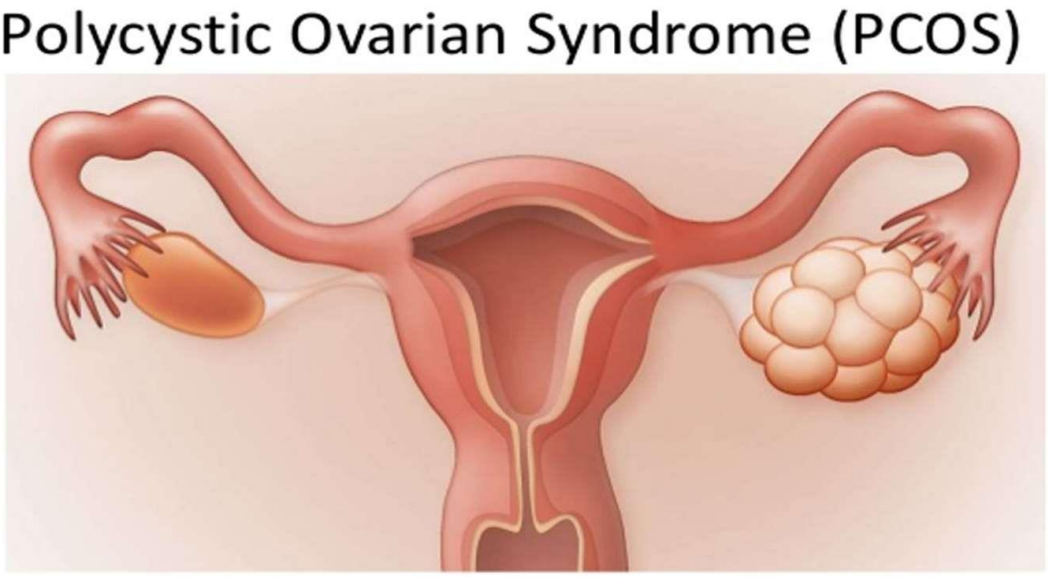
CSE3013- ARTIFICIAL INTELLIGENCE PROJECT FINAL REVIEW

SUBMITTED BY:
Arya Dubey (20BCE0908)
Prachurya Priyadarshini (20BCT0155)
Prakhar Kandpal (20BCE2117)
Abhinav Bijith (20BCE2149)

UNDER THE GUIDANCE OF:
PROF. GUNAVATHI C

Polycystic Ovarian Syndrome Detecting Analyser



CERTIFICATE

This is to certify that the project entitled "Polycystic Ovarian Syndrome Detecting Analyzer", submitted by Arya Dubey (20BCE0908), Prachurya Priyadarshini (20BCT0155), Prakhar Kandpal (20BCE2117), Abhinav Bijith (20BCE2149), VIT, for the award of the degree of Bachelor of Technology in Programme, is a record of bonafide work carried out by them under my supervision during the period, 02. 08. 2021 to 10.12.2021, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date :

Signature of the Guide

Internal Examiner

External Examiner

Head of the Department Programme

DECLARATION

We thereby declare that the project entitled "Polycystic Ovarian Syndrome Detecting Analyser" submitted by us, for the award of the degree of Bachelor of Technology in Programme to VIT is a record of bonafide work carried out by us under the supervision of Prof. Gunavathi C. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

EXECUTIVE SUMMARY

In the past few decades, technology has revolutionized our universe and affected our lives, making them easier day by day. Emerging technologies are reshaping mankind in a lot of ways. These days, machine learning, a eld of study that gives computers to learn without being explicitly programmed, is playing a key role in the healthcare sector. Machine learning can deal with obscenely huge datasets, convert analysed data into clinical insights and help in the diagnosis of various ailments. Polycystic Ovary Syndrome (PCOS) is a medical condition which causes hormonal disorder in women in their childbearing years. PCOS occurs as a result of hormonal imbalances. In this disorder, the ovaries develop small collections of uids called follicles (cysts) and fail to release eggs, which is why women suffering from PCOS tend to have complications in conceiving . A lot of women have PCOS, but do not get diagnosed with it at an earlier stage. In a study, 69 to 70 percent of women did not have a pre-existing diagnosis.

