Using Machine Learning Algorithms to Predict the likelihood of PCOS based on Demographic, Clinical and Lifestyle Factors

Leticia Salazar

May 16, 2023

Contents

Abstract:

Polycystic Ovary Syndrome (PCOS) stands as a prevalent endocrine disorder affecting women of reproductive age worldwide, presenting a confluence of hormonal imbalances, reproductive irregularities, and potential metabolic complications. Characterized by irregular menstrual cycles, hyperandrogenism, and polycystic ovaries, PCOS poses multifaceted challenges that extend beyond reproductive health, encompassing metabolic disturbances and psychological implications. This complex syndrome, rooted in intricate interplays of genetic predispositions, hormonal dysregulation, and environmental factors, manifests variably among affected individuals. The clinical landscape of PCOS often requires a comprehensive, multidisciplinary approach, incorporating lifestyle modifications, pharmacological interventions, and personalized treatments to address symptoms and reduce associated health risks. Despite ongoing research efforts, elucidating the precise etiology and optimal management strategies for PCOS remains a dynamic area of exploration in contemporary medicine. This research project aims investigates the current studies on PCOS diagnosis, assessing the effectiveness of various machine learning algorithms including Linear Regression (LM), Logistic Regression, Decision Tree (rpart), Random Forest (randomForest), Gradient Boosting Machines (xgboost or gbm), Support Vector Machines (e1071), Neural Networks (neuralnet), and K-Nearest Neighbors (kknn or class). From my research

Key words:

Polysyctic Ovary Syndrome, Polycystic Ovarian Syndrome, PCOS, Women's Health, Machine Learning, Linear Regression, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Super Vector Machine, Neural Networks, K-Nearest Neighbors

The Problem:

Polycystic Ovary Syndrome (PCOS) is a hormonal imbalance disorder affecting women of reproductive age. Determining the precise global count of women affected by PCOS poses challenges due to many cases remaining undiagnosed. However, the World Health Organization (WHO) estimates that approximately 3.4% of women are affected [1]. While this percentage might seem relatively small, considering that women constitute 49.7% of today's population [2], nearly 13% of them fall within the reproductive age bracket [3]. This data suggests that approximately 17.5 million women report suffering from PCOS. Beyond the physical and emotional toll PCOS exacts, it also disrupts ovarian function, leading to challenges in maintaining a healthy menstrual cycle and can result in the formation of cysts, ultimately impacting fertility. It's important to note that the prevalence of PCOS varies by region and ethnic groups, with some studies suggesting higher rates of PCOS in certain populations [4]. Early identification of risk factors associated with PCOS can assist in timely interventions and lifestyle adjustments. Tailoring suggestions or treatments based on individualized risk profiles has the potential to enhance patient outcomes. Employing predictive models can play a pivotal role in increasing awareness regarding PCOS risk factors and preventive measures. However, limitations may exist in accessing comprehensive and varied datasets containing accurate demographic, clinical, and lifestyle data. It is crucial to ensure the model's transparency and interpretability to facilitate well-informed decisions and recommendations. Additionally, it's imperative that the model demonstrates proficiency across diverse demographic groups and populations. Addressing these challenges involves the development of a robust predictive model using machine learning techniques. Such an endeavor holds the promise of significantly contributing to the identification of individuals at risk of PCOS, thereby enabling early interventions and guiding personalized healthcare strategies for improved management of the condition. This project will investigate publicly available datasets that will be used to develop machine learning models that predicts the probability or risk of an individual having or developing PCOS based on demographic information (age, ethnicity, geographical location), clinical data (hormonal levels, BMI, menstrual irregularities), and lifestyle factors (dietary habits, exercise routine, stress levels).

Objectives:

Construct predictive models using machine learning techniques that utilize a dataset comprising demographic, clinical, and lifestyle variables as features and PCOS diagnosis as the target variable. Machine learning algorithms have shown promise in advancing our understanding of the disease and improving its diagnosis and treatment.

I anticipate answering the following questions with my data:

1. Are there commonalities women with and without PCOS have that can be easily dismissed as normal?
2. Are there differences for women of different race/ethnic background when it comes to having PCOS? What about women without PCOS?
3. What is the likelihood of a woman developing PCOS based on her age, ethnicity, and BMI history?
4. Can we predict the risk of insulin resistance, diabetes, and cardiovascular disease in women with PCOS based on their medical history, hormone levels, and lifestyle factors?

5. Can we predict the likelihood of successful pregnancy outcomes in women with PCOS based on their age, weight, hormone levels, and treatment history?

6. Can we predict the long-term health outcomes and quality of life of women with PCOS based on their age, lifestyle factors, hormone levels, and treatment history?

Literature Review:

For this project, my emphasis was on discovering literature reviews that validate the relevance of the dataset utilized alongside the chosen machine learning algorithms. In pursuit of this goal, I've come across various articles that make reference to the Kaggle dataset. These articles investigate studies associated with PCOS, presenting findings aimed at providing a deeper comprehension of the dataset I'm analyzing.

The literature reviews collectively delve into various aspects of Polycystic Ovarian Syndrome (PCOS), employing diverse methodologies and approaches for diagnosis, classification, understanding clinical manifestations, and proposing potential treatments. Researchers across these studies have primarily utilized data mining, machine learning, and clinical investigations to address the complexity of PCOS. Below are the key characteristics, achievements, advantages, and drawbacks across these reviews:

Common Focus Areas: Multiple studies emphasize the use of machine learning algorithms, such as Naïve Bayes, Decision Trees, Artificial Neural Networks, Support Vector Machines, and ensemble methods, for PCOS diagnosis. They explore the accuracy and predictive power of these models using diverse datasets, including clinical, lifestyle, and physiological parameters. Investigations into clinical parameters and anthropometric measures aim to identify potential predictors or indicators of PCOS, such as hormonal imbalances, insulin resistance, metabolic traits, obesity, and associated risks like infertility and cardiovascular issues. Studies examine the diverse clinical presentations and phenotypic variations within PCOS, shedding light on how different subgroups may manifest the syndrome and respond to various treatments.

Achievements: Machine learning models, particularly those utilizing Convolutional Neural Networks (CNNs) and ensemble methods, have shown high accuracy in diagnosing PCOS. These models effectively utilize diverse datasets, ranging from ultrasound images to clinical parameters. Research has identified several potential predictive factors for PCOS, including hormonal markers, lifestyle attributes, and metabolic indicators, offering insights into early detection and tailored treatment strategies. Studies evaluating PCOS awareness among women have highlighted the importance of education and awareness campaigns in enhancing understanding and facilitating early diagnosis.

Advantages: Machine learning algorithms offer promising accuracy rates, particularly CNNs and ensemble models, in diagnosing PCOS using various non-invasive parameters. Understanding phenotypic variations aids in tailoring treatments based on specific subgroups, potentially improving patient outcomes and management. Efforts toward identifying early markers or predictive factors can facilitate early intervention and lifestyle modifications, mitigating long-term health risks associated with PCOS.

Drawbacks and Recommendations: Some studies may suffer from limited sample sizes or datasets, impacting the generalizability of findings. Larger and more diverse datasets are recommended for robust model development and validation. While certain machine learning models showcase high accuracy, the variability in dataset characteristics and preprocessing methods could influence their effectiveness across different populations or settings. The multifaceted nature of PCOS, influenced by both genetic and environmental factors, presents challenges in pinpointing a singular cause or standard diagnostic criteria.

