Machine Learning for Fertility Diagnosis

Medea

5/12/2021

Contents

1 Introduction

The topic of this study is trying to predict fertility : normal or altered. We have used data from https://archive.ics.uci.edu/ml/datasets/Fertility, we downloaded data, added header & save as csv. The modified version is now located on the github. This is location from where we will access it throughout the project.

```
if(!require(tidyverse)) install.packages("tidyverse",
                                         repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret",
                                     repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table",
                                          repos = "http://cran.us.r-project.org")
if(!require(rvest)) install.packages("rvest",
                                     repos = "http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart",
                                     repos = "http://cran.us.r-project.org")
if(!require(rpart.plot)) install.packages("rpart.plot",
                                          repos = "http://cran.us.r-project.org")


library(caret)
library(data.table)
library(tidyverse)
library(rvest)
library(rpart.plot)

url_csv <- paste0("https://raw.githubusercontent.com/medeag/fertility-capstone/",
                  "main/dataset/fertility_diagnosis.csv")
diagnosis <- read.csv(url_csv, header = TRUE)
```

We first confirm how many samples we have in the dataset and what are variable names.

```
dim(diagnosis)
```

```
## [1] 100  10
```

```
names(diagnosis)
```

```
##  [1] "Season"               "Age"                  "Childish_Diseases"
##  [4] "Trauma"               "Surgeon"              "Fevers_Last_Year"
##  [7] "Alcohol_Consumption" "Smoking"              "Sitting_Per_Day"
## [10] "Output"
```

So we have 100 samples and 10 variables. Here we copy descriptions of variables from archive.ics.uci.edu.

1. Season: in which the analysis was performed. 1) winter, 2) spring, 3) summer, 4) fall. (-1, -0.33, 0.33, 1)
2. Age: at the time of analysis. 18-36 (0, 1)
3. Diseases: Childish diseases (i.e., chicken pox, measles, mumps, polio) 1) yes, 2) no. (0, 1)
4. Trauma: Accident or serious trauma 1) yes, 2) no. (0, 1)
5. Surgeon: Surgical intervention 1) yes, 2) no. (0, 1)
6. Fevers_Last_Year: High fevers in the last year 1) less than three months ago, 2) more than three months ago, 3) no. (-1, 0, 1)
7. Alcohol_Consumption: Frequency of alcohol consumption 1) several times a day, 2) every day, 3) several times a week, 4) once a week, 5) hardly ever or never (0, 1)
8. Smoking: Smoking habit 1) never, 2) occasional 3) daily. (-1, 0, 1)
9. Sitting_Per_Day: Number of hours spent sitting per day ene-16 (0, 1)
10. Output: Diagnosis normal (N), altered (O)

Before starting any analysis let's review how many altered & normal cases do we have.

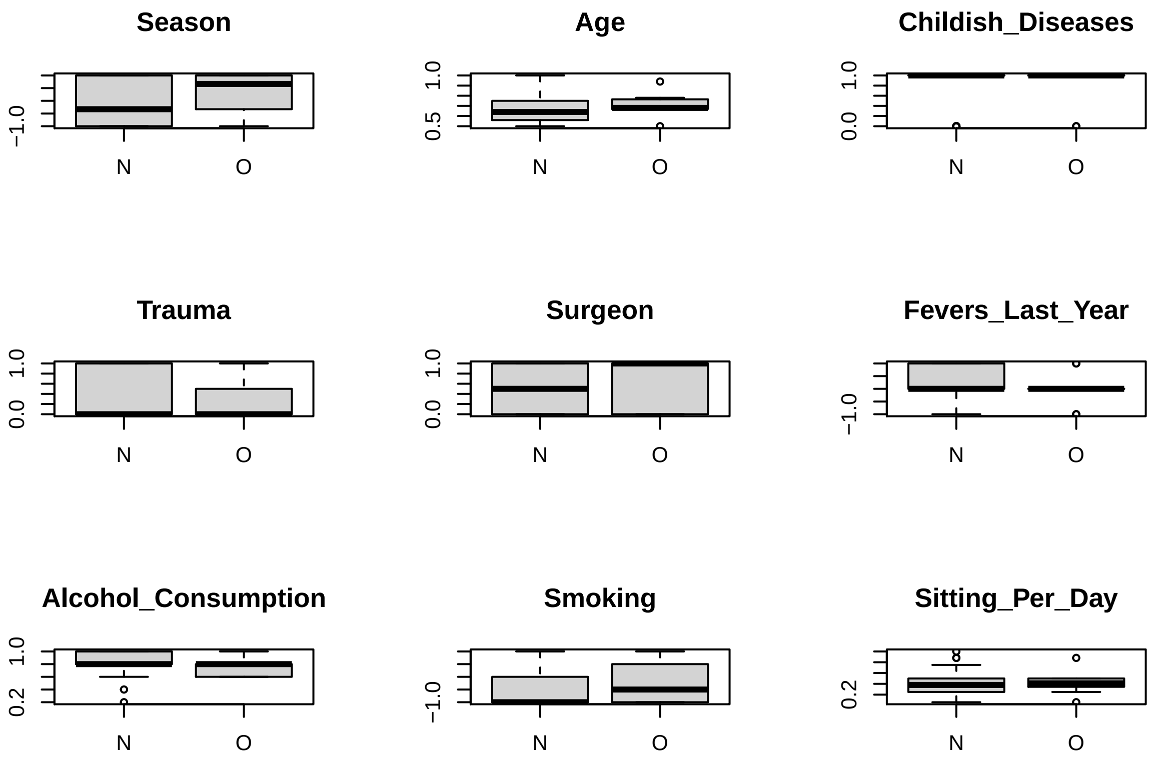
```
table(diagnosis$Output)
```

```
##
##  N  O
## 88 12
```

