DEPARTMENT OF MATHEMATICS AND STATISTICS

Project Report

Paper : MTH552A – Statistical and AI Techniques in Data Mining

Semester II

Detection of Presence of Chronic Kidney Disease using Random Forest Classifier

Submitted by -

Anis Pakrashi :	**211264**
Sovon Gayen :	**211465**

-under the supervision of

Prof. Dr. Amit Mitra

ABSTRACT

Chronic kidney disease is a prevalent health issue among many people around the world. It’s always good to be aware of the possible risk factors and detecting the disease at the earliest is crucial for a healthy living. In this project, we look into a data taken over 2-month period in India, which consists of a categorical class variable, *classification* having two classes: ‘notckd’ for no chronic kidney disease and ‘ckd’ for presence of the disease; and 25 other variables which may possibly affect the occurrence of the disease, some of which are age, bp, rbc and pcv. There are 400 observations in all. The objective of this project is to fit a random forest classifier model using 70% of the data and test its performance on the rest 30% of the data using performance diagnostics from confusion matrix, like accuracy, specificity rate and sensitivity rate. At each step we compare the efficacy of the random forest classifier with the common decision tree classifier. In the process of building the problem, we shall perform initial exploratory data analysis and check model adequacy measures on the 70% data. Using the remaining 30% of data, we see how well our classification problem works. Finally, we mention the important variables which have significant impact in the proper classification.

Keywords: Chronic Kidney Disease, Classification, Random Forest Classifier, Decision Tree Classifier

ACKNOWLEDGEMENT

We present our project report on "detection of presence of chronic kidney disease using Random Forest classifier." However, as mere students of statistics, any achievement, in terms of a substantial project, would have been an uphill task, considering just our own efforts. It is apparent that there has always been a constant encouragement of noble minds for our efforts to bring this project to a successful completion. The satisfaction that accompanies the effectiveness of any task would be incomplete without mentioning those who made it possible for us to complete our desired project.

First and foremost, we want to express our sincere gratitude to our instructor Dr. Amit Mitra for his constant help, support and advice throughout the project preparation, which acted as a catalyst for effective completion of our work. Without his valuable guidance and motivation, it would have been nearly impossible to work as a team and imbibe the practical aspects of the course "MTH552A: Statistical and AI Techniques in Data Mining".

Besides, we are thankful to all faculty members of our department and our seniors, because without their support at various stages, this project would not have materialized.

We are also thankful to our friends for their critical appreciation and their questions and counter-questions solved a huge spectrum of doubts related to our work.

Last but not the least, we are grateful to our parents for their constant motivation on the way of materializing the project.

Anis Pakrashi
Sovon Gayen

CONTENTS

1. INTRODUCTION-- 1

1.1 Context--- 1

1.2 Objectives -- 2

2. DATASET DESCRIPTION--- 2

3. DATA CLEANING --- 4

3.1 Encoding of binary variables-- 4

3.2 Missing Value Imputation --- 5

4. DATA EXPLORATION--- 6

5. THEORY OF CLASSIFICATION PROBLEM--10

5.1. What is Classification? ---10

5.2. Classification Trees ---10

5.3. Random Forest Classifier---11

6. DATA SPLITTING---11

7. MODEL FITTING--11

7.1 Fitting Classification tree to the Training data --12

7.2 Fitting Random Forest to the Training data---12

8. MODEL ADEQUACY USING TRAIN DATA---15

8.1 The Classification Tree setup ---15

8.2 The Random Forest setup --17

8.3 Comparison---18

9. MODEL ADEQUACY USING TEST DATA--18

9.1 The Classification tree setup --18

9.2 The Random Forest setup --20

9.3 Comparison---21

10. VARIABLE IMPORTANCE IN THE MODEL --21

10.1 Determining important variables from the Random Forest classifier model ---21

10.1 Variable Importance Plot---23

11. CONCLUSION---24

12. REFERENCES --24

1. INTRODUCTION

1.1 Context

The kidneys are the most important part of the sewerage system of the human body. They form the backbone behind filtering out the waste materials from the blood and making the circulation system pure once again. Thus, as kidneys fail, wastes build up inside the body and their accumulation causes a host of different problems. Longstanding problems lead to renal failure and chronic kidney disease. The main risk factors that possibly lead to kidney disease are diabetes, high blood pressure or even a family history of renal failure.

Early diagnosis can lead to making predictive models for the risk of the disease. However, it is to be noted that there may be finer reasons for the presence or absence of the disease in a certain individual. These may include variables like age, blood pressure and such smaller aspects rather than just broader aspects like cardiovascular or diabetic disorder. This may be a hot recipe for the classification modelling problem.

Various modelling techniques have been developed to address the lack of standardized processes that incorporate the perspectives of all healthcare investors. Such models can back the decision-making process aimed at achieving specific clinical outcomes, and at the same time, guide the allocation of healthcare resources and reduce costs. Classification models are predominantly relevant to healthcare, mainly in the clinical sector, with implications for payers, patients, and providers. Significant improvisation in the clinical decision-making process leads to a boom in the usage of such models to achieve better outcomes, while reducing overall healthcare costs. Some models are aimed at predicting a clinical outcome, whereas others focus on identifying patients who may be at risk for the development of a particular condition. We, in the subsequent work, try to focus on the classification perspective at a slightly deeper level.

A plethora of different classification methods are available, like the Fisher Linear Discriminant Function (FLDF) classifier, the multiple logistic classifier, the k-nearest neighbour classifier, the decision tree classifier and the Random forest classifier, to name a few.

1.2 Objectives

In our project we consider a host of possible variables which may be important in classifying individuals into two classes – one with presence of chronic kidney disease and the other without. We perform exploratory data analysis and also data cleaning, with missing value imputation and other tasks. We then fit the Random Forest classifier model and at each stage we compare the results with the ordinary Classification tree model. We also report the variables which turned out to be important in our classification problem.

2. DATASET DESCRIPTION

The dataset related to our project is a survey data and secondary data. The dataset is taken over 2-months period time in India. It consists of 400 responses and 26 feature variables. The classification is done based on an attribute which takes either 'ckd'(Chronic kidney decease) or 'notckd'. The feature variables are listed below:

- **id:** Count of the entries in the dataset.
- **age:** Age of the respondents.
- **bp:** Measurement of blood pressure of the individuals.
- **sg:** Specific gravity of urine which normally ranges between 1.005 to 1.030.
- **al:** Albumin level in an individual's blood. Normally albumin level ranges from 3.5 to 5.5 grams per decilitre.
- **su:** Amount of sugar in urine (in millimoles per litre)
- **rbc:** Red blood corpuscle count of the individuals. Responses are binary in terms of "normal" or "abnormal."
- **pc: C**ount of pus cells in the urine. The normal range of pus cells ranges from 0-5. Responses are recorded as binary in terms of "normal" or "abnormal."
- **pcc:** Presence of pus cells in urine with clumps which indicates pyuria and presence of pyuria often occurs in a urinary tract infection. Responses are taken as "present" or "notpresent."
- **ba:** If urine contains bacteria or not. Responses are taken as "present" or "notpresent."
- **bgr:** Value of Blood Glucose random test of the respondents. responses are of continuous type.
- **bu:** Blood urea nitrogen test measures the amount of urea nitrogen in blood.

- **sc:** Serum creatinine level is based on a blood test that measures the amount of creatinine in our blood. Healthy kidneys filter creatinine from our blood through urine.
- **sod:** Blood sodium level count that normally ranges between 135 to 145 milliequivalents per litre. Responses are of continuous type.
- **pot:** Count of potassium level in blood that normally ranges from 3.6 to 5.2 millimoles per litre.
- **hemo:** Haemoglobin count in blood. For men, normal hemoglobin counts are 14 to 17 gm/dL while it's 12 to 15 gm/dL for women.
- **pcv:** The packed cell volume is a measurement of the proportion of blood that is made up of cells. The value is expressed as a percentage of cells in blood. For example, a PCV of 40% means that there are 40 millilitres of cells in 100 millilitres of blood.
- **wc:** White blood cell count in blood of the individuals.
- **rc:** Red blood cell count in blood of the individuals.
- **htn:** High blood pressure or hypertension is a condition in which the long-term force of the blood against artery walls is high enough for causing health problems. Responses are taken as "yes" or "no" according to the fact that if an individual is suffering from hypertension or not.
- **dm:** Responses are recorded as if the respondents are suffering from diabetes mellitus or not. Diabetes mellitus refers to a group of diseases that affect how our body uses blood sugar. Responses are taken as "yes" or "no."
- **cad:** If an individual is suffering from coronary artery disease or not. Responses are binary in nature. ("yes" or "no")
- **appet:** Appetite refers to the eating behaviour or a natural desire to satisfy a bodily need for food. In our data appet is taken as "poor" or "good" according to the respondents' appetite condition.
- **pe:** Pedal edema causes an abnormal accumulation of fluid in the ankles, feet and lower legs causing swelling of the feet and ankles. Responses are "yes" or "no."
- **ane:** Anaemia is a condition in which there is a deficiency in red blood cells or haemoglobin in blood. We got the responses as "yes" or "no" for an individual is suffering from anaemia or not.
- **classification:** Classification takes either 'ckd'(presence of chronic kidney disease) or 'notckd'. Our prime objective is classification which is based on this attribute.