The collective body of literature reviews signifies advancements in PCOS diagnosis, understanding clinical manifestations, and potential avenues for tailored treatments. The use of machine learning models, especially those employing CNNs and ensemble methods, showcases significant promise in accurate PCOS diagnosis based on non-invasive parameters. However, further research is necessary, emphasizing larger and more diverse datasets, refined models, and a multidisciplinary approach to fully comprehend the complexity of PCOS and enhance diagnostic and treatment strategies.

Dataset:

The Polycystic ovary syndrome (PCOS) dataset, available on Kaggle.com, is comprised of two csv files labeled `PCOS_data_without_infertility` and `PCOS_infertility`. In total, these files encompass 48 variables and 541 data entries all collected from 10 different hospitals across Kerala, India. The dataset contains all physical and clinical parameters to determine PCOS and infertility related issues.

Full description of the variables below:

- Units used range from imperial to metric system of measurement
- For Yes | No questions
  - Yes = 1
  - No = 0

Variables	Description
"Sl..No"	unique identification number assigned to each entry
"Patient.File.No."	file number for each patient's record.
"PCOS..Y.N."	indicates the presence or absence of PCOS, with "Y" denoting "1 or Yes" and "N" for "0 or No
"I…beta.HCG.mIU.mL."	pregnancy hormone case I measured in milli-international units per liter (mIU/L)
"II….beta.HCG.mIU.mL."	pregnancy hormone case II measured in milli-international units per liter (mIU/L)
"AMH.ng.mL."	detects ovarian reserve (egg count)
"Age..yrs."	age of patient in years
"Weight..Kg."	weight of patient in kg
"Height.Cm."	height of patient in cm
"BMI"	body mass index
"Blood.Group"	Blood Groups: A+ = 11, A- = 12, B+ = 13, B- = 14, O+ = 15, O- = 16, AB+ = 17, AB- = 1
"Pulse.rate.bpm."	beats per minute
"RR..breaths.min."	respiration rates per minute
"Hb.g.dl."	hemoglobin concentration measured in grams per deciliter (g/dL).
"Cycle.R.I."	cycle Regularity Index used to assess the regularity or irregularity of menstrual cycles in women
"Cycle.length.days."	length of menstrual cycle
"Marraige.Status..Yrs."	years married
"Pregnant.Y.N."	pregnant yes or no
"No..of.aborptions"	number of abortions
"FSH.mIU.mL."	follicle stimulating hormone measured in milli-international units per liter (mIU/L)
"LH.mIU.mL."	luteinizing hormone (increases during ovulation) measured in milli-international units per liter
"FSH.LH"	ratio between Follicle-Stimulating Hormone (FSH) and Luteinizing Hormone (LH)
"Hip.inch."	measurement of hips in inches
"Waist.inch."	measurement of waist in inches
"Waist.Hip.Ratio"	ratio of measurement of waist and hip
"TSH..mIU.L."	thyroid stimulating hormone measured in milli-international units per liter (mIU/L)
"AMH.ng.mL."	Anti-Müllerian Hormone (AMH) measured in nanograms per milliliter (ng/mL); a marker used

Variables	Description
"PRL.ng.mL."	Prolactin measured in nanograms per milliliter (ng/mL); a hormone produced by the pituitary
"Vit.D3..ng.mL."	Vitamin D3 measured in nanograms per milliliter (ng/mL); is essential for bone health, immun
"PRG.ng.mL."	Progesterone measured in nanograms per milliliter (ng/mL); a hormone involved in the menstru
"RBS.mg.dl."	Random Blood Sugar measured in milligrams per deciliter (mg/dL); it represents the level of gl
"Weight.gain.Y.N."	weight gain yes or no
"hair.growth.Y.N."	hair growth yes or no (hirsutism)
"Skin.darkening..Y.N."	darkening of skin yes or no
"Hair.loss.Y.N."	hair loss yes or no
"Pimples.Y.N."	pimples (acne) yes or no
"Fast.food..Y.N."	consumption of fast food yes or no
"Reg.Exercise.Y.N."	regularly exercise yes or no
"BP._Systolic..mmHg."	systolic blood pressure measured in millimeters of mercury (mmHg)
"BP._Diastolic..mmHg."	diastolic blood pressure measured in millimeters of mercury (mmHg)
"Follicle.No...L."	number of follicles on left ovary
"Follicle.No...R."	number of follicles in right ovary
"Avg..F.size..L...mm."	average size of follicles in left ovary measured in millimeters (mm)
"Avg..F.size..R...mm."	average size of follicles in right ovary measured in millimeters (mm)
"Endometrium..mm."	size of the endometrial thickness in millimeters (mm)

Methodology:

Data Collection and Preparation: Gather and preprocess a comprehensive dataset containing demographic details, clinical measurements (hormone levels, BMI, menstrual history), and lifestyle information (diet, exercise, stress levels) from a diverse population.

Feature Selection and Engineering: Identify the most relevant features through exploratory analysis and feature engineering techniques, ensuring that the model focuses on key predictors for PCOS.

Model Training and Evaluation: Train the machine learning model using appropriate algorithms (such as Logistic Regression, Random Forest, Support Vector Machines) on a subset of the dataset, validate its performance using cross-validation techniques, and evaluate its accuracy, precision, recall, and F1-score.

Prediction and Risk Assessment: Deploy the trained model to predict the likelihood or risk of PCOS in new, unseen data, providing valuable insights into individuals who might be predisposed to or already have the condition.

Interpretation and Recommendations: Interpret the model's findings, analyze the significance of various factors contributing to the prediction, and offer recommendations or interventions based on identified risk factors to potentially prevent or manage PCOS.

This project will investigate mainly publicly available datasets that will be used to create predictive models on markers in routine test results to make a diagnosis. Some variables that are included in these datasets are:

- Age
- Weight
- BMI
- Race/ethnicity
- Family history of PCOS
- Menstrual cycle irregularity
- Hormone levels (e.g., testosterone, LH, FSH)

- Insulin resistance
- Physical activity level
- Diet

I will be using R (statistical performing language) to perform exploratory data analysis to process and analyze the data to check for structural errors and be able to create graphs and perform tests with minimal errors. Once the data is ready to use, I will be splitting the data into a training and test set to be able to use a machine learning algorithm such as logistic regression to create a predictive model. The predictive model will be based on markers (variables mentioned above) used to identify individuals who are at high risk for PCOS and target interventions to manage the condition.

To answer my research question: * The datasets publicly available and the NICHD Dash looking to obtain all have PCOS and non-PCOS patients including demographic information, medical history, and laboratory tests. Preprocess the data by removing missing values, outliers, and redundant variables. Perform feature selection to identify the most informative variables for prediction. * Split the dataset into training, validation, and testing sets. The training set is used to train the machine learning algorithm, the validation set is used to tune hyperparameters and prevent overfitting, and the testing set is used to evaluate the performance of the final model. * Select an appropriate machine learning algorithm for the task at hand, such as logistic regression, decision trees, random forests, support vector machines, or neural networks. Train the algorithm on the training set using various techniques, such as cross-validation and regularization, to optimize its performance. * Evaluate the performance of the trained model on the validation set using various metrics, such as accuracy, precision, recall, F1 score, and area under the curve. Use feature importance analysis to identify the most influential variables for prediction. * Tune the hyperparameters of the machine learning algorithm using grid search, random search, or Bayesian optimization to improve its performance on the validation set. * Select the final model based on its performance on the validation set. Evaluate its performance on the testing set to assess its generalization ability. * Interpret the results of the machine learning algorithm using various techniques, such as decision trees, feature importance analysis, and partial dependence plots. Visualize the results using graphs, charts, and heatmaps to facilitate understanding and communication. * Deploy the trained model on new data and disseminate the findings through scientific publications, presentations, and online platforms.

