2020

COSC2670 Assignment 2

PRACTICAL DATA SCIENCE WITH PYTHON

NIKHIL SHARMA S3833151

S3833151@studnt.rmit.edu.au | RMIT UNIVERSITY

Student ID: S3833151

Student Name: Nikhil Sharma

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

Down syndrome, also known as trisomy 21 is a kind of genetic disorder which is caused because of the presence of an extra copy of chromone 21. Approximately 0.1% of the total live births (worldwide) suffer through down syndrome. It is not curable, but some early treatment programs can help to improve skills. Some of the treatments may include educational, speech or physical therapy. People with down syndrome can have physical problems along with intellectual abilities. Apart from these problems, people may also some other health problems like heart disease, hearing problems, problems with intestines, eyes and skeleton. Down syndrome causes learning and memory deficiency. The purpose of this report is to assess associative learning in 8 classes of control and down syndrome mice. We are provided 77 protein levels of these mice on the basis of a previous study. Our goal is to select the most important proteins which are discriminant between the classes and based on which we can consider some medication which would help in improving the learning and memory deficiency. The technique used is classification, by building a classification model we can predict what would be the class of the mice depending on the protein levels. After selecting the best subsets of proteins which included 40 proteins we build a classification model which is based on the K-Nearest Neighbours algorithm we can predict the class of a mice with 99.63% accuracy.

INTRODUCTION

Down syndrome describes a set of cognitive and physical symptoms that result from an extra copy or part of a copy of chromosome 21. It is because of chromosomes (which carry genes) that the body is able to develop in certain ways and performs certain functions. Almost all the cells of the human body contain 23 pairs of chromosomes (one set from each parent) for a total of 46 chromosomes. The people who suffer from down syndrome have either a full or partial extra copy of chromosome 21, for a total of 47 chromosomes. Down Syndrome is sometimes called as 47,XX,+21 or 47,XY,+21 to indicate the extra chromosome 21.

The extra chromosome interferes with the normal course of development and results in physical features and developmental and intellectual disabilities associated with the syndrome.

The degree of intellectual disability in people with down syndrome varies from person to person but is usually mild to moderate. Children with Down syndrome reach the key developmental milestone later than other children. People who are suffering from Down syndrome are also more likely to be born with heart abnormalities and they are at increased risk of developing vision and hearing problems. However, with appropriate support and treatment, many people with down syndrome lead happy and productive lives.

The dataset contains the expression levels of 77 proteins/protein modifications that produced some detectable signals in the nuclear fraction of cortex. This experiment was done on 38 control mice and 34 trisomic mice (mice which had down syndrome), so a total of 72 mice were in the experiment. The dataset categorised, the mice on the basis of their genotype, drug treatment they were given and the behaviour. Some mice were stimulated to learn by shock-context while others were simulated to learn by context-shock. Some of the mice were injected with memantine while some were injected with saline In this experiment, 15 measurements were registered of each protein per mouse. For the 38 control mice we get 38X15 or 570 measurements and the 34 trisomic mice we get 38X15 or 510 measurements. Each measurement of the sample is an independent measurement.

Below are the 8 classes of mice that we will use for our analysis:

1. c-CS-s: control mice, stimulated to learn, injected with saline (9 mice)
2. c-CS-m: control mice, stimulated to learn, injected with memantine (10 mice)
3. c-SC-s: control mice, not stimulated to learn, injected with saline (9 mice)
4. c-SC-m: control mice, not stimulated to learn, injected with memantine (10 mice)
5. t-CS-s: trisomy mice, stimulated to learn, injected with saline (7 mice)
6. t-CS-m: trisomy mice, stimulated to learn, injected with memantine (9 mice)
7. t-SC-s: trisomy mice, not stimulated to learn, injected with saline (9 mice)
8. t-SC-m: trisomy mice, not stimulated to learn, injected with memantine (9 mice)

In this report we will identify the best subset of proteins that will distinguish between the classes of mice.

METHODOLOGY

In this report, we will do a comparison between two machine learning models using the classification approach and choose the best model for us. The two models that we will be comparing for classification are K-Nearest Neighbours and Decision Tree. We use classification to solve the research question which is to identify the best subset of proteins that will distinguish between the classes. We use classification because, we can identify the variable which we want to predict (class) made up of 8 values. On the basis of the protein measurements we can categorize a measurement to one class. We do this by using two machine learning models and them comparing them with each other and finalizing the one which would give us the best results.

