DIABETES PREDICTION

Harshit Mehta(hm23388)

Sanjit Paliwal(Sp42626)

Department of Operation Research and Industrial Engineering

University of Texas at Austin

TABLE OF CONTENTS

Objective:

The utmost aim of the project is to devise a method for selecting the patients for the intervention program to minimise the total cost for all the prediabetics. Another task is to build a model which can predict the patient as diabetic or non-diabetic most accurately and to discuss the distinctive set of characteristics of Type II diabetics.

Description:

In Type II Diabetes, the Insulin hormones are not able to work effectively to convert the glucose we get from the food intake into energy. It is also known as Insulin Resistance. As the Type II Diabetes gets worse, the Pancreas makes less insulin and leads to Insulin deficiency. As per **National Diabetes Statistics Report 2014,** In USA, 86 Million people are prediabetic.

The Intervention program is started by the Health care organisation to reduce the likelihood of Type II Diabetes. Currently, the organisation is selecting patients based on their Fasting Blood Glucose level for the Intervention program. This current practice of selecting patients having FBC greater than 110 incurs total cost $**35,639,660.** Cost structure is mentioned in the Appendix. Our first task is to build a model from the data available to us which can reduce the total cost.

Data Summary:

The data available has 89903 observations and 47 variables including the response variable “Class”. Some of the variables have missing data as well. “NA” represents the missing values in the data. The structure of the original data is as follows:

```
data.frame':   89903 obs. of  47 variables:
 $ Class         : Factor w/ 2 levels "DIABETIC","NON-DIABETIC": 2 2 2 2 2 2 2 1 2 1 ...
 $ ifg           : int  NA NA NA NA NA NA NA NA NA NA ...
 $ igt           : int  NA NA NA NA NA NA NA NA NA NA ...
 $ weight        : num  NA NA NA NA NA NA 70 NA NA NA ...
 $ height        : num  NA NA NA NA NA ...
 $ BMI           : num  NA NA NA NA NA ...
 $ glucose       : num  NA 81 NA NA NA NA 86 109 111 NA ...
 $ A1C           : num  NA NA NA NA NA NA 5.7 NA NA NA ...
 $ insulin       : num  NA NA NA NA NA ...
 $ trig          : num  NA NA NA NA NA NA 209 129 166 NA ...
 $ albumin       : num  NA NA NA NA NA NA 4.4 NA NA NA ...
 $ cholesterol   : num  NA NA NA NA NA NA 194 179 196 NA ...
 $ age           : num  29.9 26.6 43 25.8 53 ...
 $ gender        : int  0 1 0 0 0 0 0 1 1 0 ...
```

Data Analysis:

Number of missing observations per attributes are calculated to check the sparsity of the attributes. All the attributes along with their missing value count is given below:

Class	IFG	IGT	Weight	Height	BMI	Glucose	A1C
0	88203	89774	79384	79384	79834	36539	83667
Insulin	Trig	Albumin	Cholesterol	Age	Gender	Hemoglobin	Bpd
89406	38758	71009	38662	0	0	33278	51244
History	Bp_med	Chol_med	Quicki	Homa_ir	Homa_b	NumDoctorVis	T3
89710	83041	82284	89417	89417	89417	0	84342
T4	TSH	cPept	LDL	HDL	CRP	numDiagnosis	MCV
79510	48098	89877	42979	43066	80004	0	33278
aGAD	Uric	creatinine	WBC	RBC	PLT	numPrescrips	HMT
89903	44249	42800	33278	33278	33278	0	33278
MCH	MCHC	sodium	potassium	Bps	BUN	SocialSecurity	
33278	33278	50538	50539	51244	50716	68843	

Attribute **"aGAD"** has all instances missing, so it won't add any information to the model, hence we removed this attribute. Furthermore, when we ran the J48 model on the original data, the following attributes did not affect the classification accuracy and area under ROC curve of the model, so we removed them.

