Classification Task Using Machine Learning Techniques

Overview: **Student ID: 17230755**

This task involves searching for machine learning packages and prediction of **test result** column in illness data set which contains 376 rows and 8 attributes. I will be selecting two classification machine learning techniques **Decision Tree** and **Random Forest** for prediction and finally comparing their performance by measuring accuracy of the models.

Preparation of Data Set/Cleansing:

Data cleansing and visualisation plays an important role before proceeding with evaluation of the any type of data. In given Illness data set, below are the steps Involved in cleansing data.

- ✓ Import the data set to excel from comma delimited txt file.
- ✓ Transpose the rows into columns. This can be done through excel by using the Transpose(T) while pasting.
- ✓ Giving the column names in the following order: plasma_glucose, bp, test_result, skin_thickness, num_pregnancies, insulin, bmi, pedigree, age.
- ✓ Factoring (2 levels) the **test_result** column in illness data set, which is used of classification.

Techniques/Algorithm Used:

- Decision Tree and Random Forest

Machine Learning Packages and Software Selection:

- ✓ **Software used:** R, RStudio.
- ✓ **Packages Used for decision tree and random forest:**

  - **readxl**: Used to read/import illness dataset to R which is stored as xlsx file. However, there are several other packages which does the same as XLConnect, read.table etc.
  - **rpart and rpart.plot:** Widely used to design and plotting the decision tree model. In addition, rpart.plot gives the good visualisation of tree over tree() package.
  - **FSelector:** Used for finding information gain of different attributes for analysing purpose. However, there are other packages for finding the information gain like RWeka, CORElearn etc. The reason why I chose FSelector over other packages is I can give the logarithm base value (2) for finding IG as a parameter to the function.
  - **randomForest:** Widely used to design the random forest tree model and this also offers several plots like variable importance plot etc.
  - **caret:** This package contains several functions to find the accuracy and tune the model (e.g. K fold cv, auto tuning the model parameters, to find the model accuracy by resampling procedure. In addition, this package also offers several plots to visualize the model performance.

Model Evaluation:

Decision Tree:

Decision tree learning uses a decision tree to go from observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values are called **regression trees** *(Wikipedia, n.d.)****[1]***.

Node selection to split the data into subset in decision tree, is based on the information gain of each attribute. Information gain is the function of entropy. Entropy is defined as below (Wikipedia, n.d.) -

$$H(T) = I_E(p_1, p_2, \dots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

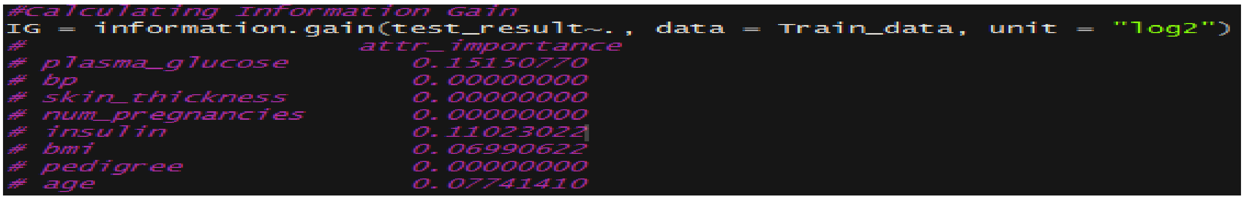
(Wikipedia, n.d.)[2]

Information Gain = Entropy(parent) - Weighted Sum of Entropy(Children)

$$IG(T, a) = H(T) - H(T|a)$$

(Wikipedia, n.d.)[3]

The above can be calculated in R using FSelector package. Below is R command to get the Information gain and its value for illness data set.



```
#Calculating Information Gain
IG = information.gain(test_result~. , data = Train_data, unit = "log2")
#                    attr_importance
# plasma_glucose          0.15150770
# bp                      0.00000000
# skin_thickness          0.00000000
# num_pregnancies         0.00000000
# insulin                 0.11023022
# bmi                     0.06990622
# pedigree                0.00000000
# age                     0.07741410
```

Procedure followed in constructing decision tree:

- ✓ Splitting the data into training data (70%) and testing data (30%) as Train_data and Test_data.
- ✓ Designing the decision tree model using rpart package and plotting the decision tree for given Illness data set. Test_result is our response variable and rest of the attributes is our predictors. **Rpart package automatically finds the best attributes and split**. Below is the decision tree for given data set. Attribute plasma_glucose has highest IG hence Tree starts splitting from plasma_glucose.

