**CMP3751M - Machine Learning**

**Assessment 2 - Written Report**

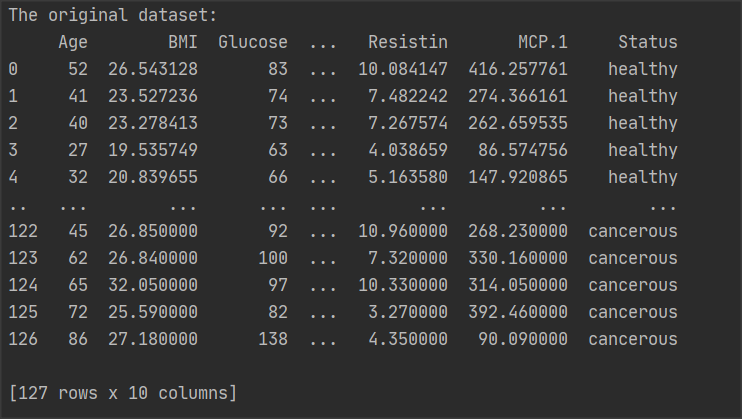
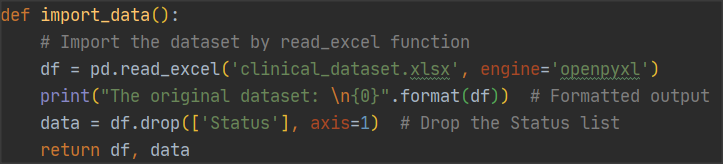
Shangyuan Liu

University of Lincoln, School of Computer Science

[25344136@students.lincoln.ac.uk](mailto:25344136@students.lincoln.ac.uk)

# Section I: **Data import, summary, pre-processing and visualisation**

## Load the Data



Function import\_data is used to load the data. The approach of the function is importing Pandas, which is a data analysis package for python providing functions and methods to manipulate data quickly and easily. And use read\_excel to an Excel file into a DataFrame, the original dataset size is 127 rows × 10 cols.

Figure 1: Function import\_data() - Load the dataset

## Summary of the Dataset

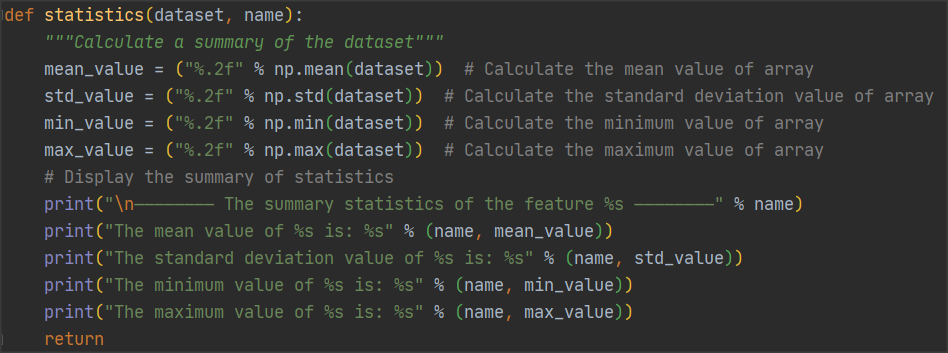
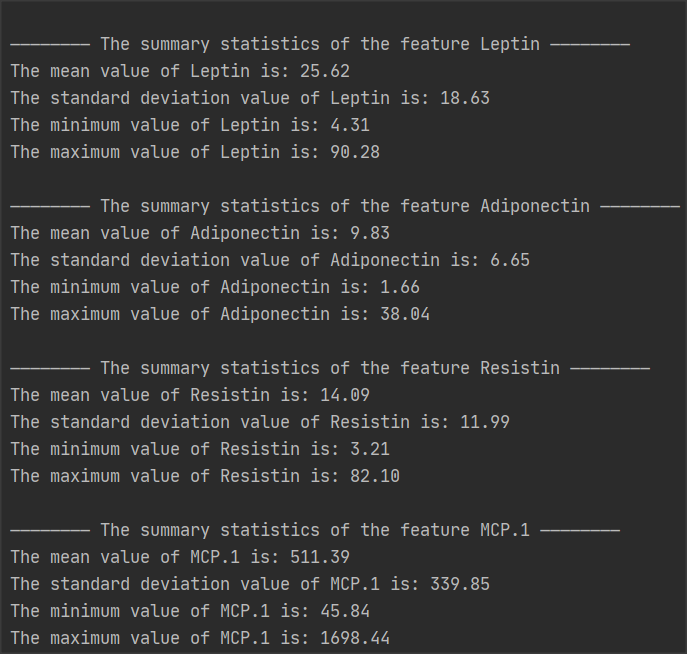
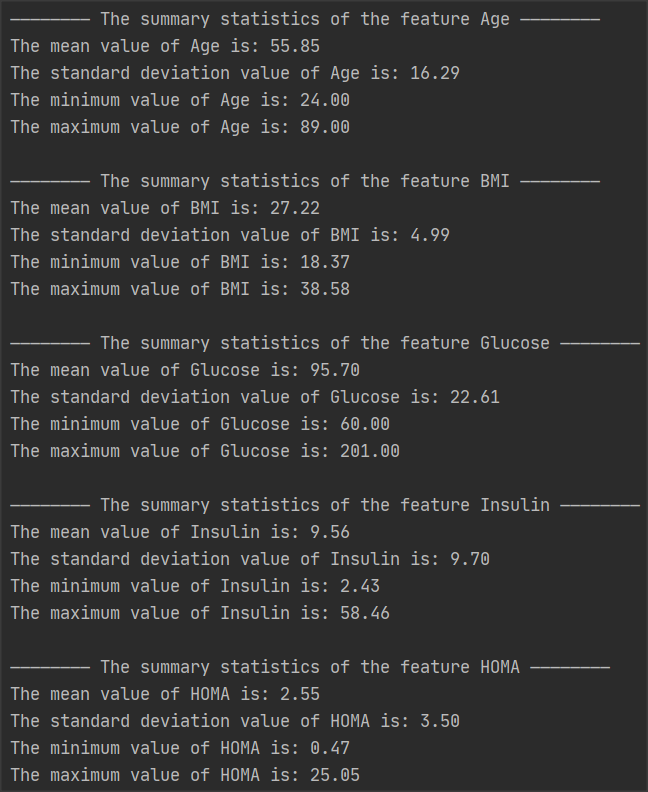
This function statistics() (figure 2) provides a summary (mean, standard deviations, min/max values) of each feature in the dataset. Using the mean(), max(), and min() methods in the numpy library in the function to provide a summary. And the result of the summary is shown in Figure 3.

