Diagnosing PCOS

Zeba Khadhijah

2023-03-01

About the Data

Polycystic ovary syndrome (PCOS) is a common condition that affect women and are characterized by having two or more of the following features: irregular periods, excess male hormones that may lead to excess facial and body hair growth, polycystic ovaries (enlarged ovaries containing fluid filled sacks called follicles).

The dataset used in this project is contains all physical and clinical parameters to determine PCOS and infertility related issues. The data has been collected from 10 different hospital across Kerala,India and is available to access freely from https://www.kaggle.com/datasets/prasoonkottarathil/polycystic-ovary-syndrome-pcos (https://www.kaggle.com/datasets/prasoonkottarathil/polycystic-ovary-syndrome-pcos)

The aim of this project is to use an appropriate classification model to diagnose PCOS. The use of machine learning in situations like these can help process large amounts of data to gain accurate diagnosis and thus possibly help reduce healthcare costs.

First, we load the required packages. The readme file provides information about these packages:

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.4.0      v purrr   0.3.4
## v tibble  3.1.6      v dplyr   1.0.10
## v tidyr   1.1.4      v stringr 1.4.0
## v readr   2.1.3      v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(dplyr)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(stats)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:Hmisc':
##
##     describe
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(DescTools)
```

```
##
## Attaching package: 'DescTools'
```

```
## The following objects are masked from 'package:psych':
##
##     AUC, ICC, SD
```

```
## The following objects are masked from 'package:Hmisc':
##
##     %nin%, Label, Mean, Quantile
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(tree)
library(rpart)
library(rattle)
```

```
## Loading required package: bitops
```

```
##
## Attaching package: 'bitops'
```

```
## The following object is masked from 'package:DescTools':
##
##     %^%
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

Next, we read the data:

```
df <- read_excel("PCOS_data_without_infertility.xlsx", sheet = "Full_new")
```

```
## New names:
## * `` -> `...45`
```

```
head(df,5)
```

```
## # A tibble: 5 x 45
##   `Sl. No` `Patient File No~ `PCOS (Y/N)` `Age (yrs)` `Weight (Kg)` `Height(Cm)`
##      <dbl>             <dbl>        <dbl>       <dbl>         <dbl>        <dbl>
## 1        1                 1            0          28          44.6          152
## 2        2                 2            0          36          65            162.
## 3        3                 3            1          33          68.8          165
## 4        4                 4            0          37          65            148
## 5        5                 5            0          25          52            161
## # ... with 39 more variables: BMI <dbl>, Blood Group <dbl>,
## #   Pulse rate(bpm) <dbl>, RR (breaths/min) <dbl>, Hb(g/dl) <dbl>,
## #   Cycle(R/I) <dbl>, Cycle length(days) <dbl>, Marraige Status (Yrs) <dbl>,
## #   Pregnant(Y/N) <dbl>, No. of aborptions <dbl>, I   beta-HCG(mIU/mL) <dbl>,
## #   II    beta-HCG(mIU/mL) <chr>, FSH(mIU/mL) <dbl>, LH(mIU/mL) <dbl>,
## #   FSH/LH <dbl>, Hip(inch) <dbl>, Waist(inch) <dbl>, Waist:Hip Ratio <dbl>,
## #   TSH (mIU/L) <dbl>, AMH(ng/mL) <chr>, PRL(ng/mL) <dbl>, ...
```