Note: this methodology plan is not exhaustive and may vary depending on the specific research question, dataset, and machine learning algorithm used.

Note 2: data has already been collected, there is no need for me to gather participants, perform exams (such as bloodwork), use medical equipment to collect the data, perform surveys, have a location to perform a study, etc. I will be the sole person studying the data set and conducting the analysis.

Assumptions:

While there's limited information available in the medical field and even less data sets available to analyze, I have some concerns on being successful in predicting a PCOS diagnosis. Yet there are justifications exploring PCOS in depth:

- Identify diagnostic biomarkers that can distinguish PCOS patients from healthy individuals or those with other disorders. These biomarkers can aid in earlier diagnosis and better management of the disease.
- Predict the likelihood of disease progression and the risk of developing complications, such as diabetes and cardiovascular disease, in PCOS patients. This information can guide treatment decisions and improve patient outcomes.
- Develop personalized treatment plans for PCOS patients based on their individual characteristics and medical history. This approach can lead to more effective and targeted interventions.

- Integrate data from various sources, such as electronic health records, imaging studies, and genetic analyses, to provide a more comprehensive understanding of PCOS. This can help identify new pathways involved in the disease and potential targets for therapy.
- Aid in the design and analysis of clinical trials, leading to more efficient and informative studies. This can accelerate the development of new treatments for PCOS.

Early diagnosis and management of PCOS can lead to better health outcomes, improved quality of life, and reduced long-term health risks. Therefore, predicting PCOS diagnosis can have several societal benefits, including:

- Predicting PCOS diagnosis can help healthcare providers identify women at risk of developing PCOS and intervene early with appropriate treatment, such as lifestyle modifications and medication, to prevent or minimize the long-term health consequences of the disorder.
- Early diagnosis and treatment of PCOS can help manage symptoms such as irregular periods, infertility, acne, and excess hair growth, leading to improved physical and mental health outcomes for affected women.
- By predicting PCOS diagnosis and intervening early, healthcare providers can prevent or reduce the need for more expensive treatments or surgeries later in life, resulting in cost savings for individuals, healthcare systems, and society.
- Predicting PCOS diagnosis can increase awareness of the disorder among healthcare providers, patients, and the public, leading to more education, research, and advocacy efforts aimed at improving PCOS diagnosis, treatment, and management.
- Early intervention and management of PCOS can improve the quality of life for affected women, leading to increased productivity, better mental health, and greater overall well-being.

Overall, I'll be able to explore the insights into PCOS pathophysiology, diagnosis, and treatment. Their use in PCOS research can lead to more personalized and effective care for patients with this complex disorder.

Experimentation and Results:

Conclusion:

My initial assumption was that while there's limited information available in the medical field and even less data sets available to analyze, I have some concerns on being successful in predicting a PCOS diagnosis.

References:

1. Bulsara, J., Patel, P., Soni, A., & Acharya, S. (2021, February 10). A review: Brief insight into polycystic ovarian syndrome. Endocrine and Metabolic Science. Retrieved February 23, 2023, from https://www.sciencedirect.com
2. World female population, 1960-2022. Knoema. (2022). Retrieved February 24, 2023, from https://knoema.com.

3. MarchofDimes. (2022). Population of women 15-44 years by age: United States, 2020. March of Dimes | PeriStats. Retrieved February 24, 2023, from https://www.marchofdimes.org

4. Engmann, L., Jin, S., Sun, F., Legro, R. S., Polotsky, A. J., Hansen, K. R., Coutifaris, C., Diamond, M. P., Eisenberg, E., Zhang, H., Santoro, N., & Reproductive Medicine Network (2017). Racial and ethnic differences in the polycystic ovary syndrome metabolic phenotype. American journal of obstetrics and gynecology, 216(5), 493.e1–493.e13. https://doi.org

5. Fehring, Richard J., "Menstrual Cycle Data" (2012). Randomized Comparison of Two Internet-Supported Methods of Natural Family Planning. 7. https://epublications.marquette.edu

6. Khan, M. J., Ullah, A., & Basit, S. (2019). Genetic Basis of Polycystic Ovary Syndrome (PCOS): Current Perspectives. The application of clinical genetics, 12, 249–260. https://doi.org/

Appendices:

Appendix A - Figures:

Appendix B - Tables:

Appendix C - R Code:

```
# load libraries
library(tidyverse) # data prep
library(DataExplorer) # histograms for datasets
library(skimr) # data prep
library(rpart) # decision tree package
library(rpart.plot) # decision tree display package
library(kableExtra) # kable function for tables
library(knitr) # kable function for table
library(tidyr) # splitting data
library(ggplot2) # graphing
library(hrbrthemes) # chart customization
library(gridExtra) # layering charts
library(stringr) # data prep
library(tidymodels) # predictions
library(corrplot) # correlation plot
library(randomForest) # for the random forest
library(caret) # confusion matrix
library("e1071") #svm
library(formattable)
library(corrplot) # correlation plot
library(caret) # confusion matrix
library(neuralnet) # neural network
library(stats) # linear and logistic regression
```

```r
library(gbm) # generalized boosted models
library(xgboost) # extreme gradient boosting
library(kknn) # weighted k-Nearest neighbors
library(jtools)  # use of summ()
library(patchwork) # ggplot2 multiplot title
library(class) # knn function

# load the dataset from github
pcos <- read.csv("https://raw.githubusercontent.com/letisalba/Data-698/master/Data-Collection-and-Analys
pcos2 <- read.csv("https://raw.githubusercontent.com/letisalba/Data-698/master/Data-Collection-and-Analy

# display the `pcos` dataset
pcos %>%
  kable(caption = "<font color=#000000><b>Table 1.</b>`pcos` data </font>", format = "html", col.names =
  kable_styling(bootstrap_options = c("hover", "condensed"), font_size = 13) %>%
  kableExtra::scroll_box(width = "100%", height = "400px")

# display the `pcos2` dataset
pcos2 %>%
kable(caption = "<font color=#000000><b>Table 2.</b>`pcos2` data </font>", format = "html", col.names =
  kable_styling(bootstrap_options = c("hover", "condensed"), font_size = 13) %>%
  kableExtra::scroll_box(width = "100%", height = "400px")

# summary of the pcos dataset
skim(pcos)

# summary of the pcos2 dataset
skim(pcos2)


DataExplorer::plot_histogram(
  geom_histogram_args = list(alpha = 1, fill = "#7e102c"),
  title = "Fig. 1 - Histogram of `pcos` data",
   data = pcos,
         ggtheme=theme_ipsum())

DataExplorer::plot_histogram(
  geom_histogram_args = list(alpha = 1, fill = "#a86800"),
  title = "Fig. 1 - Histogram of `pcos2` data",
   data = pcos2,
         ggtheme=theme_ipsum())

# change variables to numeric
pcos <- mutate_all(pcos, function(x) as.numeric(as.character(x)))
pcos2 <- mutate_all(pcos2, function(x) as.numeric(as.character(x)))


# missing data
colSums(is.na(pcos))

# missing data
colSums(is.na(pcos2))
```

```r
# removing first two column for `pcos` data
pcos <- dplyr::select(pcos, -c(2:6))

# renaming columns for `pcos` data
pcos <- pcos %>%
  rename("Sl.No" = "Sl..No")

# removing columns not needed for `pcos_infertility` data
pcos2 <- dplyr::select(pcos2, -c(2, 45))

# renaming columns for `pcos_infertility` data
pcos2 <- pcos2 %>%
  rename("Sl.No" = "Sl..No",
         "PCOS" = "PCOS..Y.N.",
         "Age_yrs" = "Age..yrs.",
         "Weight" = "Weight..Kg.",
         "Height" = "Height.Cm.",
         "BMI" = "BMI",
         "Blood_Group" = "Blood.Group",
         "Pulse_rate_bpm" = "Pulse.rate.bpm.",
         "RR_breaths_min" = "RR..breaths.min.",
         "Hb_gdl" = "Hb.g.dl.",
         "Cycle_RI" = "Cycle.R.I.",
         "Cycle_length_days" = "Cycle.length.days.",
         "Married_yrs" = "Marraige.Status..Yrs.",
         "Pregnant" = "Pregnant.Y.N.",
         "No_of_abortions" = "No..of.aborptions",
         "IbetaHCG_mIUmL" = "I...beta.HCG.mIU.mL.",
         "IIbetaHCG_mIUmL" = "II....beta.HCG.mIU.mL.",
         "FSH_mIUmL" = "FSH.mIU.mL.",
         "LH_mIUmL" = "LH.mIU.mL.",
         "FSH_LH" = "FSH.LH",
         "Hip" = "Hip.inch.",
         "Waist" = "Waist.inch.",
         "Waist_Hip_Ratio" = "Waist.Hip.Ratio",
         "TSH_mIUmL" = "TSH..mIU.L.",
         "AMH_ngmL" = "AMH.ng.mL.",
         "PRL_ngmL" = "PRL.ng.mL.",
         "Vit_D3_ngmL" = "Vit.D3..ng.mL.",
         "PRG_ngmL" = "PRG.ng.mL.",
         "RBS_mgdl" = "RBS.mg.dl.",
         "Weight_gain" = "Weight.gain.Y.N.",
         "Hair_growth" = "hair.growth.Y.N.",
         "Skin_darkening" = "Skin.darkening..Y.N.",
         "Hair_loss" = "Hair.loss.Y.N.",
         "Pimples" = "Pimples.Y.N.",
         "Fast_food" = "Fast.food..Y.N.",
         "Reg_Exercise" = "Reg.Exercise.Y.N.",
         "BP_Systolic_mmHg" = "BP._Systolic..mmHg.",
         "BP_Diastolic_mmHg" = "BP._Diastolic..mmHg.",
         "Follicle_NoL" = "Follicle.No...L.",
         "Follicle_NoR" = "Follicle.No...R.",
         "Avg_F_sizeL_mm" = "Avg..F.size..L...mm.",
```

```
         "Avg_F_sizeR_mm" = "Avg..F.size..R...mm.",
         "Endometrium_mm" = "Endometrium..mm.")

# merge data sets
pcos_data <- merge(pcos, pcos2,  by=c("Sl.No"))
# display the merged dataset
pcos_data %>%
kable(caption = "<font color=#000000><b>Table 3.</b>`pcos_data` merged data </font>", format = "html", c
  kable_styling(bootstrap_options = c("hover", "condensed"), font_size = 13) %>%
  kableExtra::scroll_box(width = "100%", height = "400px")

# convert Height from cm to
pcos_data$"Height" <- round((pcos_data$"Height" * 0.01),1)

# convert hip and waist from inches to cm
pcos_data$"Hip" <- round((pcos_data$"Hip" * 2.54),1)
pcos_data$"Waist" <- round((pcos_data$"Waist" * 2.54),1)

#calculate BMI
pcos_data$"BMI" <- round((pcos_data$"Weight" / pcos_data$"Height"^2), 1)

# calculate waist-hip ratio
pcos_data$"Waist_Hip_Ratio" <- round((pcos_data$"Waist" / pcos_data$"Hip"),2)

# calculate FSH/LH
pcos_data$"FSH_LH" <- round((pcos_data$"FSH_mIUmL"/pcos_data$"LH_mIUmL"),2)

# calculate Married years
pcos_data$"Married(yrs)"[is.na(pcos_data$"Married(yrs)")] <- median(pcos_data$"Married(yrs)", na.rm = T

# calculate Fast food
pcos_data$"Fast_food"[is.na(pcos_data$"Fast_food")] <- median(pcos_data$"Fast_food", na.rm = T)

# calculate
pcos_data$"AMH_ngmL"[is.na(pcos_data$"AMH_ngmL")] <- median(pcos_data$"AMH_ngmL", na.rm = T)

# List of variables to round
vars_to_round <- c("Hb_gdl", "Married_yrs", "IbetaHCG_mIUmL", "IbetaHCG_mIUmL",
                   "FSH_mIUmL", "LH_mIUmL", "FSH_LH", "TSH_mIUmL", "AMH_ngmL", "PRL_ngmL",
                   "Vit_D3_ngmL", "PRG_ngmL", "Avg_F_sizeR_mm", "Endometrium_mm")

# Rounding the variables to 1 decimal places
pcos_data <- pcos_data %>%
  mutate_at(vars(vars_to_round), ~ round(., digits = 1))


# remove 1st column
pcos_cleaned <- pcos_data[-1]

# display results of cleaned pcos
pcos_cleaned %>%
kable(caption = "<font color=#000000><b>Table 4.</b>`pcos_cleaned` dataset </font>", format = "html", c
  kable_styling(bootstrap_options = c("hover", "condensed"), font_size = 13) %>%
```

```r
  kableExtra::scroll_box(width = "100%", height = "400px")

DataExplorer::plot_histogram(
  geom_histogram_args = list(alpha = 1, fill = "dark blue"),
  title = "Fig. 3 - Histogram of `pcos_cleaned` data",
   data = pcos_cleaned,
         ggtheme=theme_ipsum())

# Selecting only the numerical variables for correlation
numerical_data <- pcos_cleaned[, sapply(pcos_cleaned, is.numeric)]

# Calculating the correlation matrix
cor_matrix <- cor(numerical_data)

# Print the correlation matrix
corrplot(cor_matrix, method = "color", type = "lower",
         tl.col = "black", tl.cex = 0.9, title = "Fig. 4 Correlation plot of `pcos_cleaned` data",mar=c

# boxplot of the variables with the outlier parameters
pcos_df2 <- pcos_cleaned %>%
              gather(variable, values, 1:dim(pcos_cleaned)[2])
pcos_df2 %>%
  ggplot() +
  geom_boxplot(aes(x = variable, y = values)) +