Figure 2: Function statistics() - Statistical summary of data set

Figure 3: The result of statistical summary



**Report the Data Size and Features**

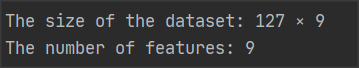
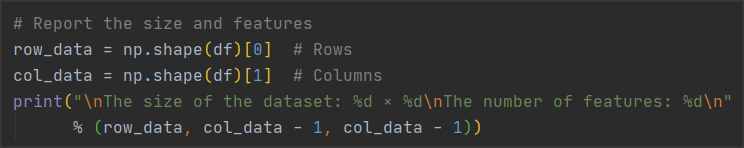
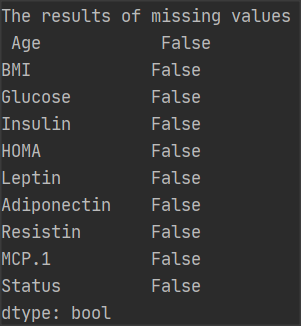


Figure 4: Report the size of the data and number of features

The np.shape method is used in this step, which is to report the dimension of a matrix or array. Shape()[0] is for counting the number of rows, and Shape()[1] is for counting the number of cols. As can be seen from figure 4 that the size of data is 127 rows × 9 cols, there are 9 clinical features that can classify patients as healthy or cancerous, and there is another feature as a prediction label.



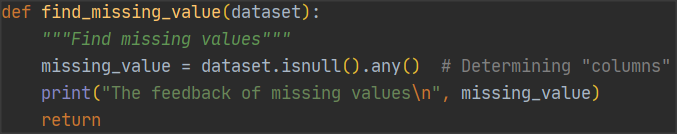
**Find Missing Values**

Figure 5: Function find\_missing\_value and feedback

In the pandas library, there is an effective function isnull() , which can be used to identify missing values. And isnull().any() will determine which columns contain missing values, and return True if there are missing values in the column, otherwise False. Therefore, as shown in the figure on the right, there is no missing value in this dataset.

**Find Categorical Variables**

There are any categorical variables in the dataset, the categorical variables are 'cancerous' and 'healthy' respectively in the 'Status' list It is clear from figure 6 that value\_counts() is a method to check how many different variables are in a column of the table, and then use the list method to store them in the list.