So we have 88 normal and 12 altered cases.

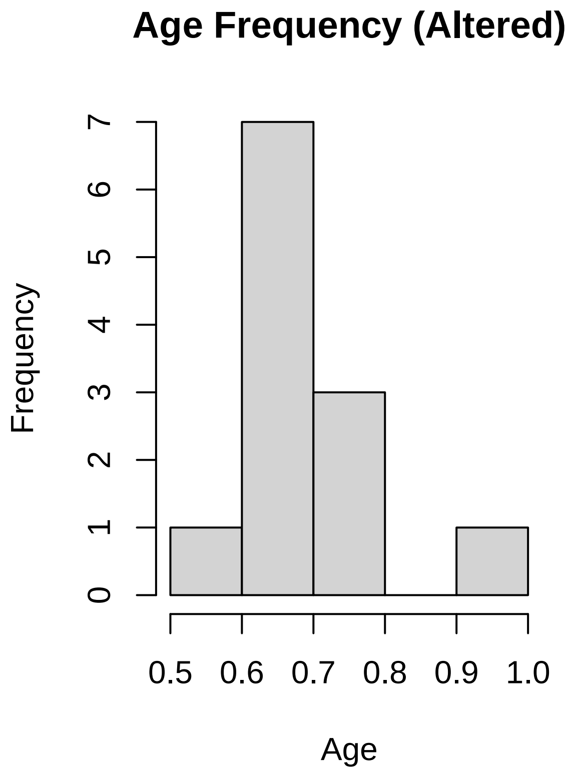
We use boxplot to see how each variable relates to normal/altered cases.

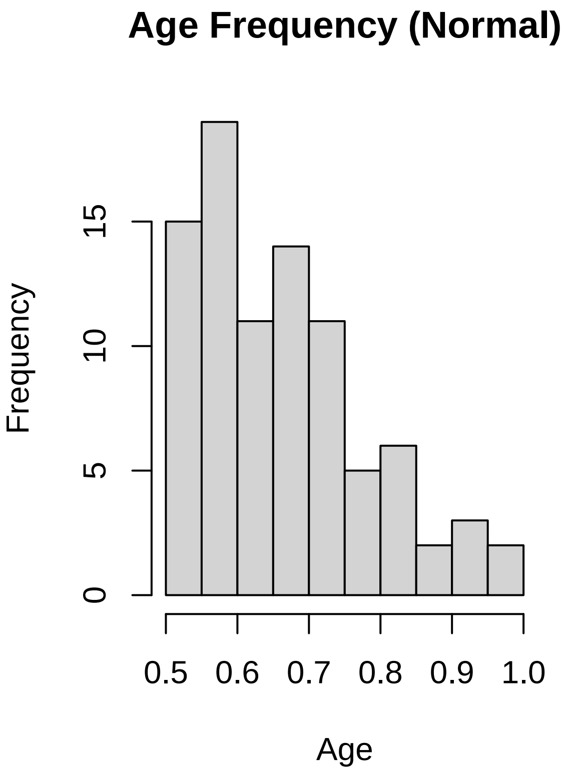
```
# boxplot for each column
par(mfrow = c(3, 3))
for(i in 1:9){
    boxplot(split(diagnosis[,i], diagnosis$Output), main=names(diagnosis)[i])
}
```



Interesting take away from these graphs is that altered cases cluster more in the upper age limit than normal ones. Let's review age frequency in dataset.

```
par(mfrow = c(1, 2))
diagnosis_o <- diagnosis %>% filter(Output==’O’)
hist(diagnosis_o$Age, main = "Age Frequency (Altered)", xlab = "Age")
diagnosis_n <- diagnosis %>% filter(Output==’N’)
hist(diagnosis_n$Age, main = "Age Frequency (Normal)", xlab = "Age")
```





2 Analysis

Our goal is to see if we can predict results using different kind of methods. For simplicity sake we will use following methods:

1. Linear Discriminant Analysis (LDA)
2. K-NN
3. Decision Trees (we will draw the decision tree here)
4. Random Forest (we will list most important predictors)

We will use train method from caret package.

