Deep in a disease that 1 in 10 women of childbearing age are afflicted

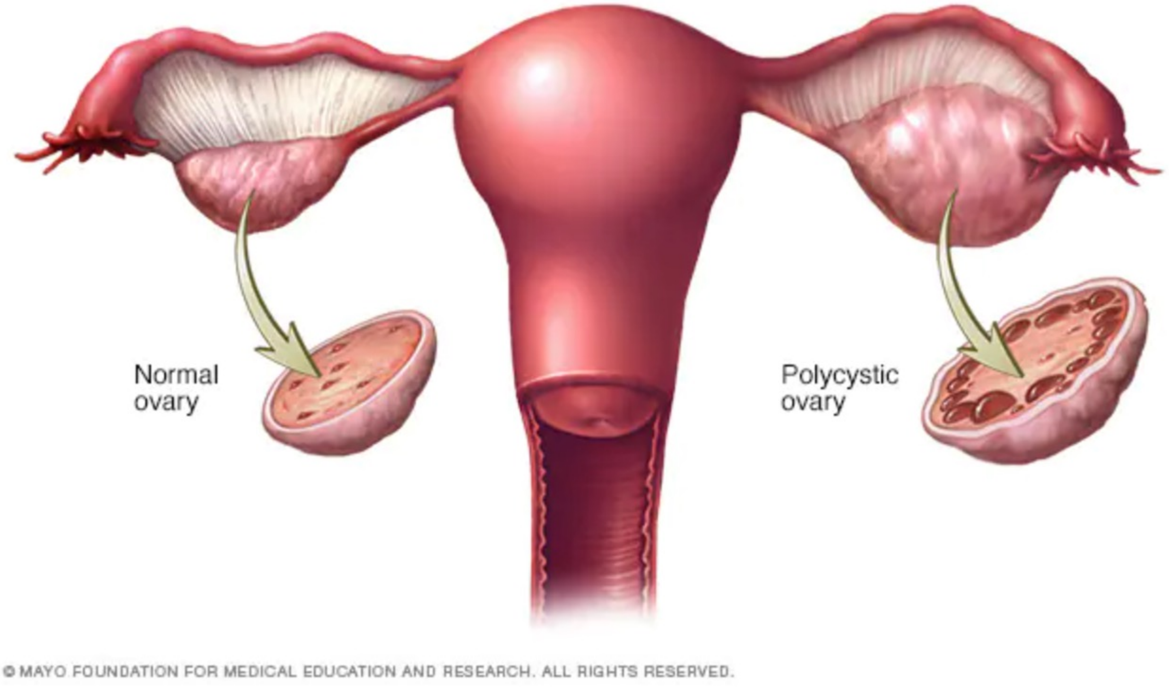
How to accurately diagnose PCOS

Yueming Xu

Introduction

The project aims to build a machine learning model to detect PCOS with a bunch of physical and clinical parameters.

What is PCOS?



© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

Polycystic ovary syndrome (PCOS) is one of most common female endocrine disorder that affects 6-15% of the female population. It is primarily characterized by an extremely irregular menstrual cycle which means one may not have a period for an abnormal long time. Women with PCOS may be at higher risk for type 2 diabetes, high blood pressure, heart problems, and endometrial cancer. Besides, PCOS can also cause excess hair growth, acne, infertility, and weight gain, which also affect patients' mental health.

However, despite that it is common and severe, the cause of PCOS remains unknown and given its complexity, there is not a golden rule in the diagnose and a list of factors should be taken into consideration.

Motive

I was once diagnosed as PCOS in year 2021 and as a young and healthy girl for 20 years, I felt panic at that time. Moreover, since there is also no effective medicine to treat it, the only method is to change my lifestyle. Therefore, I forced myself to sleep early, eat less and exercised more, which made me struggle a lot.

Nevertheless, it eventually turns out to be an incorrect diagnosis and actually I got another kind of disease, meaning that my efforts for a half year is towards a wrong direction. As a patient, I am frustrated.

Thus, I would like to raise the accuracy of the diagnosis of PCOS, which is the motive of the project.

Goal

The main, as mentioned above, is to raise the accuracy of the diagnosis by gathering a comprehensive profile of a patient' parameters and predicting. Additionally, there are more goals that I hope to achieve: * find out the decisive factor of the diagnosis * detect the high-risk group of having PCOS and alert them * cluster patients diagnosed with PCOS

Loading packages and the raw data

First, let us load in all packages and the raw data.

```
library(tibble)
library(kableExtra)
library(naniar)
library(ggfortify)
library(MASS)
library(parsnip)
library(discrim)
library(vip)
library(tidyverse)
library(tidymodels)
library(ggplot2)
library(readxl)
library(pROC)
library(rpart.plot)
library(corrplot)
library(dplyr)
tidymodels_prefer()
theme_set(theme_bw())
```

```
raw_data <- read_excel("D:/mac/ /  /131 machine learning/final project/archive/PCOS_data_without_inferti
    sheet = "Full_new")
head(raw_data)
```

```
## # A tibble: 6 x 45
##   `Sl. No` `Patient File No.` `PCOS (Y/N)` `Age (yrs)` `Weight (Kg)`
##      <dbl>                <dbl>        <dbl>       <dbl>         <dbl>
## 1        1                    1            0          28          44.6
## 2        2                    2            0          36          65
## 3        3                    3            1          33          68.8
## 4        4                    4            0          37          65
## 5        5                    5            0          25          52
## 6        6                    6            0          36          74.1
## # i 40 more variables: `Height(Cm)` <dbl>, 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>, ...
```

The data is from Kaggle data set "Polycystic ovary syndrome (PCOS), which is collected from 10 different hospital across Kerala,India.

Exploring the Raw Data and tidying the data

```r
dim(raw_data)
```

```
## [1] 541  45
```

```r
numeric_cols<-raw_data%>%
  select_if(is.numeric)
str_cols<-raw_data%>%
  select_if(is.character)
ncol(numeric_cols);ncol(str_cols)
```

```
## [1] 42
```

```
## [1] 3
```

```r
colnames(str_cols)
```

```
## [1] "II    beta-HCG(mIU/mL)" "AMH(ng/mL)"             "...45"
```

There are 541 rows and 45 columns in the data, meaning 541 observations and 44 predictors. 41 of predictors are numeric while the rest of them are strings.

For numeric predictors, we need to notice that some are actually dummy variables, or more precisely, binary variables since the level of the initial categorical factors are all 2. Therefore, we do not need to dummy them by ourselves when creating recipes.

Correcting entry error and converting the type of variables

The column names of strings are II beta-HCG(mIU/mL), AMH(ng/mL), …45, the last of which seems to have no meanings.

```r
head(str_cols$...45)
```

```
## [1] NA NA NA NA NA NA
```

```r
sum(!is.na(str_cols$...45))
```

```
## [1] 2
```

By deepening into the ... 45, we can find that only 2 rows in the vector are not NA, suggesting that it is simply an entry mistake. Thus, we remove the columns from the data.

```r
raw_data<-raw_data[,-45]
tidy_data<-raw_data%>%
  mutate(`PCOS (Y/N)`=as.factor(`PCOS (Y/N)`))
```

Besides, the first two predictors that are string are supposed to be numeric.

```r
wr<-which(is.na(as.double(tidy_data$`II    beta-HCG(mIU/mL)`)))
wr
```

```
## [1] 124
```

```r
tidy_data$`II    beta-HCG(mIU/mL)`[wr]
```

```
## [1] "1.99."
```

```r
wr2<-which(is.na(as.double(tidy_data$`AMH(ng/mL)`)))
wr2
```

```
## [1] 306
```

```r
tidy_data$`AMH(ng/mL)`[wr2]
```

```
## [1] "a"
```

```r
tidy_data$`II    beta-HCG(mIU/mL)`[wr]<-1.99
tidy_data$`AMH(ng/mL)`[wr2]<-NA
tidy_data$`II    beta-HCG(mIU/mL)`<-as.double(tidy_data$`II    beta-HCG(mIU/mL)`)
tidy_data$`AMH(ng/mL)`<-as.double(tidy_data$`AMH(ng/mL)`)
```

When we attempt to transform the `II    beta-HCG(mIU/mL)` and `AMH(ng/mL)` into double, warning message occurs. To figure out the reasons, we locate the newly introduced NA and their original forms, discovering the input error. Thus, we delete the”.” in “1.99.” and change the meaningless character into NA. Finally, we get the tidy data.

For `Blood Group` and `Cycle (R/I)`, although they are shown as numeric, they are actually categorical variable with limited levels. According to the instruction along with the data, `Blood Group` indicates blood group by number. Since there are already a large number of predictors in the model as well as the type of `Blood Group` , for the sake of simplicity, we will keep using `Blood Group` as a numeric.

- A+=11
- A-=12
- B+=13
- B-=14
- O+=15
- O-=16
- AB+=17
- AB-=18

When it comes to `Cycle (R/I)` , which indicates whether the cycle is regular, the levels are supposed to be 2.

```r
unique(tidy_data$`Cycle(R/I)`)
```

```
## [1] 2 4 5
```

```r
nrow(tidy_data[tidy_data$`Cycle(R/I)`==5,])
```

```
## [1] 1
```

However, it is confusing that the variable has 3 levels. Deeply looking into the data, only one observation is with `Cycle (R/I)` taking value as 5. Thus, it is rational to suspect that it is a slip of the pen and is supposed to be 4. Consequently, we correct it and for the sake of simplicity, we change `Cycle (R/I)` from a continuous numeric to a binary factor, which corresponds to its true meaning.

```
tidy_data[tidy_data$`Cycle(R/I)`==5,]$`Cycle(R/I)`<-4
tidy_data%>%
  mutate(`Cycle(R/I)`=as.factor(`Cycle(R/I)`/2))
```

```
## # A tibble: 541 x 44
##    `Sl. No` `Patient File No.` `PCOS (Y/N)` `Age (yrs)` `Weight (Kg)`
##       <dbl>              <dbl> <fct>              <dbl>         <dbl>
##  1        1                  1 0                     28          44.6
##  2        2                  2 0                     36          65
##  3        3                  3 1                     33          68.8
##  4        4                  4 0                     37          65
##  5        5                  5 0                     25          52
##  6        6                  6 0                     36          74.1
##  7        7                  7 0                     34          64
##  8        8                  8 0                     33          58.5
##  9        9                  9 0                     32          40
## 10       10                 10 0                     36          52
## # i 531 more rows
## # i 39 more variables: `Height(Cm)` <dbl>, BMI <dbl>, `Blood Group` <dbl>,
## #   `Pulse rate(bpm)` <dbl>, `RR (breaths/min)` <dbl>, `Hb(g/dl)` <dbl>,
## #   `Cycle(R/I)` <fct>, `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)` <dbl>, `FSH(mIU/mL)` <dbl>, ...
```