1). **"ifg"** (Impaired Fasting Glycaemia)

2). **"igt"** (Impaired Glucose tolerance)

3). **"Chol_med"**

4). **"history"**

Data Imputation:

We have done the imputation on the data available after removing the above attributes using the package **"mice"** (**Multivariate imputation by chained equation**) available in **Rstudio** software. The imputed data looks as follows:

```
'data.frame':   89903 obs. of  42 variables:
 $ Class          : Factor w/ 2 levels "DIABETIC","NON-DIABETIC": 2 2 2 2 2 2 2 1 2 1 ...
 $ weight         : num  90 92 112 86 102 ...
 $ height         : num  172 171 180 168 179 ...
 $ BMI            : num  30.1 31.5 33.2 30.3 31.6 ...
 $ glucose        : num  100 81 86 86 91 84 86 109 111 96 ...
 $ A1C            : num  6 5.5 5.5 5.3 5.3 5.3 5.7 5.6 6 5.8 ...
 $ insulin        : num  34.4 13.2 10.8 10.8 5.2 ...
 $ trig           : num  158 102 140 41 124 ...
 $ albumin        : num  4.04 4.6 4.3 4.79 4.2 4.4 4.4 4.5 4.4 4.8 ...
 $ cholesterol    : num  180 233 191 211 220 ...
 $ age            : num  29.9 26.6 43 25.8 53 ...
 $ gender         : int  0 1 0 0 0 0 0 1 1 0 ...
```

Comparison of Model on Original and Imputed Data:

For the comparison between unimputed data and imputed data, we used the **cross validation with N fold=10** for different models. Following are the characteristics of the models we used:

Model	Accuracy rate		Area under ROC curve	
	Original Data	Imputed Data	Original Data	Imputed Data
J48	94.92%	94.965%	0.616	0.624
Logistic Regression	94.91%	94.914%	0.667	0.686
Naïve Bayes	91.58%	91.980%	0.667	0.672
Bagging with 70 iterations	94.53%	95.024%	0.752	0.760
Random_forest with 100 iterations	94.78%	95.032%	0.748	0.754

We have witnessed that for each algorithm, the imputed data has better Classification Accuracy and area under ROC curve, hence we used the Imputed data for our task.

