DIAGNOSIS OF DIABETES USING CLASSIFICATION MINING TECHNIQUES

Problem Statement

Design of prediction model for diabetes in the Females by minimizing the cost.

Diabetes

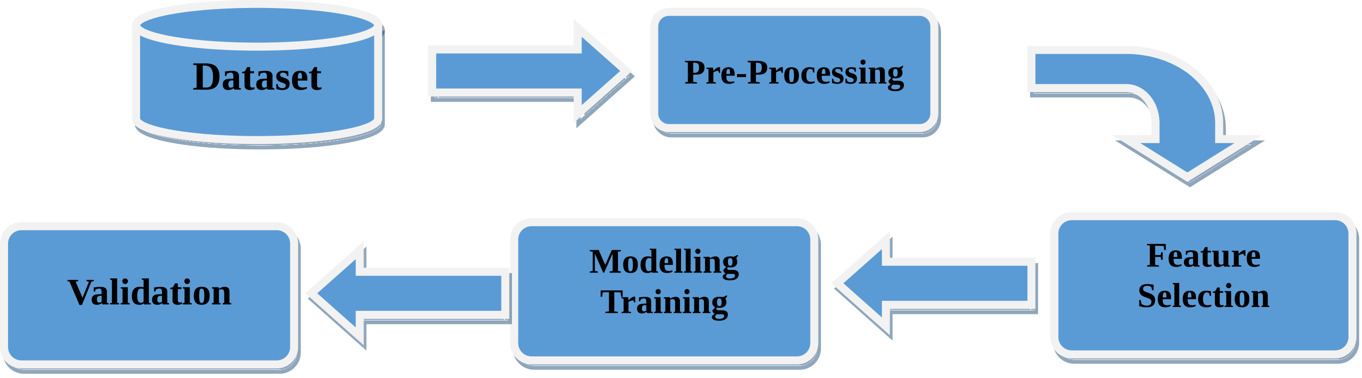
Diabetes is a disease that occurs when the insulin production in the body is inadequate or the body is unable to use the produced insulin in a proper manner, as a result, this leads to high blood glucose. The body cells break down the food into glucose and this glucose needs to be transported to all the cells of the body. Any change in the production of insulin leads to an increase in the blood sugar levels and this can lead to damage to the tissues and failure of the organs. Generally, a person is considered to be suffering from diabetes, when blood sugar levels are above normal (4.4 to 6.1 mmol/L).

Effects of diabetes have been reported to have a more fatal and worsening impact on women than on men because of their lower survival rate and poorer quality of life. WHO reports state that almost one – third of the women who suffer from diabetes have no knowledge about it. The effect of diabetes is unique in case of mothers because the disease is transmitted to their unborn children. Strokes, miscarriages, blindness, kidney failure and amputations are just some of the complications that arise from this disease.

Nowadays, large amount of information is collected in the form of patient records by the hospitals. Knowledge discovery for predictive purposes is done through data mining, which is a analysis technique that helps in proposing inferences. This method helps in decision-making through algorithms from large amounts of data generated by these medical centres. Considering the importance of early medical diagnosis of this disease, data mining techniques can be applied to help the women in detection of diabetes at an early stage and treatment, which may help in avoiding complications.

