COMP3009 Machine Learning

Assignment 2: Report

November 25, 2021

Group 18

Name	Student ID	Username
Teo Shi Bin	20183717	hcyst2
How Khai Chuin	20210654	hcykh1
Loh Qian Kai	20194664	hcyql1
Mohamad Arif Bin Mohamed Abu Baker	20116042	hfymm5
Muhammad Fikri Bin Mohd Roslee	20116065	hfymm4

Contents

Introduction

In this coursework, we are using decision tree to do both classification and regression. The data sets we used are both from previous coursework.

Data Set for Classification

This data set consists of 12 features and 1 column of class labels with a total of 299 instances within the data set. The goal is to classify the death event based on the 12 features.

Data Set for Regression

In this data set, 8 features and a total of 1030 instances were consisted. The goal is to predict the Concrete Compressive Strength based on the 8 features.

Implementations

Folder Structure

```
.
└── cw2/
    ├── datasets/
    ├── functions/
    │   ├── data_processing/
    │   ├── decision_tree/
    │   ├── metrics/
    └── demo.m
    └── cross_validation_classification.m
    └── cross_validation_regression.m
```

`demo.m` show both tree are working for the datasets.
`cross_validation_classification.m` show the evaluation of decision tree on the classification task.
`cross_validation_regression.m` show the evalutation of decision tree on the regression task.
`data_processing` for data validation split and normalisation
`decision_tree` for all decision tree related functions
`metrics` for accuracy, f1, confusion matrix etc.

Decision Tree for Classification

Figure below denoting the acquired decision tree for classification dataset, constructed using the best tuned depth of 2.

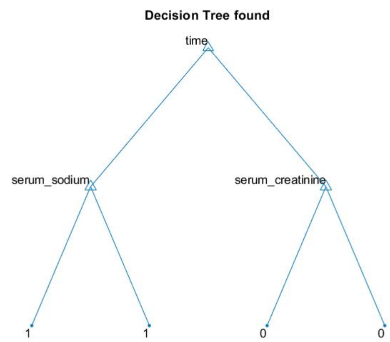


Figure 1: Classification Decision Tree

Cross-validation classification results

The result below consists of F1 score, recall and precision shown below is acquired from performing cross validation.

```
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.755556 BestD: 9
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.833333 F1Score: 0.909091 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.814815 F1Score: 0.608696 BestD: 6
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.756757 BestD: 4
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.792453 F1Score: 0.782609 BestD: 2
Outer: 1 TestSize: 30 TrainSize: 269 Accuracy: 0.733333 F1Score: 0.500000 Precision: 0.571429 Recall: 0.444444 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.722222 F1Score: 0.600000 BestD: 2
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.909091 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.833333 F1Score: 0.636364 BestD: 7
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.705882 BestD: 4
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.830189 F1Score: 0.782609 BestD: 2
Outer: 2 TestSize: 30 TrainSize: 269 Accuracy: 0.733333 F1Score: 0.714286 Precision: 0.909091 Recall: 0.588235 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.722222 F1Score: 0.680851 BestD: 6
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.796296 F1Score: 0.838710 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.600000 BestD: 2
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.796296 F1Score: 0.685714 BestD: 5
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.754717 F1Score: 0.782609 BestD: 2
Outer: 3 TestSize: 30 TrainSize: 269 Accuracy: 0.866667 F1Score: 0.818182 Precision: 1.000000 Recall: 0.692308 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.717949 BestD: 7
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.796296 F1Score: 0.800000 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.851852 F1Score: 0.733333 BestD: 4
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.777778 F1Score: 0.702703 BestD: 5
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.698113 F1Score: 0.782609 BestD: 2
Outer: 4 TestSize: 30 TrainSize: 269 Accuracy: 0.966667 F1Score: 0.923077 Precision: 1.000000 Recall: 0.857143 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.666667 F1Score: 0.652174 BestD: 7
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.685185 F1Score: 0.812500 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.796296 F1Score: 0.645161 BestD: 8
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.722222 F1Score: 0.648649 BestD: 3
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.754717 F1Score: 0.782609 BestD: 2
```