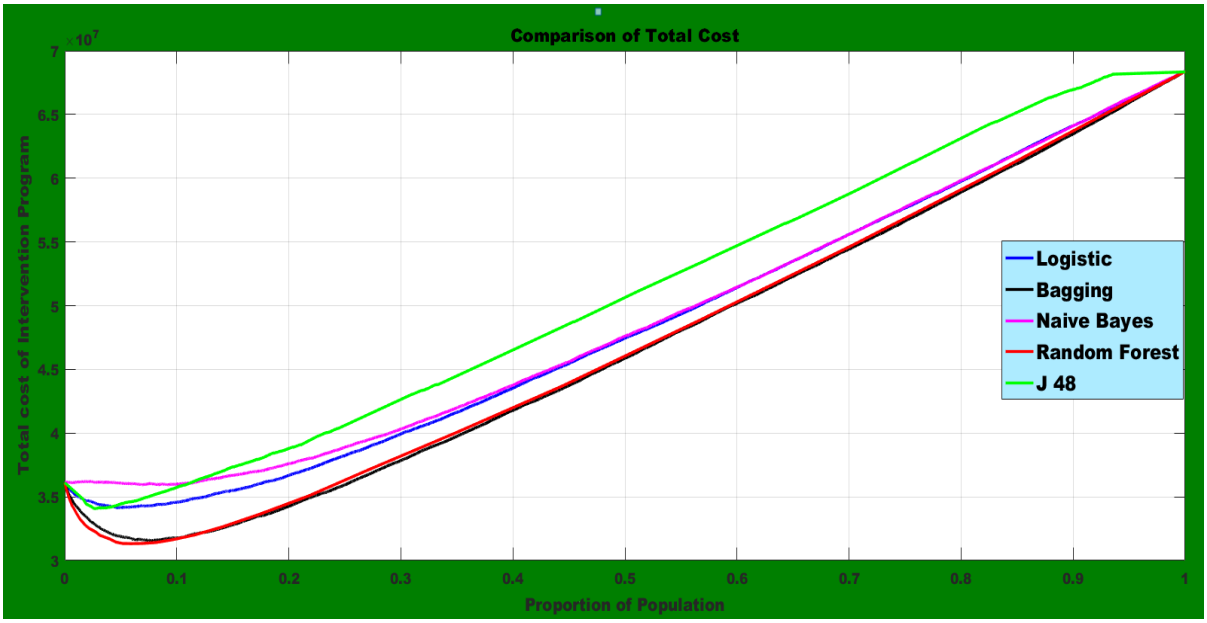
PART 1: Minimising the Total Cost

We have used different algorithms like J48, Naïve Bayes, Logistic Regression, Bagging, Random Forest to estimate the total cost of all the prediabetics. For each model, we have different score threshold at which the Total Cost would be minimum. After evaluating the Total cost for each model, the one with the lowest estimated cost was compared against current practice model.

Model	True Positive	False Positive	False Negative	True Negative	Threshold Probability	% of Pop Targeted	Total Cost
J48	737	1699	3834	83633	0.125	2.7	$34,079,120
Logistic Regression	965	3362	3606	81970	0.129	4.8	$34,133,000
Naïve Bayes	826	5420	3745	79912	0.2397	6.94	$35,943,260
Bagging with 70 iterations	1833	5053	2738	80279	0.1127	7.66	$31,554,080
Random Forest with 100 iterations	1684	3649	2887	81683	0.15	5.93	$31,328,540

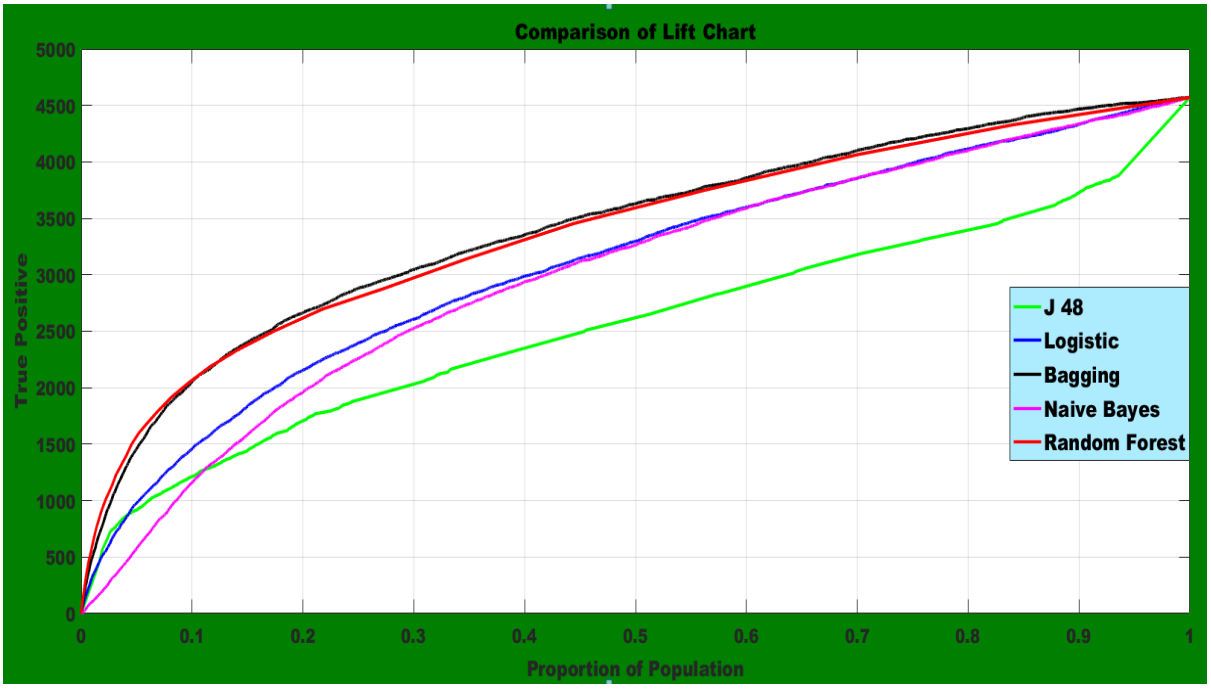
Among the above models, the best one is **Random Forest with 100 iterations** as it has the lowest estimated total cost. The total cost for all the prediabetics according to the current practice is $**35,639,660.** By implementing the Random Forest with 100 iterations model we can save the cost by **$4,311,120** which is equivalent to saving **$ 943.15** per prediabetic. In US, no. of prediabetics is 86 Million. Using the same model, we can save **$81 Million** over the current practice for 86 Million prediabetics.