3. DATA CLEANING

3.1 Encoding of binary variables

Our dataset contains many binary variables with categories labelled as strings. We change the levels from text to numeric to make calculations and interpretations more understandable. The detailing is given in the following table:

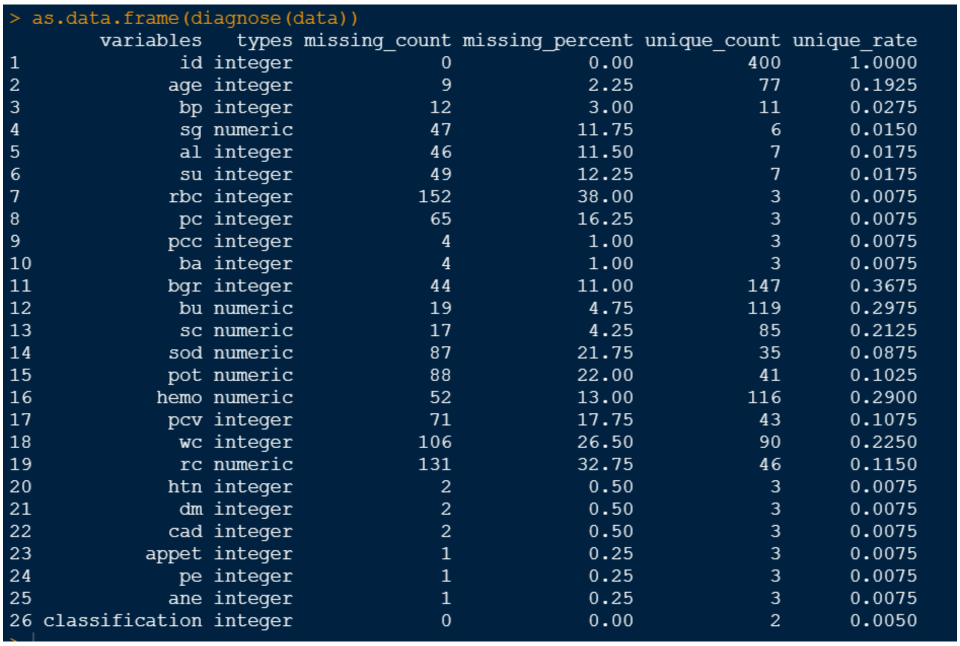
Table 1 – Encoding of binary variables

Binary Variable	Original levels	Encoded levels
rbc	normal	1
	abnormal	0
pc	normal	1
	abnormal	0
pcc	present	1
	notpresent	0
ba	present	1
	notpresent	0
htn	yes	1
	no	0
dm	yes	1
	no	0
cad	yes	1
	no	0
appet	good	1
	poor	0
pe	yes	1
	no	0
ane	yes	1
	no	0
classification	ckd	1
	notckd	0

3.2 Missing Value Imputation

Our dataset contains missing values and in fact there are several of them though out the dataset. The following snap gives a clear picture of variable wise count of missing values.

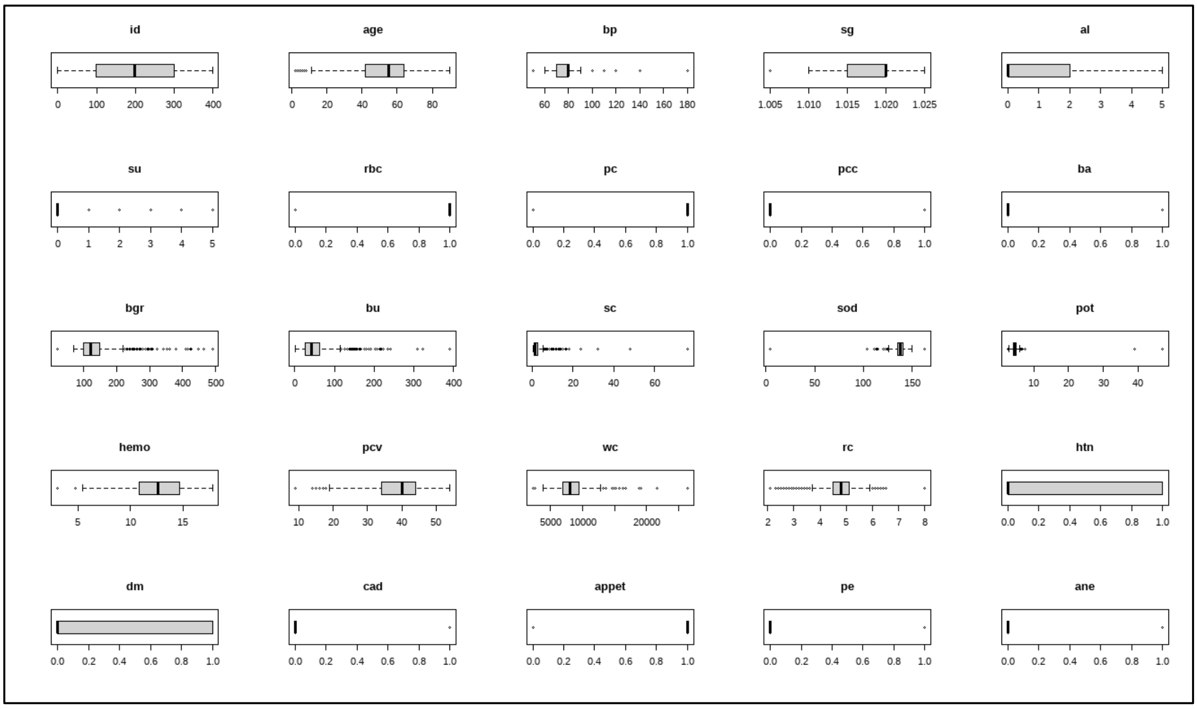
Figure 1



```
> as.data.frame(diagnose(data))
        variables   types missing_count missing_percent unique_count unique_rate
1              id integer             0            0.00          400      1.0000
2             age integer             9            2.25           77      0.1925
3              bp integer            12            3.00           11      0.0275
4              sg numeric            47           11.75            6      0.0150
5              al integer            46           11.50            7      0.0175
6              su integer            49           12.25            7      0.0175
7             rbc integer           152           38.00            3      0.0075
8              pc integer            65           16.25            3      0.0075
9             pcc integer             4            1.00            3      0.0075
10             ba integer             4            1.00            3      0.0075
11            bgr integer            44           11.00          147      0.3675
12             bu numeric            19            4.75          119      0.2975
13             sc numeric            17            4.25           85      0.2125
14            sod numeric            87           21.75           35      0.0875
15            pot numeric            88           22.00           41      0.1025
16           hemo numeric            52           13.00          116      0.2900
17            pcv integer            71           17.75           43      0.1075
18             wc integer           106           26.50           90      0.2250
19             rc numeric           131           32.75           46      0.1150
20            htn integer             2            0.50            3      0.0075
21             dm integer             2            0.50            3      0.0075
22            cad integer             2            0.50            3      0.0075
23          appet integer             1            0.25            3      0.0075
24             pe integer             1            0.25            3      0.0075
25            ane integer             1            0.25            3      0.0075
26 classification integer             0            0.00            2      0.0050
```

Consider the following collection of boxplots for all variables, other than *id*:

Figure 2

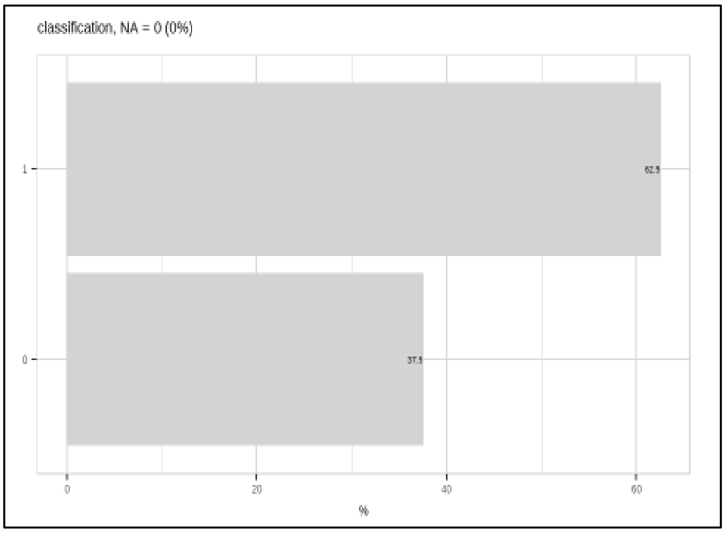


Instead of discarding the rows with missing values, which would lead to missing out on valuable information, we estimate the missing values on the basis of the remaining values (values which are already present) in that column. Now, to estimate the missing values our first choice should be the mean. However, the box plots reveal that in some variables, we have many outliers present. So, mean will not be a good measure and hence we use median as a more robust measure.

