Assignment 2

Data Modelling and Presentation
COSC2760: Practical Data Science

Wednesday, 10 June 2020

s3694372

Labiba Islam

Email: s3694372@student.rmit.edu.au

Contents

Abstract

The goal of the assignment was to discuss which model should be used to find the "subset of proteins that are discriminant between the classes" *(1, UCI Machine Learning Repository: Mice Protein Expression Data Set, n.d.)*. The relationship between some proteins and the other features of the Mice Protein Expression Data Set was explored. To achieve the aforementioned goal, I built two models, a Decision Tree Classifier and a K-Nearest Neighbour (KNN) Classifier, that can predict which class a mouse belongs to with reasonable accuracy. From the analysis of the classification reports, it has been recommended that to predict with better accuracy, a KNN classifier should be used.

1. Introduction

The expression level of proteins gives a distinctive understanding of all proteins in a biological system. Changes in the normal levels may occur due to any condition present in the system. Protein levels can either increase or decrease, indicating that there is something wrong in the system. Protein expression levels are measured by "immunoblotting using the 9E10 antibody, ECL (Amersham) and densitometry" *(2, Protein Expression Level - an overview | ScienceDirect Topics, n.d.).*

The Mice Protein Expression Dataset is available in UCI Machine Learning Repository. This dataset contains the expression level of 77 proteins "measured in the nuclear fraction of cerebral cortex". There is a total of 72 mice, 38 of which are control mice. The remaining 34 mice have a condition called Down Syndrome and are called trisomic mice. For each protein per mouse sample, 15 measurements have been registered. Therefore, the dataset contains 1080 measurements of the 72 mice. There are four other columns in the dataset which are Genotype, Behavior, Treatment and Class. The Class column has 8 classes of mice whose features are described by Behavior, Treatment and Genotype. Genotype is either control or trisomic. Behavior is either C/S (context-shock, meaning these mice have been stimulated to learn) or S/C (shock-context, meaning these mice have not been stimulated to run). Treatment is either Memantine, a drug that has been injected to some mice, or Saline. *(1, UCI Machine Learning Repository: Mice Protein Expression Data Set, n.d.)*.

This report will discuss how I prepared the data for the analysis and some interesting visualisations that have been made in the data exploration stage. It will also discuss how I tuned the parameters to build the models to predict the class of the mice with fair accuracy.

2. Methodology

2.1 Retrieving and preparing the data

Since the downloaded dataset was in xls format, I used Microsoft Excel to convert the xls format to csv format. The csv file is, at first, loaded in Jupyter Notebook and then read using the function read_csv(). Then, I printed the first five rows to check that the data was properly imported and the headers are also appropriate. In this pre-processing stage, I checked the datatypes using dtypes and the number of rows and columns using shape.

I dropped the MouseID column using the drop() because I would not require this column in my analysis. I used a loop to remove all redundant whitespaces using the function str.strip(). To check if there are any typos in the last 4 columns (Genotype, Behavior, Treatment and Class), I used the function value_counts() and found out that there were no typos.

Next, I used the function isnull().any() to print the names of the columns with missing values and then used functions fillna() and mean(axis=0) to replace the NaN values with the mean of the respective columns. Since there were a lot of columns, I again used isnull().any() to ensure that there were no more missing values.

2.2 Data exploration

I used the python Matplotlib library to visualise the data. As mentioned in the specification, at first I explored 10 columns individually. I used pie charts and bar charts for the categorical data and box plots for numerical data. In the next part of data exploration, I explored the relationship between pairs of attributes. To help me with this, I plotted two scatter matrices at first with some features, so that I could find some interesting relationships and use them to plot scatter plots. I removed the scatter matrices from my code so that the execution of the code does not slow down. I also made some box plots to show the numerical data grouped by categorical data. Screenshots of the graphs will be attached in the Results section.

2.3 Data modelling

The classification models, Decision Tree and KNN, were imported from sklearn. I began by dropping Genotype, Behaviour, and Treatment columns as I aimed to find only the subset of proteins that are discriminant between the classes. In this section of the report, I will discuss how I tuned the parameters for each model, how feature selection was done, and how I built the classifiers and generate the classification reports.

2.3.1 Decision Tree Classifier

The parameters to be tuned for decision tree were max_depth, mim_samples_split, min_samples_leaf, max_features and max_leaf_nodes.