TOTAL COST CHART



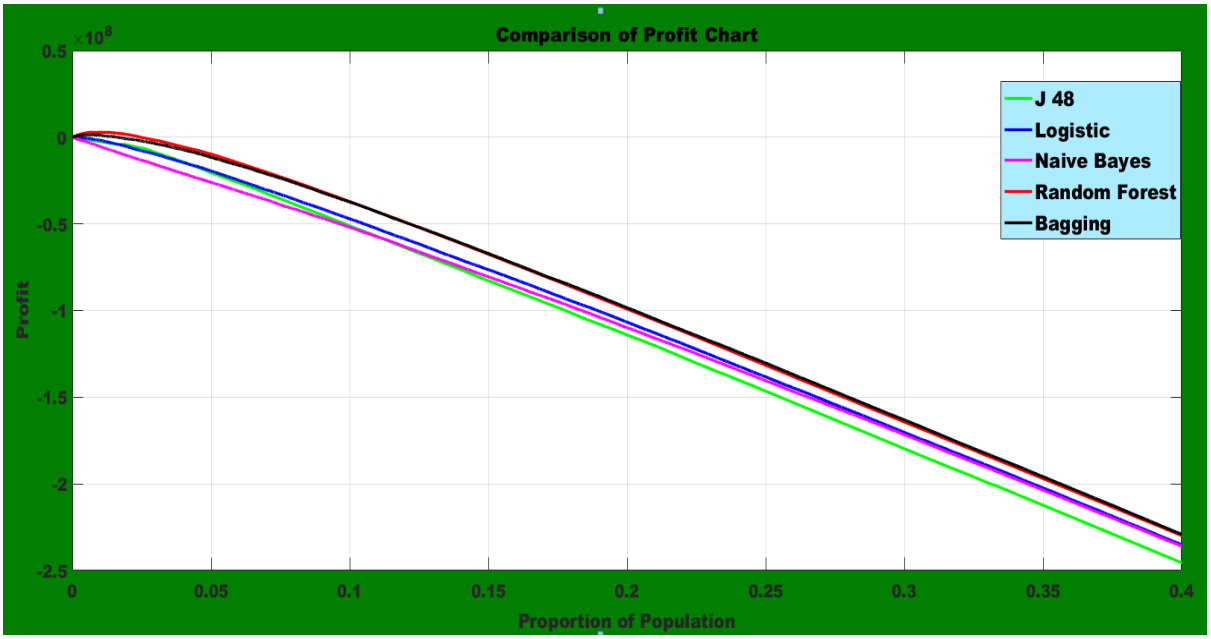
We are getting minimum cost when targeting the 5.93%(0.0593) of Population with the best model.

LIFT CHARTS



Since, we are targeting the top 5.93%(0.0593) of the population, hence Random Forest has the better lift in comparison to other models.

PROFIT CHARTS



Since, we are targeting the top 5.93%(0.0593) of the population, hence Random Forest has the maximum profit in comparison to other models.

Clearly, from the all three curve we can say that **Random Forest with 100 iterations** is the best model.

PART 2: Selecting the Most Accurate Model for Prediction

To check the out of sample accuracy of the different model we need to split the entire data into training set and testing set. Since, the number of diabetes instances are less, hence we need to split the data using **StratifiedRemoveFolds** in "weka" software. We use the number of folds equal to 3, so our training set has 70% of the entire data.