We noticed that despite the tree can fit perfectly to the training set, the model doesn't perform well in validation set within the inner cv, which indicates that it is overfitting in some cases. In general, higher depth can make more decisions but it will more susceptible to overfitting. Result shows that time is the only important feature, others may not be as important because pruning the tree will lead to an equal prediction of 89% accuracy (Figure 6).

```
Outer: 5 TestSize: 30 TrainSize: 269 Accuracy: 0.900000 F1Score: 0.842105 Precision: 0.888889 Recall: 0.800000 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.685185 F1Score: 0.631579 BestD: 2
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.740741 F1Score: 0.812500 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.796296 F1Score: 0.800000 BestD: 2
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.685185 F1Score: 0.648649 BestD: 3
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.849057 F1Score: 0.782609 BestD: 2
Outer: 6 TestSize: 30 TrainSize: 269 Accuracy: 0.800000 F1Score: 0.400000 Precision: 0.666667 Recall: 0.285714 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.740741 F1Score: 0.682927 BestD: 3
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.722222 F1Score: 0.812500 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.907407 F1Score: 0.857143 BestD: 6
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.722222 F1Score: 0.615385 BestD: 3
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.830189 F1Score: 0.782609 BestD: 2
Outer: 7 TestSize: 30 TrainSize: 269 Accuracy: 0.800000 F1Score: 0.700000 Precision: 0.777778 Recall: 0.636364 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.740741 F1Score: 0.684211 BestD: 6
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.833333 F1Score: 0.812500 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.759259 F1Score: 0.787879 BestD: 2
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.796296 F1Score: 0.764706 BestD: 3
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.811321 F1Score: 0.787879 BestD: 5
Outer: 8 TestSize: 30 TrainSize: 269 Accuracy: 0.766667 F1Score: 0.461538 Precision: 0.500000 Recall: 0.428571 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 215 Accuracy: 0.666667 F1Score: 0.700000 BestD: 5
     Inner: 2 TestSize: 54 TrainSize: 215 Accuracy: 0.703704 F1Score: 0.812500 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 215 Accuracy: 0.833333 F1Score: 0.823529 BestD: 2
     Inner: 4 TestSize: 54 TrainSize: 215 Accuracy: 0.740741 F1Score: 0.714286 BestD: 2
     Inner: 5 TestSize: 53 TrainSize: 216 Accuracy: 0.773585 F1Score: 0.600000 BestD: 2
Outer: 9 TestSize: 30 TrainSize: 269 Accuracy: 0.866667 F1Score: 0.750000 Precision: 0.857143 Recall: 0.666667 MostDepth: 2
     Inner: 1 TestSize: 54 TrainSize: 216 Accuracy: 0.722222 F1Score: 0.697674 BestD: 5
     Inner: 2 TestSize: 54 TrainSize: 216 Accuracy: 0.759259 F1Score: 0.812500 BestD: 2
     Inner: 3 TestSize: 54 TrainSize: 216 Accuracy: 0.796296 F1Score: 0.809524 BestD: 2
     Inner: 4 TestSize: 54 TrainSize: 216 Accuracy: 0.722222 F1Score: 0.700000 BestD: 3
     Inner: 5 TestSize: 54 TrainSize: 216 Accuracy: 0.777778 F1Score: 0.666667 BestD: 5


Outer: 10 TestSize: 29 TrainSize: 270 Accuracy: 0.931034 F1Score: 0.800000 Precision: 1.000000 Recall: 0.666667 MostDepth: 2
FinalAccuracy: 0.836437 FinalF1: 0.690919 Finald: 2
```

Figure 2: Result of the classification cross validation

Decision Tree for Regression

Figure below shown the acquired decision tree for regression generalized with the best tuned hyperparameter of depth 10.