But first we need to divide dataset into the train & test sets, we will divide data set into two parts: 70% for training and 30% for test. We considered to use 90%/10% but since in that case test set was a really small we discarded the idea.

Please take in consideration that although analysis was performed on “R version 4.0.2 (2020-06-22), we have not used sample.kind=”Rounding" on the seed, since we find it redundant to be compatible with 3.5

```
set.seed(197379245)

y <- diagnosis$Output
x <- diagnosis[-10]

test_index <- createDataPartition(y, times = 1, p = 0.3, list = FALSE)
```

```
test_set_x <- x[test_index,]
test_set_y <- y[test_index]
train_set_x <- x[-test_index,]
train_set_y <- y[-test_index]
# number of case: normal vs altered in train set
table(train_set_y)
```

```
## train_set_y
##  N  O
## 61  8
```

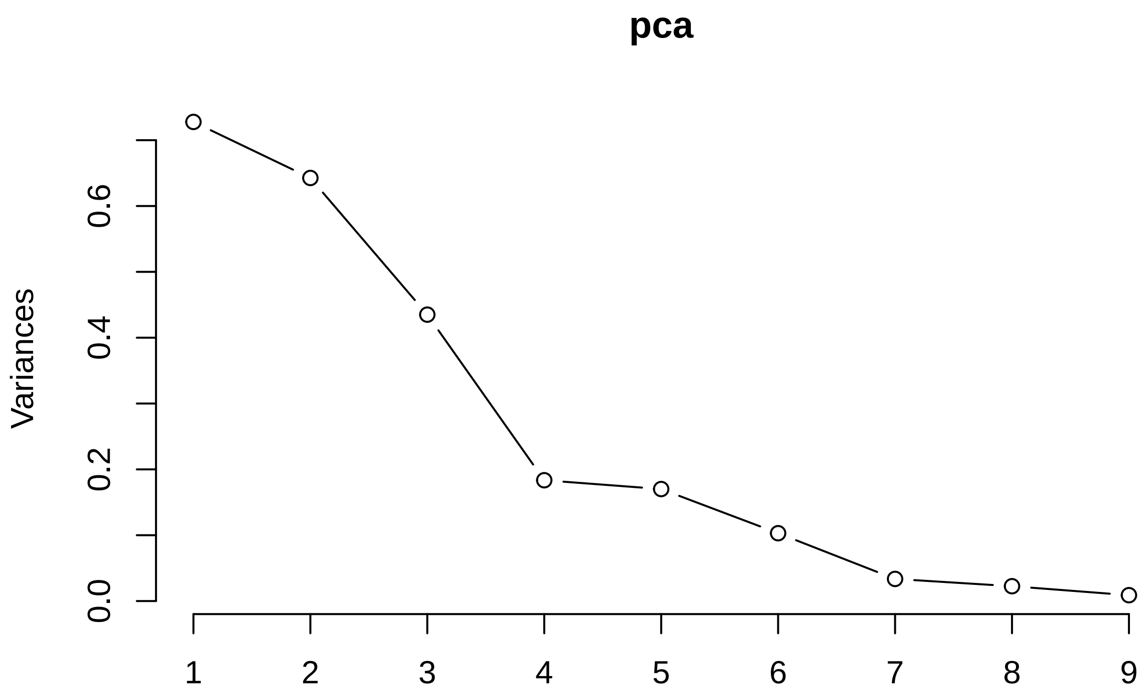
Before we move forward, we need to answer the question: can we reduce the number of features without losing much of the variance in the data?

Let's perform the principle component analysis to check.

```
tmp <- train_set_x
pca <- prcomp(tmp)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4    PC5     PC6     PC7
## Standard deviation     0.8531 0.8016 0.6596 0.42836 0.4124 0.32100 0.18328
## Proportion of Variance 0.3127 0.2761 0.1870 0.07885 0.0731 0.04428 0.01444
## Cumulative Proportion  0.3127 0.5889 0.7758 0.85467 0.9278 0.97205 0.98649
##                            PC8     PC9
## Standard deviation     0.15009 0.09445
## Proportion of Variance 0.00968 0.00383
## Cumulative Proportion  0.99617 1.00000
```

```
plot(pca, type = "l")
```



So 97% of the variance in the data can be described with 6 components. That doesn't sounds like much of an improvement.