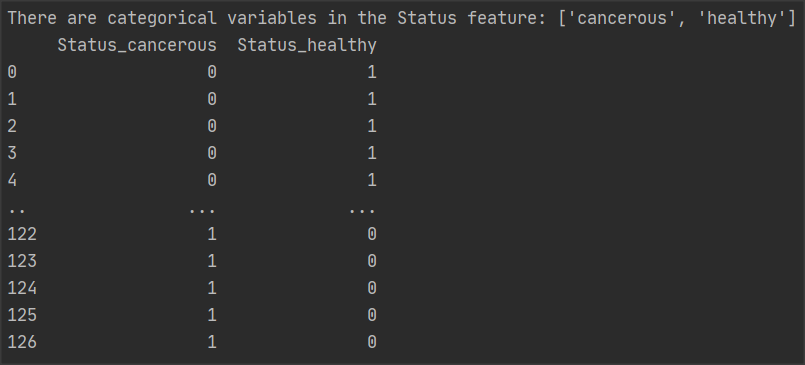
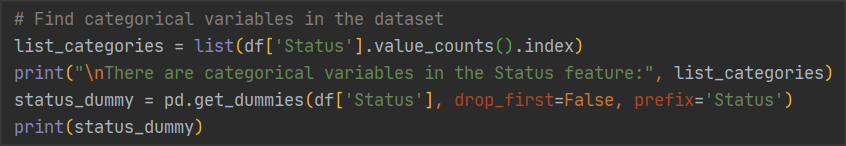


Figure 6: Find categorical variables

Finally, using function get\_dummies() for one-hot encoding to convert categorical variables into dummy/indicator variables, and the result of the change in the status column after the one-hot code is applied is as shown in figure 6.

**Data Normalisation**

Figure 7: Functions z\_score() and max\_min\_normalization() to data normalization

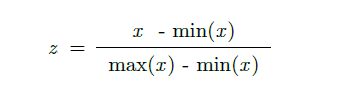
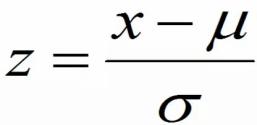
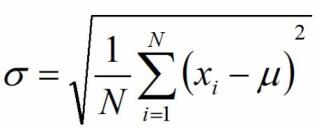
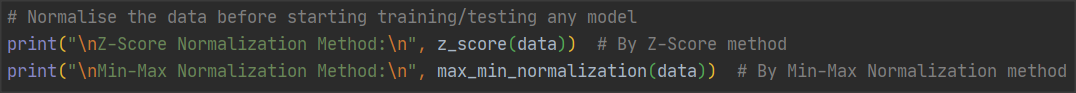
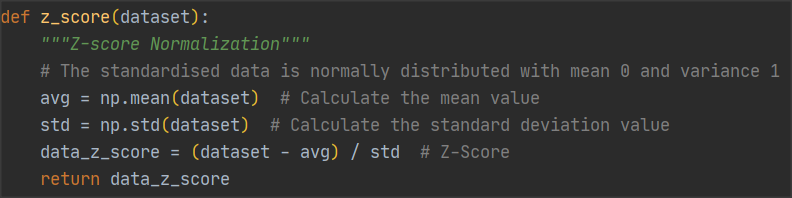
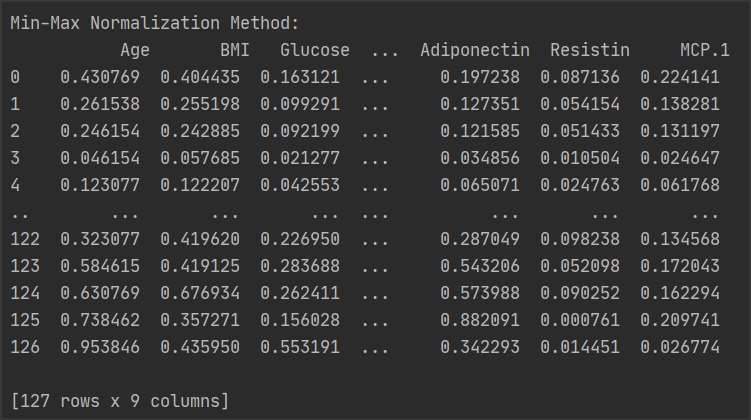
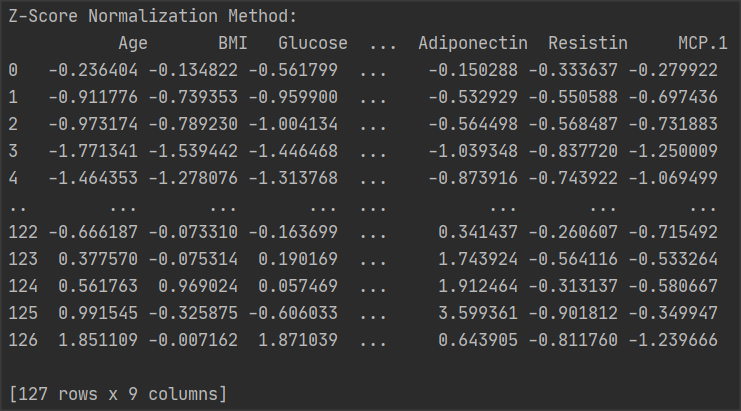