The first model that we will use is the K-Nearest Neighbours model. This model can be used for both supervised as well as unsupervised methods. In our case we will use the supervised neighbour's model for classification because our data has discrete labels. The principle behind this model is to find a predefined number of training samples closest to the new point for which we want to categorize into one of the existing class labels. The number of samples can either be a fixed constant or it may vary based on the density of points around the new point which we need to categorize. The distance can be any metric measurement but in most of the cases the standard Euclidean distance is used. There are several other parameters to the K-Nearest Neighbours algorithm. For this report the parameters that we will be using are

- metric: It is the measurement metric that we will use
- n_neighbors: It is the number of neighbours that we will use
- p: It is the power parameter for the metric
- weights: It specifies how to weigh the calculated distance

The second model that we will be using is the Decision Tree model. Decision tree is a supervised machine learning model. The goal of the model is to create a model which classifies a new point into one of the existing categories by learning simple decision rules from the training data. Like K-Nearest Neighbours, Decision tree also has several parameters., but for this report we will be focusing on the following parameters

- criterion: It is the function which is used to measure the quality of the split
- max_depth: It is the maximum depth of the tree
- max_features: It is the number of features to consider when looking for the best split
- max_leaf_nodes: It is the maximum number of terminal nodes in the tree
- min_samples_leaf: It is the minimum number of samples required to be a leaf node
- min_samples_split: It is the minimum number of samples required to split an internal node

We will use K-folds cross validation, in order to evaluate both the model's performance. The reason that we are using K-folds cross validation is that we don not have a lot of data with us, we only have 1080 observations, which we can split into training and test data and then evaluate the model only once. K-folds cross validation uses the entire data to train and test the model, it divides the data into k parts and uses each part one time as a test data while using the other part for training the model. Another reason for choosing the K-Folds cross validation is that if we use the train-test split only once there is a slight chance that similar data may fall into the test data and since the model is not trained on that data, it would not be able to predict the new values which are in the test data with a good accuracy. K-folds cross validation overcomes that issue as well by using all the data for the training and testing.

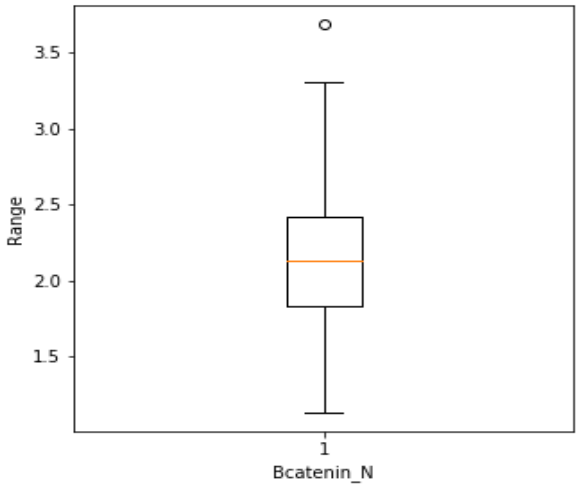
Before we do the model training and model evaluation, we would have to first do the steps in the data science process that are before the data modelling which are setting the research goal, retrieving the data, prepare the data and data exploration. Below are the steps taken in all the previous data science process stages.

1. Setting the research goal:
   We have already completed this step as we have set our research goal which is to identify the best subset of proteins that will distinguish between the classes of mice.
2. Retrieving the data:
   We do not need to perform this step since we already have the data retrieved for us, so we can skip this step.
3. Data preparation:
   The first step of the data preparation is to load the data into the python environment. Since the data provided to us in and excel format, we load the data by using the read_excel() function of the pandas library.
   Once the data is loaded, we have a look at the loaded data to make sure that it is loaded properly and if it is loaded properly, we look for any white spaces, White spaces can cause fatal errors in the project if not taken care of. To handle the white spaces, we use the strip function on our non-numerical columns of the dataset. Then, we check for any typos in the dataset and also we do the sanity checks to look for any impossible values. Both these things are done by using the value counts() function. Once this is done, we take care of the missing values. After looking for missing values, we find out that there are several missing values for several protein. One way of handling the missing values is to just delete the entry, we cannot do that because there are a lot of missing values in the data. Our data set is already pretty small and by deleting the missing values records we are reducing the size of the dataset even more which is not good. We cannot replace the missing value with a fixed value or the mean value of the column because the column has values for other categories and replacing the value with a fixed value or mean value won't do justice because if we replace it with a fixed values the results would be centred around that value and if we replace that value with the mean of the column it is also not good because, the mean value would have the readings of all the 8 classes and we just want to replace the missing value for a single class. The best way to do that and the way which is done in this report is by replacing the missing values of a protein for a particular class by replacing it with mean value of the same protein for that class. By doing this we can take care of all the missing values and the data is also not disturbed that much. After taking care of the missing values, we convert all the non-numerical columns to upper case in order to avoid any case sensitive mistakes. The last step of the data preparation is to convert the last 4 columns that are Genotype, Treatment, Behaviour and class to categorical columns because they contain categorical data in them. This is done using the Categorical() function of the pandas library.
   Once all this this is done, we are ready for tour data exploration stage.
4. Data exploration:
   First, we will explore 3 columns and look at their summary statistics. After that we will explore the relationships between 3 pairs of columns. Due to the page limit constraint of this report we will be exploring only 3 columns individually and 3 columns in pairs, for the complete data exploration of rest of other the columns, see the python notebook.
   We will be exploring the proteins 'Bcatenin_N', 'NUMB_N', 'JNK_N' because the are common in the both the machine learning models that we will use.