I have used ‘gini’ as criterion. To tune my parameters, I used a loop to calculate the values and print the best value. Then I used Hill Climbing algorithm for feature selection and created a new dataset with columns only selected by Hill Climbing. Then I proceeded to make my decision tree classifier. While tuning the parameters, I had to make sure that I minimise overfitting and under-fitting as much as possible. If the depth is too low, the model will under-fit and if the depth is too high, the model will become more complex as it will have more splits and capture more information which is one of the main reasons for overfitting in decision trees. For min_samples_split , higher values can lead to under-fitting, and since they have a range from 1 to 40 for the CART algorithm implemented by scikit-learn. For min_samples_leaf, the range is from 1 to 20 and a very small value will cause overfitting while a very large value will not allow the tree to learn the data *(3, Mithrakumar, 2019)*. I have tried different combinations of the parameters with varying training and test set size. I split the data using test_train_split. The parameters that gave me the highest accuracy with 90% training data and 10% testing data are as follows: max_depth=9, min_samples_split=10, min_samples_leaf=4, max_features=8, max_leaf_nodes=45.

2.3.2 K-Nearest Neighbour (KNN) Classifier

I started with using the Hill Climbing algorithm to find the selected features and created a new dataset from the original dataset with only the selected columns. Then I proceeded to build my KNN classifier. The parameter to be tuned for KNN was the number of neighbours only. I used ‘distance’ as weight as I did not want all points in each neighbourhood to be weighted equally. With ‘distance’, closer neighbours will have a greater impact than neighbours which are further away *(4, Lecture slides*). For p, I used 2 which is Euclidean. I did not use 1 (Manhattan) as my dataset did not have more features than samples and thus it was not high dimensional *(5, Gohrani, 2019).* I used a different number of neighbours and used the clarification report to see which one gave the highest accuracy. I split the data using test_train_splitThe parameters that gave me the highest accuracy with 90% training data and 10% testing data are as follows: number of neighbours=10, weights='distance', p=2.

3. Results

This section will show the visualisations I created during the data exploration stage. The first 10 figures are exploring individual columns. The second 10 will show 10 relationships between 10 pairs of columns. The plausible hypotheses will be mentioned in the Discussion section.

3.1 Exploring single columns

Figure 1: Showing the proportion of Control and Trisomic Genotype

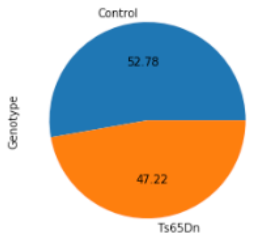


Figure 2: Showing the number of mice given Memantine and Saline treatment

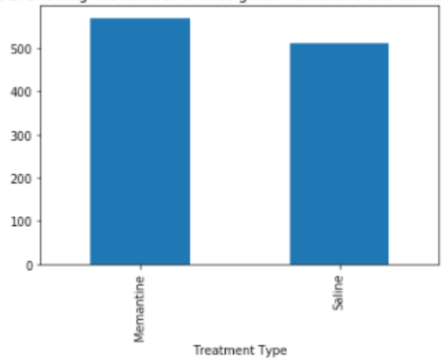


Figure 3: Showing the number of mice who have been stimulated to learn, C/S and who have not been stimulated, S/C