  facet_wrap(~variable, ncol = 4, scales = "free") +
  ggtitle("Fig. 5 - Boxplot outliers for `pcos_cleaned` data") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=12)
    )

# Histogram of Age distribution
p1 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`Age_yrs`))) +
  geom_histogram(stat = "count", fill = "#F39C12", color = "#e9ecef", alpha = 0.9) +
  geom_text(stat = "count", aes(label = ..count..), position = position_dodge(width = 1), vjust = -0.5,
  labs(title = "Age Distribution", x = "Age (years)", y = "Count") +
  theme_ipsum() +
  theme(
    plot.title = element_text(size = 10), axis.text.x = element_text(angle = 90, vjust = 1, hjust=2, si
  )

# Histogram of Weight distribution
p2 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`Weight`))) +
  geom_histogram(stat = "count", fill = "#FF5733", color = "#e9ecef", alpha = 0.9) +
  geom_text(stat = "count", aes(label = ..count..), position = position_dodge(width = 1), vjust = -0.5,
    labs(title="Weight Distribution", x ="Weight (kg)", y = "Count") +
    theme_ipsum() +
      theme(
        plot.title = element_text(size=10), axis.text.x = element_text(angle = 90, vjust = 1, hjust=2,
    )
```

```r
# Histogram of years married distribution
p3 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`Married_yrs`))) +
    geom_histogram(stat="count", show.legend = FALSE, fill = "#C70039", color = "#e9ecef", alpha=0.9) +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    labs(title="Years Married Distribution", x ="Married (yrs)", y = "count") +
    theme_ipsum() +
      theme(
        plot.title = element_text(size=10), axis.text.x = element_text(angle = 90, vjust = 1, hjust=2, s
    )

# plot all histograms
ggp_all <- (p1 + p2 + p3) +     # Create grid of plots with title
  plot_annotation(title = "Fig. 6 - Scatterplot of Biometric measures Variables") &
  theme(plot.title = element_text(hjust = 0.5))
ggp_all

# Scatterplot of Age and Weight
p4 <- pcos_cleaned %>%
  ggplot(aes(x=`Age_yrs`, y=`Weight`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Age(yrs) with Weight(kg)",
        x ="Age(yrs)", y = "Weight(kg)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of Hip and Waist
p5 <- pcos_cleaned %>%
  ggplot(aes(x=`Hip`, y=`Waist`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Relationship between Hip(cm) and Waist(cm)",
        x ="Hip (cm)", y = "Waist (cm)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of Length of Cycle and Age
p6 <- pcos_cleaned %>%
ggplot(aes(x=`Age_yrs`, y=`Cycle_length_days`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Length of cycle based on Age",
```

```
       x ="Age(yrs)", y = "Cycle Length(days)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of Type of Cycle and Age
p7 <- pcos_cleaned %>%
ggplot(aes(x=`Age_yrs`, y=`Cycle_RI`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Type of cycle based on Age",
       x ="Age(yrs)", y = "Cycle(R/I)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of BMI and Age
p8 <- pcos_cleaned %>%
ggplot(aes(x=`Age_yrs`, y=BMI, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="BMI based on Age",
       x ="Age(yrs)", y = "BMI") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of number of follicles in left and right ovaries and PCOS
p9 <- pcos_cleaned %>%
ggplot(aes(x=`Follicle_NoR`, y=`Follicle_NoL`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Number of Follicles",
       x ="Follicles Right", y = "Follicles Left") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of follicle size and PCOS
p10 <- pcos_cleaned %>%
ggplot(aes(x=`Avg_F_sizeR_mm`, y=`Avg_F_sizeL_mm`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
```

```
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Follicle size's",
       x ="Follicle size right", y = "Follicle size left") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of Endometrium and Length of Cycles
p11 <- pcos_cleaned %>%
ggplot(aes(x=`Endometrium_mm`, y=`Cycle_length_days`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Endometrium and length of cycle(days)",
       x ="Endometrium(mm)", y = "Cycle Length(days)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of pregnancy hormone levels
p12 <- pcos_cleaned %>%
ggplot(aes(x=`IbetaHCG_mIUmL`, y=`IIbetaHCG_mIUmL`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Pregnancy hormone levels",
       x ="I Beta-HCG(mIu/mL)", y = "II Beta-HCG(mIu/mL)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of blood pressure levels against PCOS
p13 <- pcos_cleaned %>%
ggplot(aes(x=`BP_Diastolic_mmHg`, y=`BP_Systolic_mmHg`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Blood Pressure levels",
       x ="BP_Diastolic(mmHg)", y = "BP_Systolic(mmHg)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot of Respiration rate and Pulse rate against PCOS
p14 <- pcos_cleaned %>%
```

```r
ggplot(aes(x=`RR_breaths_min`, y=`Pulse_rate_bpm`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Respiration rate vs Pulse rate(bpm)",
        x ="RR(breaths/min)", y = "Pulse rate(bpm)") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# Scatterplot on Ovarian reserve against PCOS
p15 <- pcos_cleaned %>%
ggplot(aes(x=`AMH_ngmL`, y=`FSH_LH`, color=as.factor(`PCOS`))) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
    scale_colour_discrete("PCOS", labels = c("No",
    "Yes")) +
  labs(title="Ovarian reserve against follicle growth",
        x ="Ovarian reserve", y = "Growth of ovarian follicles") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# plot all scatterplots
ggp_all2 <- (p4 + p5 + p6) / (p7 + p8 + p9) / (p10 + p11 + p12) / (p13 + p14 + p15) +     # Create grid
  plot_annotation(title = "Fig. 7 - Scatterplot of variables with PCOS yes or no as factor") &
  theme(plot.title = element_text(hjust = 0.5))
ggp_all2

# barchart of PCOS variable
p16 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`PCOS`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="PCOS", x ="PCOS(Yes or No)", y = "count") +
   theme_ipsum() +
      theme(
        plot.title = element_text(size=10)
      )

# barchart of Pregnant variable
p17 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Pregnant), fill = as.factor(Pregnant))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Pregnant", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Pregnant", x ="Pregnant(Yes or No)", y = "count") +
```

```
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Weight gain variable
p18 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Weight_gain), fill = as.factor(Weight_gain))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Weight Gain", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Weight Gain", x ="Weight Gain (Yes or No)", y = "count") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Hair growth variable
p19 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Hair_growth), fill = as.factor(Hair_growth))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Hair Growth", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Hair Growth", x ="Hair Growth(Yes or No)", y = "count") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Skin darkening variable
p20 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Skin_darkening), fill = as.factor(Skin_darkening))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Skin Darkening", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Skin Darkening", x ="Skin Darkening (Yes or No)", y = "count") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Hair loss variable