- Bcatenin_N: Below are the boxplot and the summary statistics of the protein Bcatenin.

We get to know that the spread of the data is pretty wide and most of the data falls in the interquartile range and there is only one outlier present in the Bcatenin protein column. Also, since the mean is greater than the median the data is skewed to the right.

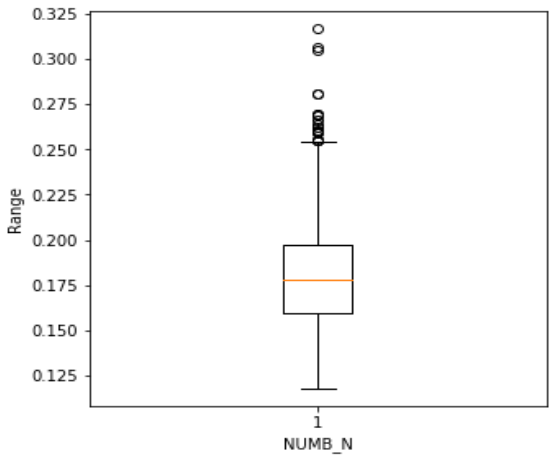
```
Sum = 2320.17902902423
Mean = 2.1483139157631763
Median = 2.1271399105
Standard deviation = 0.43210577229445163
Min = 1.134886146
Max = 3.680551799
```



- NUMB_N: Below are the boxplot and the summary statistics of the protein NUMB.

We get to know that the spread of the data is very small, it varies just from 0.118 to 0.315 and although most of the data falls in the interquartile range and there are a few outliers present in the NUMB_N protein column. Also, since the mean is greater than the median the data is skewed to the right.

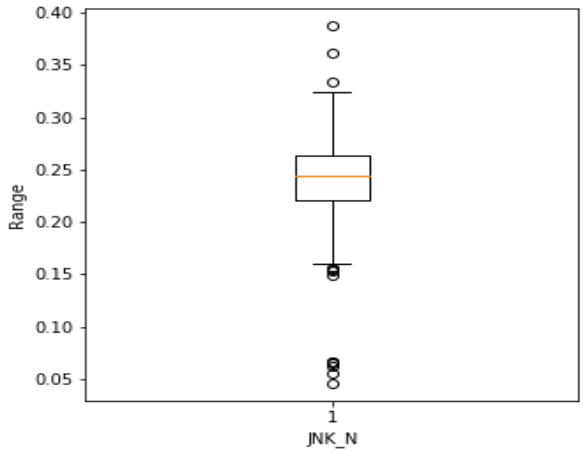
```
Sum = 195.566508338
Mean = 0.18108010031296295
Median = 0.1782350315
Standard deviation = 0.02928280939043787
Min = 0.117998506
Max = 0.316575348
```



- JNK_N: Below are the boxplot and the summary statistics of the protein JNK_N.