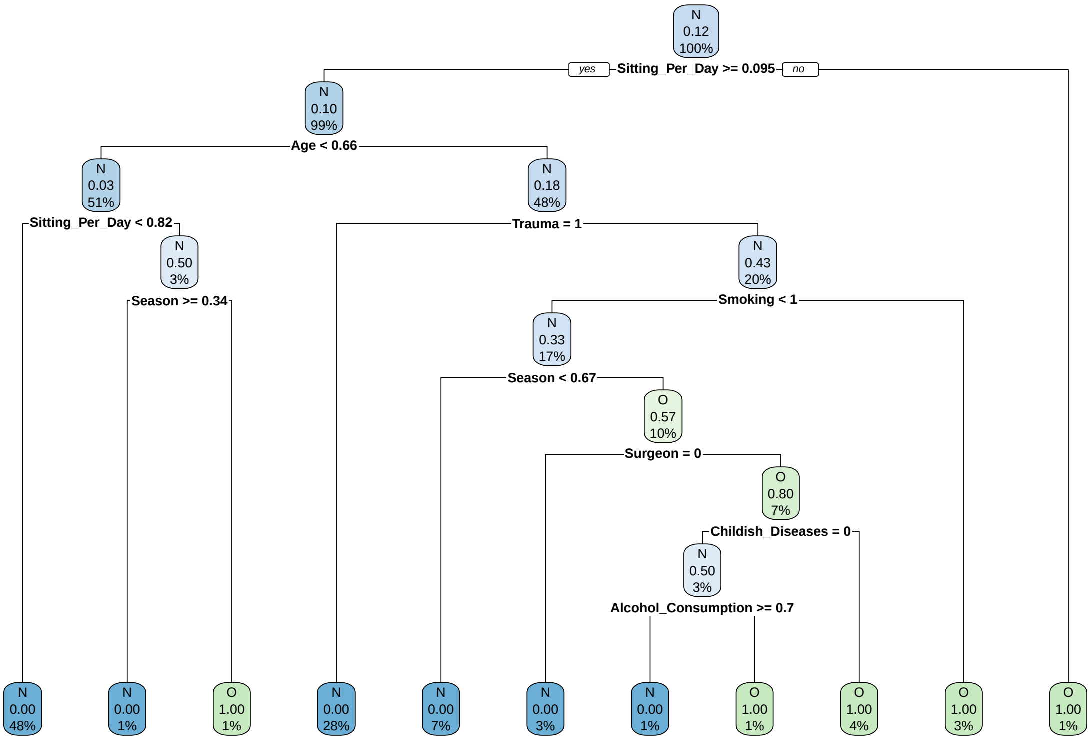
Now it is time to train our models.

```
# LDA
control <- trainControl(method = "cv", number = 5)

train_lda <- train(train_set_x, train_set_y,
                   method = "lda",
                   trControl = control)

# k-Nearest Neighbors
train_knn <- train(train_set_x, train_set_y,
                   method = "knn",
                   trControl = control)

#Decision Tree
train_rpart <- train(train_set_x, train_set_y,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0, 0.05, len = 25)),
                     control = rpart.control(minsplit = 1, minbucket = 1))
rpart.plot(train_rpart$finalModel)
```



```r
#Random Forest
train_rf <- train(train_set_x, train_set_y,
                  method = "rf",
                  trControl = control)

varImp(train_rf)
```

```
## rf variable importance
##
##                     Overall
## Age                  100.00
## Sitting_Per_Day       78.62
## Trauma                43.22
## Season                29.12
## Smoking               27.87
## Fevers_Last_Year      26.81
## Alcohol_Consumption   25.48
## Surgeon               11.90
## Childish_Diseases      0.00
```

Now it is time to do predictions. Please note that we tried to tune decision tree to create the best model.

```r
# Evaluate LDA model on test data
predictions <- predict(train_lda, test_set_x)
results <- tibble(method = "LDA",
                 accuracy = (mean(predictions==test_set_y)))


# Evaluate KNN model on test data
predictions <- predict(train_knn, test_set_x)
results <- results %>% add_row(method = "K-NN",
                               accuracy = (mean(predictions==test_set_y)))

# Evaluate Decision Tree on test data
predictions <- predict(train_rpart, test_set_x)
results <- results %>% add_row(method = "Decision Tree",
                               accuracy=(mean(predictions==test_set_y)))

# Evaluate Random Forest model on test data
predictions <- predict(train_rf, test_set_x)
results <- results %>% add_row(method = "Random Forest",
                               accuracy=(mean(predictions==test_set_y)))
```

3 Results

Below are listed the actual results we got

```
results %>% knitr::kable(digits = 3)
```

method	accuracy
LDA	0.806
K-NN	0.871
Decision Tree	0.774
Random Forest	0.806

The best method turns out to be K-NN.

4 Conclusions

Although we can predict fertility reasonably accurately, a big limitation of the study is a small sample size. It would be nice if in the future we are able to find a larger sample and with additional features, e.g. weight and genetic information. We can also try to combine/assemble several methods to see if we get better results.