On the training set we used different models with **10-Fold Cross-Validation** and compare their Classification Accuracy. The model with the best Accuracy rate was compared against the current practice. Weka calculated the accuracy rate at a score threshold of 0.5, so we also varied the score threshold to see whether there is improvement in the accuracy rate at different score threshold. We observed increase in accuracy rate at different threshold as mentioned below.

Training Set:

Model	Accuracy (Threshold =0.5)	Maximum Accuracy (threshold)
J48	94.05%	94.344% (0.93)
Naïve bayes	91.69%	94.68% (1)
Logistic Regression	94.96%	94.986% (0.36)
Random Forest with 100 Iterations	95.05%	95.248% (0.31)

Bagging with 60 iterations	95.12%	95.151% (0.41)
K Nearest Neighbors with K=5	94.85%	94.959%(0.80)

Testing Set:

Model	Accuracy (Threshold =0.5)	Maximum Accuracy (threshold)
J48	94.02%	94.23% (0.93)
Naïve bayes	91.87%	94.614% (1)
Logistic Regression	94.82%	94.81%(0.36)
Random Forest with 100 Iterations	94.88%	95.188% (0.31)
Bagging with 60 iterations	94.99%	95.018% (0.41)
K Nearest Neighbors with K=5	94.70%	94.873%(0.80)

Clearly, The **Random Forest with 100 iterations at threshold 0.31** gave the best Training and testing set accuracy rate. Since, the decline in accuracy rate from training to testing set is not very high, it implies that overfitting is absent also. The error rate of Random Forest on testing set is **4.812%.**

Comparison of Best Model with Current Practice:

Using the Current Practice, where we used only Glucose level as the classifier, the confusion Matrix of the testing set is:

True Positive	238
False Positive	1386
False Negative	1312
True Negative	27032
Accuracy rate	91.00%

The Accuracy rate of the current practice is **91%** and corresponding error rate is **9%** Hence, we recommend to use **Random Forest with 100 iterations at threshold 0.31.**

Feature Addition:

We have added the following attributes in the training set and ran Random Forest with 100 iterations on the dataset.

1) **K_5_prediction:**
   Probabilities of response variable calculated using K Nearest Neighbors.
2) **Naïve_Bayes:**
   Probabilities of response variable calculated using Naïve Bayes.

3) **Logistic_prediction:**
   Probabilities of response variable calculated using Logistic Regression.

The Classification Accuracy rate on the training set and testing set by adding the above three attributes in the dataset achieved is **95.21%** and **95.06%** at threshold score of **0.33** respectively. However, this Classification accuracy rate is still less than our Best model.

Feature Selection using Wrapper approach:

For features selection, we used the "**Boruta"** Package in **Rstudio**. It uses the wrapper approach to calculate the importance of the attributes. Attribute importance is calculated using the Z-Score which is proportional to the decrease in model accuracy when an attribute is removed. We keep the best 10 attributes in our dataset and ran the Random Forest with 100 iterations algorithm to check whether we can improve the accuracy rate of the model. The 10 important features are:

1. Age
2. numDoctorVis
3. numDiagnosis
4. numPrescrips
5. glucose
6. A1c
7. Uric
8. Creatinine
9. HDL
10. Gender

The classification accuracy increased to **95.48%** on training set at threshold score of **0.47**. The testing set accuracy at the same threshold score is **95.385%.** By keeping the ten most important attributes in the dataset only we improved the accuracy by **0.232%** and **0.197%** on training and testing set respectively over our best model.

Part 3: Selecting the Distinctive characteristics

To find out the set of highly predictive characteristics, we used "**Decrease in area under the ROC curve**" as the key factor. We calculate decrease in the ROC curve area for each attribute before and after removing it. The attributes which lead to decrease in area are classified as important features. We then kept those attributes only in the dataset and ran J48 model to get the decision rules which gave us the pure nodes.

