Assignment 2

Artificial Neural Network

Kieran Best

Artificial Intelligence Computing

MCOMD2AIC

09/04/2022

Page of Contents

[Implementation 1](#_Toc103867264)

[Data Preparation 1](#_Toc103867265)

[Artificial Neural Network 1](#_Toc103867266)

[Other Models 2](#_Toc103867267)

[Support Vector Machine 2](#_Toc103867268)

[K-Nearest Neighbour 3](#_Toc103867269)

[Ethical and legal issues 3](#_Toc103867270)

[Conclusion 3](#_Toc103867271)

[Link to Source Code 4](#_Toc103867272)

[Link to Video Walk Through 4](#_Toc103867273)

[References 5](#_Toc103867274)

[Appendix 7](#_Toc103867275)

[Appendix 1 7](#_Toc103867276)

[Appendix 2 7](#_Toc103867277)

[Appendix 3 8](#_Toc103867278)

[Appendix 4 8](#_Toc103867279)

[Appendix 5 8](#_Toc103867280)

[Appendix 6 9](#_Toc103867281)

[Appendix 7 9](#_Toc103867282)

[Appendix 8 10](#_Toc103867283)

[Appendix 9 10](#_Toc103867284)

[Appendix 10 10](#_Toc103867285)

[Appendix 11 10](#_Toc103867286)

[Appendix 12 10](#_Toc103867287)

[Appendix 13 10](#_Toc103867288)

[Appendix 14 10](#_Toc103867289)

[Appendix 15 10](#_Toc103867290)

[Appendix 16 11](#_Toc103867291)

[Appendix 17 11](#_Toc103867292)

[Appendix 18 11](#_Toc103867293)

[Appendix 19 11](#_Toc103867294)

[Appendix 20 11](#_Toc103867295)

[Appendix 21 11](#_Toc103867296)

[Appendix 22 11](#_Toc103867297)

[Appendix 23 11](#_Toc103867298)

[Appendix 24 11](#_Toc103867299)

[Appendix 25 12](#_Toc103867300)

[Appendix 26 12](#_Toc103867301)

[Appendix 27 12](#_Toc103867302)

[Appendix 28 12](#_Toc103867303)

[Appendix 29 12](#_Toc103867304)

[Appendix 30 12](#_Toc103867305)

[Appendix 31 13](#_Toc103867306)

[Appendix 32 13](#_Toc103867307)

[Appendix 33 14](#_Toc103867308)

# Implementation

The data is taken from males in a high-risk heart disease area of the Western Cape, South Africa; many of the men who have had a Coronary Heart Disease (CHD) event have undergone blood pressure reduction treatment or other programs to reduce their risk factors of having another CHD event (Mantovani, 2015). The aim of this report is to evaluate different methods of predicting CHD and see which method is better.

## Data Preparation

The first thing besides from being able to read the data, is to import the necessary libraries so that we can read and edit the data, ready to use in the machine learning approaches for prediction as seen in Appendix 1.

* “from scipy.io import arff” – enables us to manipulate and visualise data saved as an arff file as a record array.
* “pandas” – enables us to manipulate and analyse data.
* “numpy” – enables us to create and manipulate arrays and matrices and gives us access to mathematical functions.
* “from sklearn.preprocessing import LabelEncoder” – enables us to edit data and labels.
* “from sklearn.metrics import confusion\_matrix” – enables us to evaluate accuracy of a classification when comparing a predicted result against an observational result (scikit-learn, 2022).
* “from sklearn.model\_selection import train\_test\_split” – is used to create the train and testing sets to determine the efficiency of the selected methods predictive performance.

After importing the datafile (Appendix 1) we then change the names of the attributes from that seen in Appendix 1, using Appendix 2, and checking that everything is correct in Appendix 3 by listing the column headers. Now to ensure that the attributes “family history” and “CHD” are true to their meaning, binary is used as 1 equals true and 0 equals false (as can be seen in Appendix 4 and 5).