4. DATA EXPLORATION

Data Exploration is an approach to summarize the main characteristics of the data set, often with visual methods. The primary objective of this is to see what the data can tell us other than formal tasks of modelling or hypothesis testing.

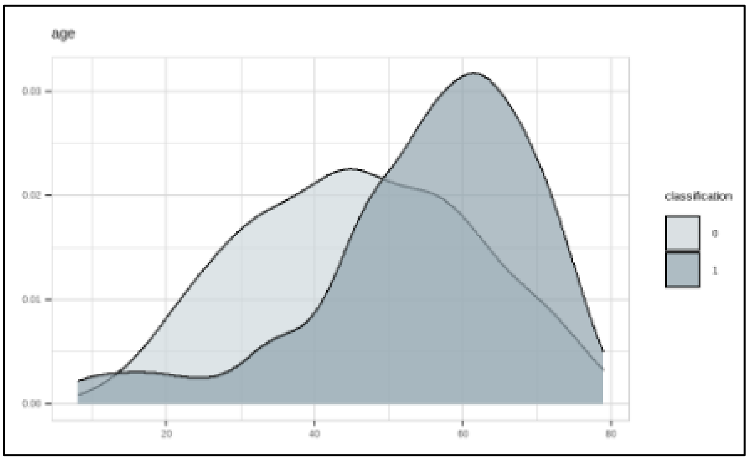
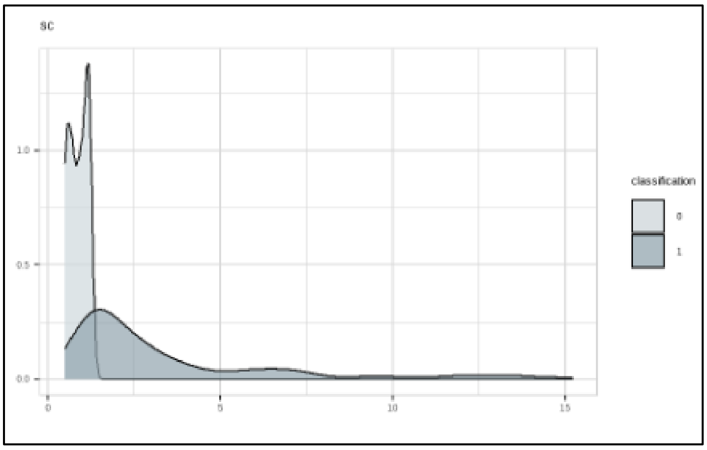
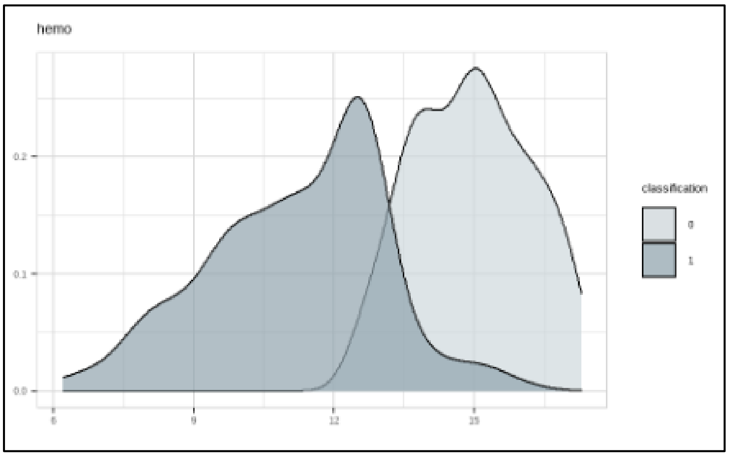
- We have total 400 observations with 26 variables.
- Among 400 observations 242 are containing missing values.
- 24 variables have missing values.

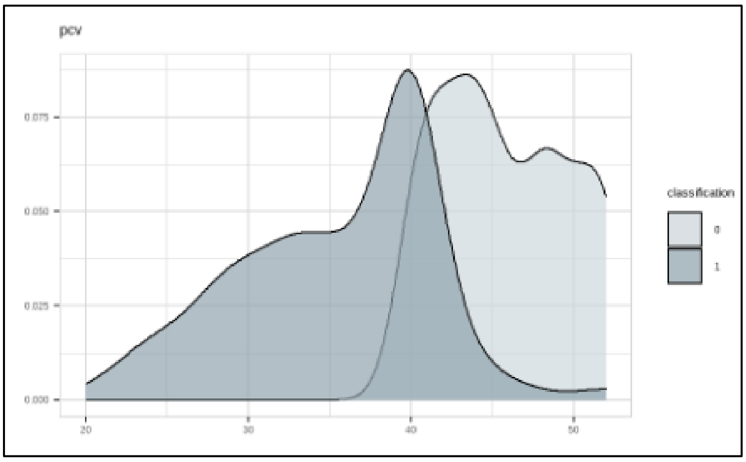
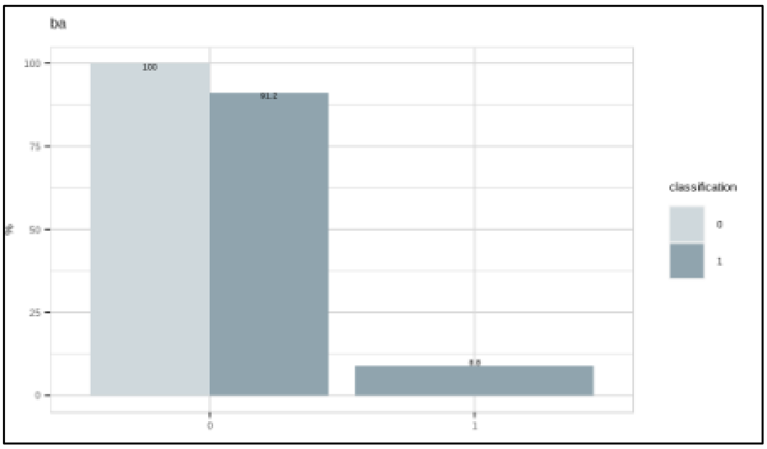
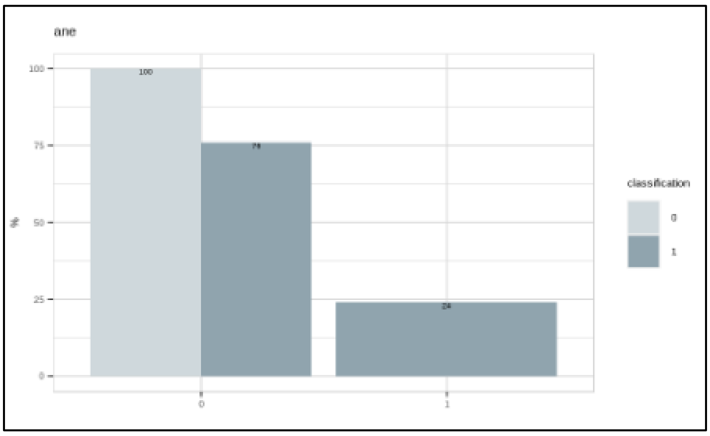