Describing the variables

```
str(tidy_data)
```

```
## tibble [541 x 44] (S3: tbl_df/tbl/data.frame)
##  $ Sl. No                : num [1:541] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Patient File No.      : num [1:541] 1 2 3 4 5 6 7 8 9 10 ...
##  $ PCOS (Y/N)            : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ Age (yrs)             : num [1:541] 28 36 33 37 25 36 34 33 32 36 ...
##  $ Weight (Kg)           : num [1:541] 44.6 65 68.8 65 52 74.1 64 58.5 40 52 ...
##  $ Height(Cm)            : num [1:541] 152 162 165 148 161 ...
##  $ BMI                   : num [1:541] 19.3 NA NA NA NA NA NA NA NA NA ...
##  $ Blood Group           : num [1:541] 15 15 11 13 11 15 11 13 11 15 ...
##  $ Pulse rate(bpm)       : num [1:541] 78 74 72 72 72 78 72 72 72 80 ...
##  $ RR (breaths/min)      : num [1:541] 22 20 18 20 18 28 18 20 18 20 ...
##  $ Hb(g/dl)              : num [1:541] 10.5 11.7 11.8 12 10 ...
##  $ Cycle(R/I)            : num [1:541] 2 2 2 2 2 2 2 2 2 4 ...
##  $ Cycle length(days)    : num [1:541] 5 5 5 5 5 5 5 5 5 2 ...
##  $ Marraige Status (Yrs) : num [1:541] 7 11 10 4 1 8 2 13 8 4 ...
##  $ Pregnant(Y/N)         : num [1:541] 0 1 1 0 1 1 0 1 0 0 ...
##  $ No. of aborptions     : num [1:541] 0 0 0 0 0 0 0 2 1 0 ...
##  $ I   beta-HCG(mIU/mL)  : num [1:541] 1.99 60.8 494.08 1.99 801.45 ...
##  $ II    beta-HCG(mIU/mL): num [1:541] 1.99 1.99 494.08 1.99 801.45 ...
##  $ FSH(mIU/mL)           : num [1:541] 7.95 6.73 5.54 8.06 3.98 3.24 2.85 4.86 3.76 2.8 ...
##  $ LH(mIU/mL)            : num [1:541] 3.68 1.09 0.88 2.36 0.9 1.07 0.31 3.07 3.02 1.51 ...
##  $ FSH/LH                : num [1:541] NA NA NA NA NA NA NA NA NA NA ...
##  $ Hip(inch)             : num [1:541] 36 38 40 42 37 44 39 44 39 40 ...
##  $ Waist(inch)           : num [1:541] 30 32 36 36 30 38 33 38 35 38 ...
##  $ Waist:Hip Ratio       : num [1:541] NA NA NA NA NA NA NA NA NA NA ...
##  $ TSH (mIU/L)          : num [1:541] 0.68 3.16 2.54 16.41 3.57 ...
##  $ AMH(ng/mL)           : num [1:541] 2.07 1.53 6.63 1.22 2.26 6.74 3.05 1.54 1 1.61 ...
##  $ PRL(ng/mL)           : num [1:541] 45.2 20.1 10.5 36.9 30.1 ...
##  $ Vit D3 (ng/mL)       : num [1:541] 17.1 61.3 49.7 33.4 43.8 52.4 42.7 38 21.8 27.7 ...
##  $ PRG(ng/mL)           : num [1:541] 0.57 0.97 0.36 0.36 0.38 0.3 0.46 0.26 0.3 0.25 ...
##  $ RBS(mg/dl)           : num [1:541] 92 92 84 76 84 76 93 91 116 125 ...
##  $ Weight gain(Y/N)     : num [1:541] 0 0 0 0 0 1 0 1 0 0 ...
##  $ hair growth(Y/N)     : num [1:541] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Skin darkening (Y/N) : num [1:541] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Hair loss(Y/N)       : num [1:541] 0 0 1 0 1 1 0 0 0 0 ...
##  $ Pimples(Y/N)         : num [1:541] 0 0 1 0 0 0 0 0 0 0 ...
##  $ Fast food (Y/N)      : num [1:541] 1 0 1 0 0 0 0 0 0 0 ...
##  $ Reg.Exercise(Y/N)    : num [1:541] 0 0 0 0 0 0 0 0 0 0 ...
##  $ BP _Systolic (mmHg)  : num [1:541] 110 120 120 120 120 110 120 120 120 110 ...
##  $ BP _Diastolic (mmHg) : num [1:541] 80 70 80 70 80 70 80 80 80 80 ...
##  $ Follicle No. (L)     : num [1:541] 3 3 13 2 3 9 6 7 5 1 ...
##  $ Follicle No. (R)     : num [1:541] 3 5 15 2 4 6 6 6 7 1 ...
##  $ Avg. F size (L) (mm) : num [1:541] 18 15 18 15 16 16 15 15 17 14 ...
##  $ Avg. F size (R) (mm) : num [1:541] 18 14 20 14 14 20 16 18 17 17 ...
##  $ Endometrium (mm)     : num [1:541] 8.5 3.7 10 7.5 7 8 6.8 7.1 4.2 2.5 ...
```

Below is the description of the variables that will be used in modelling and the predictors mainly falls into 6 categories based on its source.

1. basic physical information
2. medical history
3. clinical information
4. symptom
5. ultra examination

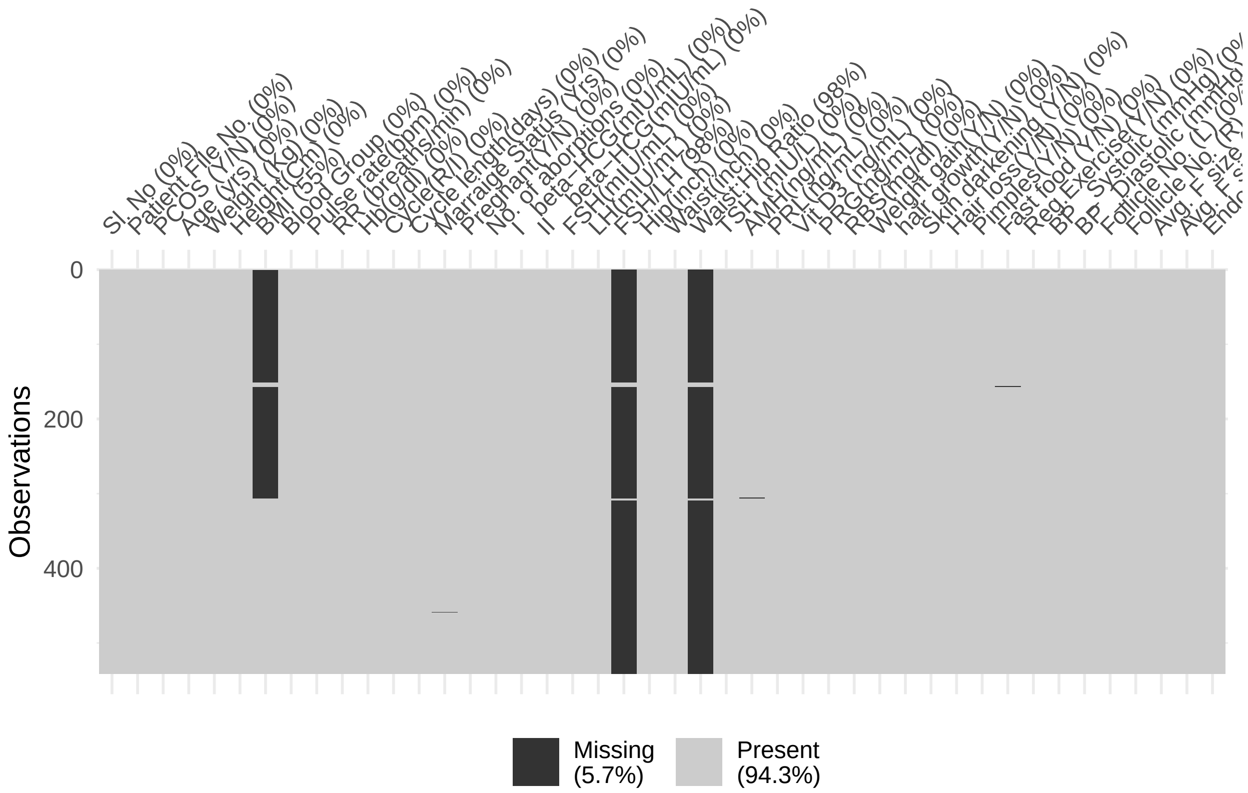
Name	Category	Meaning
Sl. No	index	the unique number for each person
PCOS (Y/N)	response variable	a binary factor which is 1 that the person is diagnosed as PCOS and is 0 otherwise
Age;Weight;Height	basic physical information	literal meaning
Blood group	basic physical information	• A+=11 • A-=12 • B+=13 • B-=14 • O+=15 • O-=16 • AB+=17 • AB-=18
Pulse rate	basic physical information	literal meaning

Name	Category	Meaning
RR	basic physical information	respiratory rate is measured by counting the number of breaths a person takes in a one-minute period
Hb	basic physical information	The number of Hemoglobin, a protein containing iron that facilitates the transport of oxygen in red blood cells.
Cycle(R/I)	medical history	Whether the cycle is regular or irregular
Cycle length;Marriage Status;Pregnant;No. of aborption	medical history	literal meaning
beta-HCG;FSH;LH;TSH;AMH;PRL;Vit D3; PRG; RBS	clinical information	represent various hormones and markers used to assess reproductive and endocrine health. They include indicators for pregnancy confirmation (Beta-HCG), ovarian function (FSH, LH, AMH), thyroid health (TSH), lactation (PRL), vitamin status (Vit D3), progesterone levels (PRG), and blood sugar levels (RBS)
Hip;Waist;Weight Gain; hair growth;skin darkening; Hair loss; Pimples	Symptom	They are the classical symptom of patient with PCOS and can be observed. Ps: Hip and waist are included because PCOS is correlated with partially obesity
Fast food; Reg. Exercise	Lifestyle	binary indicators of whether the volunteer have fast food or exercise regularly
BP _Systolic (mmHg)	Ultrasound examination	Blood pressure in arteries during heart contraction
BP _Diastolic (mmHg)	Ultrasound examination	Blood pressure in arteries when heart is at rest
Follicle No. (L)	Ultrasound examination	Quantity of developing egg follicles (left ovary)
Follicle No. (R)	Ultrasound examination	Quantity of developing egg follicles (right ovary)
Avg. F size (L) (mm)	Ultrasound examination	Average size of developing follicles (left ovary)
Avg. F size (R) (mm)	Ultrasound examination	Average size of developing follicles (right ovary)
Endometrium (mm)	Ultrasound examination	Thickness of the inner lining of the uterus

Missing Data

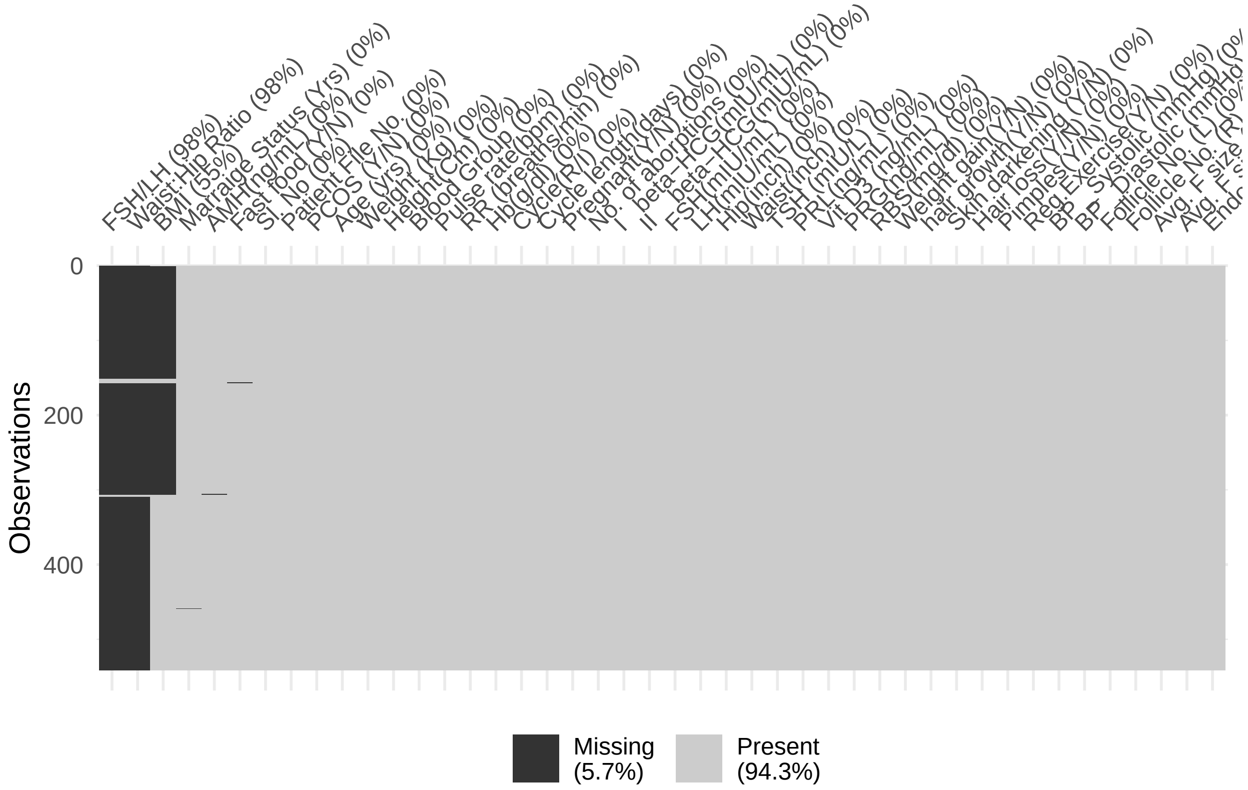
Dealing with the missing data is an essential step. We first have a general picture using `vis_miss()`.

```r
vis_miss(tidy_data)
```