Figure 8: z-score and max\_min\_normalization formulas

In my opinion, it is really necessary to have a data normalisation processing for machine learning models. Sola and Sevilla (1997) indicated that “An adequate normalization, not only for the network output variables but also for the input ones, previous to the training process is very important to obtain good results and to reduce significantly calculation time”. Therefore, it is significant to normalize the data in order to overcome the problem of model learning. Moreover, the model ensures that a similar range of values is used for different features so that gradient descent can converge more quickly.

There are two approaches used for data normalization, which are Z-Score and [Min-Max normalisation respectively. The formulas are displayed in figure 8 and implemented in the functions z\_score and max\_min\_normalisation respectively (as shown in figure 7). The Min-Max normalisation is a linear transformation of the original data to map the values to between [0, 1]. However, the Z-Score is based on the mean and standard deviation of the original data to standardise the data, the main purpose is to standardise data of different magnitudes to the same magnitude, using the calculated Z-Score value to ensure comparability between data.](https://www.codecademy.com/articles/normalization) The results of the data normalisation are displayed in figure 9.

Figure 9: The result of two methods for data normalisation



**Data Visualisation**

Seaborn is used in this function (shown in figure 10), which is a Python data visualization library based on Matplotlib. The boxplot and density plot are grouped by the categorical variables of ‘Status’, and these plots can be seen in the figure 11.

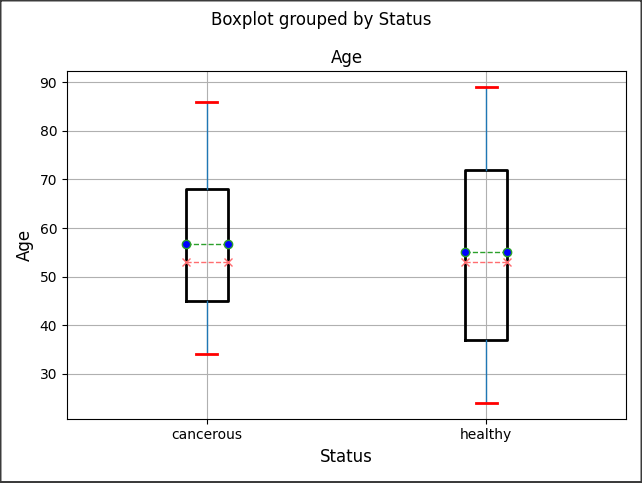
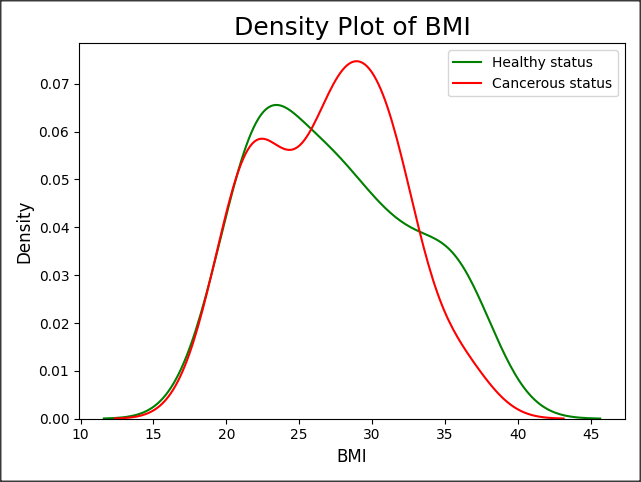
The boxplot provides significant information about the location and dispersion of the data, especially Figure 10: Function box\_density\_plots when comparing different parent data. As can be seen in figure 11 in the x-axis contains the two classes (Cancerous and Healthy). Each boxplot contains six main data nodes. The Age data are arranged from largest to smallest and its upper edge, upper quartile, mean value, median, lower quartile, and lower edge are calculated.