Methodology

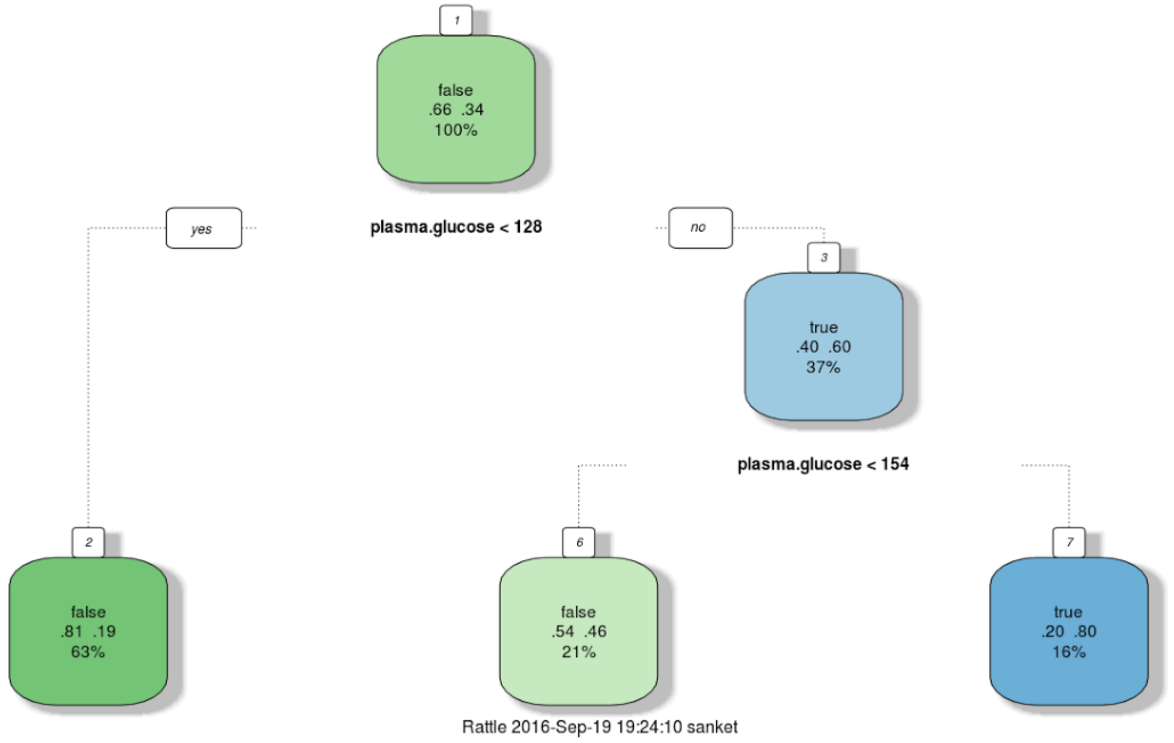
The present work intends to create a mining model based on Decision tree classification algorithm in order to provide a simpler solution to the problem of diagnosis of diabetes in women.



Modelling Cycle

Decision Trees

Decision tree is a tree structure, which is in the form of a flowchart. It is used as a method for classification and prediction with representation using nodes and internodes. The root and internal nodes are the test cases that are used to separate the instances with different features. Internal nodes themselves are the result of attribute test cases. Leaf nodes denote the class variable.



Decision Tree with Rpart

Decision tree provides a powerful technique for classification and prediction in Diabetes diagnosis problem. Various decision tree algorithms are available to classify the data, including ID3, C4.5, J48, CART. In this paper, J48 decision tree algorithm [10] has been chosen to establish the model. Each node for the decision tree is found by calculating the highest information gain for all attributes and if a specific attribute gives an unambiguous end product (explicit classification of class attribute), the branch of this attribute is terminated and target value is assigned to it.

Dataset

1 . Dataset Description:

Dataset	Number Of Attributes	Number of Instances
Women's diabetes dataset Provided By client	8	768

2. Attributes Description:

Attribute	Relabelled values
1. Number of times pregnant	Preg
2. Plasma glucose concentration	Plas
3. Diastolic blood pressure (mm Hg)	Pres
4. Triceps skin fold thickness (mm)	Skin
5. 2-Hour serum insulin	Insu
6. Body mass index (kg/m2)	Mass
7. Diabetes pedigree function	Pedi
8. Age (years)	Age
9. Class Variable (True or False)	Class

Pre-Processing:

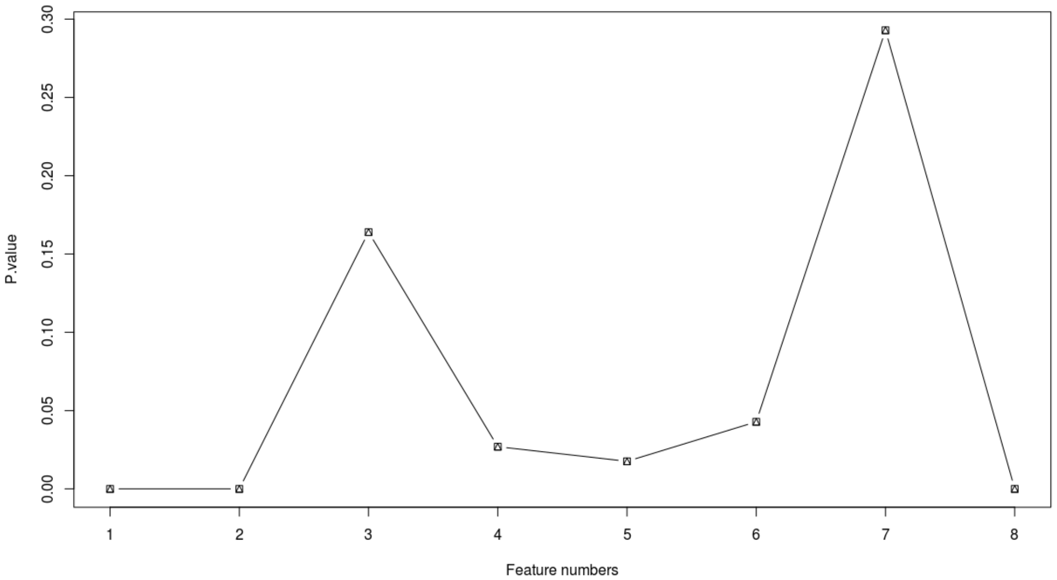
As the Dataset was having many missing values pre-processing and cleaning of the data was done. In this step we replaced the missing values with the median.