Overall, 5.7% are missing, which is acceptable. Moreover, luckily, the missingness only appear in a small number of columns.

```r
vis_miss(tidy_data,sort_miss = T)
```



```r
miss_var_summary(tidy_data)
```

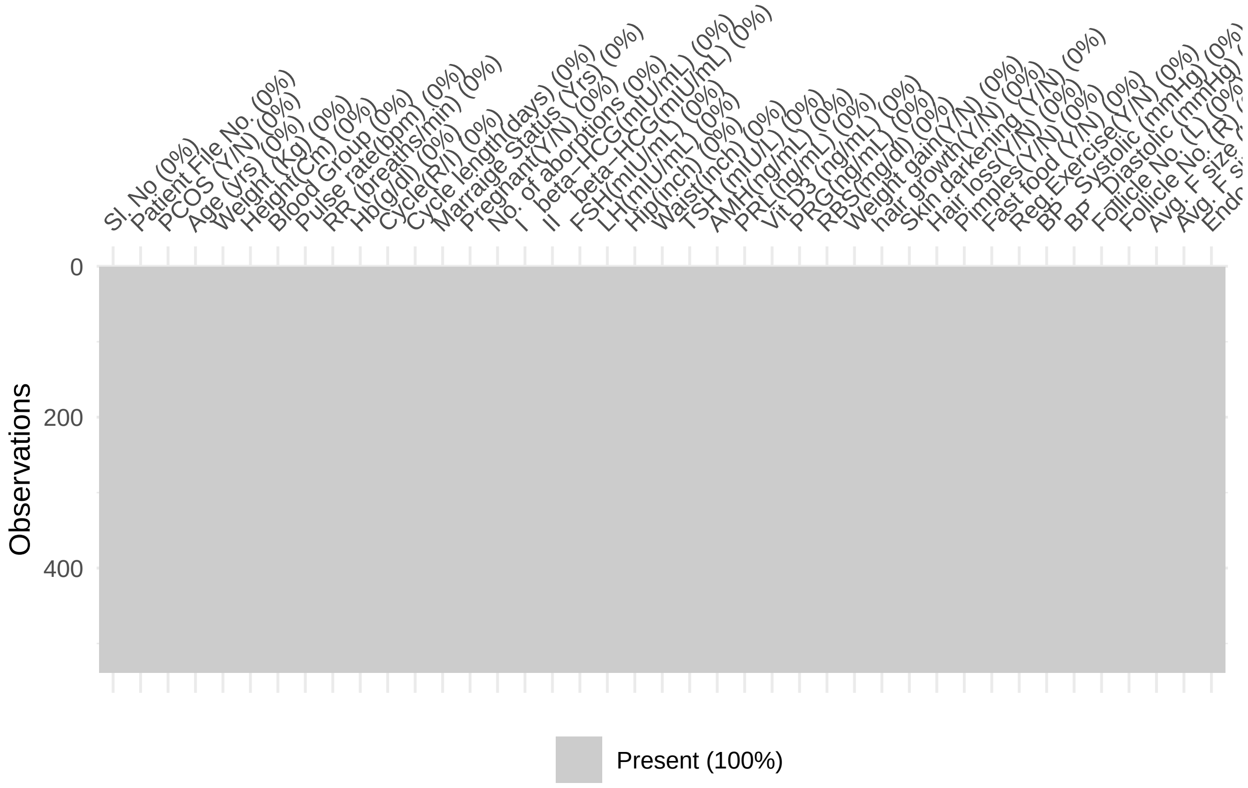
```
## # A tibble: 44 x 3
##    variable              n_miss pct_miss
##    <chr>                  <int>    <dbl>
##  1 FSH/LH                   532   98.3
##  2 Waist:Hip Ratio          532   98.3
##  3 BMI                      299   55.3
##  4 Marraige Status (Yrs)      1    0.185
##  5 AMH(ng/mL)                 1    0.185
##  6 Fast food (Y/N)            1    0.185
##  7 Sl. No                     0    0
##  8 Patient File No.           0    0
##  9 PCOS (Y/N)                 0    0
## 10 Age (yrs)                  0    0
## # i 34 more rows
```

By reordering the predictors based on its percentage of data missing, we could see that for three predictors, half of observations are missing and thus we should remove the predictors entirely. Besides, all the missing data are in fact what we can commute since we have the entire information about`FSH`,`LH`,`Waist`,`Hip`,`Weight` and `Height`. Thus, we should remove them in order to keep predictors independent. For Marriage Status and Fast food, only one observation is missing so removing the observations won't lead to much loss of influence.

```r
tidy_data<-tidy_data%>%
  dplyr:: select(-c("FSH/LH","Waist:Hip Ratio","BMI"))%>%
  filter(!is.na(`Marraige Status (Yrs)`)&!is.na(`Fast food (Y/N)`)&!is.na(`AMH(ng/mL)`))
dim(tidy_data)
```

```
## [1] 538  41
```

```r
vis_miss(tidy_data)
```



Now there is no data missing and the number of observation is 539.

Variable Selection

Among the variables that are retained, we consider their physical meaning to see whether we can eliminate some variable to simplify the model.

```r
any(tidy_data$`Sl. No`!=tidy_data$`Patient File No.`)
```

```
## [1] FALSE
```

```r
tidy_data<-tidy_data%>%
  dplyr::select(-`Patient File No.`)
tidy_data1<-tidy_data%>%
  dplyr::select(-`Sl. No`)
```

It is shown that two columns `Sl. No` and `Patient File No.` are exactly the same. Thus, to be concise, we should delete a surplus one. Also, since we assume that each observations are independent, `Sl. No` is not used in predicting.

Eventually, we achieve our tidy data.

```r
head(tidy_data1)
```

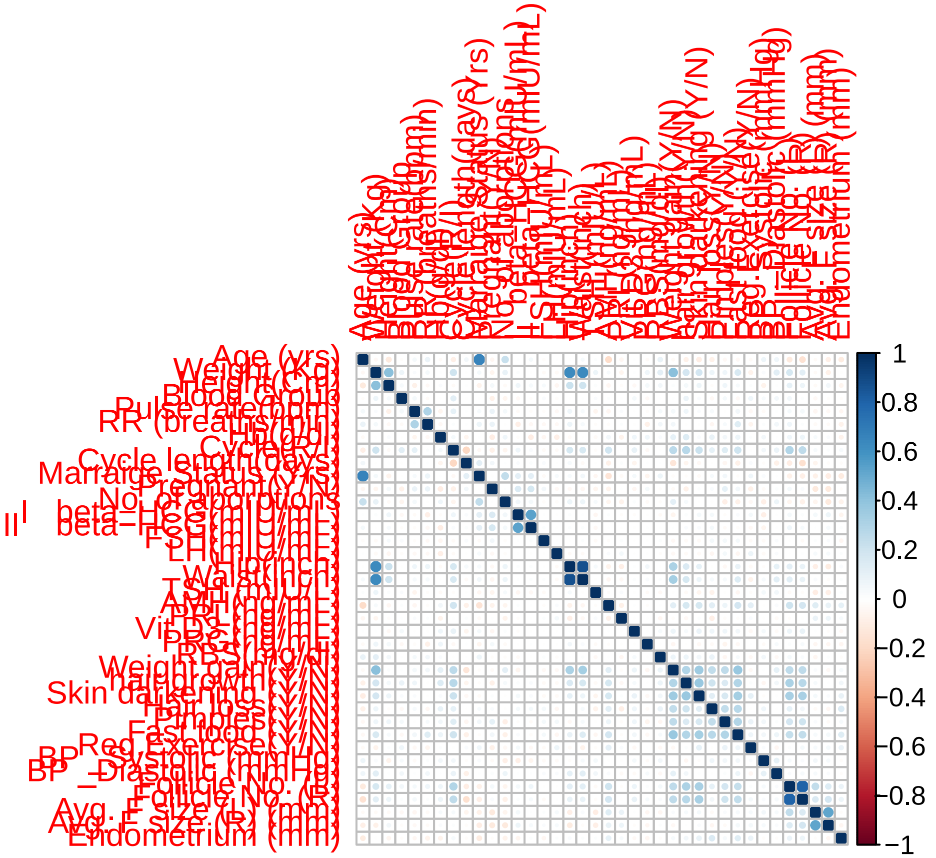
```
## # A tibble: 6 x 39
##   `PCOS (Y/N)` `Age (yrs)` `Weight (Kg)` `Height(Cm)` `Blood Group`
##   <fct>                  <dbl>         <dbl>       <dbl>         <dbl>
## 1 0                         28          44.6         152            15
## 2 0                         36          65           162.           15
## 3 1                         33          68.8         165            11
## 4 0                         37          65           148            13
## 5 0                         25          52           161            11
## 6 0                         36          74.1         165            15
## # i 34 more variables: `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)` <dbl>, `FSH(mIU/mL)` <dbl>, `LH(mIU/mL)` <dbl>,
## #   `Hip(inch)` <dbl>, `Waist(inch)` <dbl>, `TSH (mIU/L)` <dbl>,
## #   `AMH(ng/mL)` <dbl>, `PRL(ng/mL)` <dbl>, `Vit D3 (ng/mL)` <dbl>, ...
```

Visual EDA

When it comes to visual explanatory data analysis, we first have a whole picture of the predictors and next focus on the random variable. Then we exhibits the relationship between the random variable and a single predictor. Lastly, we expand the number of predictor from one to two.