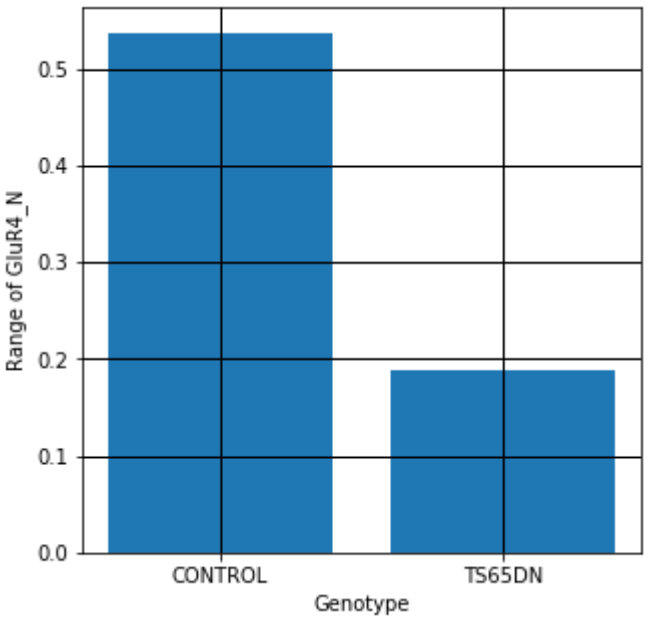
We get to know that the spread of the data is pretty wide and there are many outliers present in the JNK_N protein column. The inter quartile range of the dataset is pretty small. Also, since the mean is smaller than the median the data is skewed to the left.

```
Sum = 260.9795603501143
Mean = 0.24164774106492065
Median = 0.24481851999999998
Standard deviation = 0.033838137808961596
Min = 0.04629779
Max = 0.387190684
```

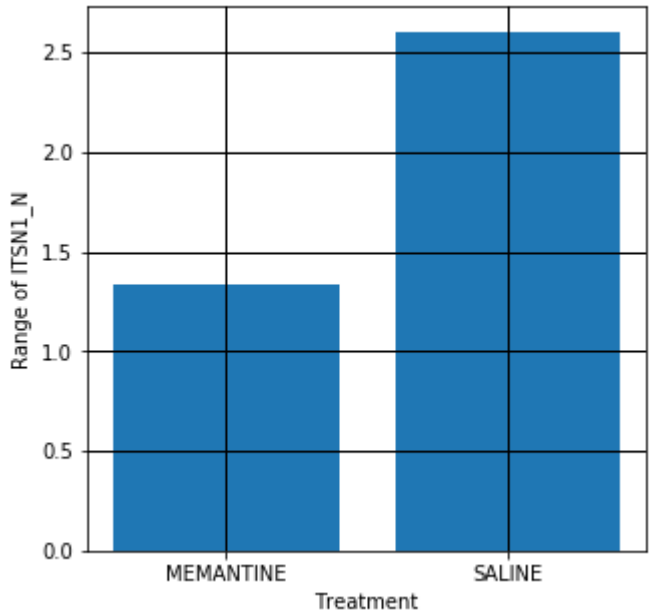


For the data exploration in pairs, we will explore the following pairs:

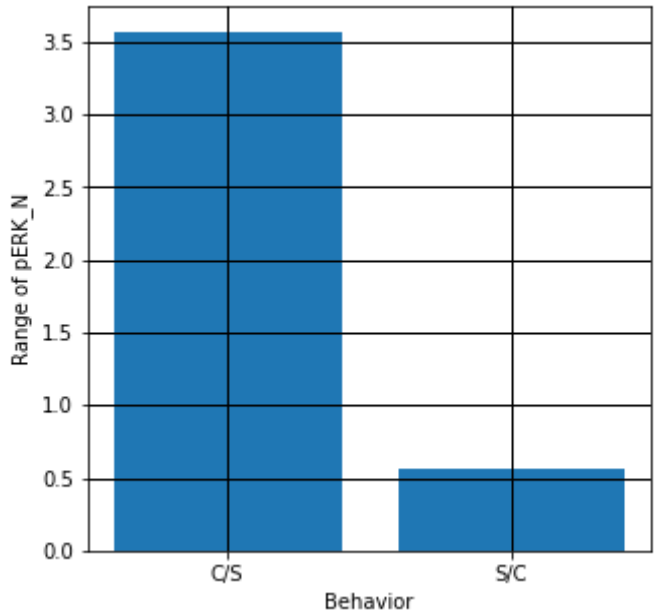
- Protein GluR4_N and Genotype: From the below graph we can say that of the value of the protein is below 1.9 there is a 50-50 chance that the reading may belong to the Control genotype or the trisomic genotype. But if the value is greater than 1.9, we can say with 100% confidence that the value will belong to the Control genotype.



- Protein ITSN1_N and Treatment: From the below graph we can say that of the value of the protein is below 1.33 there is a 50-50 chance that the reading may belong to the Memantine treatment or Saline treatment. But if the value is greater than 1.33, we can say with 100% confidence that the value will belong to the Saline treatment.