##Data Cleaning:

```
#dropping redundant columns
df <- df[-c(1,45)]

#check data type
d_type <- sapply(df, class)
print(d_type)
```

```
##       Patient File No.             PCOS (Y/N)              Age (yrs)
##             "numeric"              "numeric"              "numeric"
##           Weight (Kg)             Height(Cm)                    BMI
##             "numeric"              "numeric"              "numeric"
##           Blood Group        Pulse rate(bpm)       RR (breaths/min)
##             "numeric"              "numeric"              "numeric"
##              Hb(g/dl)             Cycle(R/I)     Cycle length(days)
##             "numeric"              "numeric"              "numeric"
##  Marraige Status (Yrs)          Pregnant(Y/N)      No. of aborptions
##             "numeric"              "numeric"              "numeric"
##   I   beta-HCG(mIU/mL) II    beta-HCG(mIU/mL)            FSH(mIU/mL)
##             "numeric"            "character"              "numeric"
##            LH(mIU/mL)                FSH/LH              Hip(inch)
##             "numeric"              "numeric"              "numeric"
##           Waist(inch)        Waist:Hip Ratio            TSH (mIU/L)
##             "numeric"              "numeric"              "numeric"
##            AMH(ng/mL)             PRL(ng/mL)         Vit D3 (ng/mL)
##           "character"              "numeric"              "numeric"
##            PRG(ng/mL)             RBS(mg/dl)       Weight gain(Y/N)
##             "numeric"              "numeric"              "numeric"
##       hair growth(Y/N)   Skin darkening (Y/N)         Hair loss(Y/N)
##             "numeric"              "numeric"              "numeric"
##          Pimples(Y/N)        Fast food (Y/N)      Reg.Exercise(Y/N)
##             "numeric"              "numeric"              "numeric"
##    BP _Systolic (mmHg)   BP _Diastolic (mmHg)       Follicle No. (L)
##             "numeric"              "numeric"              "numeric"
##       Follicle No. (R)   Avg. F size (L) (mm)   Avg. F size (R) (mm)
##             "numeric"              "numeric"              "numeric"
##       Endometrium (mm)
##             "numeric"
```

```
#change character to numeric values
df[c(17, 25)]<- lapply(df[c(17, 25)], as.numeric)

#check for missing values
missing_val <- colSums(is.na(df))
print(missing_val)
```

```
##       Patient File No.            PCOS (Y/N)             Age (yrs)
##                     0                     0                     0
##           Weight (Kg)            Height(Cm)                   BMI
##                     0                     0                     0
##           Blood Group       Pulse rate(bpm)      RR (breaths/min)
##                     0                     0                     0
##              Hb(g/dl)            Cycle(R/I)    Cycle length(days)
##                     0                     0                     0
##  Marraige Status (Yrs)         Pregnant(Y/N)     No. of aborptions
##                     1                     0                     0
##   I   beta-HCG(mIU/mL) II    beta-HCG(mIU/mL)           FSH(mIU/mL)
##                     0                     1                     0
##             LH(mIU/mL)                FSH/LH             Hip(inch)
##                     0                     0                     0
##           Waist(inch)       Waist:Hip Ratio           TSH (mIU/L)
##                     0                     0                     0
##            AMH(ng/mL)            PRL(ng/mL)        Vit D3 (ng/mL)
##                     1                     0                     0
##            PRG(ng/mL)            RBS(mg/dl)      Weight gain(Y/N)
##                     0                     0                     0
##       hair growth(Y/N)   Skin darkening (Y/N)        Hair loss(Y/N)
##                     0                     0                     0
##           Pimples(Y/N)       Fast food (Y/N)     Reg.Exercise(Y/N)
##                     0                     1                     0
##    BP _Systolic (mmHg)   BP _Diastolic (mmHg)      Follicle No. (L)
##                     0                     0                     0
##       Follicle No. (R)   Avg. F size (L) (mm)   Avg. F size (R) (mm)
##                     0                     0                     0
##       Endometrium (mm)
##                     0
```

```
#replace missing data with median value

df$"Marraige Status (Yrs)"[is.na(df$"Marraige Status (Yrs)")] <- median(
  df$"Marraige Status (Yrs)", na.rm = T)

df$`II    beta-HCG(mIU/mL)`[is.na(df$`II    beta-HCG(mIU/mL)`)]<- median(
  df$`II    beta-HCG(mIU/mL)`, na.rm = T)

df$`AMH(ng/mL)`[is.na(df$`AMH(ng/mL)`)]<- median(
  df$`AMH(ng/mL)`, na.rm = T)

df$`Fast food (Y/N)`[is.na(df$`Fast food (Y/N)`)]<- median(
  df$`Fast food (Y/N)`, na.rm = T)
```