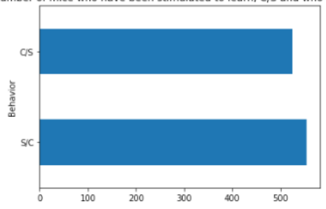


Figure 4: Showing the proportion of the different classes of mice

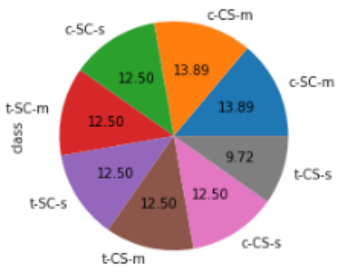


Figure 5: Box plot for the expression level of DYRK1A_N

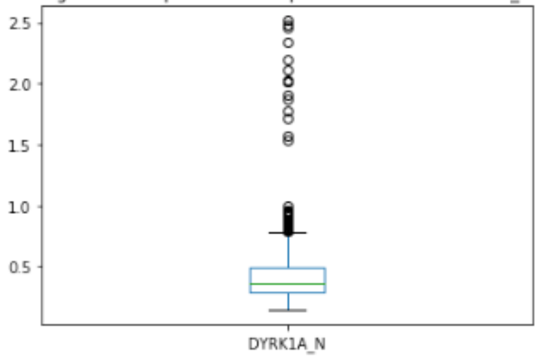


Figure 6: Box plot for the expression level of pCAMKII_N

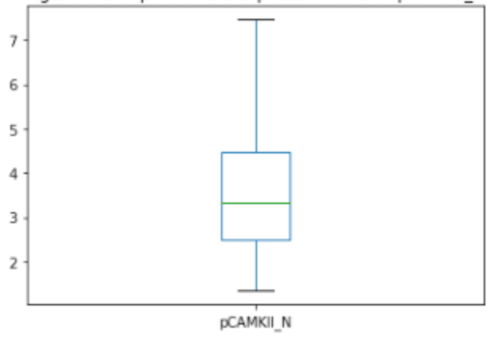


Figure 7: Box plot for the expression level of pERK_N

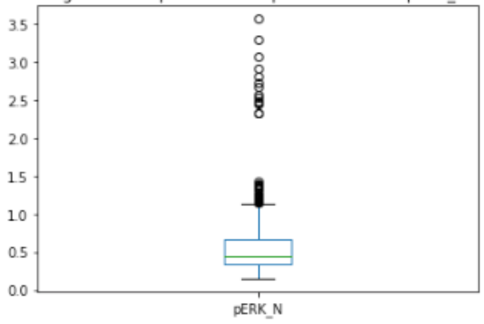
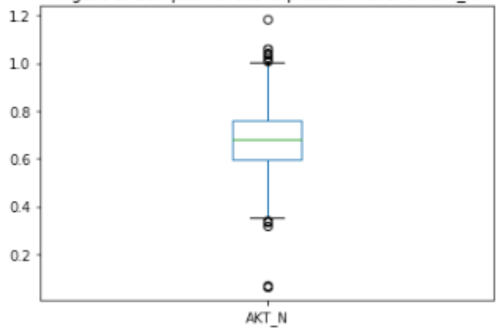