From here the next step is to create the train and test sets; according to Gholamy, Kreinovich, & Kosheleva (2018) testing sets that use between 20-30% receive the best results, therefore we shall use 20% across all methods. In Appendix 6 you can see that we have copied the ‘chd’ column that tells us if a participant has CHD or not, this is our base logic that we will compare against; it is then dropped from the rest of the data in Appendix 7. “x” and “y” are then assigned to “x\_arr” and “y\_arr” as arrays to make it accessible to create training and testing sets (Appendix 8). As “y\_arr” is the base logic being compared against, it must be a single array otherwise it cannot be compared against (Appendix 9 and 10). The training and testing sets are then created using the “x\_arr” and “y\_arr” arrays, it creates a testing set using 20% of the data.

## Artificial Neural Network

When using an Artificial Neural Network, the first thing we must do is import the necessary libraries as seen in Appendix 12.

* “from sklearn.metrics import make\_scorer, accuracy\_score” – the “make\_scorer” function “wraps scoring functions for use in GridSearchCV” using the “accuracy\_score” (scikit-learn, 2022a).
* “from sklearn.model\_selection import GridSearchCV” – “GridSearchCV” “implements a “fit” and “score” method to be used by cross validation for the training and testing sets.
* “from sklearn.neural\_network import MLPClassifier” – stands for Multi-Layer Perceptron and is a classification algorithm that relies on an underlying neural network (Nair, 2019).

We then go on to setting “ann\_clf” as the “MLPClassifier” method and setting the parameters that are used for the “GridSearchCV” in Appendix 13. In Appendix 13 we then set the type of scoring to be used, which we’ve already imported the library for (“make\_scorer” and “accuracy\_scorer”), run the grid search (using parameters we’ve already defined), and picks the best combination of parameters to be used. Appendix 14 then fits that best algorithm to the training data set and from this we can use that algorithm on the test data set in Appendix 15 to predict whether each participant has or has not had CHD as the output “y\_pred\_ann”. And lastly in Appendix 16 and 17 we determine our accuracy of this algorithm using a confusion matrix comparing our predicted outcome (“y\_pred\_ann”) to the actual CHD (“y\_test”) which gives us an accuracy of 77%.

When looking at the “parameters” included within the “grid\_obj” function in Appendix 13 which are defined in Appendix 12, there are many changes we can make to edit our outcome, in brief detail we shall look at each parameter and how changing them affects our accuracy score. “Solver” relates to the “weight optimisation over the nodes” (Fuchs, 2021) and can be changed to multiple different values that will affect the accuracy. For example, “lbfgs” relates to ‘multiclass problems, and handles multinomial loss’; whereas “liblinear” can only be used for small datasets and is limited when handling larger datasets (Point, 2022). Another parameter to change is “alpha” which corresponds to a regularisation term that combats overfitting by adding constraints onto the size of weights and allowing a sharper, more precise way of managing data (scikit-learn, 2022c). We have our “alpha” set to 0.0001 to allow virtually no space between data plots. “hidden\_layer\_sizes” relates to the number of layers and nodes that the neural network has, for example “(9, 14, 14, 2)” denotes that there will be 9 nodes, and 14 neurons in both of the 2 layers (Verma, 2018). Random state sets the experiment to be repeatable and will constantly include the same parameters no matter how many times repeated, the number can be any number at all, but the scientific community commonly use 42 as it is in reference of ‘The Hitchhiker’s Guide to the Galaxy’, “the answer to the ultimate question of life, the universe and everything”. “max\_iter” is so set the number of iterations allowed in order to complete the algorithm (scikit-learn, 2022d). “early-stopping” is designed to stop overfitting and to ensure that model does not learn the training data to the extent it impacts negatively on the testing data, however, we have already determined the weights to the training and testing data so this can be set to false.

# Other Models

There are many, many methods of machine learning that can be used for prediction purposes with heart disease data, however as with all ways of doing things, there is no “right way”. There are many factors to consider when looking at what method to use, accuracy, precision, suitability, ethical issues, legal issues etc. Firstly, we will look at the different types of machine learning models and how they work, and then we shall delve into these factors.

## Support Vector Machine

Support Vector Machine (SVM) is used commonly for the detection of heart disease; however, it is still strained by its computationally complex (Li et al., 2020) system that requires constrained optimisation programming (Wang & Hu, 2005). SVM works by creating a divide between classes using a hyper-plane to differentiate between data classes. This is then used as a deterministic rule of thumb to decide under which class new data is instilled in (Ray, 2017). SVM is very useful when it comes to large datasets with information regarding classes and is particularly effective in high-dimensional space problems (Pouriyeh et. al., 2017), however training large data sets can be a strenuous activity.