##Exploratory Data Analysis

As we can see, there are numerous variables available with us. For a comprehensive model without noise, we should use factors that we know is likely to have an impact on our outcome variable (whether PCOS is present or not). Therefore, lets start by finding all the significant correlations between the predictors and the outcome variable in the data:

```r
#checking correlations

# choose the column to correlate with the others
outcome_var <- "PCOS (Y/N)"

# initialize an empty data frame to store the results
corr_df <- data.frame(col1 = character(), col2 = character(), correlation = numeric(), p_valu
e = numeric(), stringsAsFactors = FALSE)

# loop through all columns in the data frame
for (col in names(df)) {
  if (col != outcome_var) {
    # calculate the correlation and test for significance
    corr <- cor.test(df[[outcome_var]], df[[col]], method = "pearson")
    # check if the p-value is less than 0.05
    if (corr$p.value < 0.05) {
      # add the results to the data frame
      corr_df <- rbind(corr_df, data.frame(col1 = outcome_var, col2 = col, correlation = corr
$estimate, p_value = corr$p.value, stringsAsFactors = FALSE))
    }
  }
}

# print the data frame
print(corr_df)
```

```
##             col1                  col2 correlation      p_value
## cor   PCOS (Y/N)             Age (yrs) -0.16851268 8.194488e-05
## cor1  PCOS (Y/N)           Weight (Kg)  0.21193797 6.532451e-07
## cor2  PCOS (Y/N)                   BMI  0.19953449 2.904846e-06
## cor3  PCOS (Y/N)       Pulse rate(bpm)  0.09182052 3.273951e-02
## cor4  PCOS (Y/N)              Hb(g/dl)  0.08717000 4.269392e-02
## cor5  PCOS (Y/N)            Cycle(R/I)  0.40164375 2.175428e-22
## cor6  PCOS (Y/N)    Cycle length(days) -0.17848004 2.975823e-05
## cor7  PCOS (Y/N) Marraige Status (Yrs) -0.11305575 8.489122e-03
## cor8  PCOS (Y/N)             Hip(inch)  0.16229712 1.498197e-04
## cor9  PCOS (Y/N)           Waist(inch)  0.16459775 1.201342e-04
## cor10 PCOS (Y/N)            AMH(ng/mL)  0.26414086 4.355535e-10
## cor11 PCOS (Y/N)        Vit D3 (ng/mL)  0.08549435 4.685719e-02
## cor12 PCOS (Y/N)      Weight gain(Y/N)  0.44104727 3.711617e-27
## cor13 PCOS (Y/N)      hair growth(Y/N)  0.46466663 2.493398e-30
## cor14 PCOS (Y/N)  Skin darkening (Y/N)  0.47573302 6.650877e-32
## cor15 PCOS (Y/N)        Hair loss(Y/N)  0.17287851 5.294624e-05
## cor16 PCOS (Y/N)          Pimples(Y/N)  0.28607668 1.195079e-11
## cor17 PCOS (Y/N)       Fast food (Y/N)  0.37618275 1.250200e-19
## cor18 PCOS (Y/N)      Follicle No. (L)  0.60334615 6.036530e-55
## cor19 PCOS (Y/N)      Follicle No. (R)  0.64832696 7.956425e-66
## cor20 PCOS (Y/N)  Avg. F size (L) (mm)  0.13299151 1.935669e-03
## cor21 PCOS (Y/N)  Avg. F size (R) (mm)  0.09768980 2.306304e-02
## cor22 PCOS (Y/N)      Endometrium (mm)  0.10664825 1.306768e-02
```

We can see that there are 23 factors that have a significant correlation with the presence or absence of PCOS. To avoid overfitting and get a simpler and more comprehensive model, we will select only the factors with strongest correlation to the outcome variable. Therfore this model will have 7 predictor variables.

```
new_df <-df[c(2,11,30,31,32,35,39,40)]

colnames(new_df)
```

```
## [1] "PCOS (Y/N)"           "Cycle(R/I)"           "Weight gain(Y/N)"
## [4] "hair growth(Y/N)"     "Skin darkening (Y/N)" "Fast food (Y/N)"
## [7] "Follicle No. (L)"     "Follicle No. (R)"
```