- Protein pERK_N and Behaviour: From the below graph we can say that of the value of the protein is below 0.55 there is a 50-50 chance that the reading may belong to the Context-Shock behaviour or Shock-Context behaviour. But if the value is greater than 0.55, we can say with 100% confidence that the value will belong to the Context-Shock behaviour.



5. Data modelling:

   The first thing which we do in the data modelling step is dividing the data into the independent variable (also referred as data in the python notebook) and the dependent variable (also referred as target in the python notebook).

   Our independent variable or data is only the set 77 proteins and the dependent variable or target is the class variable from the dataset.

Before fitting any model, we first normalize the data, because the K-Nearest Neighbours model predicts new values-based n distance. Some proteins have a measurement which is greater than 3, while some have measurement less than 0.5. We can not let these measurements fool our machine learning model that is why we normalize the data before fitting the model to it. To normalize the data, we use the MinMaxScaler. For the detail on how to use it you can refer to the python notebook.

Once the data is normalized, we are ready to train our model. At first, we provide the model all the features i.e. all the 77 protein measurements and train the model on it. Our goal in this step is to tune the parameters of the model in order to find those parameters based on which, the model's performance is the best. To evaluate the model's performance, we use K-Fold cross validation, because our dataset is not that large that we can just divide the dataset into training and testing and check the score only once. Because of this issue, we use K-Fold cross validation. In K-Fold cross validation, we use the complete dataset to train and test the model. At first, we specify the number of folds, in our case we use the number of folds to be 5. In our case since the number of folds is equal to 5, we will get 5 different values for the model's performance, one for each fold. The final model performance is the average score of all those 5 values. The next step is to do a hit and trial method in order to get the best parameters of the model. Every time we change a parameter, we evaluate the model's performance by the K-Fold cross validation. We do this, multiple times until we find the best parameters for our model. We do this for both K-Nearest Neighbours and Decision Tree.

There are some other methods of hyperparameter tuning available like GridSearchCV, RandomSearchCV as well instead of the hit and trial approach that we have followed. We did not use those approaches because they were beyond the scope of this report, therefore we stuck to our standard hit and trial error method.

After the hyperparameter tuning, we get the result as follows:

- K-Nearest Neighbours model with accuracy of 99.54%
- Decision Tree model with accuracy of 85.37%

RESULTS

One final step before finalizing our model is feature selection. Till now, we have chosen the best parameters for both our models, the final step is to select only those features which have the most impact on the models performance. Till now, we find out that the best model for classification for this project is the K-Nearest Neighbours model. We choose the K-Nearest Neighbours model because the accuracy of the model after parameter tuning is 99.54% while the accuracy of the Decision tree model after tuning the parameters is 85.37%. Now we will select the best features from all the features which have an impact on the model accuracy. To do this, we use hill climb techniques, by these techniques, we get only those which play a role in the model's performance. We will eliminate some features which do not play a role in the model's performance.

After applying the hill climbing technique to the K-Nearest Neighbours model, we came to know that out of the 77 proteins, only 40 had an impact on the model's performance. We now keep only those 40 features in the new dataset and evaluate the model again. The final 40 proteins that we are keeping in the dataset are 'Bcatenin_N', 'NUMB_N', 'JNK_N', 'EGR1_N', 'GluR3_N', 'GluR4_N', 'P3525_N', 'RRP1_N', 'DSCR1_N', 'GFAP_N', 'pERK_N', 'ARC_N', 'P38_N', 'BDNF_N', 'pPKCG_N', 'pNR2A_N', 'pGSK3B_N', 'AKT_N', 'CaNA_N', 'PSD95_N', 'pCASP9_N', 'pMTOR_N', 'NR2B_N', 'nNOS_N', 'MTOR_N', 'SHH_N', 'RAPTOR_N', 'CAMKII_N', 'BAD_N', 'NR1_N', 'P70S6_N', 'Tau_N', 'pNUMB_N', 'CDK5_N', 'ERK_N', 'pCREB_N', 'pPKCAB_N', 'DYRK1A_N', 'BCL2_N', 'S6_N'.

After training the K-Nearest Neighbours model on the new filtered data with the parameters that we found best previously, we finalize our model. Our K-Nearest Neighbour model is ready with 99.63% accuracy.