## K-Nearest Neighbour

K-Nearest Neighbour (K-NN) is widely used due to it being simple to use, understand and applicable to real word applications (Zhang et. al., 2017). It classifies an object based on its surrounding objects and their classes (Pouriyeh et. al., 2017), although simple to understand, it struggles from numerous classification and regression issues concerning its nearest neighbour (Elhoseny, et al., 2021).

# Ethical and legal issues

When looking into using machine learning for medical purposes, it is important to look at the ethical, legal, and social issues that may arise from doing so. As de Miguel, Sanz, & Lazcoz (2020) explain that there are so many benefits that can come from using machine learning such as diagnosis, prognosis, and even appropriate treatment; however, different issues can also arise such as:

* Customer disapproval of using said machine learning for diagnosis, prognosis, or treatment
* Storing medical history with a system to store and use for future use
* Using an automated system for decision making that may impact the patient
* Physicians inability to understand/navigate devices used for machine learning

# Conclusion

Before deciding on a machine learning method, it is important to think about legal and ethical issues when deciding which method to use; while they are all great at obtaining accurate results, we must remember that these are people’s lives that will be tremendously impacted, and if the predicted results are incorrect then it leaves you open for a breach in a large amount of legal and ethical violations.

Regarding their accuracy, precision, and suitability, as we can see in Appendix 29-32, we are able to identify that in this case ANN has the highest accuracy meaning that out of all the total observations, approximately 77.4% were accurate, which when looking in the grand scheme of things isn’t great when looking at heart disease, that could mean if you were to test 100 people, 77 of those people will have obtained accurate results, but what about those other 23? They will either have been told they have heart disease when they don’t, or don’t when actually they do.

So, we move on to another way of analysing which is the better way to determine which machine learning method is better, looking at precision. Precision relates to how many of the predicted positives were actual positives (Davis, J. & Goadrich, M., 2006), and our highest percentage for that is for the KNN method which obtained 83.6%. Looking at it another way, if 100 people were tested, the KNN method would correctly predict the outcome for 84 of those people, 16 however would have an incorrect prediction.

And finally, we test recall, looking at our predicted positives against our total actual positives (Davis, J. & Goadrich, M., 2006) we can see that SVM had the highest percentage at 91.5. Which isn’t excellent when you think about it as that’s 8 out of 100 people who have heart disease being predicted as not having it; however, it is still a lot better than the 26 obtaining incorrect results for ANN.

At this point if you’re still unsure which machine learning algorithm to use, we can do our final test called an F-score, which is regarded as the average weight between recall and precision (Elhoseny, et al., 2021). Looking at Appendix 33 we can see that KNN has a final score of 85 making it the most reliable machine learning method when looking at both precision and recall percentages.

# Link to Source Code

https://github.com/KieranBest/AI-Assignment-2

# Link to Video Walk Through

<https://youtu.be/UerrN6Hedzs>

# References

Davis, J. & Goadrich, M. (2006) 'The relationship between Precision-Recall and ROC curves', *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233-240.

de Miguel, I., Sanz, B., & Lazcoz, G. (2020) 'Machine learning in the EU health care context: exploring the ethical, legal and social issues', *Information, Communication & Society, 23*(8), pp. 1139-1153.

Elhoseny, M., Mohammed, M., Mostafa, S., Abdulkareem, K., Maashi, M., Garcia-Zapirain, B., . . . Maashi, M. (2021) 'A new multi-agent feature wrapper machine learning approach for heart disease diagnosis', *Comput. Mater. Contin, 67*, pp.51-71.

Fuchs, M. (2021) *NN - Multi-layer Perceptron Classifier (MLPClassifier)*. Available at: https://michael-fuchs-python.netlify.app/2021/02/03/nn-multi-layer-perceptron-classifier-mlpclassifier/ (Accessed: 18/05/2022)

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018) *Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation*. *Available at:* https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=2202&context=cs\_techrep (Accessed: 18/05/2022)

Jain, J., & Sheth, A. (2022) *South African Heart Disease: Trees, Forests, Boosting*. *Available at:* https://www.kaggle.com/code/arihantsheth/south-africa-heart-disease-trees-forests-boosting/notebook (Accessed: 18/05/2022)

Li, J., Haq, A., Din, S., Khan, J., & Saboor, A. (2020) 'Heart disease identification method using machine learning classification in e-healthcare', *IEEE Access, 8*, pp.107562-107582.