Multicollinearity can be a problem in simple classification models like logistic regression. Therefore let's check the predictor variables for correlations between them:

```
# calculate correlation matrix and significance values of new_df
corr_mat <- corr.test(new_df, method = "pearson")$r
p_mat <- corr.test(new_df, method = "pearson")$p

# combine correlation matrix and significance values into a single matrix
corr_p_mat <- matrix(paste(round(corr_mat, 2), ifelse(p_mat < 0.05, "*", " "), sep = ""), nco
l = ncol(new_df))
rownames(corr_p_mat) <- colnames(corr_p_mat) <- colnames(new_df)
print(corr_p_mat)
```

```
##                      PCOS (Y/N) Cycle(R/I) Weight gain(Y/N) hair growth(Y/N)
## PCOS (Y/N)           "1*"       "0.4*"     "0.44*"          "0.46*"
## Cycle(R/I)           "0.4*"     "1*"       "0.25*"          "0.28*"
## Weight gain(Y/N)     "0.44*"    "0.25*"    "1*"             "0.3*"
## hair growth(Y/N)     "0.46*"    "0.28*"    "0.3*"           "1*"
## Skin darkening (Y/N) "0.48*"    "0.22*"    "0.35*"          "0.37*"
## Fast food (Y/N)      "0.38*"    "0.21*"    "0.37*"          "0.3*"
## Follicle No. (L)     "0.6*"     "0.3*"     "0.25*"          "0.31*"
## Follicle No. (R)     "0.65*"    "0.25*"    "0.26*"          "0.27*"
##                      Skin darkening (Y/N) Fast food (Y/N) Follicle No. (L)
## PCOS (Y/N)           "0.48*"              "0.38*"         "0.6*"
## Cycle(R/I)           "0.22*"              "0.21*"         "0.3*"
## Weight gain(Y/N)     "0.35*"              "0.37*"         "0.25*"
## hair growth(Y/N)     "0.37*"              "0.3*"          "0.31*"
## Skin darkening (Y/N) "1*"                 "0.34*"         "0.32*"
## Fast food (Y/N)      "0.34*"              "1*"            "0.23*"
## Follicle No. (L)     "0.32*"              "0.23*"         "1*"
## Follicle No. (R)     "0.32*"              "0.25*"         "0.8*"
##                      Follicle No. (R)
## PCOS (Y/N)           "0.65*"
## Cycle(R/I)           "0.25*"
## Weight gain(Y/N)     "0.26*"
## hair growth(Y/N)     "0.27*"
## Skin darkening (Y/N) "0.32*"
## Fast food (Y/N)      "0.25*"
## Follicle No. (L)     "0.8*"
## Follicle No. (R)     "1*"
```

We can see that the predictors are significantly correlated with each other. Therefore we need to use a classification model that is robust to multicollinearity. We will use the decision tree classification model.

Firstly, lets check how balanced our dataset is:

```
# check class distribution
table(new_df$'PCOS (Y/N)')
```

```
##
##   0   1
## 364 177
```

We can see that there are twice as many cases of absence of PCOS than presence of PCOS. Having balanced classes in training data can improve accuracy of predictions. Therefore we will re-sample the data by under-sampling the majority class(downsampling), and then split the data into train and test data.

```
#change classifer to factor variable
new_df$'PCOS (Y/N)'<- factor(new_df$'PCOS (Y/N)')

# resample data using random undersampling of majority class
df_resampled <- downSample(new_df, new_df$'PCOS (Y/N)')

# split data into train and test sets
set.seed(123)
trainIndex <- createDataPartition(df_resampled$'PCOS (Y/N)', p = 0.7, list = FALSE)
train <- df_resampled[trainIndex, ]
test <- df_resampled[-trainIndex, ]
```

We can now train our decision tree model and make predictions on the test data:

```
# create the decision tree model
tree_model <- rpart(train$`PCOS (Y/N)` ~ ., data = train)

# Make predictions on testing set
predictions <- predict(tree_model, test, type = "class")
```

Let us now check how accurate our model is:

```
# Create confusion matrix
cm <- confusionMatrix(predictions, test$'PCOS (Y/N)')
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 53  0
##          1  0 53
##
##                Accuracy : 1
##                  95% CI : (0.9658, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : 0
##
```

The above metrics demonstrate a 100% accuracy of the model with the test data.

The confusion matrix shows the number of true positives, false positives, false negatives, and true negatives in predicting the target variable. In this case, the model correctly predicted 53 instances of class 0 and 53 instances of class 1, with no false positives or false negatives.

Accuracy is the proportion of correctly predicted instances out of the total number of instances in the testing set. In this case, the model achieved perfect accuracy of 1.0.

Sensitivity is the proportion of true positives out of the total number of positive instances in the testing set, while specificity is the proportion of true negatives out of the total number of negative instances in the testing set. In this case, both sensitivity and specificity are 1.0, indicating that the model correctly predicted all instances of both classes.

The positive predictive value is the proportion of true positives out of the total number of predicted positives, while the negative predictive value is the proportion of true negatives out of the total number of predicted negatives. Both values are 1.0 in this case, indicating that the model correctly predicted all instances of both classes.

Prevalence is the proportion of instances in the testing set that belong to the positive class. In this case, the prevalence is 0.5, meaning that half of the instances in the testing set belong to the positive class.

Detection rate is the proportion of true positives out of the total number of instances in the testing set, while detection prevalence is the proportion of predicted positives out of the total number of instances in the testing set. Both values are 0.5 in this case, indicating that the model correctly predicted the true positives.

Finally, balanced accuracy is the arithmetic mean of sensitivity and specificity, and it is also 1.0 in this case, indicating that the model correctly predicted all instances of both classes accurately.

In conclusion, by selecting the most relevant features and re sampling data to ensure that classes are balances in the training data, we have built a highly accurate classification model to diagnose PCOS.