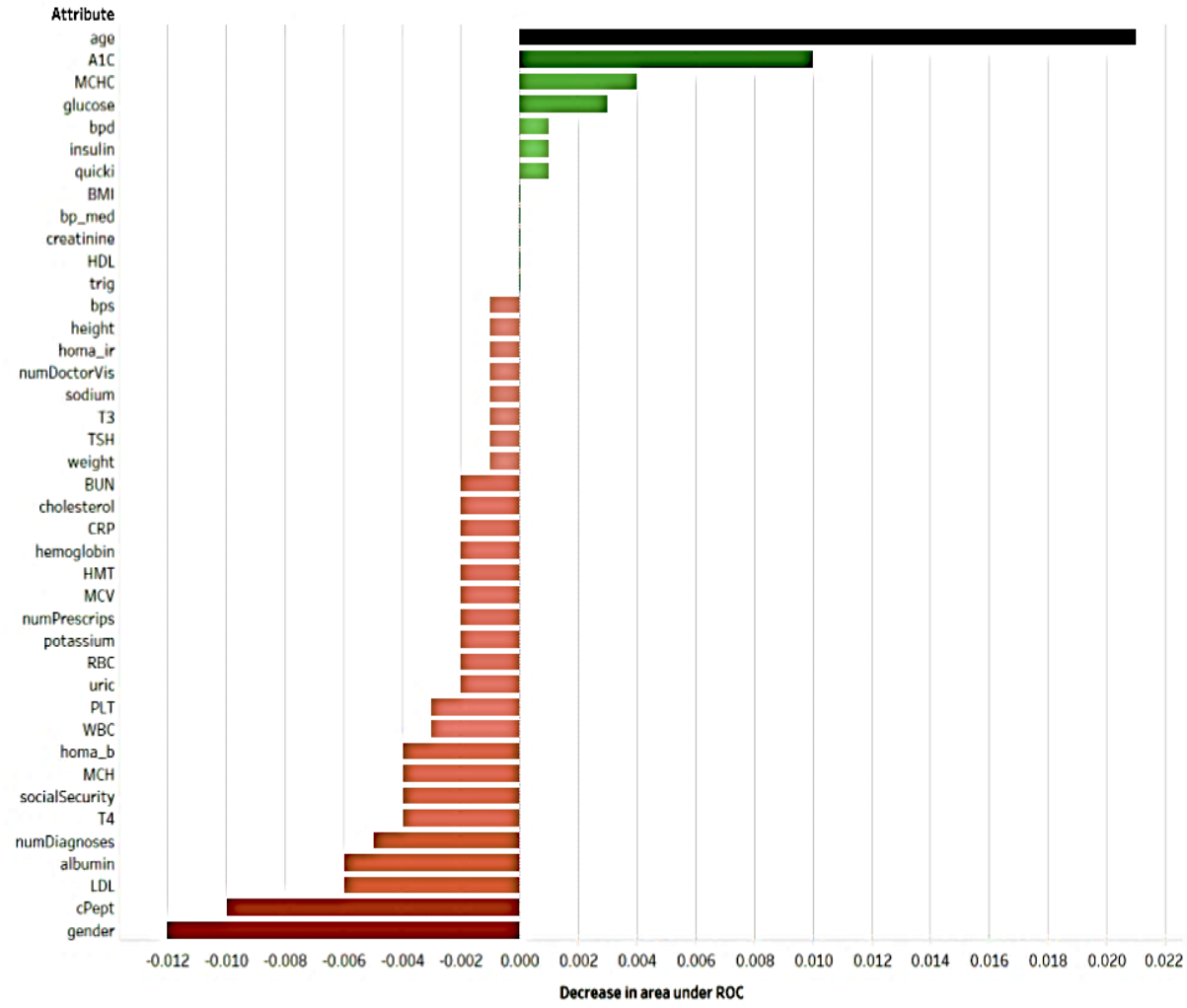
Decision Rules:

1) If (**A1C > 6.2** & **Age > 56.6744** & **glucose > 109** & **AIC > 6.1**) then probability of Diabetes is (**146/200**) = **0.73.**

2) If (**MCHC > 33.375** & **(6.1 < A1C < = 6.2)** & **Age > 56.6744** & **glucose > 109**) then probability of Diabetes is (**43/55**) = **0.78**.

Features Visualisation based on decrease in area under ROC:

To illustrate the highly predictive attributes, we construct a histogram showing the decrease in area under the ROC curve for each attribute.