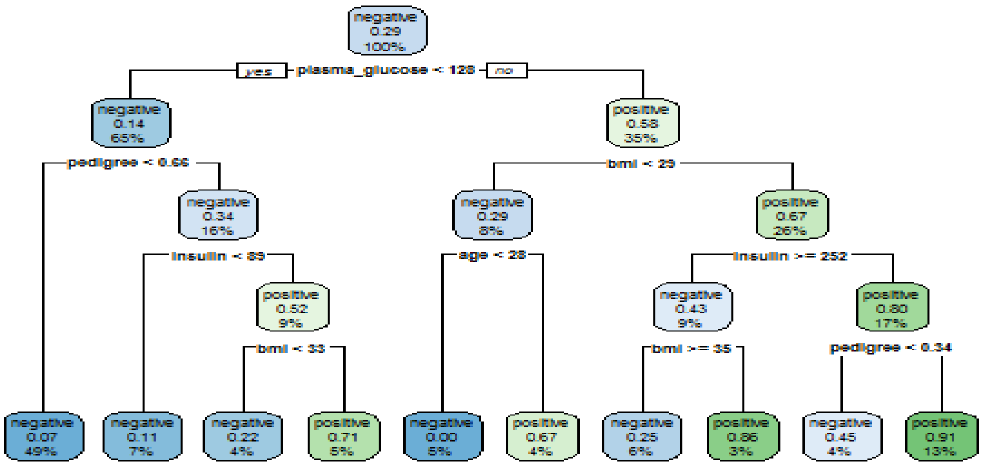


Figure 1: Decision Tree

- ✓ Manually pruned **(Manual Tuning the parameter)** the data for better prediction. Because of pruning, **accuracy** of the model on test data is increased from **70.25%** to **80.17%.** Here, selected cp = 0.054 where xerror is minimum i.e. 0.68 and nsplit = 2. cp is the function of node split. Hence, we need to select the cp where relative error or misclassification is less. Below is the pruned model and error vs cp plot. **Pruning is necessary to avoid overfitting of data**.

(Overfitting occurs when model is trained in detail (pure set in leaf node) and to the noise data which decreases the performance, where as if there is no enough data to train the model, then it might suffer from underfitting the model.

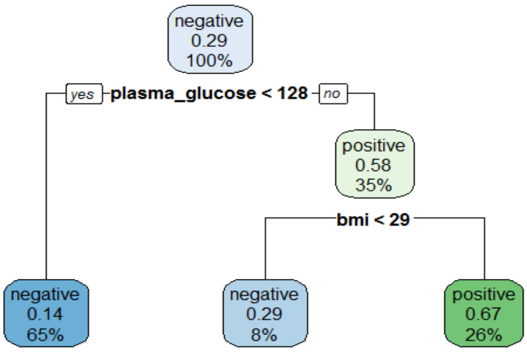


Figure 2: pruned Tree

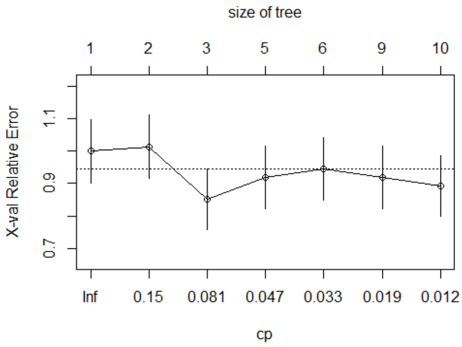


Figure 3: Relative error vs cp/size of tree

✓ **10-Fold cross validation and Auto tuning of parameter** of algorithm using **caret** package. Here, dataset is partitioned into 10 sets, keeping 1-fold for validation and rest is used for training and same is repeated for all the 10 folds. This ensures model is trained to all the combination of data sampling and **mean accuracy** of the all the folds gives better idea of model accuracy. In addition, it removes bias prediction by training all combination of data. Average accuracy obtained on test data set **(unseen data while training)** is **80.17%** same as pruned model. The final value used for model was cp = 0.054.

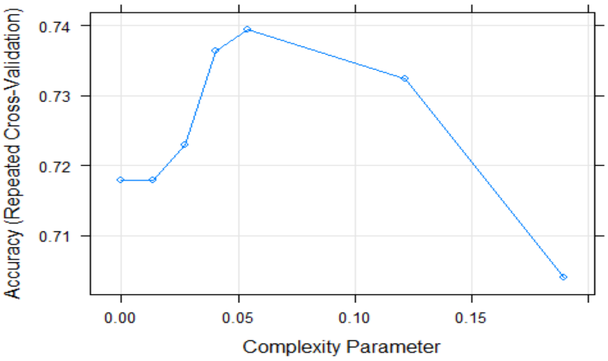


Figure 2: Accuracy vs Complexity parameter

Random Forest:

Random forests or random decision forests[4][5] are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set *(Hastie, Tibshirani, & Friedman, n.d.)*[6]**.**

Random forest model plots multiple decision tree by randomly selecting attributes at a time. For classification, prediction is based on **voting (Max votes)** of different trees and for regression it's the mean values of different trees. The parameters that we need to concentrate is **mtry** = number of random attributes selected at a time (default value is 3)**, ntree** = number of trees used for prediction (default value is 500 in randomForest and caret package) and **depth of each tree. Advantages of random forest over decision tree is**: Less sensitive to noise, avoids overfitting, less variance.