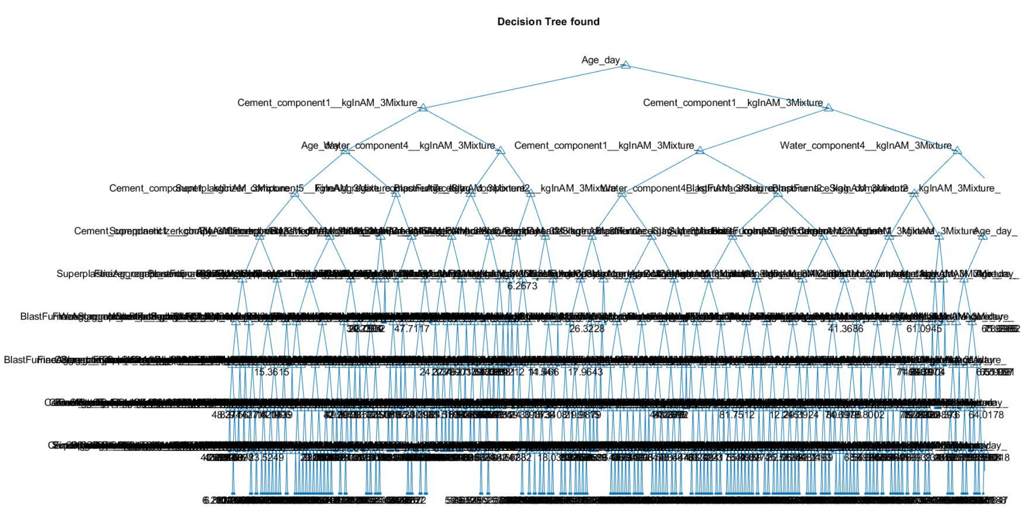


Figure 3: Regression Decision Tree

The decision tree for regression is a lot larger, not only because that this regression data set have more instances, but it is mainly due to the nature of regression task as it has infinite range of values to fit. The number nodes can grow exponentially as the number of features and data set grow. The computation will be very expensive.

Regression decision tree require a lot of node splitting for it to reach a real number. If the true regression line is not perpendicular to the axis, it will require a lot of node splitting to fit the true regression line. Worst case happened when the regression line decision tree trying to fit is a complicated polynomial line. For these 2 reasons, decision tree may not be so ideal for complicated regression task.

Cross-validation regression results

The below result which consist of RMSE shown below is acquired from performing cross validation.

```
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 5.836764 BestD: 14
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.324395 BestD: 17
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.335310 BestD: 18
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 6.135024 BestD: 15
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 6.389585 BestD: 16
Outer: 1 TestSize: 103 TrainSize: 927 RMSE: 5.771021 MostDepth: 14
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 6.211833 BestD: 9
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.145389 BestD: 11
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.649158 BestD: 13
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 6.219018 BestD: 18
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 6.915477 BestD: 11
Outer: 2 TestSize: 103 TrainSize: 927 RMSE: 5.204744 MostDepth: 11
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 6.004376 BestD: 10
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.214803 BestD: 16
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 7.870636 BestD: 12
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 6.549546 BestD: 9
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 7.846672 BestD: 8
Outer: 3 TestSize: 103 TrainSize: 927 RMSE: 5.678429 MostDepth: 9
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 7.015569 BestD: 12
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.229615 BestD: 17
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.360509 BestD: 11
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 5.926373 BestD: 19
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 7.551650 BestD: 10
Outer: 4 TestSize: 103 TrainSize: 927 RMSE: 5.023550 MostDepth: 11
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 6.847550 BestD: 10
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.021640 BestD: 12
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.596329 BestD: 12
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 6.225894 BestD: 12
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 7.239258 BestD: 11
Outer: 5 TestSize: 103 TrainSize: 927 RMSE: 5.453350 MostDepth: 12
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 6.421334 BestD: 7
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 7.282734 BestD: 7
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 5.963037 BestD: 12
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 6.127547 BestD: 10
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 7.403905 BestD: 11
Outer: 6 TestSize: 103 TrainSize: 927 RMSE: 5.324208 MostDepth: 12
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 6.175456 BestD: 10
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.143625 BestD: 10
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 5.915851 BestD: 14
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 7.319395 BestD: 17
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 6.390718 BestD: 10
Outer: 7 TestSize: 103 TrainSize: 927 RMSE: 5.507387 MostDepth: 10
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 7.508725 BestD: 10
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 6.681730 BestD: 7
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.109656 BestD: 16
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 7.539876 BestD: 13
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 6.263431 BestD: 11
Outer: 8 TestSize: 103 TrainSize: 927 RMSE: 6.275347 MostDepth: 10
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 6.607116 BestD: 9
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 5.410458 BestD: 20
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.279138 BestD: 15
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 5.963944 BestD: 16
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 6.668170 BestD: 10
Outer: 9 TestSize: 103 TrainSize: 927 RMSE: 6.556206 MostDepth: 10
     Inner: 1 TestSize: 185 TrainSize: 742 RMSE: 7.660379 BestD: 9
     Inner: 2 TestSize: 185 TrainSize: 742 RMSE: 7.319591 BestD: 13
     Inner: 3 TestSize: 185 TrainSize: 742 RMSE: 6.652584 BestD: 16
     Inner: 4 TestSize: 185 TrainSize: 742 RMSE: 7.187244 BestD: 13
     Inner: 5 TestSize: 187 TrainSize: 740 RMSE: 6.950056 BestD: 6
Outer: 10 TestSize: 103 TrainSize: 927 RMSE: 6.035104 MostDepth: 10
FinalRMSE: 5.682935 FinalD: 10
```

Figure 5: Result of the regression cross validation

Despite that, this decision tree still have impressive result with RMSE around 6, which is better than the SVM with linear kernel. SVM with polynomial kernel and gaussian kernel still yields better results with RMSE around 3 to 5.