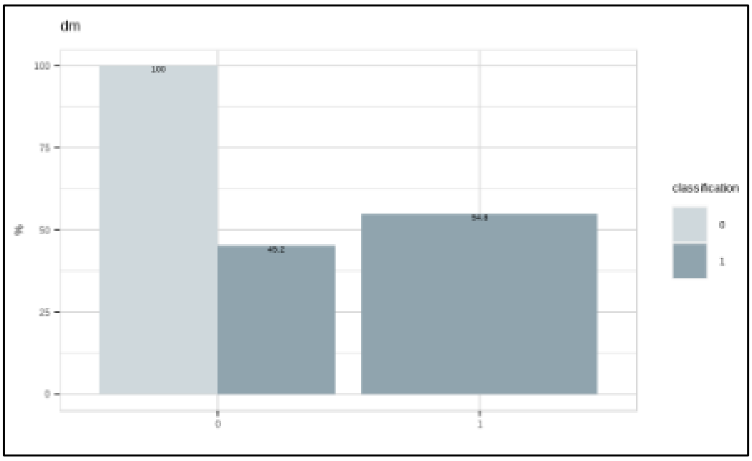
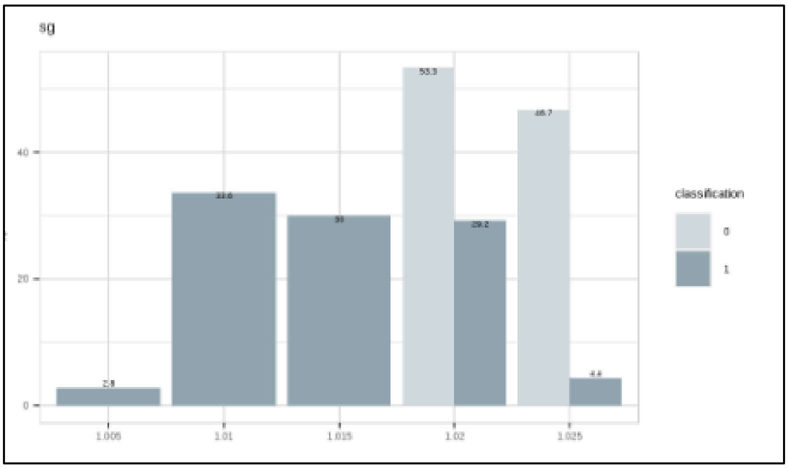
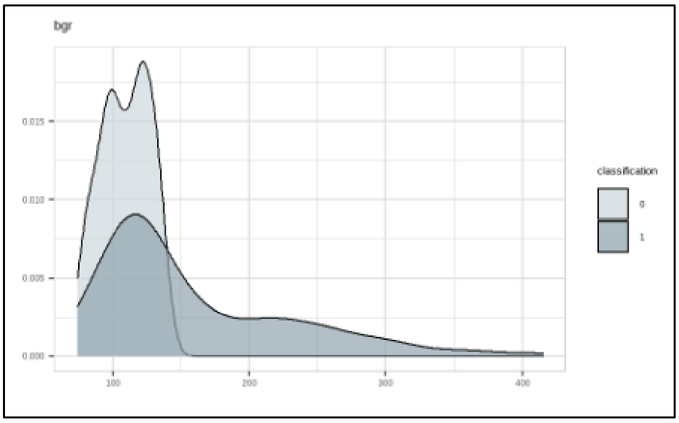
Classification is our target variable. We can see that 62.5% of the respondents are nonckd and 37.5%are ckd. We find no missing values here.

Now, we have listed a few variables and their relation with the target variable in the table below:

Table 2 – List of relationships of some variables with the variable *classification*

Plots of Target variable vs Feature variable	Interpretations
	From the plot we can infer that for nonckd individuals age is symmetrically related, most nonckd people are of 30 to 60 years old. On the other hand for the ckd individuals the graph is negatively skewed. Above 50 years of age people are more likely to be ckd.
	The typical range for serum creatinine is 0.74 to 1.35mg/dl for adult men and 0.59 to 1.04mg/dl for adult women. This is well justified from the graph that non ckd people are lying in that normal region. Whereas some ckd people's sc measure is crossing the range.
	Normal haemoglobin counts are 14 to 17 gm/dL for men and 12 to 15 gm/dL for women. We can see that hemo count of most of the nonckd individuals is normal but for ckd individuals we find there is deficiency in hemo count.

Plots of Target variable vs Feature variable	Interpretations
	PCV is the percentage of cells in the blood. The cell count is quite lower in ckd individuals than nonckd respondents. The figure shows that more than 40millilitres cell can be found in 100 millilitres of blood of a nonckd person whereas the count is less than 40millilitres for the ckd individuals.
	Presence of bacteria in urine is a serious concern. No nonckd individuals have been reported to have bacteria in urine but almost 8.8% of our ckd respondents are suffering from this condition.
	No nonckd respondent of our survey is suffering from anaemia whereas 24% our ckd respondents are suffering from anaemia.

Plots of Target variable vs Feature variable	Interpretations
 	None of our nonckd respondents is suffering from the disease Diabetes Mellitus. Almost 54.8% of ckd respondents are suffering from Diabetes Mellitus.
 	Usual range for the specific gravity of urine is 1.005 to 1.030. All the nonckd respondents are lying in the level of 1.02 and 1.025. A very few number of nonckd respondents have sg below 1.005.
 	Normal blood glucose random test ranges 110 to 126 and normal glucose tolerance level is 70-110. Most of the nonckd respondents' bgr is in normal range on the other hand some of the ckd respondents' bgr is >126.

5. THEORY OF CLASSIFICATION PROBLEM

5.1. What is Classification?

Classification is a supervised learning approach. This is the study of approaches for predicting qualitative responses. It is a process of categorizing a given set of observation into certain classes. We predict the probability of each of the categories of a qualitative variable as the basis for making the classification. Both structured or unstructured data can be used for this task. We can frame this technique as the task of approximating the mapping function from input variables to discrete output variables. The main task is to predict which class the new data will fall into.

5.2. Classification Trees

For a classification tree we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. from the results of a classification tree, we are interested not only in the class prediction corresponding to a particular terminal node region, but also in the class proportions also. In the classification setting we use classification error rate to assign an observation in a given region to the most commonly occurring class of training observations in that region. The classification error rate is the proportion of the training observations in that region that do not belong to the most common class.

- The misclassification error rate(MER) at a certain node t is given by,

  MER(t) = 1- P($\pi_j|t$) , Where P($\pi_j|t$) is the estimate of P(Xε U(t)|Y_i = π_j) based on the learning sample. U(t) : Partition of feature space induced by T with terminal node set $\tilde{T}$.

- Gini Index(t)= $\sum_{i,j}$ P($\pi_i|t$) P($\pi_j|t$)
- Cross Entropy or Deviance = - $\sum_i$ P($\pi_i|t$)log P($\pi_i|t$)

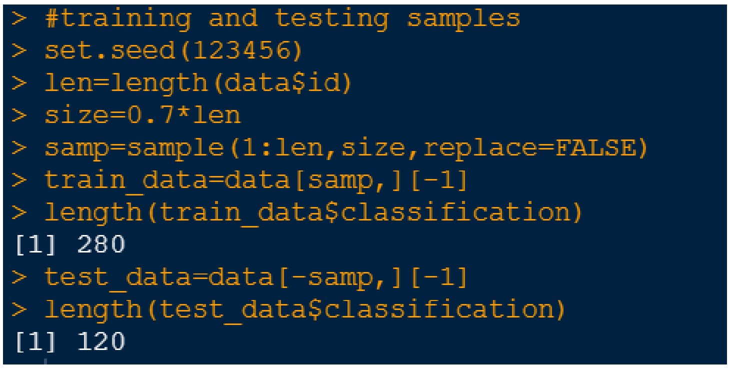
5.3. Random Forest Classifier

Random forest is an ensemble learning method for both classification and regression. Random forest operates by constructing a multitude of decision trees at training time. For our classification purpose the output of the random forest is the class selected by most trees. Random forests are a way of averaging multiple decision trees which are trained on different parts of the training set with the goal of reducing the variance.

6. DATA SPLITTING

We randomly divide the whole dataset into two parts. We take 70% of the total observations as train dataset on which the classifier model is initially fit. Remaining 30% of the total observations is used as test dataset. The number of observations in 2 parts are 280 and 120 respectively, as shown below.

