Assignment 2

Artificial Neural Network

Kieran Best

Artificial Intelligence Computing

MCOMD2AIC

09/04/2022

Table of Contents

[Implementation 1](#_Toc101123961)

[Other Models 1](#_Toc101123962)

[Artificial Neural Network 1](#_Toc101123963)

[Support Vector Machine 1](#_Toc101123964)

[K-Nearest Neighbour 1](#_Toc101123965)

[Conclusion 1](#_Toc101123966)

[Link to Source Code 1](#_Toc101123967)

[References 2](#_Toc101123968)

[Appendices 2](#_Toc101123969)

[Appendix 1 2](#_Toc101123970)

[Appendix 2 2](#_Toc101123971)

[Appendix 3 2](#_Toc101123972)

[Appendix 4 3](#_Toc101123973)

[Appendix 5 3](#_Toc101123974)

[Appendix 6 3](#_Toc101123975)

[Appendix 7 4](#_Toc101123976)

[Appendix 8 4](#_Toc101123977)

[Appendix 9 4](#_Toc101123978)

[Appendix 10 4](#_Toc101123979)

[Appendix 11 4](#_Toc101123980)

[Appendix 12 4](#_Toc101123981)

[Appendix 13 5](#_Toc101123982)

[Appendix 14 5](#_Toc101123983)

[Appendix 15 5](#_Toc101123984)

[Appendix 16 5](#_Toc101123985)

[Appendix 17 5](#_Toc101123986)

[Appendix 18 5](#_Toc101123987)

[Appendix 19 5](#_Toc101123988)

[Appendix 20 5](#_Toc101123989)

[Appendix 21 6](#_Toc101123990)

[Appendix 22 6](#_Toc101123991)

[Appendix 23 6](#_Toc101123992)

[Appendix 24 6](#_Toc101123993)

[Appendix 25 6](#_Toc101123994)

[Appendix 26 6](#_Toc101123995)

[Appendix 27 6](#_Toc101123996)

[Appendix 28 6](#_Toc101123997)

[Appendix 29 7](#_Toc101123998)

# 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 and 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”). Therefore, leaving us with an accuracy of 77% (Appendix 17).

When looking at the parameters included within the “grid\_obj” function in Appendix 13, 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

## Conclusion

In conclusion, using an Artificial Neural Network allows us to obtain an accuracy of 77%, however this can be changed when altering the parameters.

# Other Models

There are many, many methods of machine learning that can be used for prediction purposes, 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, then we shall delve into these factors and then conclude with which model is more suitable and why.

## Support Vector Machine

## K-Nearest Neighbour

# they look into different parameters, even just the solver in ANN can change the results drastically.

## Conclusion

# Link to Source Code

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

# References

Elhoseny, M. e. (2021). A new multi-agent feature wrapper machine learning approach for heart disease diagnosis. *Comput. Mater. Contin, 67*, 51-71.

Fuchs, M. (2021). NN - Multi-layer Perceptron Classifier (MLPClassifier). Retrieved from https://michael-fuchs-python.netlify.app/2021/02/03/nn-multi-layer-perceptron-classifier-mlpclassifier/

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation. *Departmental Technical Reports*. Retrieved from https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=2202&context=cs\_techrep

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

Mantovani, R. (2015). sa-heart. *OpenML*. Retrieved from https://www.openml.org/search?type=data&sort=runs&id=1498&status=active

Nair, A. (2019). A Beginner's Guide To Scikit-Learn's MLPClassifier. *Developers Corner*. Retrieved from https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/

Point, T. (2022). Scikit Learn - Logistic Regression. Retrieved from https://www.tutorialspoint.com/scikit\_learn/scikit\_learn\_logistic\_regression.htm

Pouriyeh, S. e. (2017). A comprehensive investigation and comparison of machine learning techniques in the domain of heart disease. *IEEE symposium on computers and communications (ISCC)*, 204-207.

scikit-learn. (2022a). sklearn.metrics.confusion\_matrix. *scikit learn*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\_matrix.html

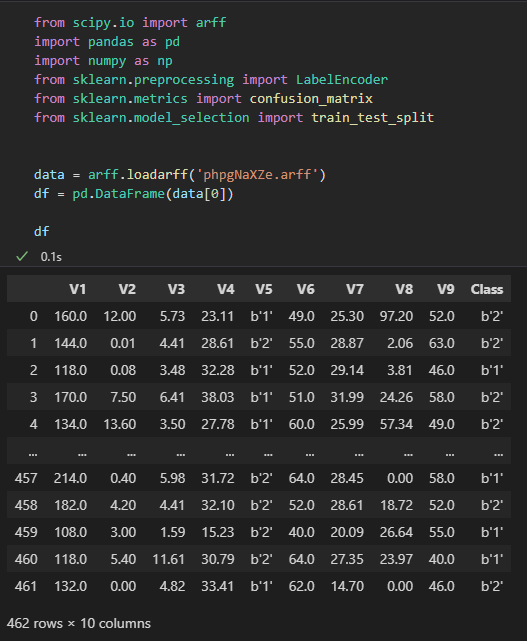
scikit-learn. (2022b). sklearn.metrics.make\_scorer. *scikit learn*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.make\_scorer.html

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

Verma, S. (2018). heart disease prediction. *Good Audience*. Retrieved from https://blog.goodaudience.com/heart-disease-prediction-aa656f2db585