Figure 10: box\_density\_plots function

The density diagram reflects the proportion of cancerous and healthy patients in the different BMI ranges. As can be seen from the graph below, the green curve represents Healthy Status and the red curve represents Cancerous Status, with the highest proportion of healthy patients in the BMI range of 20 to 25 and the highest proportion of cancerous patients between 25 and 33, which accounts for the highest proportion is over 7%.

Figure 11: The boxplot and density plot

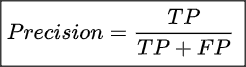
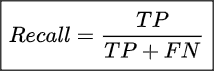
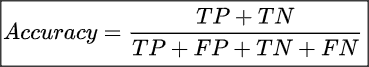


**Section II: Discussion on Selecting an Algorithm**

I disagree with this view, the classification models are only as good as the scenario in which it is applied. Although in most research scenarios high precision is pursued, in the field of disease diagnosis a test set achieving 90% precision is not a good enough measure of how well a model works in a scenario. I will provide a detailed analysis and explanation of why I disagree with this view in the following points.

1. In terms of the dataset split ratio, 70% of the data as training set, 10% as test set, and there are 20% as a validation set, however, it is not known what the approach to model validation is. If the trainee uses Holdout validation, which is not a cross-validation method, there is an obvious disadvantage for Holdout that is sensitive to the proportion of the training set and test set split, that is, the final evaluation index calculated on the validation set is highly dependent on the original grouping. Therefore, the method of cross-validation can avoid overfitting effectively.
2. In model evaluation, Accuracy (as seen in figure 12) is the proportion of the correct data (TP+TN) to the total data (TP+TN+FP+FN). However, accuracy is not the only best measure for assessing classification models. Ganesan (2014) indicated that “accuracy alone is sometimes quite misleading as you may have a model with relatively ‘high’ accuracy with the model predicting the ‘not so important’ class labels fairly accurately but the model may be making all sorts of mistakes on the classes that are actually critical to the application”.

Figure 12: Accuracy, Recall and Precision formula



1. Especially in the disease diagnosis scenario, false positives can be tolerated for classification results, but false negatives must be reduced that means would rather misdiagnose a healthy patient, also don't miss any genuine confirmed cancerous patient. Therefore, sensitivity, accuracy, and other parameters should also be used as criteria for evaluating models. Recall (Sensitivity) represents the percentage of the population that actually has the disease that is tested "positive", also known as the true positive rate. And precision means how many of all predicted positives are correctly classified as positive. The closer these values (especially the sensitivity) to 1, the better the classifier model, As Hakama et al. (2007) said that “sensitivity is the indicator on the ability of screening to find cancer in the detectable preclinical phase (DPCP). The ability is usually specified as to the screening test”.



Figure 13: F1-Score formula

Therefore, for this type of cancer patient prediction model, we need to rely on parameters and criteria to analyse and evaluate the model in order to select the best algorithm. In addition to accuracy, precision, sensitivity, and specificity, the F1-Score can also be used for evaluation (Shown in figure 13). F1-Score is viewed as a superior indicator of the classifier’s performance than the conventional accuracy measure. The most desirable value for the F1-score is to close to 1, by having both precision and recall have high values. A value of 1 for both means that the algorithm has the best accuracy.

Moreover, there is another measure that can be used to estimate the classifier's performance that is the receiver operating characteristic curve (ROC) and the area under the curve ( AUC). The ROC space is defined by the FPR and TPR as the x and y axes respectively, which describe the relative trade-off between true positives (TP) and false positives (FP). TPR represents the percentage of all samples that are actually positive that is correctly judged to be positive, and FPR is the percentage of all samples that are actually negative that are incorrectly judged to be positive. As a consequence, “the quantitative metrics of accuracy and AUC are used for assessing the overall performance of a classifier. Specifically, accuracy is a measure related to the total number of correct predictions. On the contrary, AUC is a measure of the model's performance which is based on the ROC curve that plots the tradeoffs between sensitivity and 1-specificity (figure 14)” (Kourou et al., 2015).

When comparing different classification models, the ROC curve for each model can be plotted, and the area under the curve can be calculated and compared, as an indicator to judge the pros and cons of the model.