Figure 3



```
> #training and testing samples
> set.seed(123456)
> len=length(data$id)
> size=0.7*len
> samp=sample(1:len,size,replace=FALSE)
> train_data=data[samp,][-1]
> length(train_data$classification)
[1] 280
> test_data=data[-samp,][-1]
> length(test_data$classification)
[1] 120
```

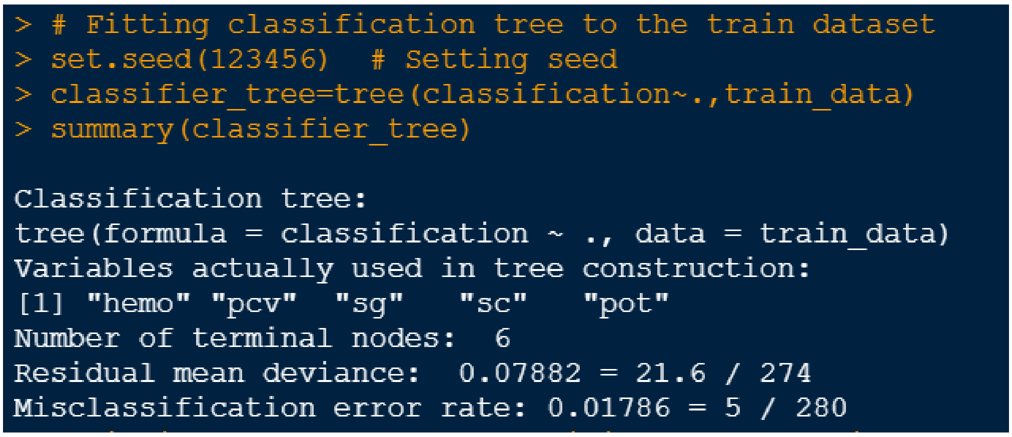
7. MODEL FITTING

We use the training set to fit the Classification models. Even though our primary task is to fit the Random Forest classifier, we initially fit the ordinary decision tree classifier and then the Random Forest to bring about a close comparison between the two.

7.1 Fitting Classification tree to the Training data

Consider the following code snippet with results:

Figure 4



```
> # Fitting classification tree to the train dataset
> set.seed(123456)  # Setting seed
> classifier_tree=tree(classification~.,train_data)
> summary(classifier_tree)

Classification tree:
tree(formula = classification ~ ., data = train_data)
Variables actually used in tree construction:
[1] "hemo" "pcv"  "sg"   "sc"   "pot"
Number of terminal nodes:  6
Residual mean deviance:  0.07882 = 21.6 / 274
Misclassification error rate: 0.01786 = 5 / 280
```

As a model diagnostic measure, we define

Misclassification Error Rate (MER) – the rate at which feature vectors are classified into wrong classes. For a tree T, it is denoted by

$$R(T) = \sum_{t \in \tilde{T}} \frac{N(t)}{N}(1 - max\ (p(\pi_i|t)))$$

Where, N(t) is the number of sample points reaching node t,

N is the total number of points in the learning sample,

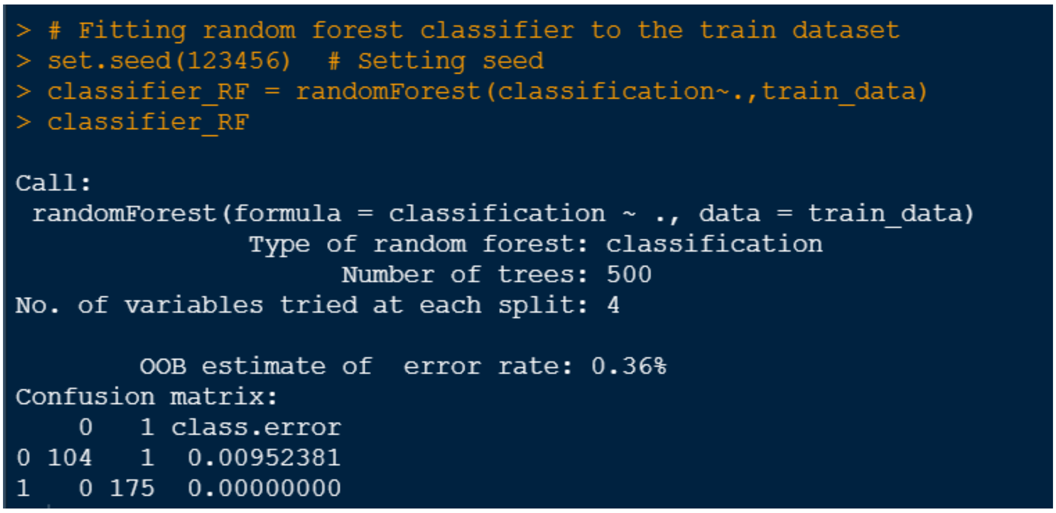
$\pi_1 and\ \pi_2\ are\ the\ two\ classes\ under\ consideration$

In this case, **MER = 0.01786**, which is a significantly low value. It basically means that only 5 out of 280 sample points in the training feature space have been incorrectly assigned class labels.

7.2 Fitting Random Forest to the Training data

Consider the following code snippet with results:

Figure 5



```
> # Fitting random forest classifier to the train dataset
> set.seed(123456)  # Setting seed
> classifier_RF = randomForest(classification~.,train_data)
> classifier_RF

Call:
 randomForest(formula = classification ~ ., data = train_data)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 0.36%
Confusion matrix:
    0   1 class.error
0 104   1  0.00952381
1   0 175  0.00000000
```

As a model diagnostic measure, we define

OOB Estimate of Error Rate – The method of Random Forest involves bootstrap aggregation, where each new tree is fit from a bootstrap sample of the training observations. The out-of-bag (OOB) error is the average error for each such observation, calculated using predictions from the trees that do not contain that observation in their respective bootstrap sample. The rule is to find all trees that are not trained by the OOB instance. Now, taking the majority vote of these trees returns the result of error for the OOB instance, compared to the true value of the OOB instance. Finally, compile the OOB error for all instances in the dataset.

Here, OOB estimate is obtained as 0.36% (=0.0036), which is quite a low value too.

Now, the Confusion Matrix from the model fit is:

Table 3 – Confusion matrix from model

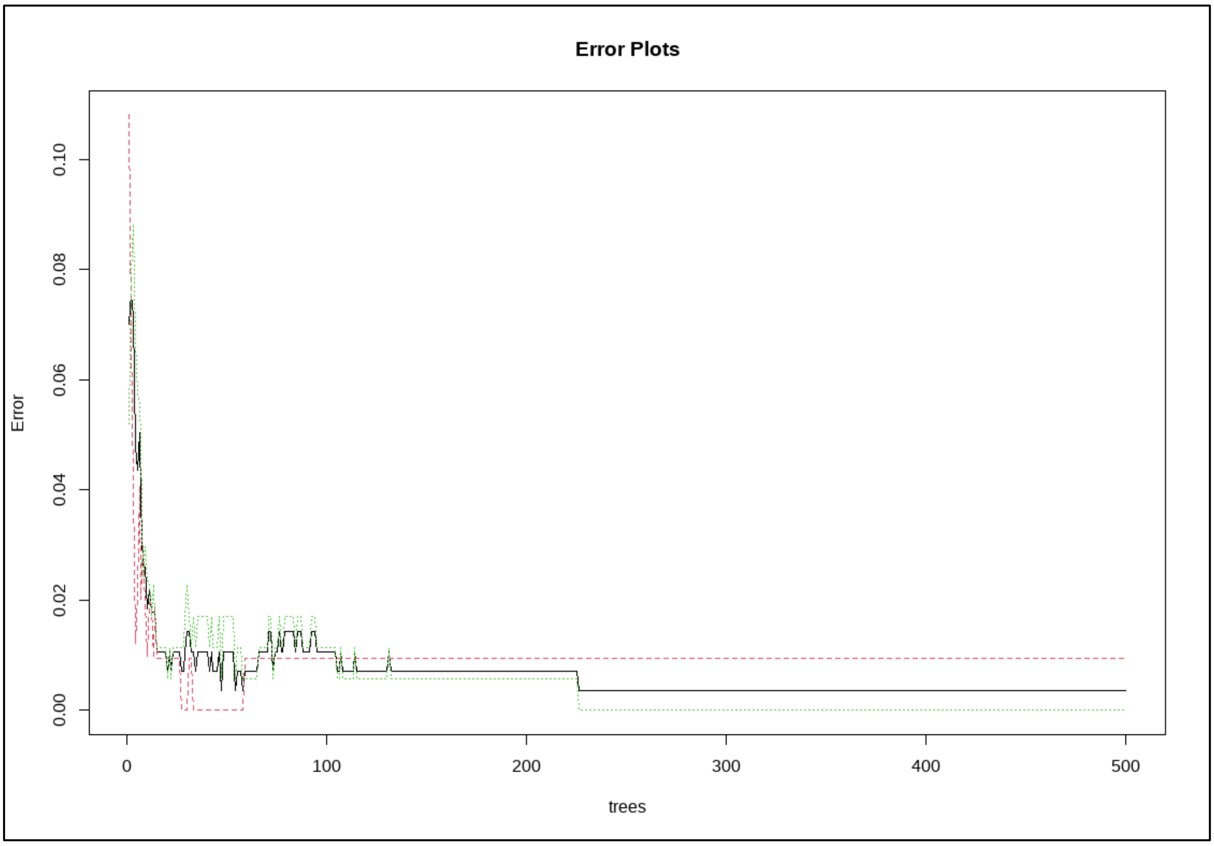
	Predicted Class	
True Class	0	1
0	104	1
1	0	175

We see that there is a near perfect classification situation in the model. 104 people not having the symptoms are also predicted to be not having the disease while 175 of them actually having the disease are also predicted to be so. The estimate of the **percentage of misclassification** from the above matrix is

$$PMC \ = \frac{Total\ number\ of\ misclassifications}{Total\ number\ of\ samples} x\ 100 = \frac{100}{280} \approx \mathbf{0.357}\ \%$$

If we plot the OOB error estimate and the two class error estimates against number of trees, we obtain something as given below:

Figure 6



The black line indicates the OOB error estimate, the green line represents the error estimate for the class signifying presence of *ckd*, while the red line indicates the class error for the class signifying *notckd*.