p21 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Hair_loss), fill = as.factor(Hair_loss))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Hair Loss", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Hair Loss", x ="Hair Loss(Yes or No)", y = "count") +
  theme_ipsum() +
```

```
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Pimples variable
p22 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Pimples), fill = as.factor(Pimples))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Pimples", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Pimples", x ="Pimples(Yes or No)", y = "count") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Fast food variable
p23 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Fast_food), fill = as.factor(Fast_food))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Fast Food", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Fast Food", x ="Fast Food(Yes or No)", y = "count") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# barchart of Regularly Exercise variable
p24 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Reg_Exercise), fill = as.factor(Reg_Exercise))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    scale_fill_discrete(name = "Regular Exercise", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    labs(title="Regularly Exercise", x ="Regular Exercise(Yes or No)", y = "count") +
  theme_ipsum() +
    theme(
      plot.title = element_text(size=10)
    )

# plot all barcharts
ggp_all3 <- (p16 + p17 + p18) / (p19 + p20 + p21) / (p22 + p23 + p24) +    # Create grid of plots with
  plot_annotation(title = "Fig. 8 - Bar charts of Bloodwork variables") &
  theme(plot.title = element_text(hjust = 0.5))
ggp_all3

# barchart of Blood Group variable against PCOS
p25 <- pcos_cleaned %>%
  ggplot(aes(x = Blood_Group, fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
```

```
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Blood Group with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Pregnant variable against PCOS
p26 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Pregnant), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Pregnant with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No', 'Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Weight gain variable against PCOS
p27 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Weight_gain), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Weight gain with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Hair growth variable against PCOS
p28 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Hair_growth), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Hair growth with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Skin darkening variable against PCOS
p29 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Skin_darkening), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Skin darkening with PCOS") +
```

```
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Hair loss variable against PCOS
p30 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Hair_loss), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Hair loss with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Pimples variable against PCOS
p31 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Pimples), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Pimples with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Fast food variable against PCOS
p32 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Fast_food), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Fast food consumption with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Reg. Exercise variable against PCOS
p33 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(Reg_Exercise), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.5
    ggtitle("Regularly exercises with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
```

```
    scale_x_discrete(labels=c('No','Yes')) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Length of Cycle variable against PCOS
p34 <- pcos_cleaned %>%
  ggplot(aes(x = `Cycle_length_days`, fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Cycle length in days with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# barchart of Number of abortions variable against PCOS
p35 <- pcos_cleaned %>%
  ggplot(aes(x = No_of_abortions, fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    ggtitle("Number of abortions with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45)
    )

# plot all barcharts
ggp_all4 <- (p25 + p26 + p27) / (p28 + p29 + p30) / (p31 + p32 + p33) / (p34 + p35) +    # Create grid
  plot_annotation(title = "Fig. 9 - Bar charts of yes or no variables") &
  theme(plot.title = element_text(hjust = 0.5))
ggp_all4

# Barchart for Vitamin D3 levels with PCOS
p36 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`Vit_D3_ngmL`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Vitamin D3 levels with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
    )

# Barchart for FSH/LH levels with PCOS
p37 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`FSH_LH`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("FSH/LH levels with PCOS") +
```

```
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
    )


# Barchart for Thyroid Hormone levels with PCOS
p38 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`TSH_mIUmL`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Thyroid Hormone levels with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
    )

# Barchart of hemoglobin levels with PCOS
p39 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`Hb_gdl`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Hemoglobin levels with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
    )

# Barchart of Prolactin levels with PCOS
p40 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`PRL_ngmL`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Prolactin levels with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
    )

# Barchart of Progesterone levels with PCOS
p41 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`PRG_ngmL`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Progesterone levels with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
```

```
    )

# Barchart of Glucose levels with PCOS
p42 <- pcos_cleaned %>%
  ggplot(aes(x = as.factor(`RBS_mgdl`), fill = as.factor(`PCOS`))) +
    geom_bar(position = "dodge") +
    #geom_text(aes(label = ..count..), position = position_dodge(width = 1), stat = "count", vjust = 1.
    ggtitle("Glucose levels with PCOS") +
    scale_fill_discrete(name = "PCOS", labels = c("No", "Yes")) +
    theme_ipsum() +
    theme(
      plot.title = element_text(size=10), axis.text.x = element_text(angle = 45, size = 2)
    )

# plot barcharts
ggp_all5 <- (p36 + p37 + p38 + p39) / (p40 + p41 + p42)  +    # Create grid of plots with title
  plot_annotation(title = "Fig. 10 - Bar charts of Yes or No variables") &
  theme(plot.title = element_text(hjust = 0.5))
ggp_all5

# DECISION TREE:

# create some random numbers for reproduction
set.seed(29)

# Cross Validation Set-up
inTrain <- createDataPartition(pcos_cleaned$`PCOS`, p=.75, list = F)
train <- pcos_cleaned[inTrain,]
valid <- pcos_cleaned[-inTrain,]

# create the decision tree
rpart_model <- rpart(`PCOS` ~ ., method = "class", data = train)

# display the decision tree
prp(rpart_model, main = "Fig. 12 - Decision Tree with entire dataset", extra=1, faclen=0,  nn=T, box.pal

# creating our prediction
rpart_result <- predict(rpart_model, newdata = valid[, !colnames(valid) %in% "PCOS"], type = 'class')