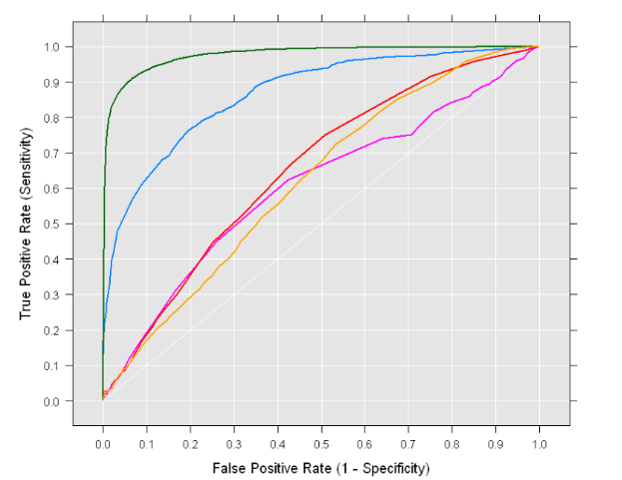
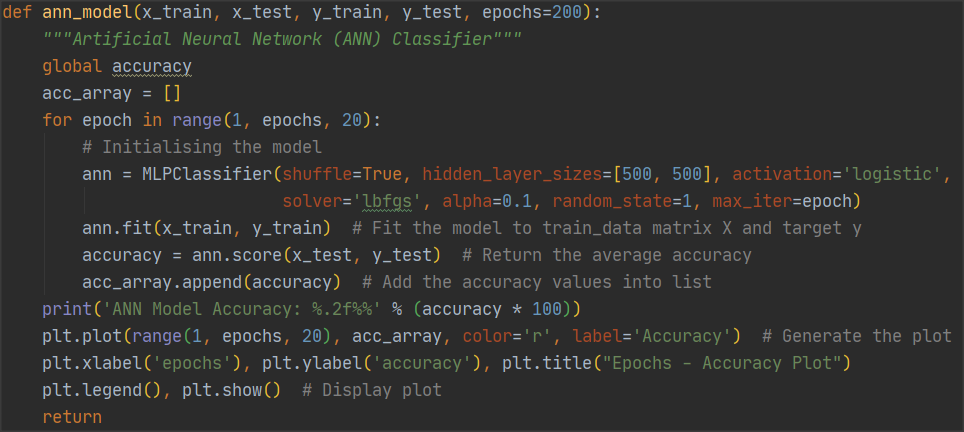
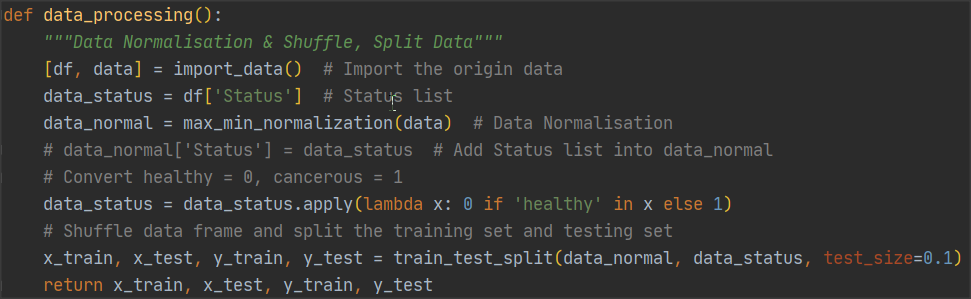
****

Figure 14: ROC Space and AUC

**Section III: Designing Algorithms**

***Artificial Neural Network (ANN)***

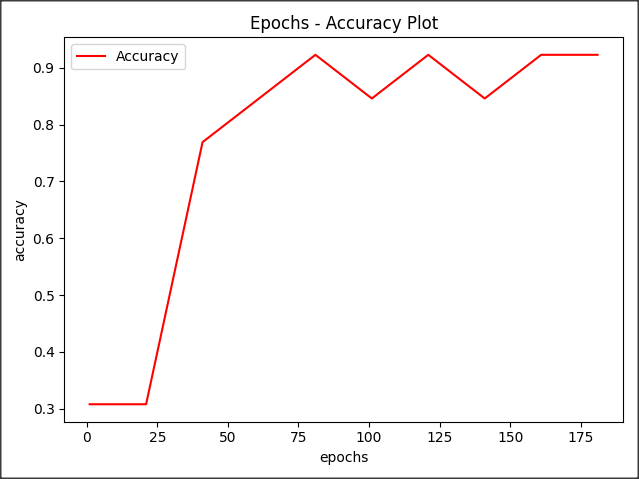
Figure 15: Functions data\_processing() and ann\_model()



Before design an artificial neural network (ANN) classifier, need to use function ‘data\_processing()’ to normalize the data set, and turn the elements in "Status" to 0 or 1. Then Shuffle the data set and split 90% of the data as the training set and 10 % as the testing set. In this function, I used the customized function ‘max\_min\_normalization()’ to achieve data normalisation. For data split processing, train\_test\_split() is a function to randomly divide the training set and test set provided in the model\_selection module of the sklearn package, which was used in this function and output the data and labels of the training set and test set separately.