Figure 8: Box plot for the expression level of AKT_N



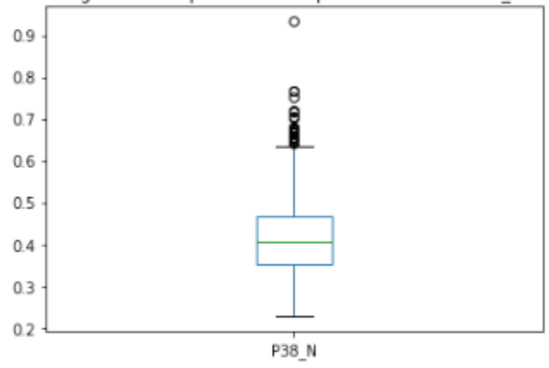


Figure 9: Box plot for the expression level of P38_N

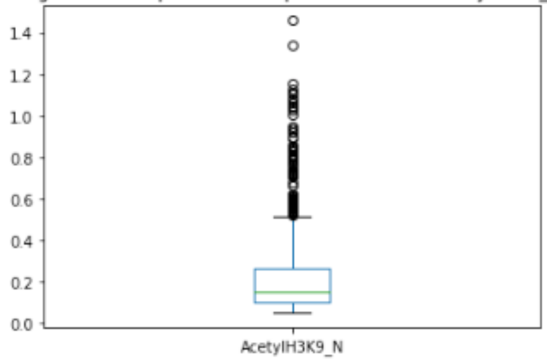


Figure 10: Box plot for the expression level of AcetylH3K9_N

3.2 Exploring relationship between columns

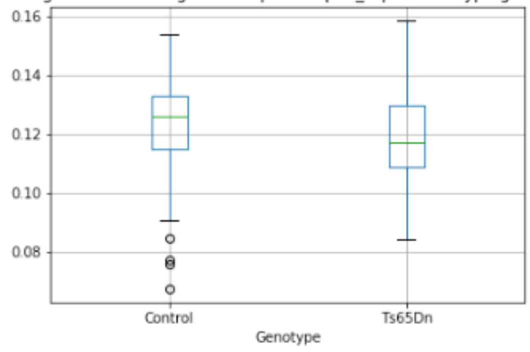


Figure 11: Showing the box plot for pS6_N per Genotype group

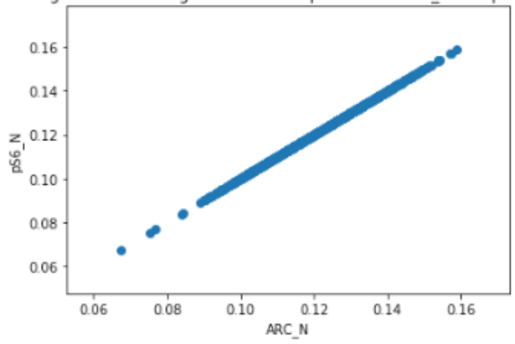


Figure 12: Showing the relationship between ARC_N and pS6_N

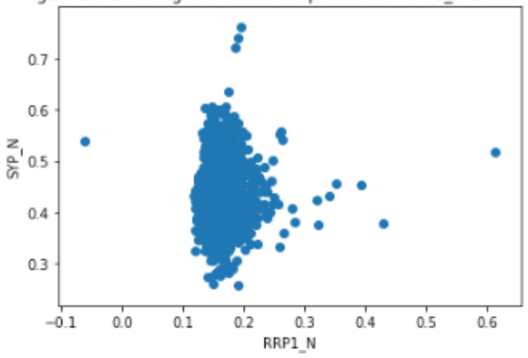


Figure 13: Showing the relationship between RRP1_N and SYP_N

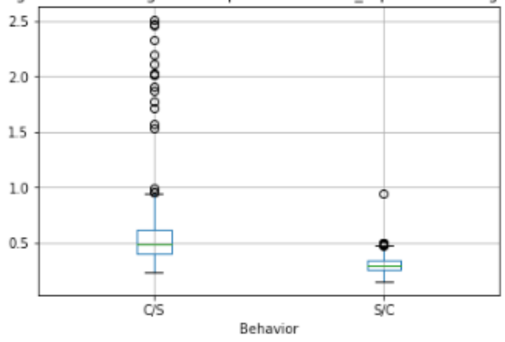


Figure 14: Showing the box plot for DYRK1A_N per Behavior group

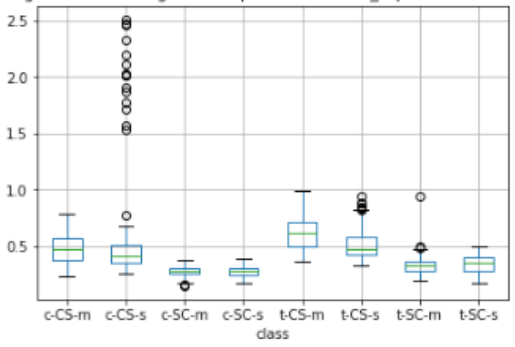


Figure 15: Showing the box plot for DYRK1A_N per class of mice

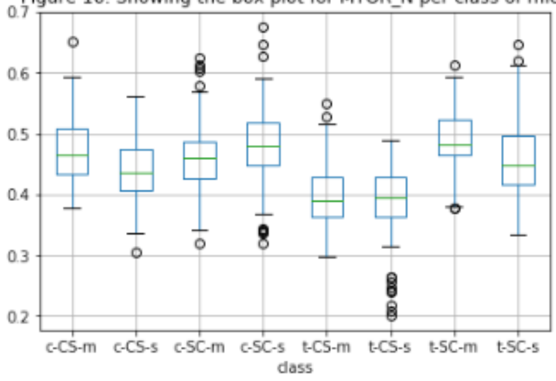


Figure 16: Showing the box plot for MTOR_N per class of mice

Figure 17: Showing the relationship between RRP1_N and MTOR_N

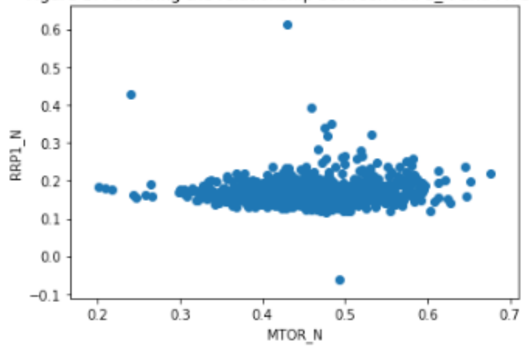


Figure 18: Showing the relationship between ARC_N and GFAP_N

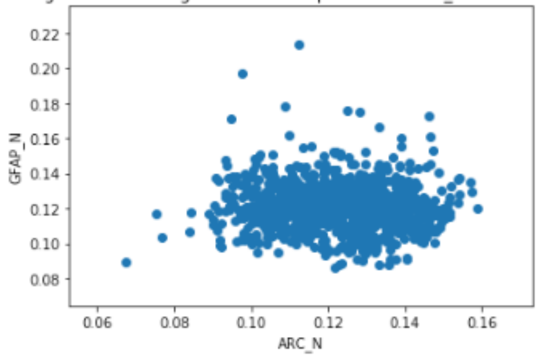


Figure 19: Showing the relationship between ARC_N and pCASP9_N

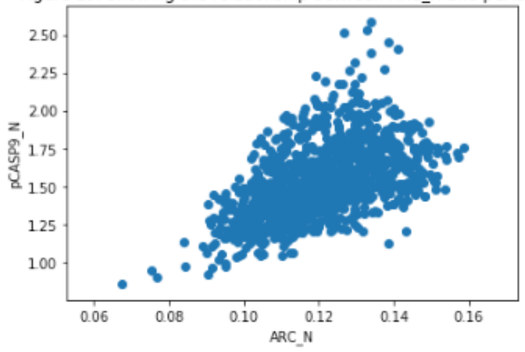
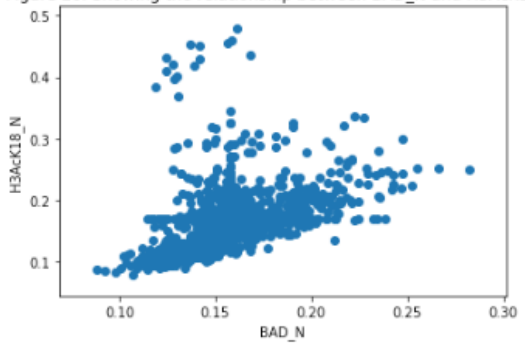


Figure 20: Showing the relationship between BAD_N and H3AcK18_N



3.3 Classification report of decision tree and KNN classifiers

```