Summary:

There were different objectives within the project. The first one was to minimize the total cost for all the prediabetics. For this the best model was **Random Forest (with 100 iterations)** that provided a cost saving of **$4,311,120,** which is equivalent to **12.1%** saving. The second one was to accurately predict whether a patient will have diabetes. For this the best model was **Random Forest** (with **100** iterations at threshold score of **0.31**) with a test accuracy rate of **95.188%.** The last objective was to find the most important predictors as well as finding the characteristics of patients with diabetes. For this to find the most important predictors we used **"Decrease in area under ROC curve"** as the key factor and to find the distinctive set of characteristics and decision rules of patients with diabetes, a **J48 tree** model was built by only keeping the important attributes in the dataset.

Appendix

A) Total Cost formula for a given Predictive Model

To calculate total cost of the prediabetics for a given model we used the following formula:

Total Cost = True Positive (0.6x$600 + 0.4x$8500) + False positive ($600) + False Negative ($7900)

Here the Positive class represents the patient being Type II Diabetic

B) R Studio code for Data Imputation and Feature Selection

Data Imputation

```
# code for data imputation
library(RCpp)
library(mice)
imputed-mice (data, method='cart', m=1)
```

Feature Selection

```
# code for feature selection using wrapper search
library(ranger)
library(Boruta)
important-Boruta(Class~.,data)
plot(important)
```

