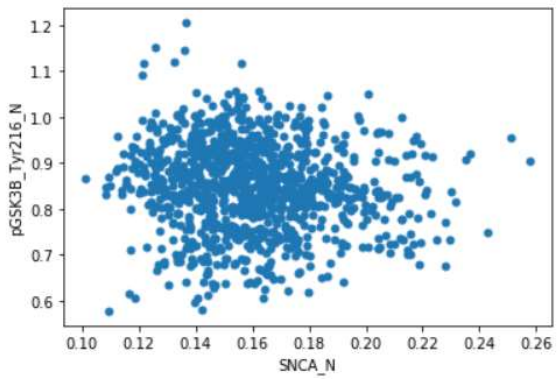
Hypothesis 7: Investigation to determine the similar distribution of protein Ubiquitin_N across all class

Observation: The hypothesis is not true as the above violin plot shows the different distribution of protein Ubiquitin_N across all class.



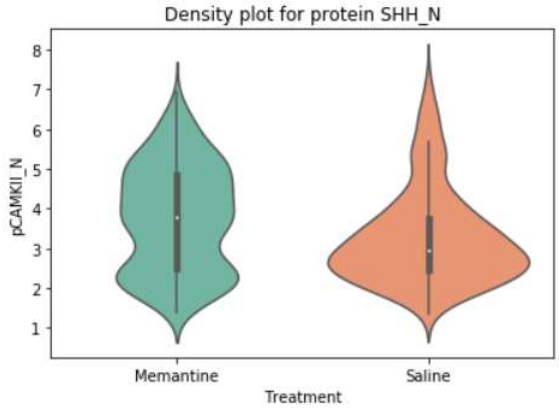
Hypothesis-8: Investigation to determine that protein SNCA_N and protein pGSK3B_Tyr216_N have a linear relationship between them

Observation: The above scatter plot shows that there is no relation between protein SNCA_N and protein pGSK3B_Tyr216_N as the points are well scatterd in the graph.



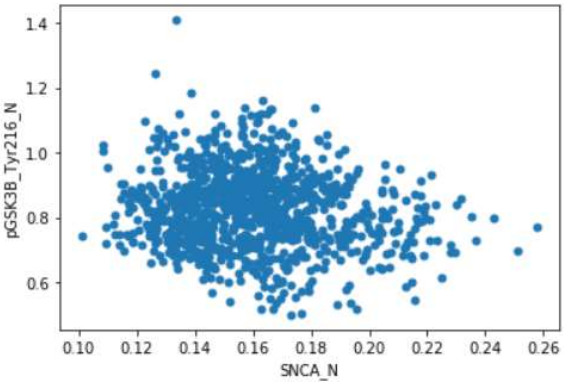
Hypothesis 9: Investigation to determine the similar distribution of protein pCAMKII_N across both Treatment type

Observation: The violin density plot suggests that the distribution of protein is more concentrated about the median in case of saline and better distributed in case of Memantine. Hence the hypothesis is not true.



Hypothesis-10: Investigation to determine that protein SNCA_N and protein pNR1_N have a linear relationship between them

Observation: The above scatter plot shows that there is no relation between protein SNCA_N and protein pNR1_N as the points are well scatterd in the graph



Data Modelling

In this report, classification was used to predict the 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). I have used K-Nearest Neighbor and Decision Tree Classification techniques to predict the class variable. Starting from the data preparation where we clean the dataset and treat the missing values using Simple Imputer from Scikit Learn. Then we drop multiple columns from the dataframe like mouseID which is unique variable and serves no purpose in prediction of the class. Furthermore, we drop the columns of 'Genotype', 'Treatment', 'Behaviour' from the dataframe as they provide redundant information and is same as provided by the 'class' - target column. Now we separate the descriptive features and target feature in a different dataframe an variable respectively. We are now left with descriptive dataset that is all numeric and which is scaled using MinMax Scaler of scikit learn to normalize the values of the dataframe. Now the dataset is ready for modelling.

We follow a specific process where the data is split into 3 categories- 50-50, 60-40 and 80-20 ratio. We train KNN and Decision tree models (default, hyperparameter tuning and feature Selection) in each category split and determine their accuracy scores

K-Nearest Neighbour Classifier

For KNN, the parameter tuning is done by varying **p**, **k** and **weights** parameter of the classifier. Following 6 types of models were trained to predict the target feature.