# confusion matrix
confusionMatrix(rpart_result, as.factor(valid$`PCOS`))

# contribution of variables
varImp(rpart_model) %>% kable()

# Extract accuracy from the confusion matrix
accuracy_rpart <- confusionMatrix(rpart_result, as.factor(valid$`PCOS`))$overall["Accuracy"]
kable(accuracy_rpart, align = "l")

# creating the second dataset from the original
pcos_cleaned2 <- pcos_cleaned %>%
  select(`PCOS`, `Follicle_NoR`, `Follicle_NoL`, `Weight_gain`, `Skin_darkening`, `Hair_growth`)
```

```r
# create some random number for reproduction
set.seed(28)

# Second Cross Validation Set-up
inTrain2 <- createDataPartition(pcos_cleaned2$`PCOS`, p=.75, list = F)
train2 <- pcos_cleaned2[inTrain2,]
valid2 <- pcos_cleaned2[-inTrain2,]

# create the second decision tree
rpart_model2 <- rpart(`PCOS` ~ ., method = "class", data = train2)

# display the decision tree
prp(rpart_model2, main = "Fig. 13 - Second Decision Tree with 6 variables", extra=1, faclen=0,  nn=T, bo

# creating our prediction
rpart_result2 <- predict(rpart_model2, newdata = valid2[, !colnames(valid2) %in% "PCOS"], type = 'class

# creating the second confusion matrix
confusionMatrix(rpart_result2, as.factor(valid2$`PCOS`))

# contribution of variables
varImp(rpart_model2) %>% kable()

# Extract accuracy from the confusion matrix
accuracy_rpart2 <- confusionMatrix(rpart_result2, as.factor(valid2$`PCOS`))$overall["Accuracy"]
kable(accuracy_rpart2, align = "l")

# RANDOM FOREST:

# create some random numbers for reproduction
set.seed(30)

# Cross Validation Set-up
rf_inTrain <- createDataPartition(pcos_cleaned$`PCOS`, p=.75, list = F)
rf_train <- pcos_cleaned[rf_inTrain,]
rf_valid <- pcos_cleaned[-rf_inTrain,]

# check the levels of PCOS using levels()
levels(rf_train$PCOS)
levels(rf_valid$PCOS)

# Convert PCOS to factor in rf_train
rf_train$PCOS <- factor(rf_train$PCOS)

# Convert PCOS to factor in rf_valid
rf_valid$PCOS <- factor(rf_valid$PCOS)

# rechecking levels again to ensure no NULL values
levels(rf_train$PCOS)
levels(rf_valid$PCOS)

# explicitly set the levels to match the levels in srf_train.
rf_valid$PCOS <- factor(rf_valid$PCOS, levels = levels(rf_train$PCOS))
```

```r
levels(rf_valid$PCOS)

# # Check the length of rf_result and rf_valid$PCOS
# length_rf_result <- length(rf_result)
# length_rf_valid <- length(rf_valid$PCOS)
#
# # Print the lengths for comparison
# print(length_rf_result)
# print(length_rf_valid)

#create some random number for reproduction
set.seed(39)

# create random forest model using the training data
rf_model <- randomForest(PCOS~., rf_train)
rf_model

# prediction
rf_result <- predict(rf_model, newdata = valid[, !colnames(valid) %in% "PCOS"])

# Create a confusion matrix
confusionMatrix(data = rf_result, reference = rf_valid$PCOS)

# plot for rf_model
varImpPlot(rf_model)

# table for rf_model variable contribution
varImp(rf_model) %>% kable()

# Extract accuracy from the confusion matrix for the rf_model
accuracy_rf <- confusionMatrix(rf_result, valid$PCOS)$overall["Accuracy"]
accuracy_rf

# create some random numbers for reproduction
set.seed(78)

# Second RF Cross Validation Set-up
rf_inTrain2 <- createDataPartition(pcos_cleaned2$`PCOS`, p=.75, list = F)
rf_train2 <- pcos_cleaned2[rf_inTrain2,]
rf_valid2 <- pcos_cleaned2[-rf_inTrain2,]

# check the levels of PCOS using levels()
levels(rf_train2$PCOS)
levels(rf_valid2$PCOS)

# Convert PCOS to factor in rf_train
rf_train2$PCOS <- factor(rf_train2$PCOS)

# Convert PCOS to factor in rf_valid
rf_valid2$PCOS <- factor(rf_valid2$PCOS)

# rechecking levels again to ensure no NULL values
levels(rf_train2$PCOS)
```

```r
levels(rf_valid2$PCOS)

# explicitly set the levels to match the levels in rf_train.
rf_valid2$PCOS <- factor(rf_valid2$PCOS, levels = levels(rf_train2$PCOS))
levels(rf_valid2$PCOS)

# # Check the length of rf_result and rf_valid$PCOS
# length_rf_result2 <- length(rf_result2)
# length_rf_valid2 <- length(rf_valid2$PCOS)
#
# # Print the lengths for comparison
# print(length_rf_result2)
# print(length_rf_valid2)

# create some random number for reproduction
set.seed(7)

# create the second random forest model using the training data from the third decision tree
rf_model2 <- randomForest(PCOS ~ Follicle_NoR + Follicle_NoL + Weight_gain + Skin_darkening + Hair_grow
rf_model2

# creating the prediction for the third decision tree
rf_result2 <- predict(rf_model2, newdata = rf_valid2[, !colnames(rf_valid2) %in% "PCOS"])

# Convert PCOS column to factor in rf_train2 and rf_valid2
rf_train2$PCOS <- factor(rf_train2$PCOS)
rf_valid2$PCOS <- factor(rf_valid2$PCOS)

# # Check unique levels in rf_result2 and rf_valid2$PCOS
# unique_levels_result <- unique(rf_result2)
# unique_levels_valid <- unique(rf_valid2$PCOS)
#
# # Check if the levels match
# identical(unique_levels_result, unique_levels_valid)
#
# # If levels do not match, manually set levels in rf_result2 to match those in rf_valid2$PCOS
# levels(rf_result2) <- levels(rf_valid2$PCOS)
#
# Convert rf_result2 to factor and align levels with rf_valid2$PCOS
rf_result2_factor <- factor(rf_result2, levels = levels(rf_valid2$PCOS))

# Create a confusion matrix
confusionMatrix(data = rf_result2_factor, reference = rf_valid2$PCOS)

# plot for the second rf_model
varImpPlot(rf_model2)

# table for rf_model2 variable contribution
varImp(rf_model2) %>% kable()

# Extract accuracy from the confusion matrix for the rf_model2
accuracy_rf2 <- confusionMatrix(data = rf_result2_factor, reference = rf_valid2$PCOS)$overall["Accuracy
accuracy_rf2
```

```r
# GRADIENT BOOSTING MACHINES:

# Set seed for reproducibility
set.seed(67)

# Train the GBM model
gbm_model <- gbm(`PCOS` ~ ., data = train, distribution = "bernoulli", n.trees = 100, interaction.depth

# Print the summary of the trained model
summary(gbm_model)

# Predict on the validation dataset (assuming 'valid' contains your validation dataset)
gbm_pred <- predict(gbm_model, newdata = valid, type = "response")

# Calculate predicted classes (0 or 1) based on the predicted probabilities
predicted_classes <- ifelse(gbm_pred > 0.5, 1, 0)

# Create confusion matrix
confusionMatrix(data = factor(predicted_classes), reference = factor(valid$`PCOS`))

# Calculate accuracy
gbm_accuracy <- sum(predicted_classes == valid$`PCOS`) / length(valid$`PCOS`)
cat("Accuracy:", gbm_accuracy)

# creating the second dataset from the original
pcos_cleaned3 <- pcos_cleaned %>%
  select(`PCOS`, `Follicle_NoR`, `Follicle_NoL`, `Weight_gain`, `Skin_darkening`, `Hair_growth`)

# Set seed for reproducibility
set.seed(68)

# Cross Validation Set-up
inTrain3 <- createDataPartition(pcos_cleaned3$`PCOS`, p=.75, list = F)
train3 <- pcos_cleaned3[inTrain3,]
valid3 <- pcos_cleaned3[-inTrain3,]

# Train the GBM model
gbm_model2 <- gbm(`PCOS` ~ ., data = train3, distribution = "bernoulli", n.trees = 100, interaction.dep

# Print the summary of the trained model
summary(gbm_model2)

# Predict on the validation dataset (assuming 'valid' contains your validation dataset)
gbm_pred2 <- predict(gbm_model2, newdata = valid3, type = "response")

# Calculate predicted classes (0 or 1) based on the predicted probabilities
predicted_classes2 <- ifelse(gbm_pred2 > 0.5, 1, 0)

# Create confusion matrix
confusionMatrix(data = factor(predicted_classes2), reference = factor(valid3$`PCOS`))

# Calculate accuracy
gbm_accuracy2 <- sum(predicted_classes2 == valid3$`PCOS`) / length(valid3$`PCOS`)
```

```r
cat("Accuracy:", gbm_accuracy2)

# SUPPORT VECTOR MACHINES:

# check the levels of PCOS using levels()
levels(train$PCOS)
levels(valid$PCOS)

# Convert PCOS to factor in svm_train
train$PCOS <- factor(train$PCOS)