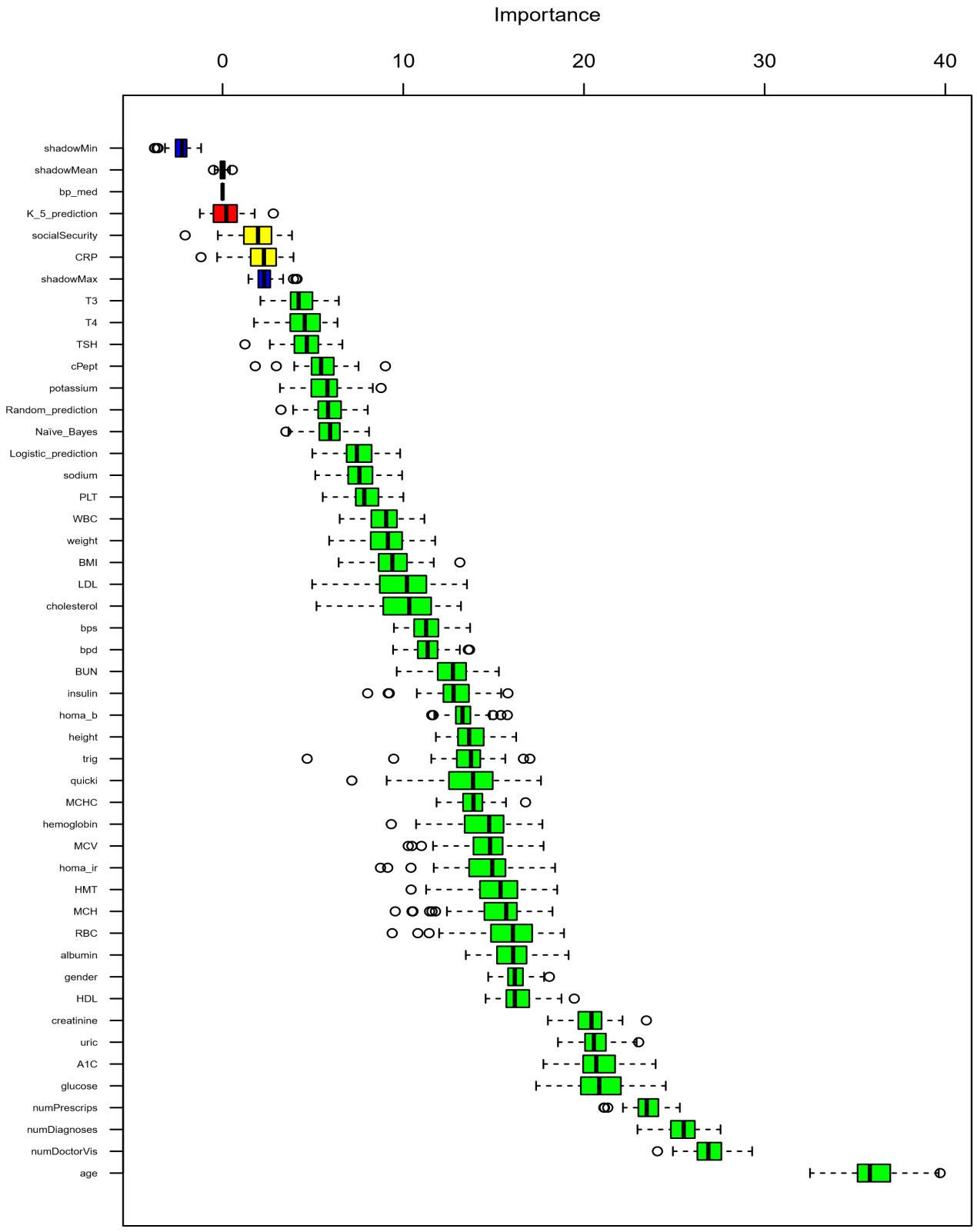
C) Attribute Description

Attribute	Type	Description
Class	Nominal	"Diabetic" or "Non-Diabetic"
ifg	Nominal	Impaired Fasting Glycaemia
igt	Nominal	Impaired Glucose Tolerance
weight	Numeric	Weight in kg
height	Numeric	Height in cm
BMI	Numeric	Body Mass Index
glucose	Numeric	Blood glucose level in mg/dL
A1C	Numeric	Glycated Hemoglobin in %
insulin	Numeric	Blood insulin level in mcu/mL
trig	Numeric	Triglyceride level in mg/dL
albumin	Numeric	albumin level in g/dL
cholesterol	Numeric	Cholesterol level in mg/dL
age	Numeric	In Years
gender	Nominal	
bps	Numeric	Beraprost Sodium

bpd	Numeric	Borderline personality disorder
history	Nominal	
bp_med	Nominal	
chol_med	Nominal	
quicki	Numeric	Quantitative insulin sensitivity check index
homa_ir	Numeric	Homeostatic Model Assessment of Insulin Resistance
homa_b	Numeric	Homeostatic Model Assessment of Beta cell function
numDoctorVis	Numeric	Number of Doctor visits
numDiagnoses	Numeric	Number of Diagnosis
numPrescrips	Numeric	Number of Prescriptions
socialSecurity	Numeric	
T3	Numeric	Triiodothyronine level in nano moles per litre
T4	Numeric	Thyroxin Harmone level in microgram/dL
TSH	Numeric	Thyroid Stimulating Harmone in mIU/L
cPept	Numeric	C peptide
LDL	Numeric	Low density Lipoprtein level in mg/dL
HDL	Numeric	High density Lipoprotein level in mg/dL
CRP	Numeric	C-Reactive Protein level in mg/dL
aGAD		Glutamic acid decarboxylase antibody test (measures whether the body is producing a type of antibody which destroys its own GAD cells
uric	Numeric	Uric acid in the blood in mg/dL
hemoglobin	Numeric	Haemoglobin level in g/L
WBC	Numeric	White Blood Cell (10^9 per litre)
RBC	Numeric	Red Blood Cell (10^6 per microlitre)
PLT	Numeric	Platelet Count in Thousands per microLitre
HMT	Numeric	Hexamethylenetetramine
MCV	Numeric	Mean corpuscular Volume
MCH	Numeric	Mean corpuscular hemoglobin
MCHC	Numeric	Mean corpuscular hemoglobin concentration
sodium	Numeric	Blood sodium level in milli equivalents per Litre
potassium	Numeric	Blood potassium level in mmol/L
BUN	Numeric	Blood Urea nitrogen level in mg/dL
creatinine	Numeric	Creatinine level in mg/dL

D) Wrapper Approach Plot

Below is the plot made using Boruta package for feature selection using wrapper approach when criteria is maximizing accuracy



Tree Visualisation for Decision Rules