For example, Blood pressure was having many zero entries, as biologically blood pressure of anyone cannot be zero we replaced it with median value from the data.

Feature Selection

Based on the Hypothesis test of features with the class variable (True or false), we selected the optimum features from the given 8 features.

Null Hypothesis: Feature does not contain any information about the desired class.
Alternate Hypothesis: Feature contains information about the desired class variable.



P-values of the features

Lower the P-value means we are Rejecting the Null hypothesis and accepting Alternate hypothesis. Based on the above P-value we selected the three features Pregnancies, Plasma glucose levels and Age based on the P-value.

Feature	P-Value
1. Pregnancies	8.648349e-08
2. Plasma glucose	4.295488e-11
8. Age	2.306982e-10

Data Modelling

We used 3 different decision tree algorithms as stated below and compared their Cost, Accuracy and different validation parameters. In each data technique mentioned below we used 5-fold cross validation technique.

1. C5.0
2. RPART
3. J48

Table showing performance:

Sr.No	Models	Cost	Kappa value	Accuracy (%)
1	C5.0	1100	0.6267	82.35
2	RPART	1500	0.5022	76.47
3	J48	1200	0.4017	73.52

Result Analysis

Based on the cost provided by the client we are using the model created with C5.0 with the below characteristics.

C5.0 Algorithm details:

Sr.No	Models	Cost	Kappa value	Accuracy (%)
1	C5.0	1100	0.6267	82.35

Predicted Matrix:

	Predicted Cases		
Actual Cases		**Diabetes**	**No Diabetes**
	Diabetes	20	7
	No Diabetes	5	36

Cost Matrix:

	Predicted Cases		
Actual Cases		**Diabetes**	**No Diabetes**
	Diabetes	0	50
	No Diabetes	150	0

Annexure

Code:

```
library(caret)

#creating dataset
link = "/home/sanket/Sanket/Praxis/R_study/DM1
Assignment/Custom_Diabetes_Dataset.csv"
dataset = read.csv(link, header = T )
dataset.clean = dataset

#Cleaning the dataset
dataset.clean[dataset.clean$blood.pressure == 0,3] = median(dataset.clean$blood.pressure)
dataset.clean[dataset.clean$triceps.skin.thickness == 0,4] =
median(dataset.clean$triceps.skin.thickness)
dataset.clean[dataset.clean$plasma.glucose == 0,2] = median(dataset.clean$plasma.glucose)
dataset.clean[dataset.clean$bmi == 0,6] = median(dataset.clean$bmi)

#creating trainset and testset
trainset = dataset.clean[1:700,]
testset = dataset.clean[701:768,]

names(trainset)

#featrure selection based on p-value
P_val = c()
for (j in 1:8)
{
  a = summary(table(dataset.clean[,c(9,j)]))
  P_val =rbind(P_val,a$p.value)
}
print(P_val)

# modeling
Cost_matrix = c()
Accuracy= c()
Kappa_value = c()
models = c("C5.0","rpart", "J48")
for (i in models)
{
  cntrl=trainControl(method = "cv",number=5)
  model_C50_car=train(diabetes~ pregnancies + plasma.glucose + age,data=trainset,method
= "C5.0" ,trControl=cntrl,metric="Kappa")
  #model_C50_car=train(diabetes~ plasma.glucose + age + bmi + blood.pressure +
diabetes.pedigree,data=trainset,method = i ,trControl=cntrl,metric="Kappa")
  pred_C50_car=predict(model_C50_car,newdata = testset)
  con = confusionMatrix(pred_C50_car,testset$diabetes)
  cost = con$table[2] * 150 +  con$table[3] * 50
  Cost_matrix = rbind(Cost_matrix, cost)
 Accuracy = rbind(Accuracy,con$overall[1])
  Kappa_value = rbind(Kappa_value,con$overall[2])
}
Compare_Matrix = data.frame(Models =
c("C5.0","RPART","J48"),Cost_matrix,Kappa_value,Accuracy)
print(Compare_Matrix)
```