Correlation

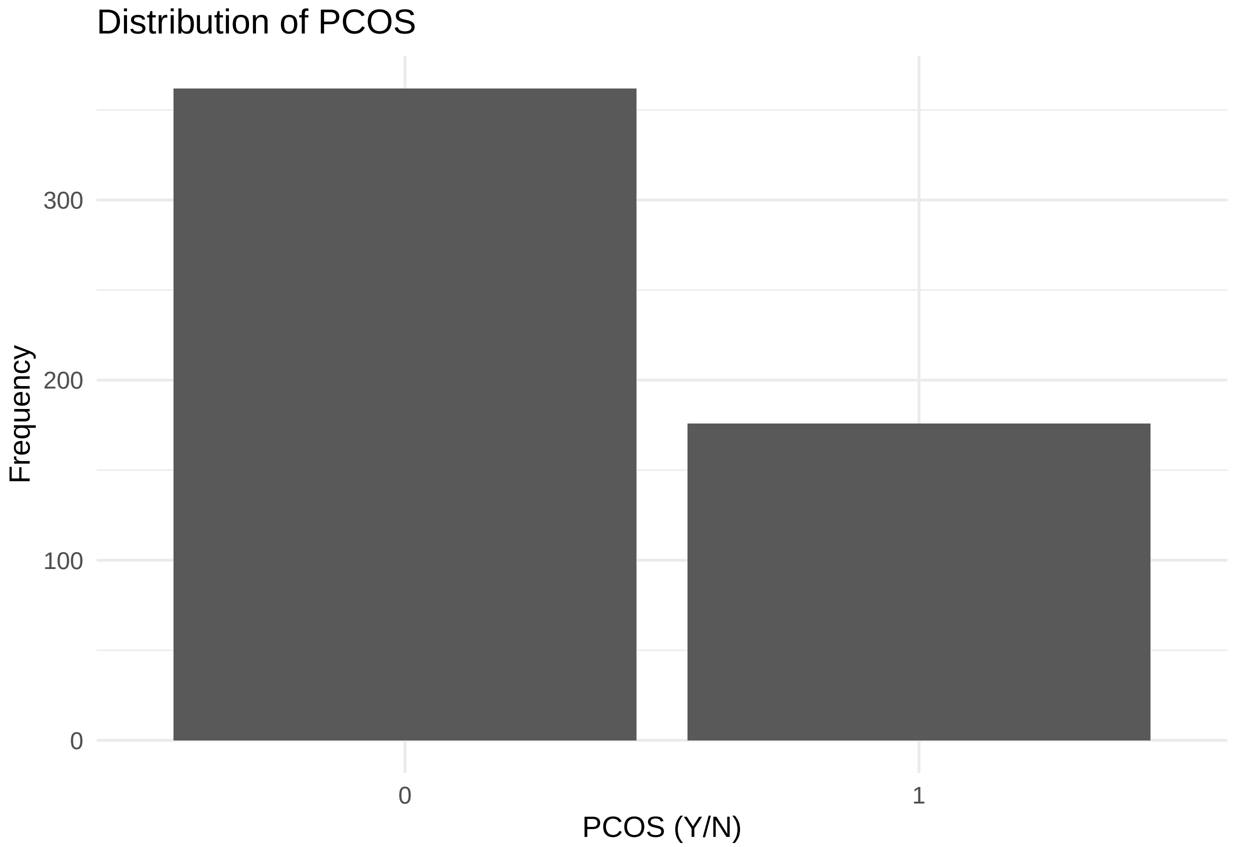
```r
cor_matrix<-tidy_data %>%
  dplyr::select(where(is.numeric) )%>%
  dplyr::select(-`Sl. No`)%>%
  cor()
cor_matrix%>%
  corrplot()
```



We are glad to see that most of the predictors have little correlation and are quite independent. On the other hand, it also means that there is no much room for PCA. Thus, we need to turn to other method to do the variable selection.

Response variable

```r
ggplot(tidy_data, aes(x = factor(`PCOS (Y/N)`))) +
  geom_bar() +
  labs(title = "Distribution of PCOS", x = "PCOS (Y/N)", y = "Frequency")+
  theme_minimal()
```

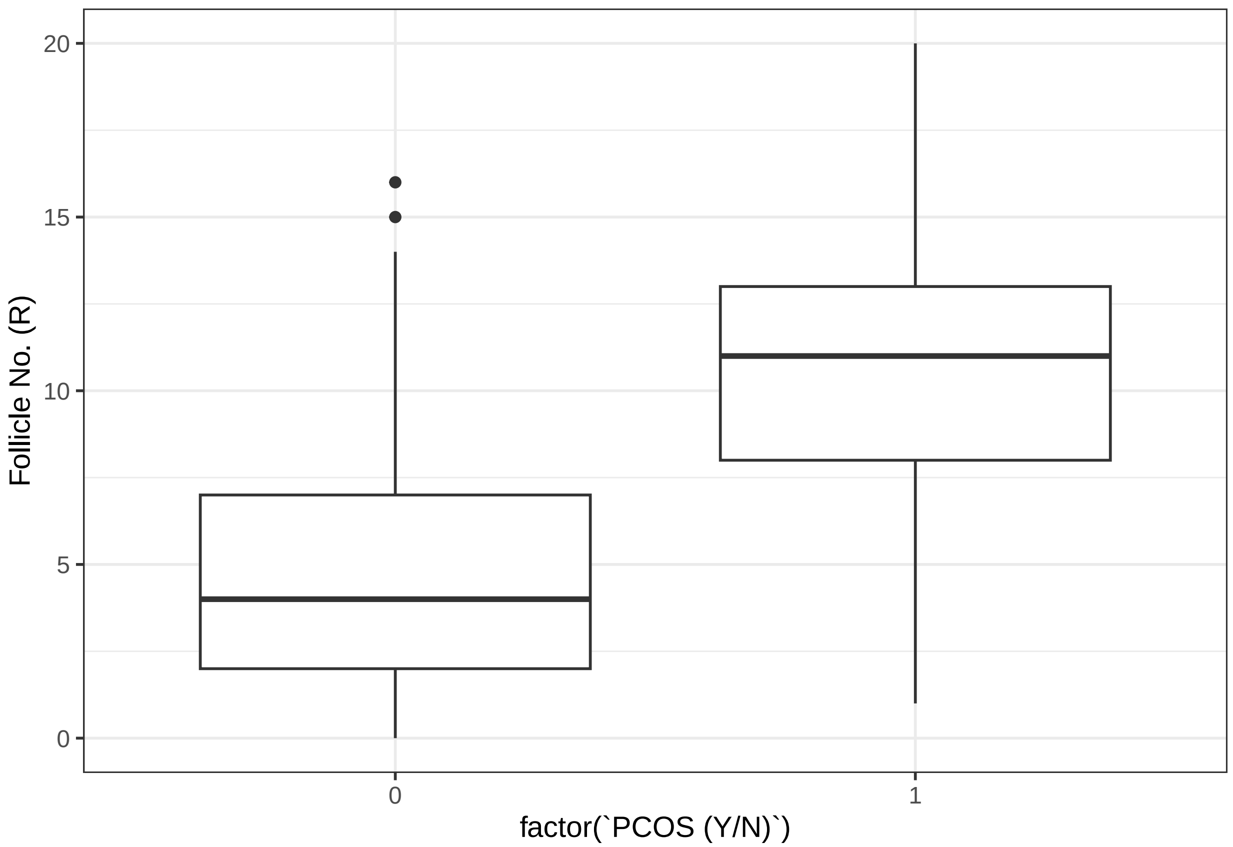


First of all, we simply check the distribution of the response variable `PCOS (Y/N)`. Corresponding with our intuition, those who do not have PCOS is much more than those who have. Since it is imbalanced, it is worth noticing that when doing initial split and k-folds validation, `strata` should be set as the response variable.

Single predictor - Follicle Number

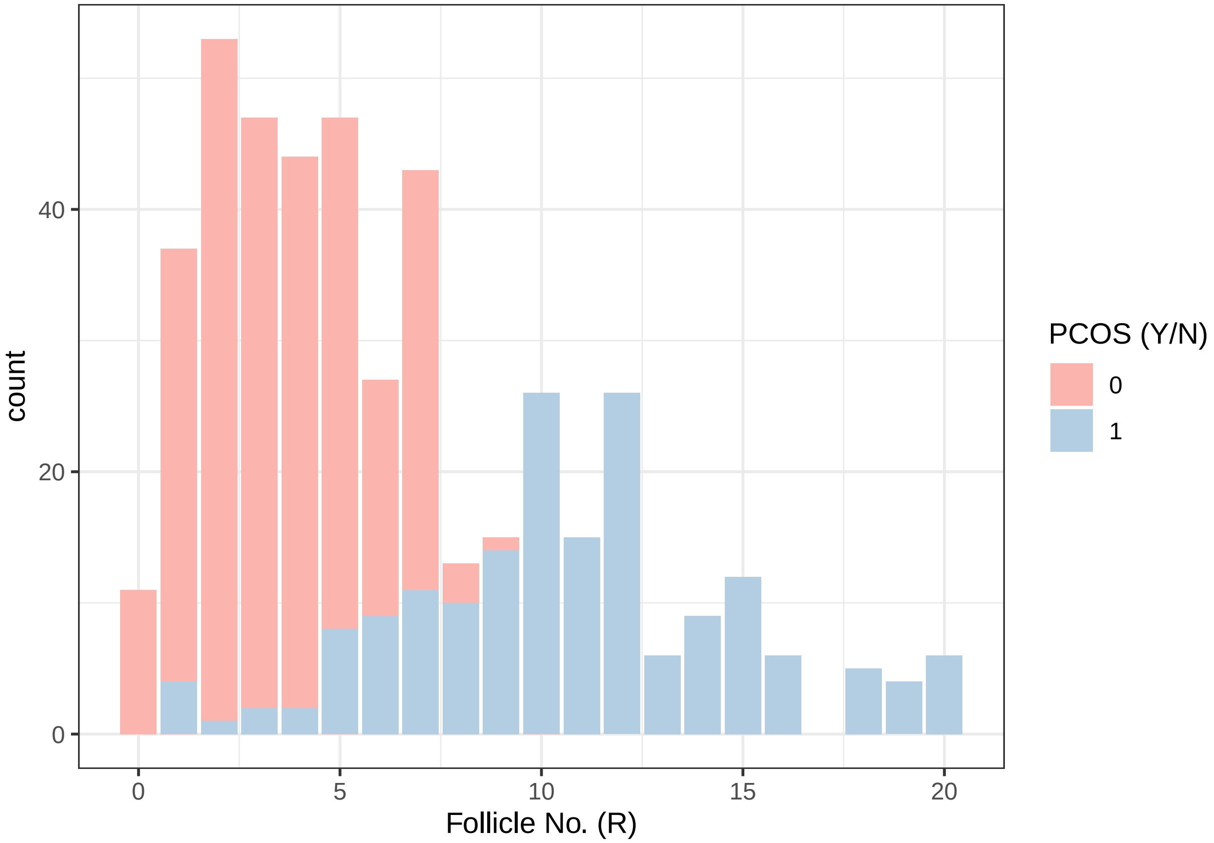
Then we explore a single predictor's relationship with the response variable. Given that `Follicle No. (R)` is a continuous variable, box plot is the first choice considering convenience.

```r
ggplot(tidy_data,aes(x=factor(`PCOS (Y/N)`),y=`Follicle No. (R)`))+
  geom_boxplot()+ scale_color_brewer(palette = "Pastel1")
```



It is apparent that the number of follicles on average for patients is larger than those who are healthy. Also, two box have little overlap area. To observe the distribution in a more detailed way, we draw another plot.