Figure 21: Classification report for Decision Tree Classifier
               precision    recall  f1-score   support

       c-CS-m       0.79      0.62      0.70        24
       c-CS-s       0.50      0.73      0.59        11
       c-SC-m       0.89      0.76      0.82        21
       c-SC-s       0.60      0.75      0.67        12
       t-CS-m       0.73      0.85      0.79        13
       t-CS-s       0.88      0.78      0.82         9
       t-SC-m       0.90      0.75      0.82        12
       t-SC-s       0.86      1.00      0.92         6

     accuracy                           0.75       108
    macro avg       0.77      0.78      0.77       108
 weighted avg       0.77      0.75      0.75       108
```

```
Figure 22: Classification report for KNN Classifier
               precision    recall  f1-score   support

       c-CS-m       0.96      0.92      0.94        24
       c-CS-s       0.91      0.91      0.91        11
       c-SC-m       0.86      0.90      0.88        21
       c-SC-s       0.62      0.67      0.64        12
       t-CS-m       0.91      0.77      0.83        13
       t-CS-s       0.80      0.89      0.84         9
       t-SC-m       0.82      0.75      0.78        12
       t-SC-s       0.57      0.67      0.62         6

     accuracy                           0.83       108
    macro avg       0.81      0.81      0.81       108
 weighted avg       0.84      0.83      0.84       108
```