It is interesting to note that with more and more trees being used in this ensemble learning method, the error rates become more and more stable and the magnitude also drops significantly.

8. MODEL ADEQUACY USING TRAIN DATA

We initially use the train dataset for checking the adequacy of the fitted classification tree model and the random forest model, through construction of confusion matrices and using the subsequent model diagnostics.

Some common diagnostic measures are:

1. **Accuracy**: It is the number of correct predictions divided by the total number of predictions made by the model.

   $$Accuracy = \frac{f_{11} + f_{00}}{n}$$

   The higher the accuracy, the better is the model.

2. **Sensitivity Rate / True Positive Rate**: It is the proportion of positives correctly identified as positive.

   It is defined as: $\frac{f_{11}}{f_{01}+f_{11}}$

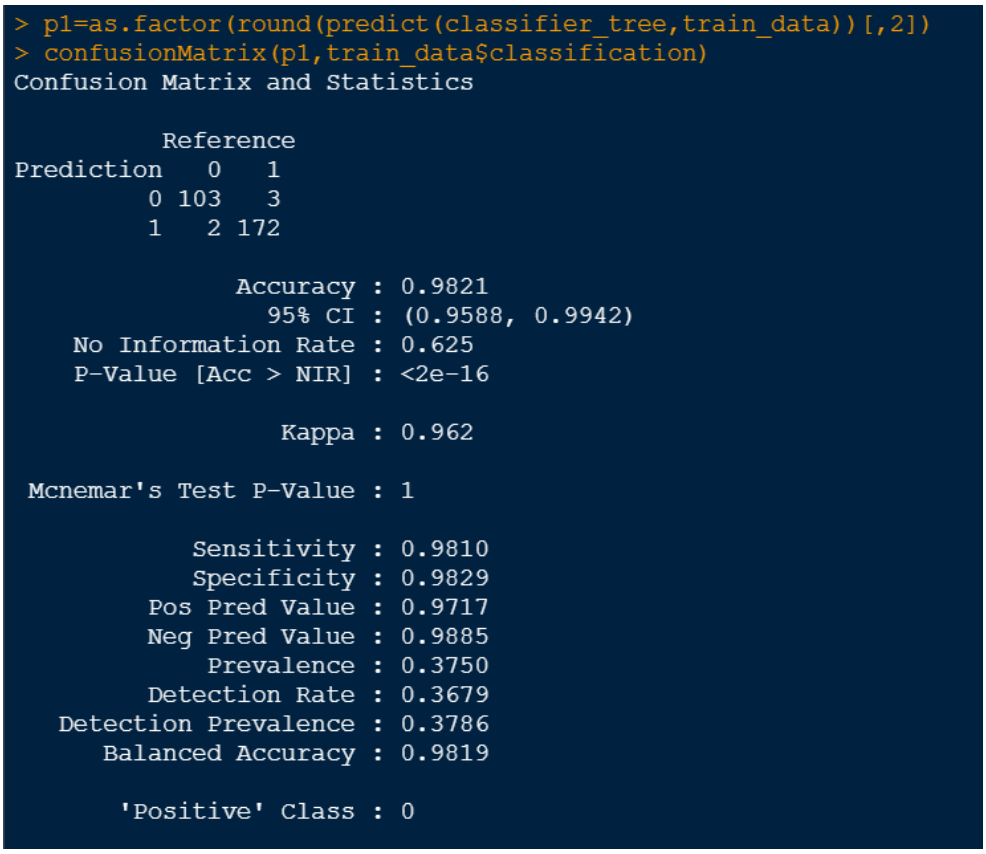
3. **Specificity Rate / True Negative Rate:** It is the proportion of negatives correctly identified as negative.

   It is defined as: $\frac{f_{00}}{f_{00}+f_{10}}$

8.1 The Classification Tree setup

Consider the following code snippet with results:

Figure 7



```
> p1=as.factor(round(predict(classifier_tree,train_data))[,2])
> confusionMatrix(p1,train_data$classification)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 103   3
         1   2 172

               Accuracy : 0.9821
                 95% CI : (0.9588, 0.9942)
    No Information Rate : 0.625
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.962

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9810
            Specificity : 0.9829
         Pos Pred Value : 0.9717
         Neg Pred Value : 0.9885
             Prevalence : 0.3750
         Detection Rate : 0.3679
   Detection Prevalence : 0.3786
      Balanced Accuracy : 0.9819

       'Positive' Class : 0
```

The Confusion Matrix :

Table 4 – Confusion Matrix from training set using decision tree

	Predicted Class	
True Class	0	1
0	103	2
1	3	172

The Model Diagnostics:

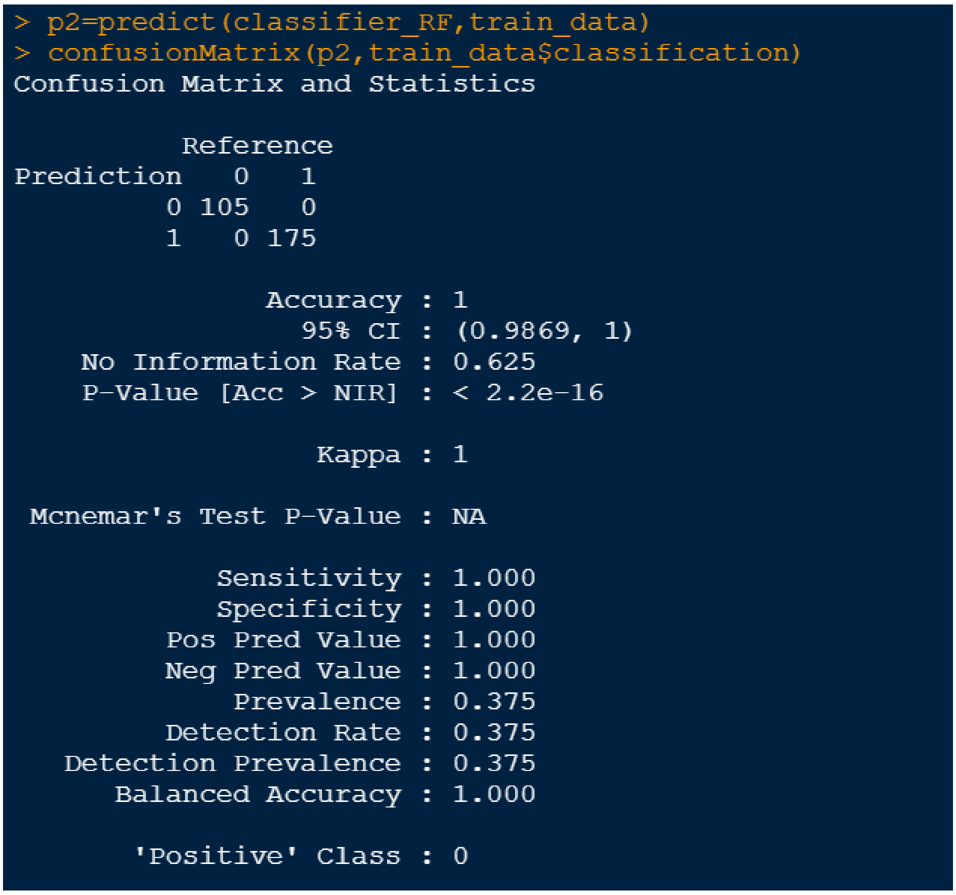
- Accuracy = 0.9821
- Sensitivity = 0.9810
- Specificity = 0.9829

Each of the above measures are having high values, which indicates that the model seems to perform well.