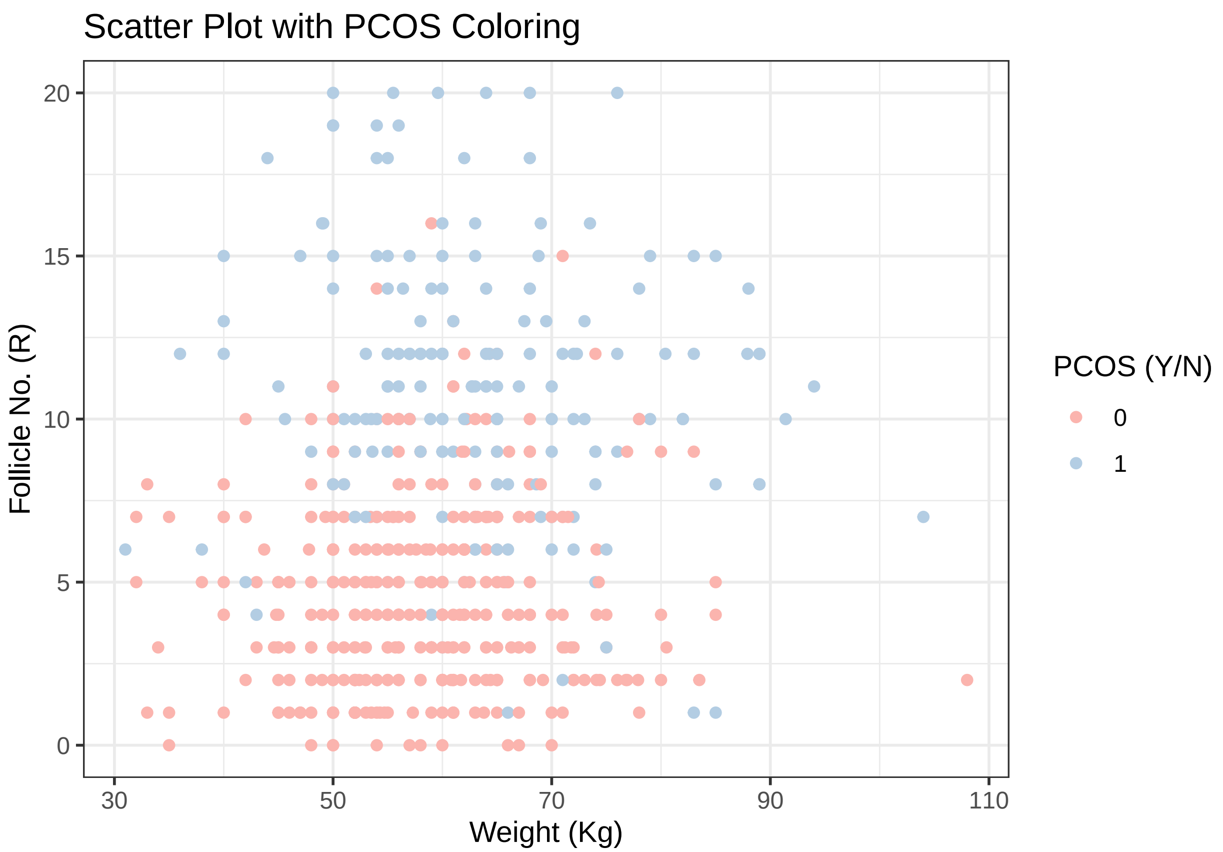
```
tidy_data1%>%
  dplyr::select(`Follicle No. (R)`,`PCOS (Y/N)`)%>%
  group_by(`Follicle No. (R)`)%>%
  ggplot(aes(x = `Follicle No. (R)`, fill = `PCOS (Y/N)`))+
  geom_bar(position = "identity")+ scale_fill_brewer(palette = "Pastel1")
```



It presents the difference in a more vivid way, suggesting `Follicle No. (R)` as a good indicator to separate two groups.

Two predictors - Follicle number and Weight

```
ggplot(tidy_data1,aes(x =`Weight (Kg)`, y = `Follicle No. (R)`, color = `PCOS (Y/N)`)) +
  geom_point()+
  labs(title = "Scatter Plot with PCOS Coloring") + scale_color_brewer(palette = "Pastel1")
```



Selecting two predictors to create a scatter plot and coloring the points based on its `PCOS (Y/N)`, we can see that there seems to be a line that can separate two groups roughly. Therefore, it is plausible to build a QDA and a SVM model later.

Setting up Models

Split the data and k-fold cross validation

When it comes to building a model, the first thing to do is to split the data. In this way, we can evaluate the performance of our final model accurately with a pure testing set in order to prevent overfitting. Since there are538 observations which is relatively adequate, we can set `prop=0.75`.

Next, to prepare a criteria for the model selection across different types of models and to get a better estimate of testing accuracy, we need to resample and here we turn to k-fold cross validation to achieve the goal since it makes full use of the training data.

As mentioned above, the response variable is imbalanced so we need to set a strata split the data.

```
set.seed(660)
pcos_insplit<-initial_split(tidy_data1,prop=0.75,strata = `PCOS (Y/N)`)
pcos_training1<-training(pcos_insplit)
pcos_testing<-testing(pcos_insplit)
pcos_folds<-vfold_cv(pcos_training1,v=10,strata = `PCOS (Y/N)`)
```

Create a recipe

We build a recipe based on the training set. To improve the accuracy of the prediction, we include all the meaningful predictors and since there are a number of numeric predictors whose scales vary a lot, we normalize them. Now our recipe is ready.

```r
recipe1<-recipe(`PCOS (Y/N)`~.,data = pcos_training1)%>%
  step_normalize(all_numeric_predictors())
prep(recipe1)
b<-bake(prep(recipe1),pcos_training1)
b
```

```
## # A tibble: 403 x 39
##    `Age (yrs)` `Weight (Kg)` `Height(Cm)` `Blood Group` `Pulse rate(bpm)`
##          <dbl>         <dbl>        <dbl>         <dbl>             <dbl>
##  1      -0.598       -1.34         -0.780         0.620             1.61
##  2       1.06         0.523        -1.45         -0.471            -0.556
##  3      -1.15        -0.665         0.734        -1.56             -0.556
##  4       0.879        1.35          1.41          0.620             1.61
##  5       0.509        0.431        -0.107        -1.56             -0.556
##  6       0.325       -0.0710        0.397        -0.471            -0.556
##  7       0.140       -1.76          0.229        -1.56             -0.556
##  8       0.879       -0.665        -1.12          0.620             2.34
##  9      -2.08         1.07          1.07          0.620             2.34
## 10      -0.968       -0.938         0.565        -0.471            -0.556
## # i 393 more rows
## # i 34 more variables: `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)` <dbl>, `FSH(mIU/mL)` <dbl>, `LH(mIU/mL)` <dbl>,
## #   `Hip(inch)` <dbl>, `Waist(inch)` <dbl>, `TSH (mIU/L)` <dbl>, ...
```

Model building and visualizing the result

In this part, we build different types of models, following the procedure summarized below:

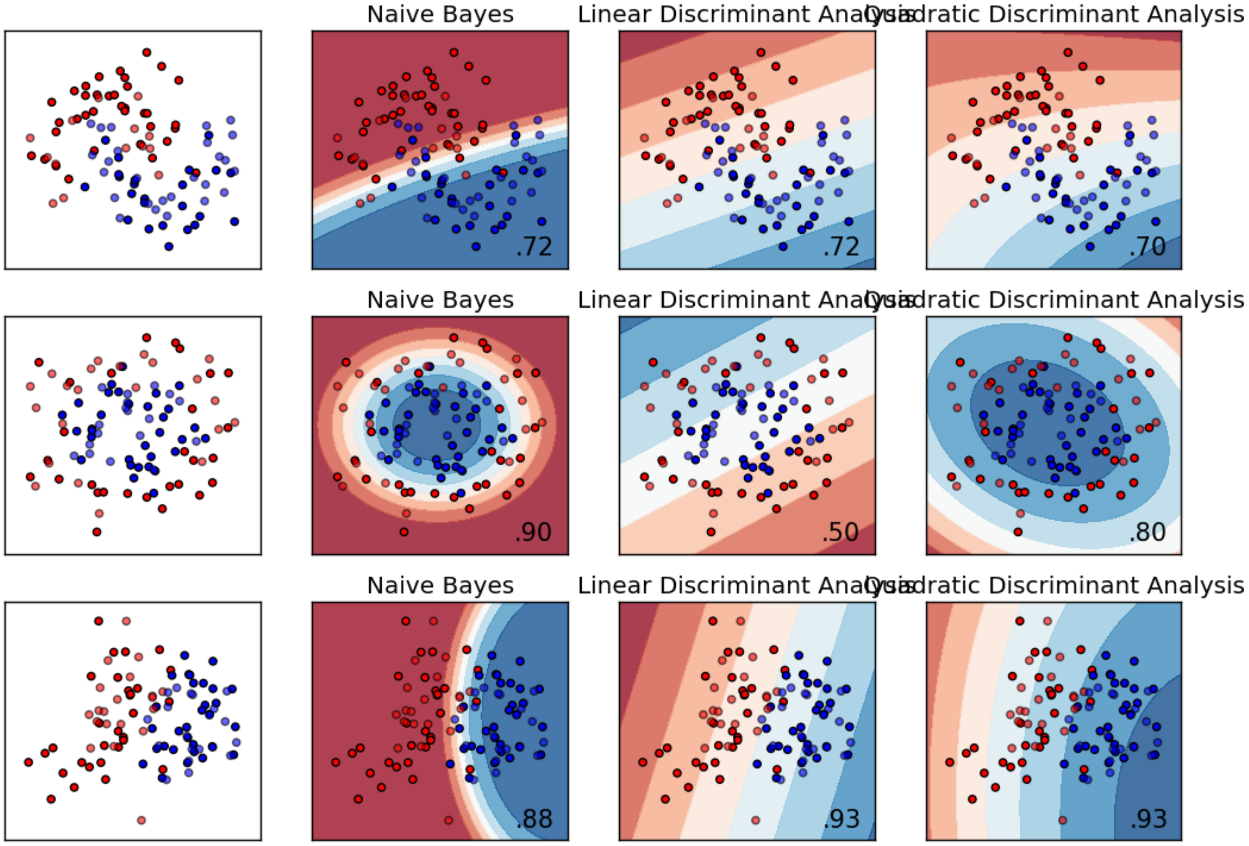
1. Specify the kind of model, setting its engine, and setting its mode (which is always classification in the project)
2. Create a workflow by adding the model and the recipe we have built
3. Set up the tuning grid for each hyper-parameters and levels
4. Tune the certain parameters
5. Save the tuning result, or it will cost a long time to knit
6. Load the tuning result and visualize the result
7. Select the best model from all of the tuning based on the `roc_auc` because it comprehensively reflects the accuracy and specificity
8. Finalize the workflow with the best selected model
9. Fit the model to the overall training set
10. Evaluate the model's performance by visualizing its `roc curve` and calculating its `roc_auc`

For the QDA model, there is no parameter for tuning, so we skip the step 3 to step 7.

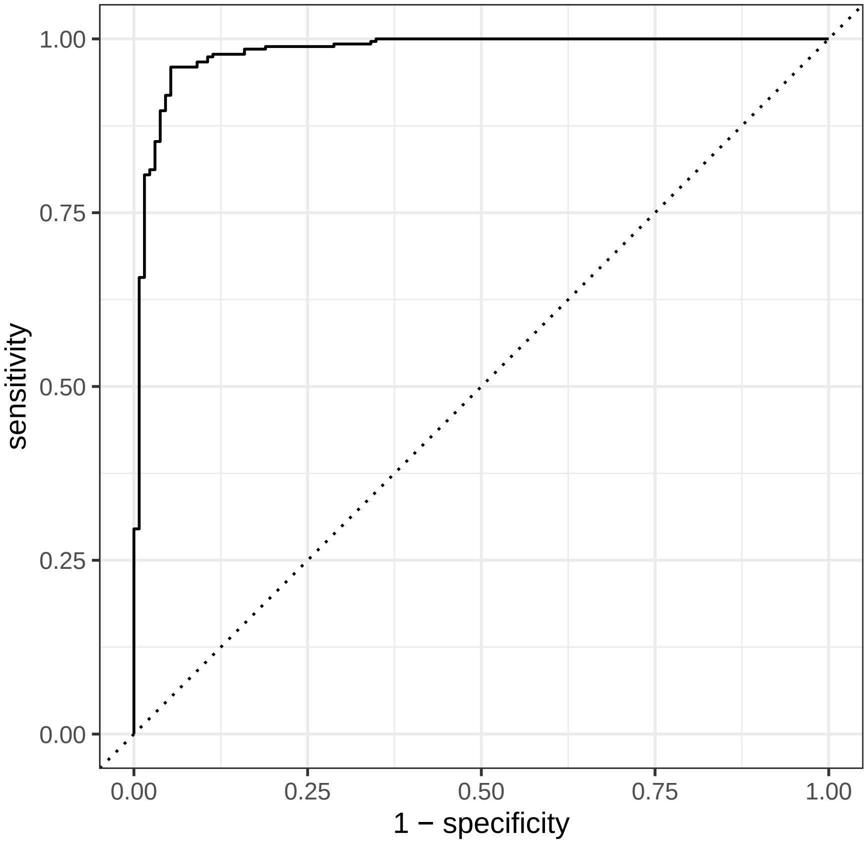
At the beginning of each model, I will explain the reason I choose it. Here I first illustrate why some models that can be used for classification is not suitable. Logistic linear regression is not chosen because our main goal is to predict, what requires enough flexibility. Besides, K-nearest neighbors is a common method but given the large number of predictors, the calculation is too complex. Thus, we set up QDA, Elastic Net Glm, Random Forest and SVM models.