1. KNN Model (k=5, p=2, weights= uniform) - Default

2. KNN Model (k=5, p=1, weights= uniform)

3. KNN Model (k=3, p=2, weights= distance)

4. KNN Model (k=4, p=2, weights= uniform)

5. KNN Model (k=4, p=2, weights= distance)

Feature Selection using Hill Climbing Technique: Simple Hill climbing to feature selection: It examines the neighboring features one by one and selects the first neighboring feature which optimizes the current cost as next feature. Feature selection is important as it can help in saving a lot of computational power.

And the 6th model was trained with feature selection using **Hill climbing Technique** on the best model that we get from hyperparameter tuning.

Reason to choose values: We chose the above values for ease of understanding and to understand the effect of changing the parameters. There is a possibility that a better model could have been built using other combinations of K,p and weights using Gridsearch of scikitlearn module. We started with k=5 and tried reducing with k=4,3 which resulted in an improved performance. The basic idea is to try and find the best combination of parameters which gives us an optima accuracy score and simultaneously avoid overfitting.

Decision Tree:

For Decision Tree, the parameter tuning was done using **criterion** and **max_features** parameters of the classifier. Following 5 models were trained under each category of Train-Test Split and the best model was selected.

1. DT Model (criterion=Gini, max_featutres=None) - Default

2. DT Model (criterion=Entropy, max_featutres=None)

3. DT Model (criterion=Gini, max_featutres=5)

4. DT Model (criterion=Entropy, max_featutres=5)

And the 5th model was trained with feature selection using **Hill climbing Technique** on the best model that we get after hyperparameter tuning.

Reason to choose values: We chose the above values for ease of understanding and to understand the effect of changing the parameters. There is a possibility that a better model could have been built using other combinations of max_features and other parameters could have been tuned using Gridsearch of scikitlearn module. We tried varying parameters of max_features(None, 5) and criterion(gini, entropy) and also their combination to get an improved performance. The basic idea is to try and find the best combination of parameters which gives us an optima accuracy score and simultaneously avoid overfitting.

Results

There are multiple metrics to measure the performance of the model. Using Scikit Learn Classifiaction report we get scores of all these metrics which helps in model comparison. These metrics are **Accuracy**, **Precision**, **Recall** and **F1-score**.

Precision: The precision is the ratio `TP / (TP + FP)` where `TP` is the number of true positives and `FP` the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

Recall : The recall is the ratio `tp / (tp + fn)` where `tp` is the number of true positives and `fn` the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

Accuracy: It is simply a ratio of correctly predicted observation to the total observations. Accuracy = TP+TN/TP+FP+FN+TN.

F1-score: It is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. F1 Score = 2*(Recall * Precision) / (Recall + Precision).

We further take into account the confusion matrix that helps in understanding the above metric calculation.

Below is the performance of each classifier under each Category split that we obtain for the Mice Protein Expression Dataset.

Train-Test Split 50-50:

Model	Accuracy
1. KNN Model(k=5, p=2, weights= uniform)-	0.91
2. KNN Model(k=5, p=1, weights= uniform)-	0.92
3. KNN Model(k=3, p=2, weights= distance)-	0.97
4. KNN Model(k=4, p=2, weights= uniform)-	0.94
5. KNN Model(k=4, p=2, weights= distance)-	0.97
6. KNN Model(k=3, p=2, weights= distance) with feature selection -	0.98

Model	Accuracy
1. DT Model(criterion=Gini, max_featutres=None)-	0.78
2. DT Model(criterion=Entropy, max_featutres=None)-	0.82
3. DT Model(criterion=Gini, max_featutres=5)-	0.73
4. DT Model(criterion=Entropy, max_featutres=5)-	0.70
5. DT Model(criterion=Entropy, max_featutres=None) with feature selection -	0.83

Train-Test Split 60-40:

Model	Accuracy
1. KNN Model(k=5, p=2, weights= uniform)-	0.94
2. KNN Model(k=5, p=1, weights= uniform)-	0.95
3. KNN Model(k=3, p=2, weights= distance)-	0.97
4. KNN Model(k=4, p=2, weights= uniform)-	0.95
5. KNN Model(k=4, p=2, weights= distance)-	0.97
6. KNN Model(k=3, p=2, weights= distance) with feature selection -	0.99