8.2 The Random Forest setup

Consider the following code snippet with results:

Figure 8



```
> p2=predict(classifier_RF,train_data)
> confusionMatrix(p2,train_data$classification)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 105   0
         1   0 175

               Accuracy : 1
                 95% CI : (0.9869, 1)
    No Information Rate : 0.625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.000
            Specificity : 1.000
         Pos Pred Value : 1.000
         Neg Pred Value : 1.000
             Prevalence : 0.375
         Detection Rate : 0.375
   Detection Prevalence : 0.375
      Balanced Accuracy : 1.000

       'Positive' Class : 0
```

The Confusion Matrix :

Table 5 – Confusion Matrix from training set using random forest

	Predicted Class	
True Class	0	1
0	105	0
1	0	175

The Model Diagnostics:

- Accuracy = 1
- Sensitivity = 1
- Specificity = 1

Each of the above measures are having values, which are exactly 1, which indicates that the model seems to fit perfectly.

8.3 Comparison

Comparing the three measures of model adequacy, we find that although the values are close enough, yet the random forest model performs better than the ordinary decision tree classifier model, at least to some extent.

9. MODEL ADEQUACY USING TEST DATA

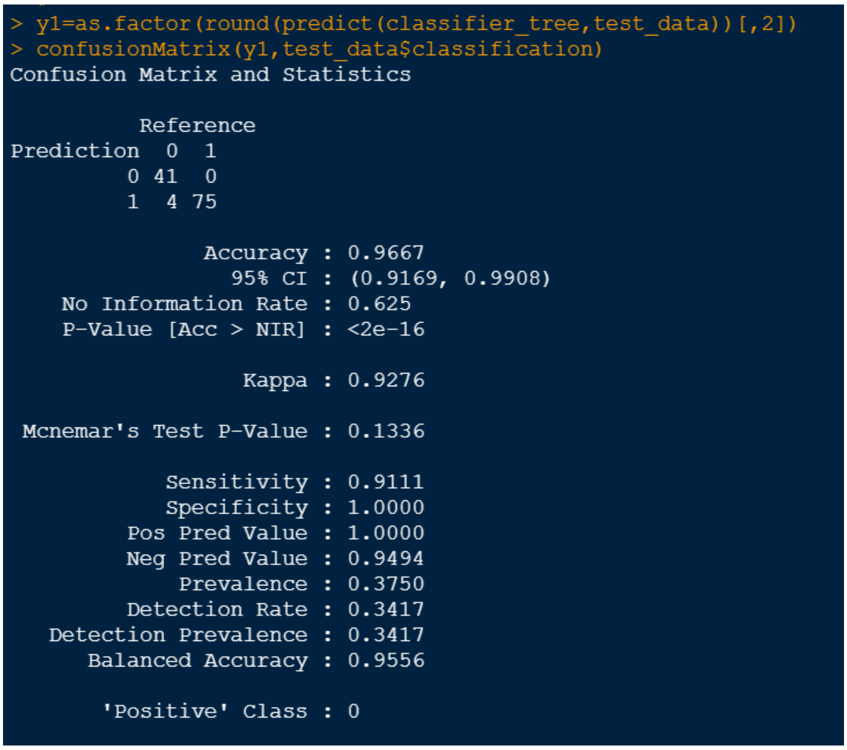
Now, we use the test dataset for predicting the classification accuracy and also testing the model adequacy of the fitted classification tree model and the random forest model, through construction of confusion matrices and using the subsequent model diagnostics. The diagnostics used are same as those used in section 8 above, that is,

- Accuracy
- Sensitivity
- Specificity

9.1 The Classification tree setup

Consider the following code snippet with results:

Figure 9



```
> y1=as.factor(round(predict(classifier_tree,test_data))[,2])
> confusionMatrix(y1,test_data$classification)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 41  0
         1  4 75

               Accuracy : 0.9667
                 95% CI : (0.9169, 0.9908)
    No Information Rate : 0.625
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9276

 Mcnemar's Test P-Value : 0.1336

            Sensitivity : 0.9111
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9494
             Prevalence : 0.3750
         Detection Rate : 0.3417
   Detection Prevalence : 0.3417
      Balanced Accuracy : 0.9556

       'Positive' Class : 0
```

The Confusion Matrix :

Table 6 – Confusion Matrix from test set using decision tree

	Predicted Class	
True Class	0	1
0	41	4
1	0	75

The Model Diagnostics:

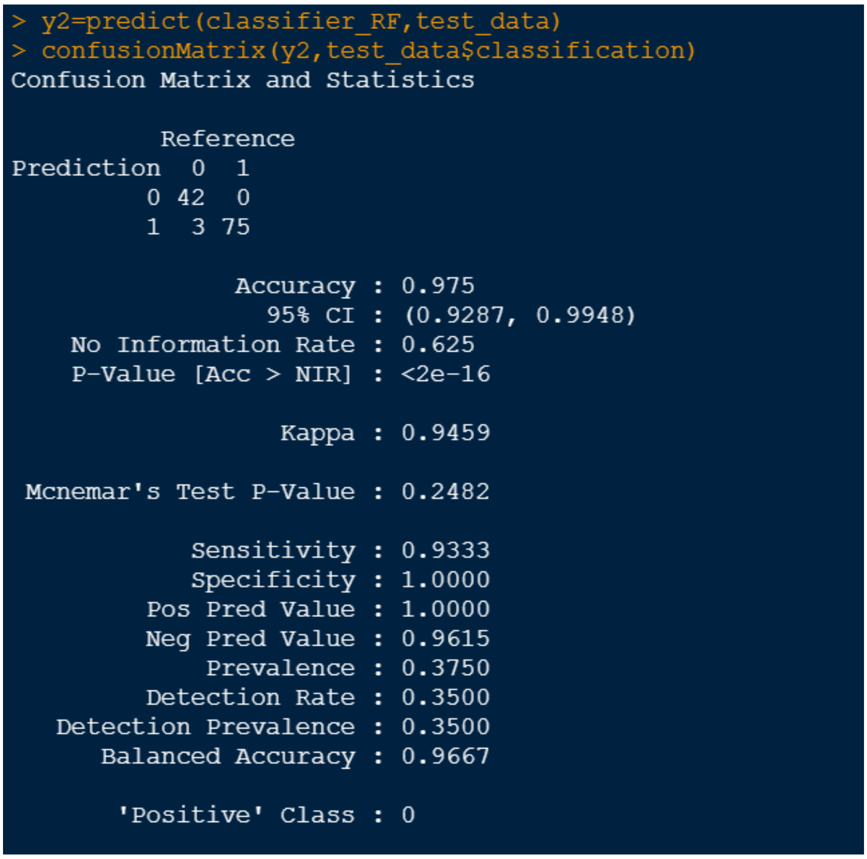
- Accuracy = 0.9667
- Sensitivity = 0.9111
- Specificity = 1

The diagnostic measures take significantly high values, which imply that the classifier model acts very well on the testing sample.

9.2 The Random Forest setup

Consider the following code snippet with results:

Figure 10



```
> y2=predict(classifier_RF,test_data)
> confusionMatrix(y2,test_data$classification)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 42  0
         1  3 75
                                          
               Accuracy : 0.975           
                 95% CI : (0.9287, 0.9948)
    No Information Rate : 0.625           
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.9459          
                                          
 Mcnemar's Test P-Value : 0.2482          
                                          
            Sensitivity : 0.9333          
            Specificity : 1.0000          
         Pos Pred Value : 1.0000          
         Neg Pred Value : 0.9615          
             Prevalence : 0.3750          
         Detection Rate : 0.3500          
   Detection Prevalence : 0.3500          
      Balanced Accuracy : 0.9667          
                                          
       'Positive' Class : 0               
```

The Confusion Matrix :

Table 7 – Confusion Matrix from test set using random forest

	Predicted Class	
True Class	0	1
0	42	3
1	0	75

The Model Diagnostics:

- Accuracy = 0.975
- Sensitivity = 0.9333
- Specificity = 1

The Random Forest model also returns very high values in terms of the model adequacy measures, which signify that the model is highly accurate in this perspective.

9.3 Comparison

Comparing the three measures of model adequacy, we find that although the values are close enough, yet the random forest model performs better than the ordinary decision tree classifier model, in terms of slightly higher accuracy and sensitivity.

10. VARIABLE IMPORTANCE IN THE MODEL

Here, we look into the Random Forest model to determine the possible important variables.

10.1 Determining important variables from the Random Forest classifier model

A classification model may involve numerous variables. However, not all variables contribute equally to the success of the classifier. In our Random Forest model, we have 25 feature variables other than the binary classification variable. It is thereby important to know which components of the model are the most effective in classifying a feature vector to one of the two classes. We shall use the Mean Decrease in Gini measure to determine variable importance.