The purpose of function ann\_model is building an ANN classify mode and using training data to train this model, in the end, calculate the accuracy of the test set through the trained model. For initialising the model, as shown in figure 15, function MLPClassifier is a multi-layer Perceptron classifier. Setting the number of neurons in two hidden layers to 500, and use the logistic function as the non-linear activation function for the hidden layers. Moreover, this model optimizes the log-loss function using lbfgs (an optimizer in the family of quasi-Newton methods), in terms of the small dataset, it can converge faster and perform better. The maximum number of iterations (max\_iter) is equal to epochs. [Next](D:/Dict/8.9.6.0/resultui/html/index.html" \l "/javascript:;) [step](D:/Dict/8.9.6.0/resultui/html/index.html" \l "/javascript:;) is fitting the model to training matrix data and target label. 最终的准确率为。。。。

And set the number of epochs is 200 and the step size is 20, in order to calculate the corresponding accuracy according to different epochs by for loop, and store the multiple accuracies in a list. In the end, plot using the number of epochs in the ‘x’ axis and the accuracy in ‘y’ axis (fig. 16).



通过。。。。观察

Figure 16: The result accuracy and plot

***Random Forests Classifier***

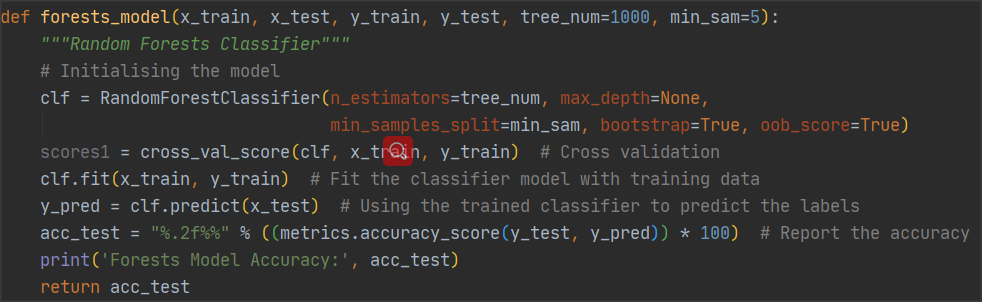


Figure 17: Function forests\_model() to build random forests and return the accuracy

Random Forest is a classification algorithm that uses a bootstrap resampling technique to repeatedly draw n samples at random from the original training sample set N to generate a new set of training samples for training decision trees, and then follows the above steps to generate m decision trees to form a random forest, with the classification result of the new data determined by the number of votes formed by the classification tree. In essence, it is a modification of the decision tree algorithm, where multiple decision trees are combined together, with the creation of each tree dependent on independently drawn samples.

In the function forests\_model (fig. 17), I used the module RandomForestClassifier from the sklearn package, which is a random forest is a meta-estimate that fits many decision tree classifiers on individual subsamples of a dataset and uses averages to improve prediction accuracy and control overfitting. In the module RandomForestClassifier, there are several parameters that need to be set. ‘n\_estimators’ stands for the number of trees in the forest, usually n\_estimators is too small that is easy to underfitting and too large for trees number, more trees will give the better model performance, therefore, this parameter is set to 1000. And min\_samples\_split indicates the minimum number of samples needed to split internal nodes, set to {5 & 50}. After model initialization, using the fit method to build a forest of trees from the training set, and then predict the test label by testing class x. Once the algorithm model has been built, it needs to be evaluated to determine its merit. Generally, a training set is used to build the model and a test set is used to evaluate the model. Therefore, return the mean accuracy on the given original test data and predicted labels.