QDA

The first model we apply is QDA model which is a more advanced and flexible version of a linear model as well as LDA as it is used to find a non-linear boundary between our classifiers, and assumes that each class follows a Gaussian distribution. Because we find a curved decision boundary might occur in the EDA part, we include the model.



```
qda_model<-discrim_quad()%>%
  set_mode("classification")%>%
  set_engine("MASS")
qda_wrkflw<-workflow()%>%
  add_model(qda_model)%>%
  add_recipe(recipe1)
qda_res<-fit(qda_wrkflw,pcos_training1)
augment(qda_res,pcos_training1)%>%
  roc_curve(`PCOS (Y/N)`,.pred_0)%>%
  autoplot()
```



```
qda_roc_auc<-augment(qda_res,pcos_training1)%>%
  roc_auc(`PCOS (Y/N)`,.pred_0)
qda_roc_auc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.981
```

We can see that QDA model performs relatively well in training data as its `roc_auc` reaches 0.981438.

Elastic Net Glm

Since there are too many predictors in the data, we prefer shrinking the coefficients toward zero. Thus, the second model we consider is the elastic neg logistic linear model. Moreover, we hope to further increase the prediction accuracy by balancing well between variance and bias.

```
en_model<-logistic_reg(mixture = tune(),penalty = tune())%>%
  set_mode("classification")%>%
  set_engine("glmnet")
en_wrkflw<-workflow()%>%
  add_recipe(recipe1)%>%
  add_model(en_model)
en_grid <- grid_regular(penalty(range = c(0, 1),
                                     trans = identity_trans()),
                        mixture(range = c(0, 1)),
                             levels = 10)
```

```r
en_tune_res<-tune_grid(
  en_wrkflw,
  resamples = pcos_folds,
  grid = en_grid,
  control = control_grid(save_pred = TRUE)
)
save(en_tune_res, file = "en_tune_class.rda")
```

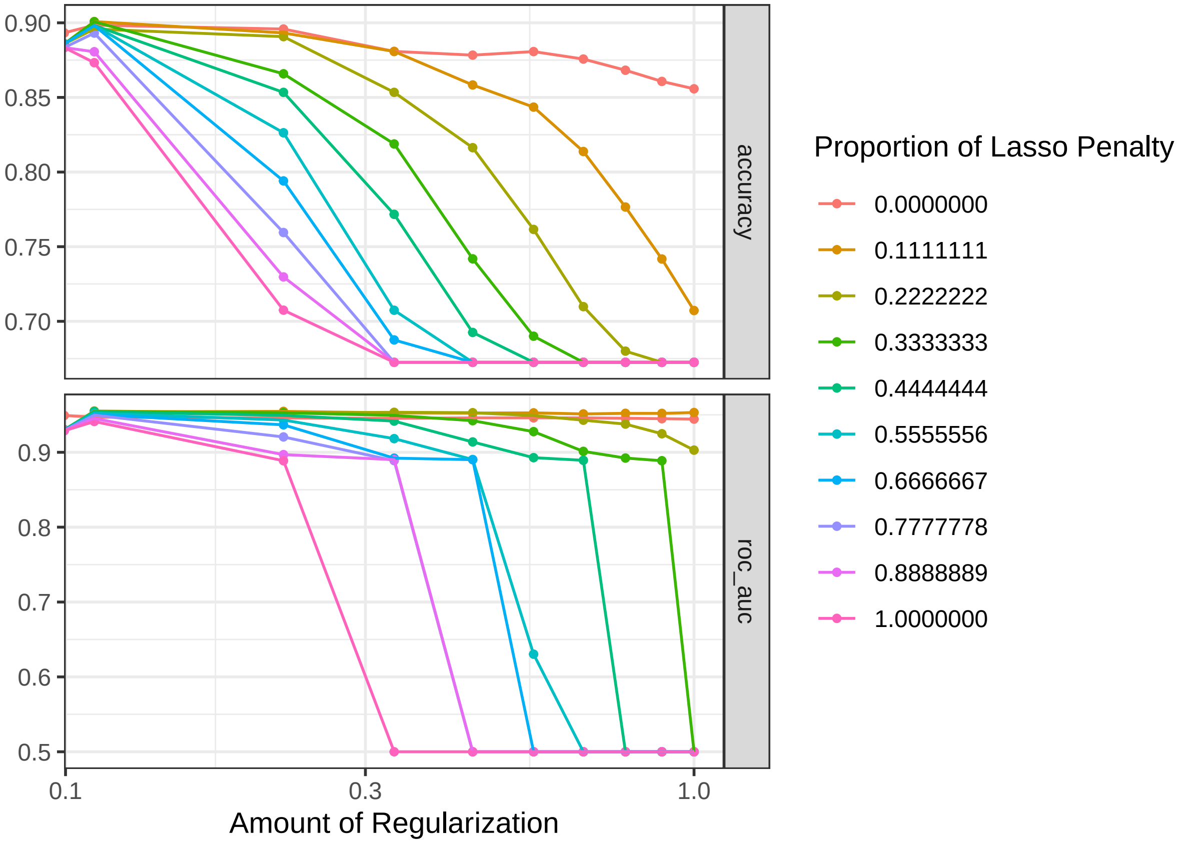
```r
load("en_tune_class.rda")
collect_metrics(en_tune_res)
```

```
## # A tibble: 200 x 8
##    penalty mixture .metric  .estimator  mean     n std_err .config
##      <dbl>   <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
##  1   0           0 accuracy binary     0.893    10  0.0153 Preprocessor1_Model0~
##  2   0           0 roc_auc  binary     0.949    10  0.0125 Preprocessor1_Model0~
##  3   0.111       0 accuracy binary     0.898    10  0.0129 Preprocessor1_Model0~
##  4   0.111       0 roc_auc  binary     0.947    10  0.0139 Preprocessor1_Model0~
##  5   0.222       0 accuracy binary     0.896    10  0.0142 Preprocessor1_Model0~
##  6   0.222       0 roc_auc  binary     0.946    10  0.0144 Preprocessor1_Model0~
##  7   0.333       0 accuracy binary     0.881    10  0.0174 Preprocessor1_Model0~
##  8   0.333       0 roc_auc  binary     0.945    10  0.0144 Preprocessor1_Model0~
##  9   0.444       0 accuracy binary     0.878    10  0.0165 Preprocessor1_Model0~
## 10   0.444       0 roc_auc  binary     0.946    10  0.0144 Preprocessor1_Model0~
## # i 190 more rows
```

```r
autoplot(en_tune_res)
```

Table 2: Best elastic net glm

penalty	mixture	.config
0.1111111	0.2222222	Preprocessor1_Model022



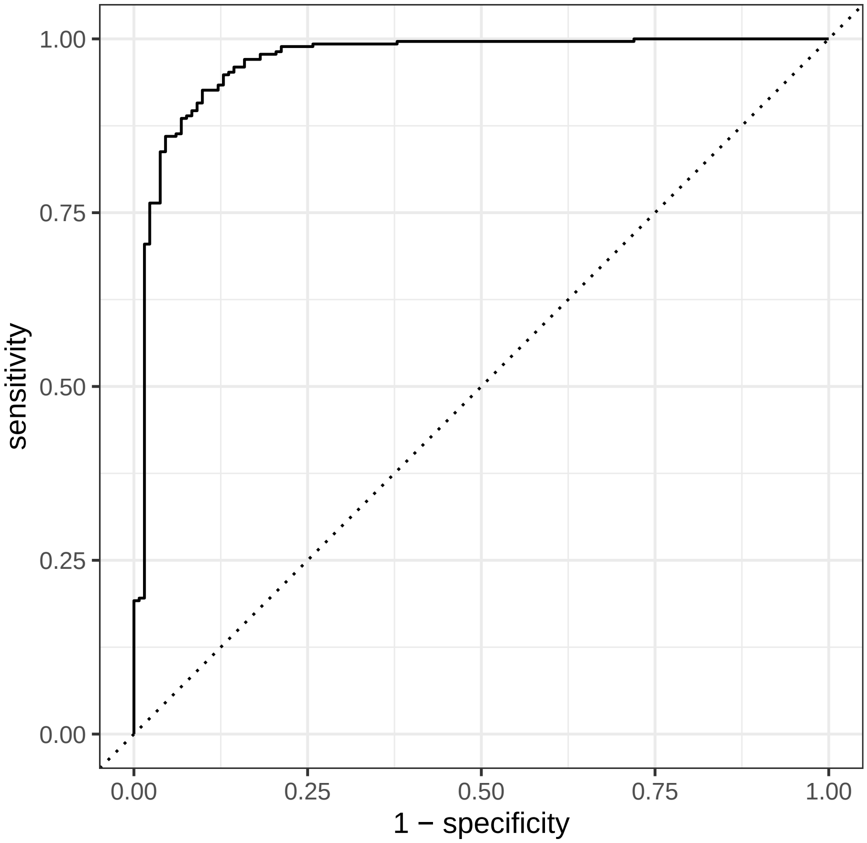
```
best_en_class<-select_best(en_tune_res,
                           metric = "roc_auc",
                           penalty,
                           mixture)
best_en_class%>%
  kable(caption="Best elastic net glm")
```

Since this is the first model with tuning process, we show how combinations of `penalty` and `mixture` affects the metrics in a table then we visualize the result.

From the plot, it is obvious that the more amount of regularization, the accuracy and `roc_auc` first increase slightly and then quickly falls. When it comes to `penalty`, generally speaking, the larger, the worse the performance is. That is to say, ridge has more weight.

The interpretation of the plot corresponds to the best model we selected.

```
final_en_model<-finalize_workflow(en_wrkflw,best_en_class)
fit(final_en_model,data=pcos_training1)%>%
  augment(new_data=pcos_training1)%>%
  roc_curve(`PCOS (Y/N)`,.pred_0)%>%
  autoplot()
```



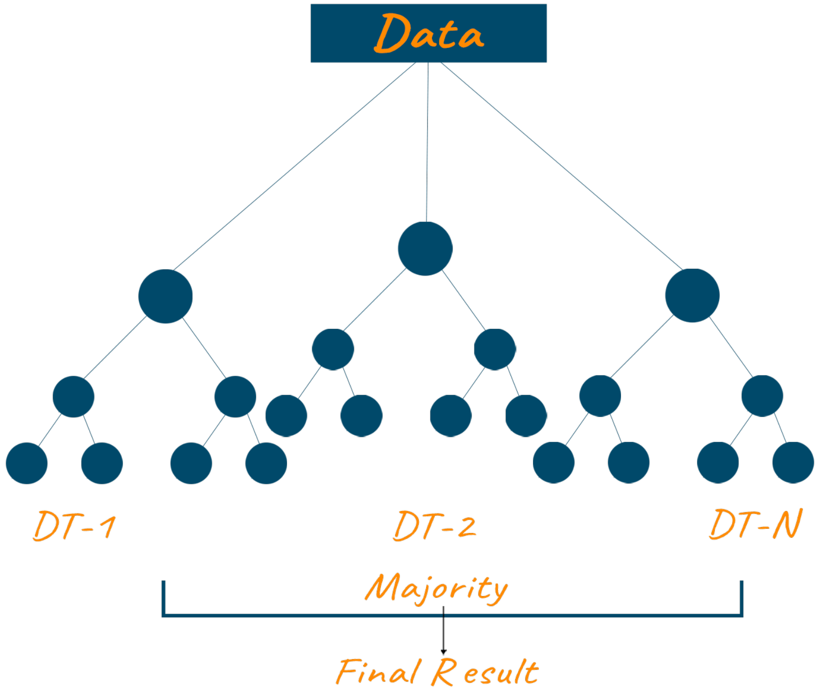
```
en_roc_auc<-fit(final_en_model,data=pcos_training1)%>%
  augment(new_data=pcos_training1)%>%
    roc_auc(`PCOS (Y/N)`,.pred_0)
en_roc_auc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.967
```

We are pleasant to find that the elastic net glm performs better than the previous model.