Mantovani, R. (2015) *sa-heart*. *Available at:* https://www.openml.org/search?type=data&sort=runs&id=1498&status=active (Accessed: 09/04/2022)

Nair, A. (2019) A Beginner's Guide To Scikit-Learn's MLPClassifier. *Available at:* https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/ (Accessed: 20/04/2022)

Point, T. (2022) *Scikit Learn - Logistic Regression*. Available at: https://www.tutorialspoint.com/scikit\_learn/scikit\_learn\_logistic\_regression.htm (Accessed: 20/04/2022)

Pouriyeh, S., Vahid, S., Sannino, G., De Pietro, G., Arabnia, H., & Gutierrez, J. (2017) 'A comprehensive investigation and comparison of machine learning techniques in the domain of heart disease', *IEEE symposium on computers and communications (ISCC)*, pp.204-207.

Ray, S. (2017) *Understanding Support Vector Machine(SVM) algorithm from examples*. Available at: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ (Accessed: 18/05/2022)

scikit-learn (2022a) *sklearn.metrics.confusion\_matrix*. *Available at:* https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\_matrix.html (Accessed: 20/04/2022)

scikit-learn (2022b) *sklearn.metrics.make\_scorer*. *Available at:* https://scikit-learn.org/stable/modules/generated/sklearn.metrics.make\_scorer.html (Accessed: 20/04/2022)

scikit-learn (2022c) *Varying regularization in Multi-layer Perceptron*. *Available at:* https://scikit-learn.org/stable/auto\_examples/neural\_networks/plot\_mlp\_alpha.html (Accessed: 20/04/2022)

scikit-learn (2022d) *sklearn.metrics.logistic\_regression*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html (Accessed: 18/05/2022)

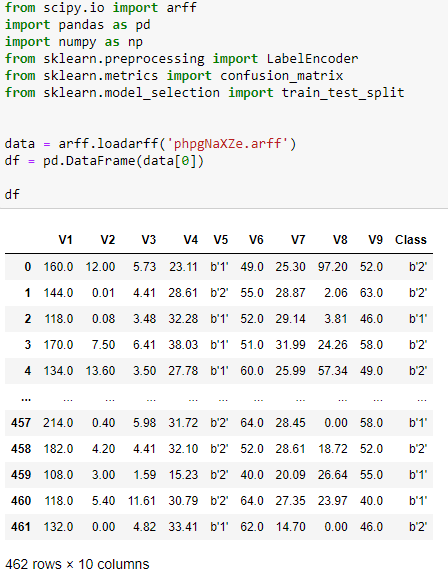
Verma, S. (2018) *heart disease prediction*. *Available at:* https://blog.goodaudience.com/heart-disease-prediction-aa656f2db585 (Accessed: 09/04/2022)

Wang, H., & Hu, D. (2005) 'Comparison of SVM and LS-SVM for Regression', *2005 International Conference on Neural Networks and Brain*, 1, pp. 279-283.

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017) 'Efficient kNN classification with different numbers of nearest neighbors', *IEEE transactions on neural networks and learning systems*, pp. 1774-1785.