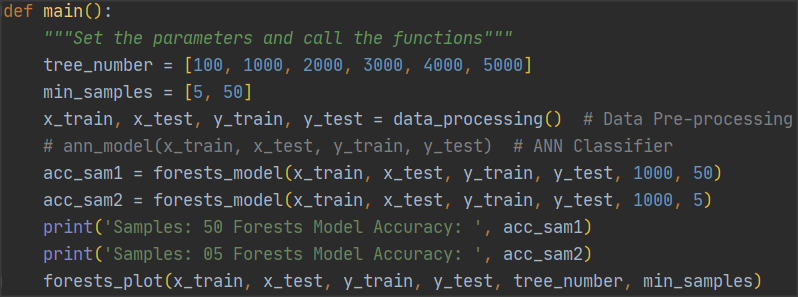
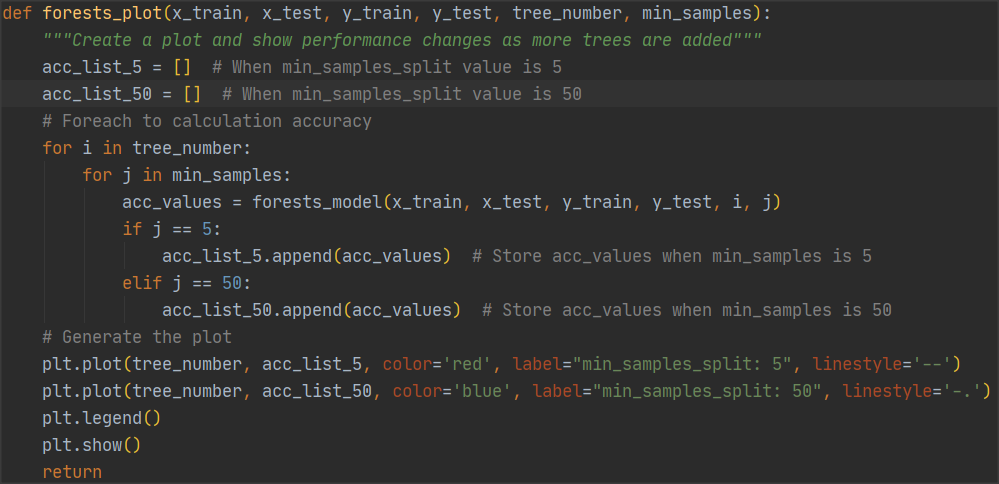


Figure 18: Function forests\_plot() to create the plot

Function forests\_plot is used for create a plot and show performance changes as more trees are added. First, set up two arrays to hold the accuracy, then iterate through the number of samples and number of trees arrays set up in the main function via a for loop (figure 18). The values inside the arrays are then assigned to the forests\_model function called to return the accuracy rates, and the accuracy rates are stored in the arrays. In the end, generate plots from accuracy and number of trees. (Figure 19)

根据对这个图的观察，当树的数量越大，则

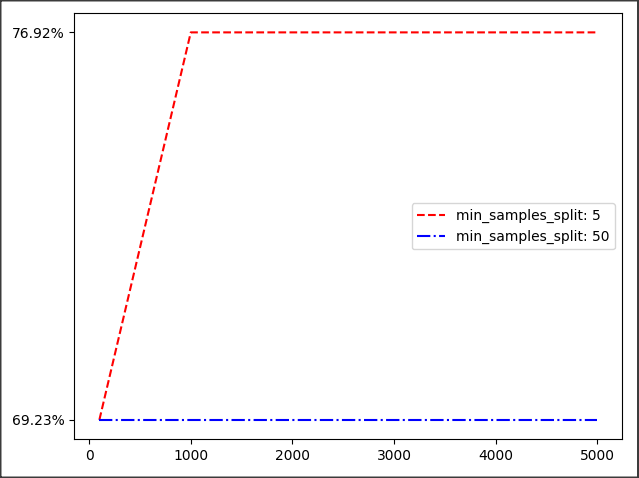


Figure 19: Function forests\_plot() to create the plot

**Reference List**

Ganesan, K. (2014). *How to compute precision and recall for a multi-class classification problem*. [online] Kavita Ganesan, Ph.D. Available at: https://kavita-ganesan.com/how-to-compute-precision-and-recall-for-a-multi-class-classification-problem/#.YBa2pNj7SUl [Accessed 31 Jan. 2021].

Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V. and Fotiadis, D.I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, [online] 13, pp.8–17. Available at: https://www.sciencedirect.com/science/article/pii/S2001037014000464 [Accessed 30 Jan. 2021].

Hakama, M., Auvinen, A., Day, N.E. and Miller, A.B. (2007). Sensitivity in cancer screening. *Journal of Medical Screening*, 14(4), pp.174–177.

Sola, J. and Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3), pp.1464–1468.