# Appendices

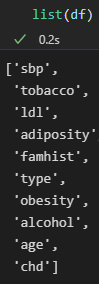
## Appendix 1



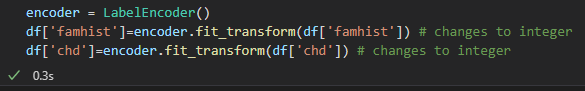
## Appendix 2



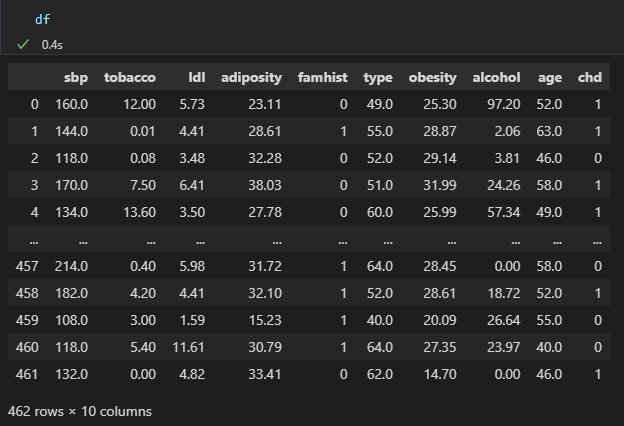
## Appendix 3



## Appendix 4



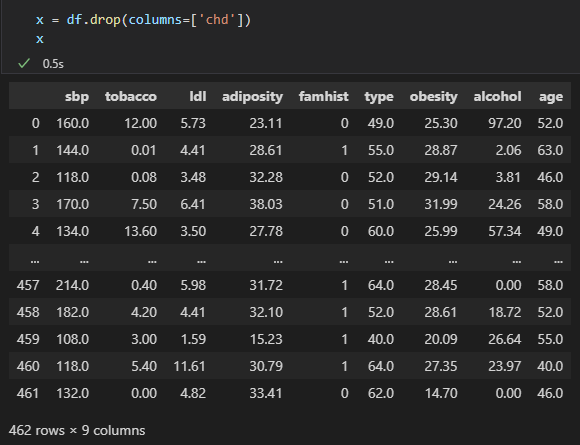
## Appendix 5



## Appendix 6



## Appendix 7



## Appendix 8



## Appendix 9



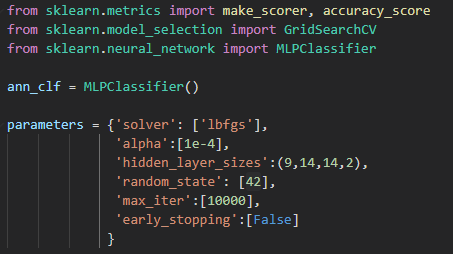
## Appendix 10



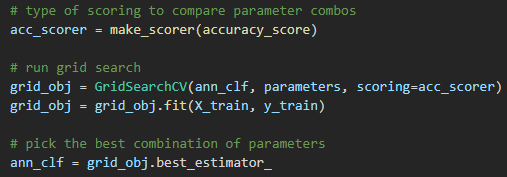
## Appendix 11



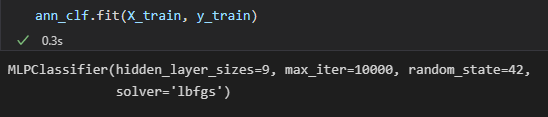
## Appendix 12



## Appendix 13



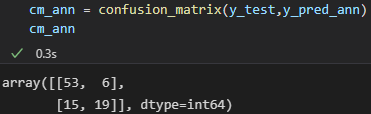
## Appendix 14



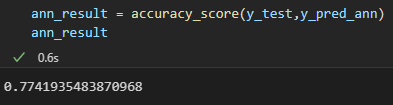
## Appendix 15



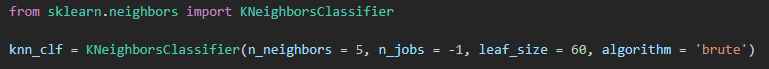
## Appendix 16



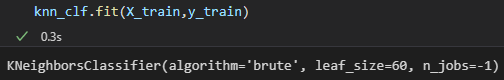
## Appendix 17



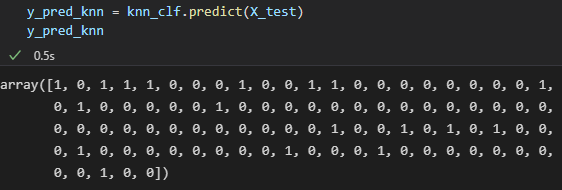
## Appendix 18



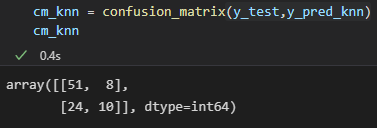
## Appendix 19



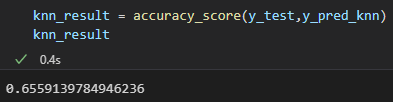
## Appendix 20



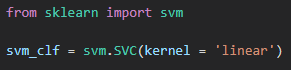
## Appendix 21



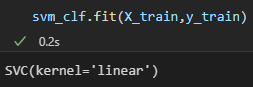
## Appendix 22



## Appendix 23



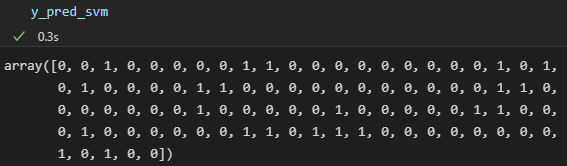
## Appendix 24



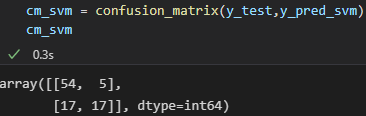
## Appendix 25



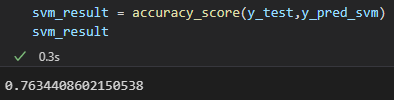
## Appendix 26



## Appendix 27



## Appendix 28



## Appendix 29