Random Forest

Now we try the random forest, which is known for model predicting. It is heavily data driven and rather flexible. Also, it can help with identifying the important predictors.



(b) Random Forest

Here we tune 3 hyper-parameters. To decide the adequate range of the hyper parameters, we follow the principles that `mtry` is smaller than `sqrt(p)`, which is roughly 6 according to our data set and the trees should be deep.

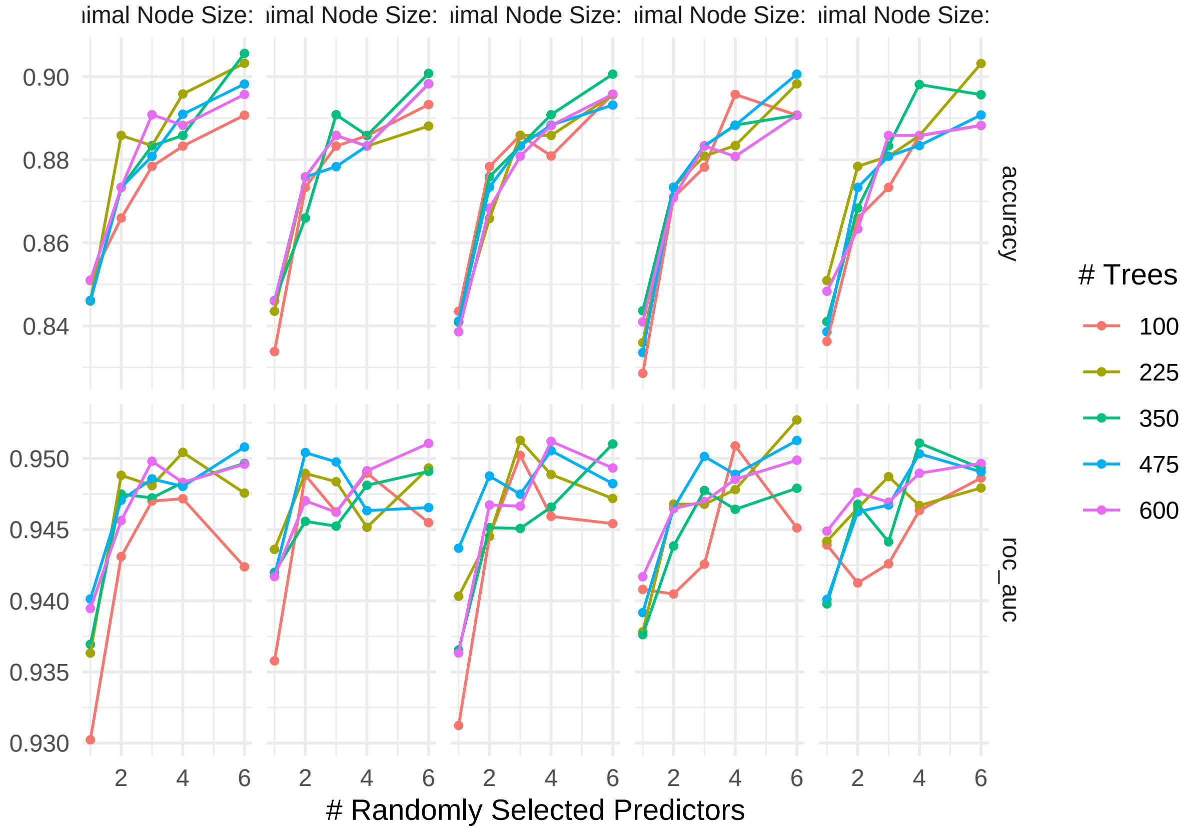
```
rf_class_spec<-rand_forest(mtry=tune(),
                           trees = tune(),
                           min_n = tune())%>%
  set_engine("ranger",importance="impurity")%>%
  set_mode("classification")
rf_class_wf<-workflow()%>%
  add_model(rf_class_spec)%>%
  add_recipe(recipe1)
rf_grid<-grid_regular(mtry(range=c(1,6)),
                      trees(range=c(100,600)),
                            min_n(range=c(20,30)),
                            levels=5)
```

```
rf_tune_class <- tune_grid(
  rf_class_wf,
  resamples = pcos_folds,
  grid = rf_grid,
  control = control_grid(save_pred = TRUE)
)
save(rf_tune_class, file = "rf_tune_class.rda")
```

Table 3: Best random forest model

mtry	trees	min_n	.config
6	225	27	Preprocessor1_Model085

```r
load("rf_tune_class.rda")
autoplot(rf_tune_class) + theme_minimal()
```

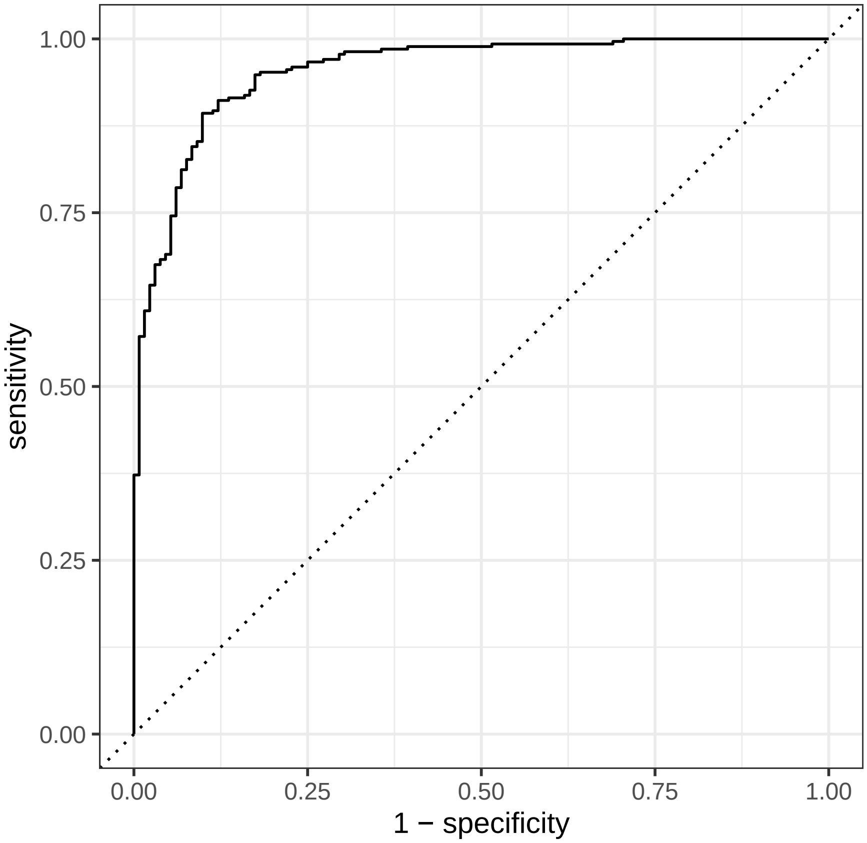


```r
best_rf_class<-select_best(rf_tune_class,"roc_auc",n=1)
best_rf_class%>%
  kable(caption="Best random forest model")
```

- mtry: As the number of randomly selected predictors increases, the metrics show a pattern of going up, then down, and up again, followed by a subsequent decline.
- trees: The number of trees does not show a significant impact on the metrics
- min_n: The number of trees does not show a significant impact on the metrics

It is satisfactory that all three parameters of best model are in the middle of the range, indicating that the selection of range is wise.

```r
rf_roc<-rf_tune_class%>%
  collect_predictions(parameters = best_rf_class)%>%
  roc_curve(`PCOS (Y/N)`,.pred_0)
rf_roc%>%autoplot()
```



```
final_rf_model<-finalize_workflow(rf_class_wf,best_rf_class)
rf_roc_auc<-fit(final_rf_model,data=pcos_training1)%>%
  augment(new_data=pcos_training1)%>%
    roc_auc(`PCOS (Y/N)`,.pred_0)
rf_roc_auc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.998
```

The `roc_auc` of Random forest becomes even higher and it can predict on the training set extremely accurately.

SVM

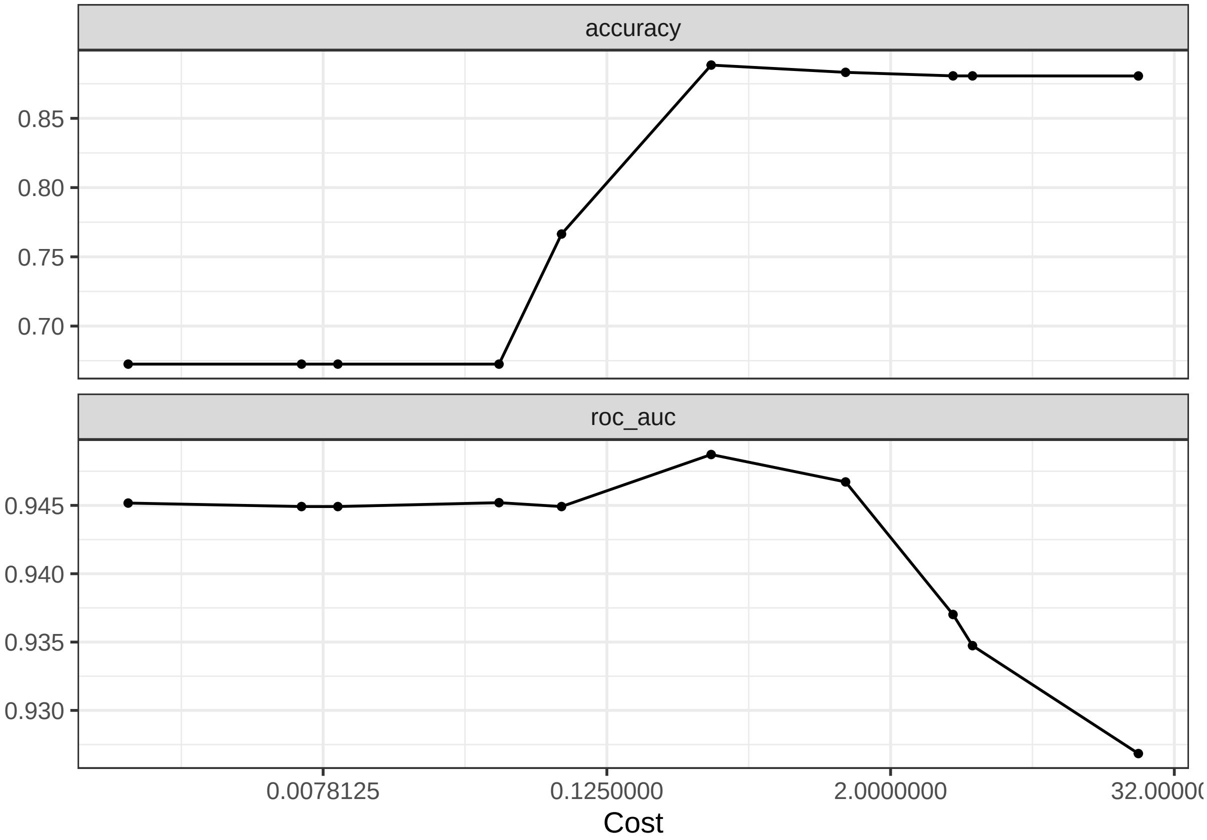
Since SVM is powerful in predicting and our model fit the requirements, the last model is SVM.

```
svm_rbf_spec<-svm_rbf(cost=tune())%>%
  set_mode("classification")%>%
  set_engine("kernlab")
svm_rbf_wkflow<-workflow()%>%
  add_recipe(recipe1)%>%
  add_model(svm_rbf_spec)
svm_rbf_grid<-grid_regular(cost(),
                           levels=5)
svm_rbf_res<-tune_grid(svm_rbf_wkflow,pcos_folds,svm_rbf_grid,
```