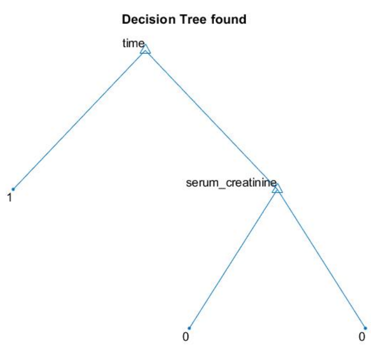
Additional Questions

1. **Pruning is an important issue in trees. Explain what pruning does and find the node(s) that would be pruned first for one of your learned trees. Explain the difference between the original and the pruned tree.**

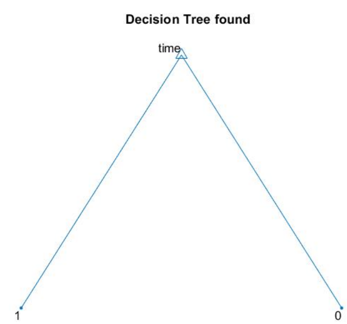
   Answer:

   Pruning is a method that is used to reduce the size and depth of the decision trees that does not impact much on the classification of the instances. Pruning a decision tree also help with reducing the overfitting and hence the model able to predict better with the unseen data. We will be implementing our pruning using post-pruning, reduced error pruning algorithm.

   Pruning our first classification tree, we are able to observe that both children are leading to the same type of labels,meaning that removing them won't affect the performance and the accuracy of before and after will be the same. Figures below demonstrating the removal of nodes without impacting the accuracy.



(a) After Pruning Once



(b) After Pruning twice

Accuracy: 89.83%

(c) Acc. Before & After Pruning

Figure 6: Pruning two Nodes

   The pruned tree should be higher or equal in performance and smaller in size.

2. **In this assignment, you trained a binary tree. Explain how you would use a single decision tree to learn to predict multiple independent binary class labels. Explain how to make decisions in leaf nodes, and how you would have to change your query search algorithm for any node.**

   Answer:

   Since this is a multi binary class problem and there is no relation between each binary classes. For example, we have 3 independent binary classes for each instances, usually we will simply construct 3 trees to solve that. However to merge 3 problems into 1, we will need some tricks to turn it into a multi class problem. we can simply concatenate 3 labels into one turning them into { [0,0,0], [0,0,1], [0,1,0], ... , [1,1,1] } of multi class, then construct the tree using the same functions we have. The prediction label at the leaf node will be assign with the maximum occurrence of the label. Here's the catch, the multi class isn't merge into one as strings but as arrays, so that we can calculate the max occurrence of each bit when the leaf node is reached. (e.g. take multi class labels, calculate max occurrence of the 1st bit, then the 2nd bit, then the 3rd, assign the array to leaf).

We also need to change the calculation of impurity. Here we present similar method which still use the same information entropy. In the following equations. i represent the label number (In this case we have 3 independent classifier), j represent the distinct value of a label(We got 2 because we are having binary classification.)

$$I(\frac{p}{p+n}, \frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$Remainder(A) = \frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{2}\frac{p_{ij}+n_{ij}}{p_i+n_i}I(\frac{p_{ij}}{p_{ij}+n_{ij}} + \frac{n_{ij}}{p_{ij}+n_{ij}})$$

$$Gain(A) = \frac{1}{3}\sum_{i=1}^{3}I(\frac{p_i}{p+n} + \frac{n_i}{p+n}) - Remainder(A)$$

Now with this, we're simply averaging 3 gains to predict 3 binary labels at once using one training dataset that constructed this single tree. The traversal of the tree will work as the way it is without changing anything.

Conclusion

Decision Tree is able to find decisive features for classification in a short time. In our experiment, decision tree with depth 2 performs the best overall with the features time, serum sodium and serum creatinine. However, we know that after pruning the tree layer 1 decisions are all leading to the same class for each of them meaning they can be removed. After all, time is what separate the label the most for our classification dataset.

Decision Tree also more capable on reasoning, we can see the decision process and try to change the decision by changing the threshold of a node. In the other hand, SVM is harder to interpret with the prediction, we can only rely on the training set it fit is a good generalization of data set of the task.