Procedure followed in constructing decision tree:

- ✓ Splitting the data into training data (70%) and testing data (30%) as Train_data and Test_data.
- ✓ Designing the decision tree model using randomForest package for given Illness data set. Test_result is our response variable and rest of the attributes is our predictors. Accuracy of random forest without tuning and k fold validation is **80.99%. Variable importance graph** is shown below: Decrease in Gini tells if we remove the variable, how much information we lose while prediction.

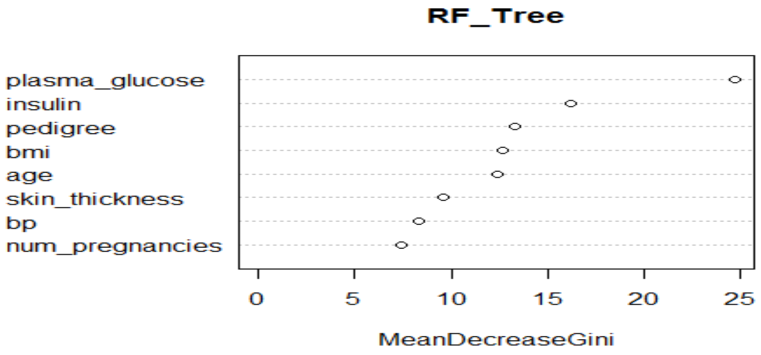


Figure 3: Attributes vs Mean Decrease Gini

- ✓ **10-Fold cross validation and Auto tuning of parameter** (**mtry**) of algorithm using **caret** package. Average accuracy obtained on test data **(unseen while training)** set is **81.82%** same as pruned model. The final value used for model was **mtry = 8, ntree = 500.** Manual tuning can also be done using tuneRF() function, and then giving the obtained mtry to RF model.

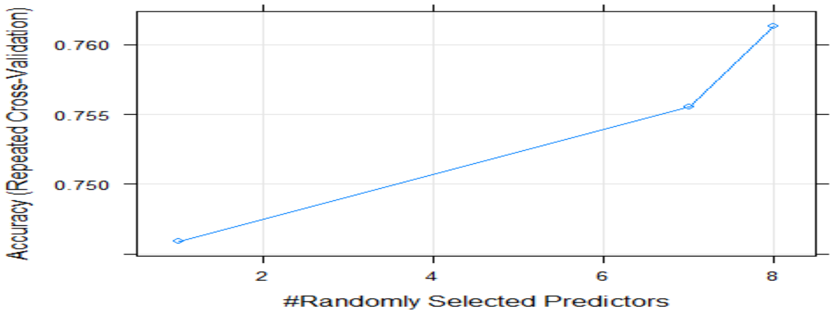


Figure 4: Accuracy vs Randomly Selected Predictors

<u>Conclusion:</u>

From the above model evaluation, we can say that both the algorithm fetches very similar result for a given data set. However, random forest has slight upper hand in prediction not only in terms of accuracy, but also on sensitivity, balanced accuracy, positive prediction rate and negative prediction rate. Moreover, random forest predicts the classification based on voting from different number of trees, effect of noise, overfitting, variance is less than decision tree. In addition, by tuning the random forest by finding best mtry, ntree, depth of tree and cv validation we can improve the performance of the model. However, decision tree has an advantage if the there is no noise in data set, less number of attributes that are used for prediction and has an advantage in terms of computational time.

Citation and References:

1. Wikipedia. (n.d.). Decision Tree Learning. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Decision_tree_learning. [1][2][3]
2. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.[4][5]
3. Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. [6]
4. Rpart: https://cran.r-project.org/web/packages/rpart/rpart.pdf
5. Randomforest:https://cran.r-project.org/web/packages/randomForest/randomForest.pdf
6. Fselector: https://cran.r-project.org/web/packages/FSelector/FSelector.pdf
7. Caret package: https://cran.r-project.org/web/packages/caret/caret.pdf
8. Readexl package: https://cran.r-project.org/web/packages/readxl/readxl.pdf
9. Quora: https://www.quora.com/What-are-some-advantages-of-using-a-random-forest-over-a-decision-tree-given-that-a-decision-tree-is-simpler