**Question 3)**

Explanations:

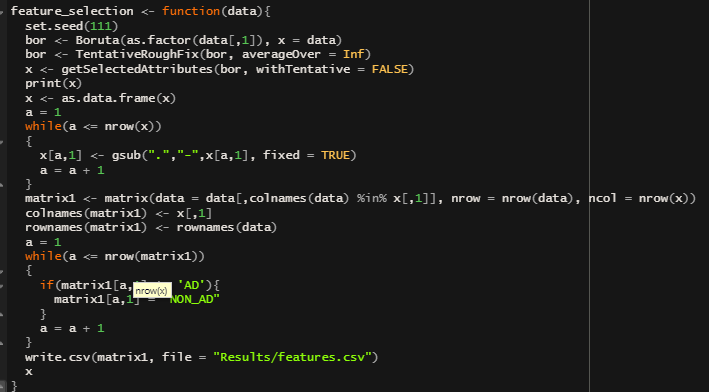
The Boruta package was used to select the features that have the largest impact on the determining the class result. Boruta is a wrapper function that applies the random forest algorithm to the dataset its given, requiring specification of the evalutated attributes and the attribute that is determined by the others, and returns the important attributes in an boruta object. The tentative rough fix function is another method of the Boruta package that tries to identify if features on the edge of important or un important should be kept or not. The getSelectedAttributes function is used to get the names of the features selected in a vector, which is then used to loop through the dataset and remove all other proteins. The CLASS column was looped through replacing any values that were not “AD” with “NON\_AD”, the matrix was then written to a file for Weka to use.  
For Classfication the Weka software was used to generate a J48 pruned tree using the features selected. The collapseTree setting within the classifier in Weka was set to true as this ensured that unneccesary checks were not being formed. Other values were mainly set to default as they didn’t have a significant impact on the classifiers perforamance.

**a)**

The following proteins were chosen as the most important for determining the outcome of the CLASS value.

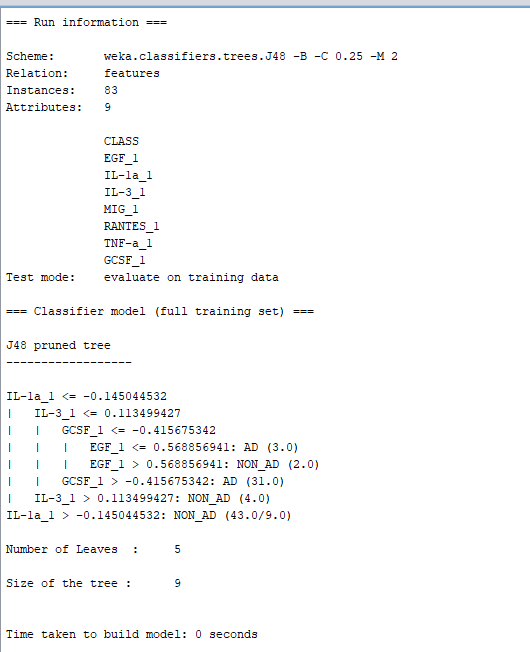
****

Code**:**

****

**b)**

Weka outputs the classification system below, with the pruned tree being the actual system that will be used to test on the different datasets.

****

**c)**

Weka outputs with the classification system.

Training Set Results:

Table

Description automatically generated

Test Set AD Results:

Table

Description automatically generated with low confidence

Test Set MCI Results:

A picture containing table

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training | Test AD | Test MCI |
| Sensitivity | 0.791 | 0.81 | 0.455 |
| Specificity | 1.0 | 0.96 | 0.68 |
| Accuracy | 89.16% | 89.13% | 57.45% |
| F1-Score | 0.891 | 0.89 | 0.569 |
| MCC | 0.803 | 0.785 | 0.138 |
| Youden’s J | 0.791 | 0.77 | 0.135 |

The classifier algorithm showed promise based on its performance with the first Test set with strong scores all around, specifically in its specificity, however the second test set performed very poorly using the same classifier system, with its accuracy only being slightly better then a coin flip. These results could indicate a flaw in the classifier system however the MCI test set also has the lowest sample size and could have been a statistical outlier.