After applying the hill climbing technique to the Decision tree model, we came to know that out of the 77 proteins, only 22 had an impact on the model's performance. We now keep only those 22 features in the new dataset and evaluate the model again. The final 26 proteins that we are keeping in the dataset are Bcatenin_N', 'NUMB_N', 'JNK_N', 'EGR1_N', 'GluR3_N', 'GluR4_N', 'P3525_N', 'RRP1_N', 'DSCR1_N', 'pERK_N', 'pCFOS_N', 'pPKCG_N', 'AKT_N', 'CaNA_N', 'PSD95_N', 'pCASP9_N', 'NR2B_N', 'P70S6_N', 'CREB_N', 'NR2A_N', 'pGSK3B_Tyr216_N', 'pNR1_N'.

After training the Decision tree model on the new filtered data with the parameters that we found best previously, we finalize our model. Our Decision Tree model is ready with 82.69% accuracy.

Now we can say that our final model is the K-Nearest Neighbours model with 40 features and 99.72% accuracy. The accuracy of the model has increased since we have removed approximately 50% of the features from the dataset and have kept only the ones that matter the most. This is the benefit of feature selection. Also, the cost for training the model has been reduced significantly.

DISCUSSION

In both the classification models, we use K-Fold cross validation to assess the model's performance. We use the value of fold to be 5 which means that all the data will be used for training and testing, the model will be divided into training set and test set 5 times. In each iteration the size of the training data would be 80% of the total data and the size of the test data will be 20% of the total data. We used K-Fold cross validation because our dataset is not that big (it only has 1080 observations) that we can just divide it into training and test once and then check the model's evaluation. If we did that there was a chance that our model would not have learned properly.

The below table shows the accuracy of the two models in all the 5 iterations of the K-Fold

Model	Accuracy in Iteration 1	Accuracy in Iteration 2	Accuracy in Iteration 3	Accuracy in Iteration 4	Accuracy in Iteration 5	Average accuracy
K-Nearest Neighbours	100%	99.54%	100%	99.54%	99.07%	99.63%
Decision Tree	78.24%	87.96%	84.72%	82.41%	80.09%	82.69%

This model can be further improved by taking other parameters of the models into account as well. In this report only a few parameters have been tuned, if all the parameters were tuned a better accuracy can be obtained. Another thing that needs to be kept in mind is that the parameter tuning for this report is done by a hit and trial method. There are other methods as well for tuning the parameters like GridSeacrchCV, RandomizedSearchCV and many others which will not take as much time as the hit and trial method took. Unfortunately, these techniques are out of scope of this report. A better model might be obtained if the best values for all the parameters can be fetched from the previously mentioned techniques and all the parameters are taken into account.

CONCLUSION

Down syndrome is a condition which occurs to due to an extra copy or a part of copy of the chromosome 21. It is associated with the intellectual ability of a person. It is a very complex condition and cannot be cured. The only thing we can do to a person with Down syndrome is some sort of treatment in order to improve their skills.

In this report, we have applied two of the best classification models to the 77 set of proteins provide in order to select the best proteins that influence the learning and memory in the mice. From our models, we have come to the conclusion that we can distinguish between the classes of mice based on the 40 proteins out of the 77 that were provided. Also, from the analysis we get to know that SOD1, CaNA, nNOS and Ubiquitin played the biggest role in the memory and learning process.

We trained two models one was K-Nearest Neighbour model and the other was Decision tree model. We concluded that the K-Nearest Neighbour model was the best model for us. The model gave 99.63% accuracy as compared to the Decision tree model which gave us 82.69% accuracy.

REFERENCES

[1] U.S. National Library of Medicine 2018, *Down Syndrome*, U.S. National Library of Medicine, viewed 10 June 2020, <https://medlineplus.gov/downsyndrome.html>

[2] US Department of Health and Human Services 2017, *Down Syndrome: Condition Information*, US Department of Health and Human Services, viewed 10 June 2020, <https://www.nichd.nih.gov/health/topics/down/conditioninfo>

[3] Genetics Home Reference 2008, *Down syndrome*, Genetics Home Reference, viewed 10 June 2020, <http://ghr.nlm.nih.gov/condition/down-syndrome?>

[4] CDC 2011, *Facts about Down syndrome*, CDC, viewed 10 June 2020, <http://www.cdc.gov/ncbddd/birthdefects/DownSyndrome.html>

[5] CDC 2012, *World Down syndrome day*, CDC , viewed 10 June 2020, <http://www.cdc.gov/ncbddd/birthdefects/features/DownSyndromeWorldDay-2012.html>

[6] scikit-learn 2019, *Nearest Neighbors*, scikit-learn, viewed 10 June 2020, <https://scikit-learn.org/stable/modules/neighbors.html>