# Convert PCOS to factor in svm_valid
valid$PCOS <- factor(valid$PCOS)

# rechecking levels again to ensure no NULL values
levels(train$PCOS)
levels(valid$PCOS)

# checking the structure of both valid and train datasets
str(valid)
str(train)

# explicitly set the levels to match the levels in svm_train.
valid$PCOS <- factor(valid$PCOS, levels = levels(train$PCOS))
levels(valid$PCOS)

# # Check the length of svm_result and svm_valid$PCOS
# length_svm_result <- length(svm_result)
# length_svm_valid <- length(svm_valid$PCOS)
#
# # Print the lengths for comparison
# print(length_svm_result)
# print(length_svm_valid)

#create some random numbers for reproduction
set.seed(31)

# SVM
svm_model <- svm(PCOS ~ ., train)

# create prediction
svm_result <- predict(svm_model, newdata = valid)


# confusion matrix for svm
confusionMatrix(svm_result, valid$PCOS)

# summary of svm_result
summary(svm_result)

#Extract accuracy from the confusion matrix
accuracy_svm <- confusionMatrix(svm_result, as.factor(valid$`PCOS`))$overall["Accuracy"]
accuracy_svm
```

```r
# create some random numbers for reproduction
set.seed(8)

# Cross Validation Set-up
svm_inTrain2 <- createDataPartition(pcos_cleaned2$PCOS, p=.75, list = FALSE)
svm_train2 <- pcos_cleaned2[svm_inTrain2,]
svm_valid2 <- pcos_cleaned2[-svm_inTrain2,]

# check the levels of PCOS using levels()
levels(svm_train2$PCOS)
levels(svm_valid2$PCOS)

# Convert PCOS to factor in svm_train2
svm_train2$PCOS <- factor(svm_train2$PCOS)

# Convert PCOS to factor in svm_valid2
svm_valid2$PCOS <- factor(svm_valid2$PCOS)

# rechecking levels again to ensure no NULL values
levels(svm_train2$PCOS)
levels(svm_valid2$PCOS)

# explicitly set the levels to match the levels in svm_train2
valid$PCOS <- factor(svm_valid2$PCOS, levels = levels(svm_train2$PCOS))
levels(svm_valid2$PCOS)

# # Check the length of svm_result and svm_valid$PCOS
# length_svm_result <- length(svm_result)
# length_svm_valid2 <- length(svm_valid2$PCOS)
#
# # Print the lengths for comparison
# print(length_svm_result)
# print(length_svm_valid2)


# Second SVM
svm_model2 <- svm(PCOS ~ Follicle_NoR + Follicle_NoL + Weight_gain + Skin_darkening + Hair_growth, svm_

# create prediction
svm_result2 <- predict(svm_model2, newdata = svm_valid2)

# confusion matrix for svm_valid2
confusionMatrix(svm_result2, svm_valid2$PCOS)

# summary
summary(svm_result2)

#Extract accuracy from the confusion matrix
accuracy_svm2 <- confusionMatrix(svm_result2, svm_valid2$`PCOS`)$overall["Accuracy"]
accuracy_svm2


# NEURAL NETWORKS:
```

```r
# create some random numbers for reproduction
set.seed(67)

# Cross Validation Set-up
nn_inTrain <- createDataPartition(pcos_cleaned$PCOS, p=.75, list = F)
nn_train <- pcos_cleaned[nn_inTrain,]
nn_valid <- pcos_cleaned[-nn_inTrain,]

# set a seed for reproducibility purposes
set.seed(19)

# create the model
nn_model <- neuralnet(`PCOS`~.,
                      data = nn_train,
                      hidden = c(12, 8),  # Specify the number of hidden layers and neurons
                      linear.output = FALSE,
                      stepmax = 20000  # Increase the maximum number of iterations
)

# create the plot based on the model above
#plot(nn_model, rep = "best", main="")
#grid::grid.text("Fig. 15 - Neural Network", x = 0.5, y = 0.1)

# make predictions on the test data using a previously trained model
pred <- predict(nn_model, valid)

# create a vector of labels for the two possible `PCOS(Y/N)` status in the dataset.
labels <- c("0", "1")

# creates a data frame with the column index of the maximum value in each row of the "pred" variable
prediction_label <- data.frame(max.col(pred)) %>%
# use the mutate function to add a new column to the data frame called "pred"
mutate(pred=labels[max.col.pred.]) %>%
select(2) %>%
# convert the data frame to a vector.
unlist()

# print the table
table(valid$`PCOS`, prediction_label)

#checking the accuracy
check <- as.numeric(valid$`PCOS`) == max.col(pred)
nn_accuracy <-  (sum(check)/nrow(valid))
nn_accuracy

# set a seed for reproducibility purposes
set.seed(13)

# create the second model
nn_model2 <-  neuralnet(`PCOS`~Follicle_NoL + Follicle_NoR + Hair_growth + Skin_darkening + Weight_gain
                        data=train2,
                        hidden=c(2,1),
                        linear.output = FALSE,
```

```
                        stepmax = 10000  # Increase the maximum number of iterations
)

# create the plot based on the model above
plot(nn_model2, rep = "best", main="")
grid::grid.text("Fig. 15 - Neural Network", x = .5, y = .2)

# make predictions on the test data using a previously trained model
pred2 <- predict(nn_model2, valid2)

# create a vector of labels for the two possible `PCOS` status in the dataset.
labels2 <- c("0", "1")

# creates a data frame with the column index of the maximum value in each row of the "pred" variable
prediction_label2 <- data.frame(max.col(pred2)) %>%
# use the mutate function to add a new column to the data frame called "pred"
mutate(pred=labels2[max.col.pred2.]) %>%
select(2) %>%
# convert the data frame to a vector.
unlist()

# print the table
table(valid2$`PCOS`, prediction_label2)

# checking the accuracy
check2 <- as.numeric(valid2$`PCOS`) == max.col(pred2)
nn_accuracy2 <-  (sum(check2)/nrow(valid2))
nn_accuracy2

# Remove rows with missing values from train and valid datasets
train <- train[complete.cases(train), ]
valid <- valid[complete.cases(valid), ]

# set a seed for reproducibility purposes
set.seed(78)

# Set the value of k for kNN
k <- 5  # Change this value as needed

# Fit the kNN model using the training data
knn_model <- knn(train[, -which(names(train) == "PCOS")],
                 valid[, -which(names(valid) == "PCOS")],
                 train$`PCOS`,
                 k = k)

# Calculate accuracy
knn_accuracy <- mean(knn_model == valid$`PCOS`)
knn_accuracy

# Filter and select the desired columns for the new dataset
pcos_cleaned4 <- pcos_cleaned %>%
  select(`PCOS`, `Follicle_NoR`, `Follicle_NoL`, `Weight_gain`, `Skin_darkening`, `Hair_growth`)
```

```r
# Split the data into training and validation sets (if needed)
set.seed(123)  # Set seed for reproducibility
inTrain4 <- createDataPartition(pcos_cleaned4$`PCOS`, p = 0.75, list = FALSE)
train4 <- pcos_cleaned4[inTrain4, ]
valid4 <- pcos_cleaned4[-inTrain4, ]

# Check for missing values and remove them if present
train4 <- train4[complete.cases(train4), ]
valid4 <- valid4[complete.cases(valid4), ]

# Set the value of k for kNN
k <- 5  # Change this value as needed

# Fit the kNN model using the training data
knn_model2 <- knn(train4[, -which(names(train4) == "PCOS")],
                  valid4[, -which(names(valid4) == "PCOS")],
                  train4$`PCOS`,
                  k = k)

# Calculate accuracy for the new kNN model
knn_accuracy2 <- mean(knn_model2 == valid4$`PCOS`)
knn_accuracy2

# Compare models
model_names <- c("Decision Tree 1","Decision Tree 2", "Random Forest 1", "Random Forest 2", "Gradient Bo
accuracies <- c(0.8592593, 0.9185185, 0.6148148, 0.9037037, 0.8814815, 0.6222222, 0.9111111, 0.9185185,

# place accuracies in data frame
results <- data.frame(Model = model_names, Accuracy = accuracies)

# order in descending order
results <- results[order(results$Accuracy, decreasing = TRUE), ]

# Display the results
kable(results, caption = "<font color=#000000><b>Table 5.</b>Model Comparison </font>", format = "html"
  kable_styling(bootstrap_options = c("hover", "condensed"), font_size = 13) %>%
  kableExtra::scroll_box(width = "100%", height = "400px")
```