# Appendix

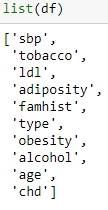
## Appendix 1



## Appendix 2



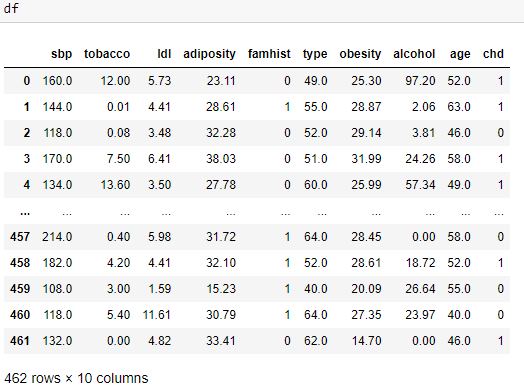
## Appendix 3



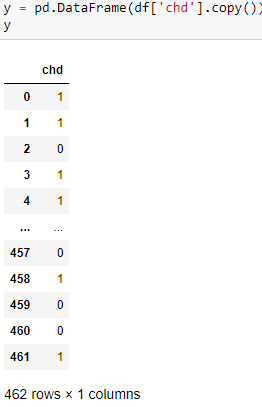
## Appendix 4



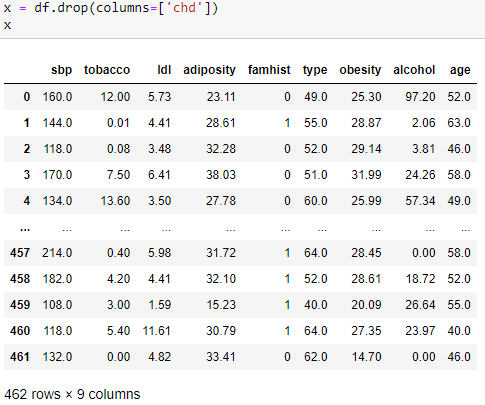
## Appendix 5



## Appendix 6



## Appendix 7



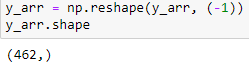
## Appendix 8



## Appendix 9



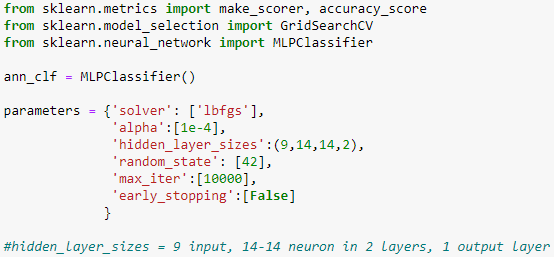
## Appendix 10



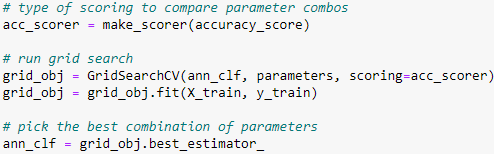
## Appendix 11



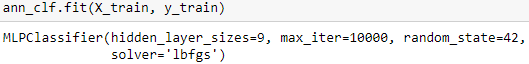
## Appendix 12



## Appendix 13



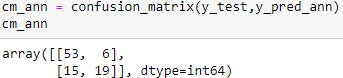
## Appendix 14



## Appendix 15



## Appendix 16



## Appendix 17



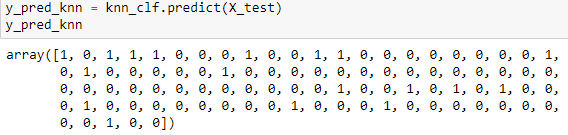
## Appendix 18



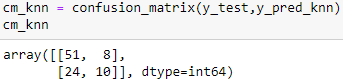
## Appendix 19



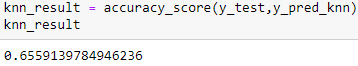
## Appendix 20



## Appendix 21



## Appendix 22



## Appendix 23



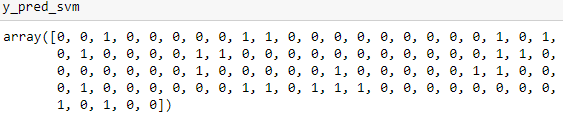
## Appendix 24



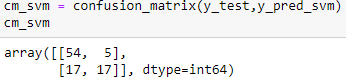
## Appendix 25



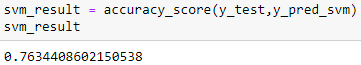
## Appendix 26



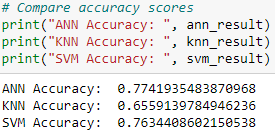
## Appendix 27



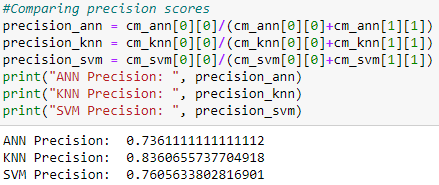
## Appendix 28



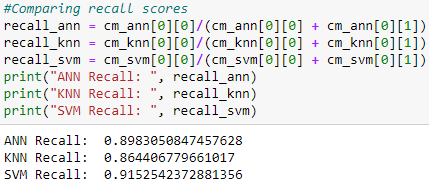
## Appendix 29



## Appendix 30



## Appendix 31



## Appendix 32



## Appendix 33