Table 4: Best svm model

cost	.config
0.3464286	Preprocessor1_Model09

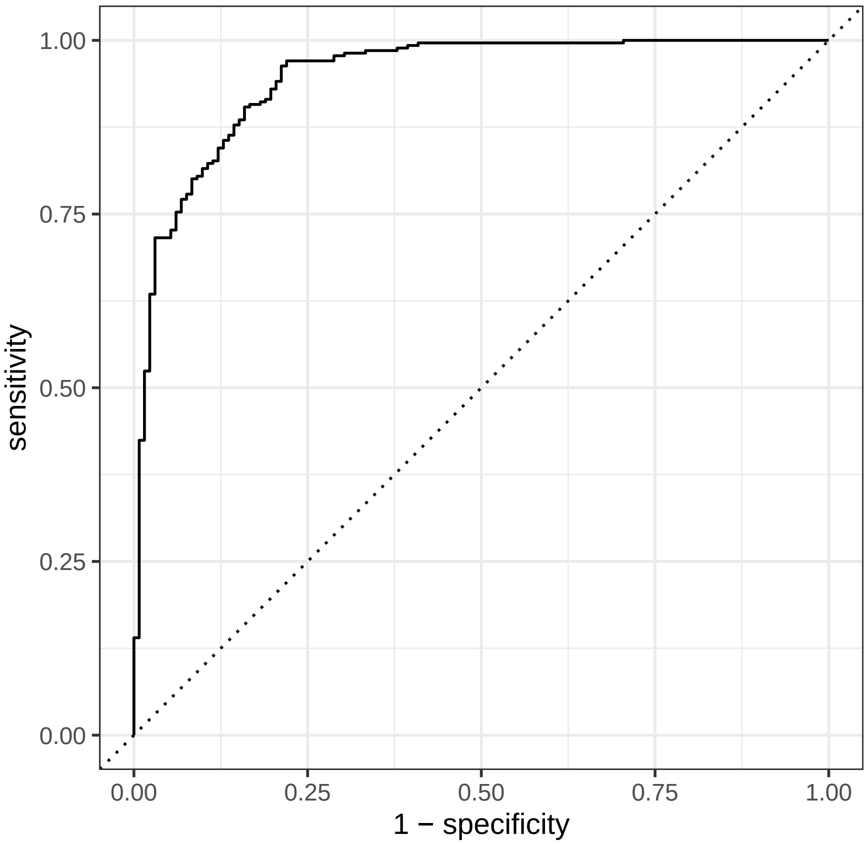
```
                           control = control_grid(save_pred = TRUE))
svm_rbf_res%>%autoplot()
```



```
best_svm_class<-select_best(svm_rbf_res,metric = "roc_auc")
best_svm_class%>%
  kable(caption="Best svm model")
```

We can see that as `cost` enlarges, the metrics first turn better, reaching a peak and then slow down. Thus we choose the peak as the best model.

```
svm_rbf_res%>%
  collect_predictions(parameters = best_svm_class)%>%
  roc_curve(`PCOS (Y/N)`,.pred_0)%>%
  autoplot()
```



```
final_svm_model<-finalize_workflow(svm_rbf_wkflow,best_svm_class)
svm_roc_auc<-fit(final_svm_model,data=pcos_training1)%>%
  augment(new_data=pcos_training1)%>%
    roc_auc(`PCOS (Y/N)`,.pred_0)
svm_roc_auc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.980
```

The SVM also does good, slightly worse than the random forest.

```
final_svm_model_last<-last_fit(final_svm_model,pcos_insplit)
final_svm_metrics<-final_svm_model_last%>%
  collect_metrics()
final_svm_auc_roc<-final_svm_metrics$.estimate[2]
```

Model Selection

After analyzing each type of models, the following step is to select across different model. The criteria is the `roc_auc` of each best model on the overall training set.

```
auc_diffmodel<-data.frame(model=c("QDA","Elastic net glm","Random forest","SVM"),
                          auc_roc=c(qda_roc_auc$.estimate,en_roc_auc$.estimate,rf_roc_auc$.estimate,svm_
auc_diffmodel
```

```
##              model   auc_roc
## 1              QDA 0.9814380
## 2 Elastic net glm 0.9668735
## 3   Random forest 0.9979313
## 4              SVM 0.9804316
```

By comparing the `auc_roc`, we find that random forest performs best, following by the SVM.

Therefore, our final model is the random forest model.

Result of the best models

Fit on the testing set

To assess the performance of the model eventually, we should fit the final model to the testing set.

```
final_rf_model_last<-last_fit(final_rf_model,pcos_insplit)
final_rf_metrics<-final_rf_model_last%>%
  collect_metrics()
final_rf_auc_roc<-final_rf_metrics$.estimate[2]
final_rf_auc_roc
```

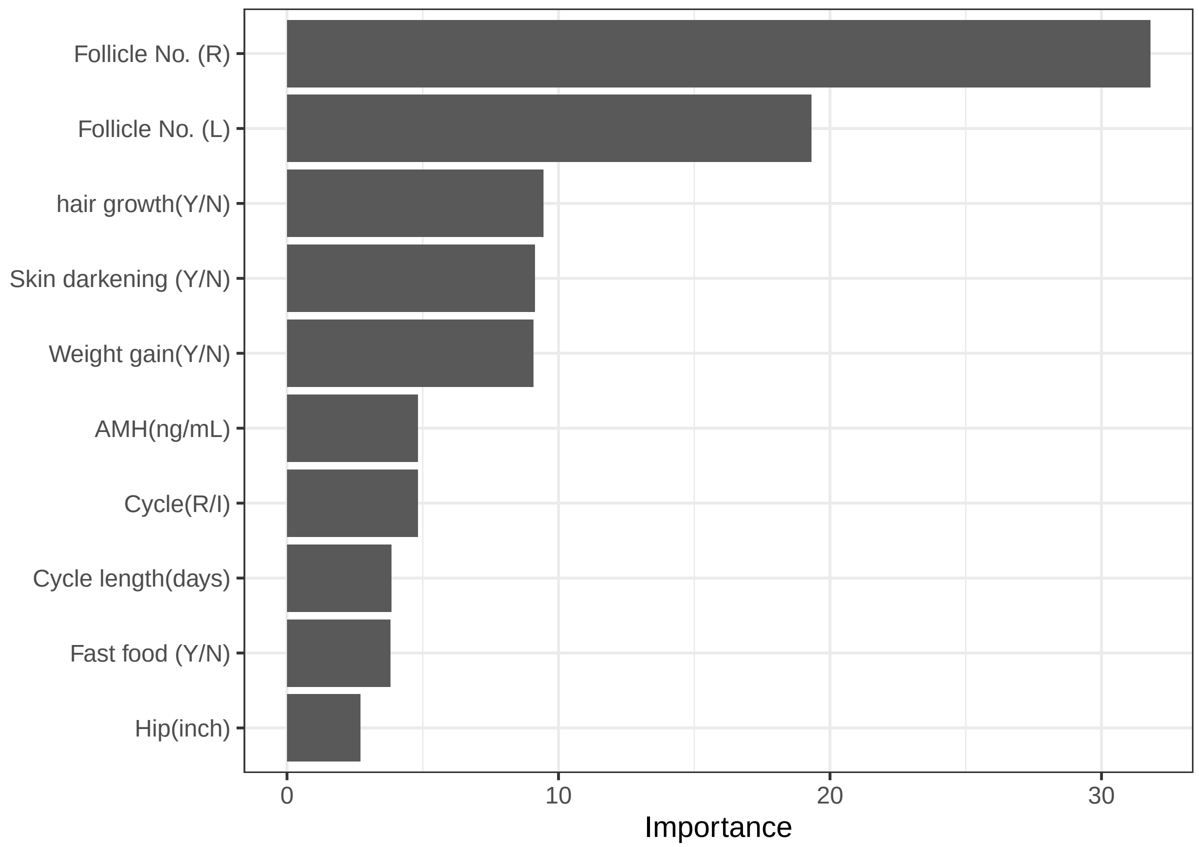
```
## [1] 0.9677822
```

Fortunately, the `roc_auc` of random forest on the testing set is 0.9677822, which is also satisfactory. It is lower than that on the training set but the gap is not large, indicating that the overfitting problem does not exist.

Variable Importance

So far, our primary goal, which is to accurately diagnose PCOS , has been fully realized. However, our model has more potential. For example, locating the decisive diagnose metrics.

```
final_rf_model_last%>%
  extract_fit_parsnip() %>%
  vip()
```



Thanks to `vip()`, we plot the importance of the predictors.

`Follicile No. (R)` and `Follicile No. (L)`rank highest, confirming the initial conclusion drawn from the EDA section. As mentioned above, they refer to the Quantity of developing egg follicles in the right and the left ovary respectively. They are both the result of the ultrasound examination, demonstrating that an ultrasound test is still the necessary and essential method to diagnose PCOS.

Additionally, three predictors from symptom categories follows closely behind, which are `Skin darkening`, `hair growth` and `Weight gain`. Such findings conveys a good news that there is a simple and convenient self-test for us to assess our risk of having PCOS and is of great importance in potential case screening,preventing and alerting.

Until now, we accomplished all the purposes we've proposed at the beginning.

Conclusion

Throughout the project, we have tidied, explored, and analyzed the data of PCOS cases and build a predictive model for it. Eventually, the random forest model outperforms other three and does a fantastic job in both training data and the testing data.

As for further extensions, one may try the neural network so that more predictors can be included. For example, for each individual, in different phase of period, the hormones and the result of the ultra soudn testing vary a lot. Therefore, it would be more accurate if there is a time series for each person.

Although I am glad that the prediction performs well in the data set, I am also confused that why predicting still remains a challenge in the reality since all the parameters in the data set are relatively easy to gather. I hypothesize that it may due to the limited example included in the data and they are from a limited number of hospitals. Consequently, the response variable, whether the woman has the disease, is defined by a relatively small group of people, namely the doctors in those hospital. By establishing the model, we nearly wonderfully detect their diagnosis pattern but here we assumes that their diagnosis is correct and accurate, which dissappointedly, may not be the case. Therefore, there is a potential concern. To improve, more observations from other areas should be also gathered. Also a cross consultation towards the same patient by different doctors would be even better and in fact, it is the normal case that is closer to the reality. Take me as an example, I visited quite a few doctors and finally draw a comprehensive conclusion.

Overall, I feel fulfilled since what I have learned in the machine learning class can be applied into my personal health issue and partially solve my confusion. Getting familiar with the data, I am acquainted to the medical term and more I learn about it, less fear I am to it and eventually more determined to conquer it.

Source

Kaggle data set “Polycystic ovary syndrome (PCOS) Mayo Foundation for Medical Education and Research