Gini Impurity

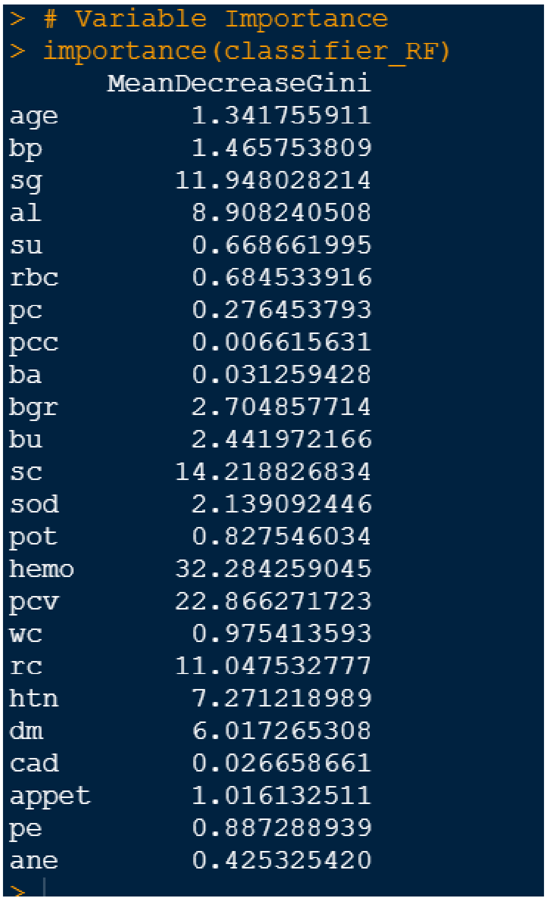
First we explain what is known as the Gini Impurity. It is a metric used in Decision Trees to decide how to split the given data into smaller groups, via determination of the variable and the threshold at every node. It measures how frequently a randomly chosen sample point from the data set will be incorrectly labelled. Gini Impurity reaches zero when all records in a group fall into the same category. This measure can be essentially related to the probability of a new record being incorrectly classified at a given node in a Decision Tree, based on the training data.

Mean Decrease in Gini

We know that Random Forests are a collection of individual Decision Trees. Thus, the Gini impurities at the nodes can be leveraged to calculate Mean Decrease in Gini. It is a measure of variable importance while estimating a target variable. It is the average of a variable's total decrease in node impurity, weighted by the proportion of samples reaching that node in each individual decision tree in the random forest. This effectively measures how important a variable is for estimating the value of the target variable across all of the trees that make up the ensemble. A higher value of Mean Decrease in Gini is equivalent to higher variable importance.

Consider the following snippet:

Figure 11



```
> # Variable Importance
> importance(classifier_RF)
      MeanDecreaseGini
age        1.341755911
bp         1.465753809
sg        11.948028214
al         8.908240508
su         0.668661995
rbc        0.684533916
pc         0.276453793
pcc        0.006615631
ba         0.031259428
bgr        2.704857714
bu         2.441972166
sc        14.218826834
sod        2.139092446
pot        0.827546034
hemo      32.284259045
pcv       22.866271723
wc         0.975413593
rc        11.047532777
htn        7.271218989
dm         6.017265308
cad        0.026658661
appet      1.016132511
pe         0.887288939
ane        0.425325420
>
```

From the above table, it is observed that the variable **hemo** has the highest value of Mean Decrease in Gini, followed by **pcv**, **sc**, **sg** and **rc**. Thus, we can say that these variables are more important than others in classifying patients to two classes – *ckd* and *notckd.*

Note that, referring back to the Figure 4 in Section 7.1, Page 12, we find that **hemo**, **pcv, sg, sc** and **pot** as the variables primarily involved in decision tree construction.

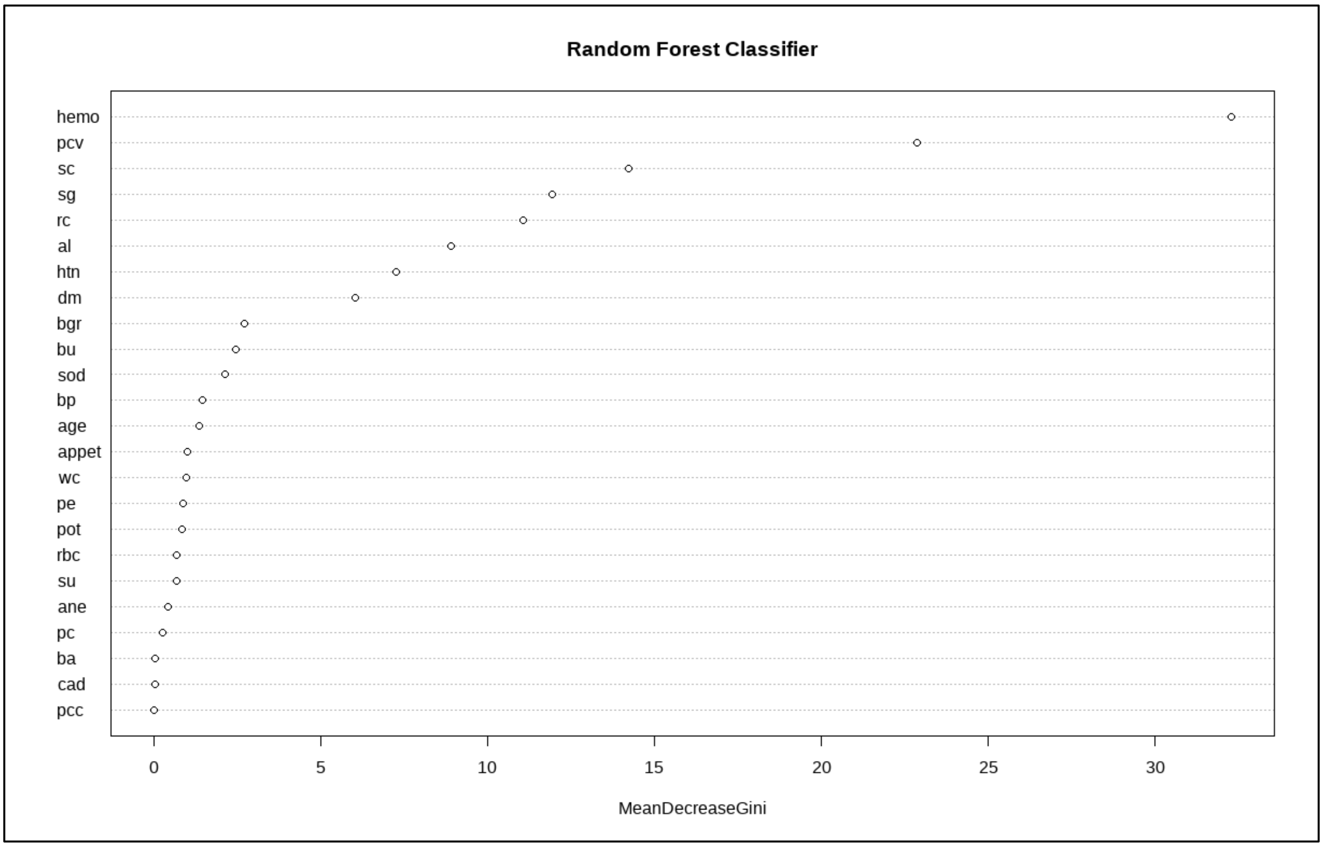
So, the variables **hemo**, **pcv, sg, sc** may be considered to be important variables in our context.

10.1 Variable Importance Plot

Based on the Mean Decrease in Gini values, the variables are sorted and displayed in the Variable Importance Plot. The most important variables to the Random Forest model will be highest in the plot and have the largest Mean Decrease in Gini Values. And, conversely, the least important variable will occupy the lowest position in the plot, and have the smallest Mean Decrease in Gini values.

Consider the following variable importance plot:

Figure 12



From the Variance Important plot, it is apparent that corresponding to **hemo**, the value of the measure is the highest. So, it can be concluded that the most important variable in this classification is **hemo,** followed by **pcv**, **sc** and **sg.**

11. CONCLUSION

From our project, we may conclude that our dataset fits quite well to both the ordinary Decision Tree Classifier as well as the Random Forest. In fact, there is not much difference in the performance of the two models. However, by finer considerations, the Random Forest model performs better than the decision tree classifier. Also, we may conclude that the haemoglobin count of the blood, the packed cell volume, the specific gravity of urine and the serum creatinine content are important considerations with respect to classification of a person as to whether he/she has chronic kidney disease or not.

12. REFERENCES

Below is a list of references we have used for the completion of this project:

- Lecture Notes of Statistical and AI Techniques in Data Mining (MTH552A), instructed by Dr. Amit Mitra
- *An Introduction to Statistical Learning with Applications in R* – Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- *The elements of statistical learning: Data Mining, Inference and Prediction* - T. Hastie, R. Tibshirani and J, Friedman
- *Applied multivariate statistical analysis* - R. A. Johnson and D.W. Wichern