3.4 Resulting decision tree

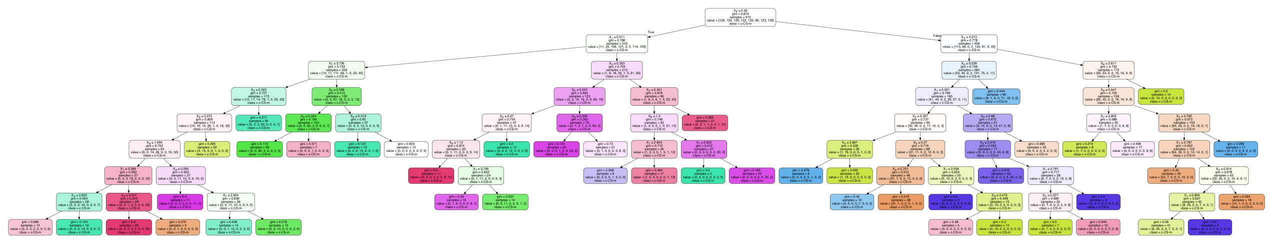


Figure 23: Decision Tree

4. Discussion

4.1 Exploring single columns

From the graphs in 3.1, we can see that there are more mice with Down Syndrome than control mice and a larger number of mice have been treated with Memantine. Compared to mice who have not been stimulated to learn, there are fewer mice who have been stimulated to learn. Figure 4 shows the proportion of different classes of mice and trisomic mice who haven stimulated to learn and not treated with Memantine is the smallest in number. Expression levels of DYRK1A_N, pCAMKII_N, pERK_N and AcetylH3K9_N are positively skewed while AKT_N and P38_N are normally distributed. DYRK1A_N, pERK_N and AcetylH3K9_N have a lot of outliers as can be seen from figures 5, 7 and 10.

4.2 Plausible hypotheses

Figure 11: Trisomic mice tend to have lower pS6_N expression levels.
Figure 12: Expression levels for ARC_N and pS6_N are more or less the same.
Figure 13: Expression level of SYP_N varies more than that of RRP1_N.
Figure 14: Mice that have been stimulated to learn has a higher expression level of DYRK1A_N.
Figure 15: Trisomic mice who have been stimulated to learn and have been treated have the highest level of DYRK1A_N.
Figure 16: Mice without Down syndrome and who has neither been treated and nor been stimulated to learn has a higher expression level of MTOR_N.
Figure 17: With varying expression levels of MTOR_N, the expression level of RRP1_N fairly remains constant.
Figure 18: Expression levels of ARC_N are relatively lower than GFAP_N.
Figure 19: Mice with higher levels of ARC_N tend to have higher levels of pCASP9_N.
Figure 20: Mice with higher levels of BAD_N tend to have higher levels of H3AcK18_N.

4.3 Classification report

Different combinations of the parameters and different test sizes gave different precisions. I will discuss some cases here. In the case of decision tree, for 90% training data size with max_features = 8 gave a weighted average precision of 0.77 and 0.69 for max_features = 10 where all the other parameters remained the same. This showed that for the dataset that I used, reducing the value for max_features increased the accuracy. On the other hand, in the case of KNN, for 90% training data size reducing the number of neighbours from 20 to 10, increased the weighted average precision from 0.76 to 0.84. The weighted average precision was the same (0.76) for 90% training set with 20 neighbours and 80% training set with 10 neighbours. In section 3.3 of this report, Figures 21 and 22 show the highest weighted average precision for decision tree and KNN classifiers.

5. Conclusion: Recommendation

According to how I tuned and tested with different parameters for both the classifiers, the weighted average precision for decision tree ranged from 0.67 to 0.77. For KNN, it ranged from 0.69 to 0.84. As mentioned in section 2.3, the parameters that gave me the highest accuracy for decision tree are as follows: max_depth=9, min_samples_split=10, min_samples_leaf=4, max_features=8, max_leaf_nodes=45 and for KNN the parameters are number of neighbours=10, weights='distance', p=2. Both the maximum weighted average precision that I got from my classification reports are for 90% training data set size and 10% testing data set size. Keeping 90% for training was also necessary as the dataset was really small (it has only 1080 samples). Considering all the factors mentioned above, in order to predict the classes from a subset of proteins for this dataset, I would recommend a KNN classifier.

6. References

1. Archive.ics.uci.edu. n.d. *UCI Machine Learning Repository: Mice Protein Expression Data Set*. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression> [Accessed 10 June 2020].

2. Sciencedirect.com. n.d. *Protein Expression Level - An Overview | Sciencedirect Topics*. [online] Available at: <https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/protein-expression-level> [Accessed 10 June 2020].

3. Mithrakumar, M., 2019. *How To Tune A Decision Tree?*. [online] Medium. Available at: <https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680> [Accessed 10 June 2020].

4. Week 6 lecture slides

5. Gohrani, K., 2019. *Different Types Of Distance Metrics Used In Machine Learning*. [online] Medium. Available at: <https://medium.com/@kunal_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7#:~:text=Manhattan%20distance%20is%20usually%20preferred,similarity%20between%20two%20data%20points.> [Accessed 10 June 2020].