1. DT Model(criterion=Gini, max_featutres=None)-	0.86
2. DT Model(criterion=Entropy, max_featutres=None)-	0.82
3. DT Model(criterion=Gini, max_featutres=5)-	0.73
4. DT Model(criterion=Entropy, max_featutres=5)-	0.78
5. DT Model(criterion=Entropy, max_featutres=None) with feature selection -	0.79

Train-Test Split 80-20:

Model	Accuracy
1. KNN Model(k=5, p=2, weights= uniform)-	0.97
2. KNN Model(k=5, p=1, weights= uniform)-	0.97
3. KNN Model(k=3, p=2, weights= distance)-	1.0
4. KNN Model(k=4, p=2, weights= uniform)-	0.97
5. KNN Model(k=4, p=2, weights= distance)-	0.99
6. KNN Model(k=3, p=2, weights= distance) with feature selection -	0.99

Model	Accuracy
1. DT Model(criterion=Gini, max_featutres=None)-	0.88
2. DT Model(criterion=Entropy, max_featutres=None)-	0.88
3. DT Model(criterion=Gini, max_featutres=5)-	0.78
4. DT Model(criterion=Entropy, max_featutres=5)-	0.75
5. DT Model(criterion=Entropy, max_featutres=None) with feature selection -	0.85

Confusion Matrix and Classification Report of the the Best Model(KNN Model-6)

```
[[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 57  0  0  1  0]
 [ 3  0  0  0 56  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]]
```

	precision	recall	f1-score	support
c-CS-m	0.95	1.00	0.97	58
c-CS-s	1.00	0.98	0.99	53
c-SC-m	1.00	1.00	1.00	62
c-SC-s	1.00	0.98	0.99	58
t-CS-m	1.00	0.95	0.97	59
t-CS-s	0.98	1.00	0.99	43
t-SC-m	0.98	1.00	0.99	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

Discussion:

The KNN and Decision Tree Model both gave more than 80% accuracy. However, KNN seems to outperform Decision tree classifier in all three Train-Test Category splits. KNN models have a very good precision score, recall score, and accuracy score in all three splits. With KNN models we will be able to better predict the classes of the mice it belongs to on the basis of mice protein expressions. Furthermore, our dataset is not too large and had a large number of missing values. With more appropriate information and even more number of observations we can train our model to perform even better and under more diverse instances. Secondly, We manually tried to tune the parameters of the models for the purpose of understanding the effect of parameter tuning. It is a possibility that a better model could have been developed with other combination of parameter using gridsearch of scikit learn.

Conclusion:

Our findings show that the KNN models performed better than Decision tree models in all three category splits. Amongst all three splits KNN model-3 in 80-20 Train-Test Split gave accuracy score of 1.0 where parameters were k=3, p=2, weights=distance. However, KNN model-6 with feature selection in 60-40 and 80-20 Train-Test split gave accuracy score of 0.99. So these two models are the best model amongst the rest as they used less number of features and still gave accuracy score of 0.99. Using this model, we can save a lot of computational power as the models use fewer models to predict the classes. Therefore, on the basis of Precision, Recall and

accuracy score we can say that the KNN model-6 with feature selection in 60-40 and 80-20 Train Test split have a comparatively similar performance and is best for the prediction of class.

References:

Higuera C, Gardiner KJ, Cios KJ . UCI Machine Learning Repository:

Mice Protein Expression Data. [online] Archive.ics.uci.edu. Available at:

https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression [Accessed 5 Jun. 2020].

Dr. Yongli Ren, 2020, ' Practical Data Science: Classification', Week 4, PowerPoint Slides, COSC2670,

RMIT University, Melbourne.

Dr. Yongli Ren, 2020, ' Practical Data Science: Classification', Week 6, PowerPoint Slides, COSC2670,

RMIT University, Melbourne.

Dr. Yongli Ren, 2020, ' Practical Data Science: Data Summarisation: Descriptive Statistics and

Visualisation, PowerPoint Slides, COSC2670, RMIT University, Melbourne.

Dr. Yongli Ren, 2020,'Data Modelling', PowerPoint Slides, COSC2670, RMIT University, Melbourne.