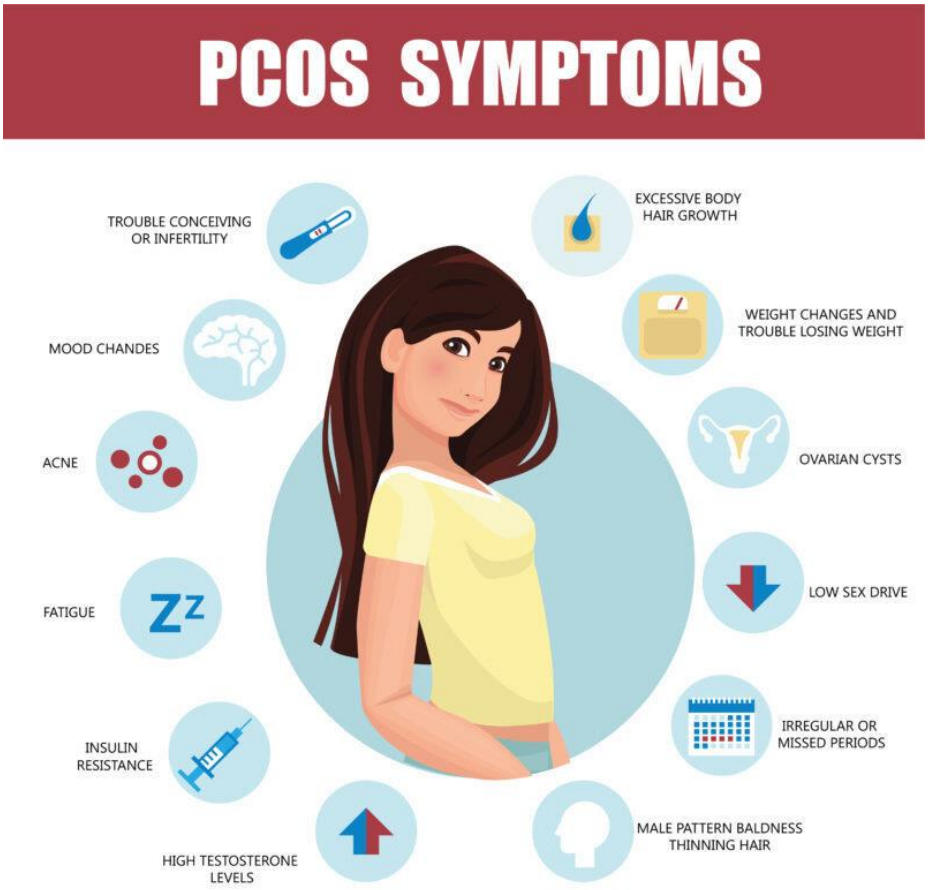
While the actual causes of PCOS remain a mystery, studies say that it is generally inherited. It is a very unpredictable condition as the cure is uncertain since there is no observable trend for this medical condition.

Table of Contents

INTRODUCTION:

Polycystic ovary syndrome (PCOS) is a condition in which the ovaries produce an abnormal amount of androgens, male hormones that are usually present in women in small amounts. The name polycystic ovary syndrome describes the numerous small cysts (fluid-filled sacs) that form in the ovaries. The basic symptoms of PCOS are:

- Cysts in ovaries.
- High levels of hormone: androgen.
- Irregular Periods
- Excessive body hair growth
- Weight gain especially around the belly



PROBLEM STATEMENT

Several factors cause infertility; one of which has been the irregular number and the dimension of follicle development in the ovulation phase . Such abnormality has been the first symptom of PCOS.

Women having PCOS are at risk of improving a diversity of symptoms/intricacies involving variant systems, l, gynecological, metabolic, physical, cardiac, and psychological. Perfect finding of the PCOS is important for the treatment.

Real-time analysis of PCOS is a major task as follicles contains different sizes and highly connected with tissues and blood vessels which results in errors. In contemporary years, transabdominal and/or transvaginal ultrasound have turn into the most generally utilized as diagnostic techniques for the recognition of PCOS.

Manual identification of follicles may cause several problems, such as more time is needed for the follicles' identification, inter and intra-observer inconsistency and sometimes they could seriously

affect women's health. Additionally, there is no assessment on the use of a better diagnosis for PCOS awareness and management. Thus, a more general machine learning procedure is needed for the better diagnosis of PCOS.

OBJECTIVE

●This project is an attempt to develop a real-time object detection using machine learning concepts. The project aims to deliver an implemented approach to detect PCOS. In the modern era, there are several new technologies available to diagnose various diseases and one of them is Machine learning algorithms because they are exposed to new data. These algorithms learn from past experiences to produce reliable and repeatable decisions. In this article, Machine learning algorithms are used to identify the important features to diagnose PCOS.

●From 1 in 10 women suffering from PCOS worldwide to currently 3-4 in 10 women, PCOS is now exponentially increasing among women due to an unhealthy lifestyle. The literature says that 1 in every 5 women in India suffers from PCOS. PCOS symptoms differ in every patient.

●The major diagnosis includes scanning for follicles, their number and sizes using Ultrasound imaging.

●Even though it is called Polycystic Ovary Syndrome, it is not essentially described by ovarian cysts. It is defined by examining at least two of three diagnostic criteria.

●In 2012, the 2003 Rotterdam criteria were endorsed by NIH for PCOS.
●For an accurate PCOS diagnosis, disorders that have specific signs and symptoms that match with those of PCOS must be dismissed.

Apollo Spectra HOSPITALS
Specialists in Surgery

POLYCYSTIC OVARY SYNDROME (PCOS): SYMPTOMS, DIAGNOSIS AND TREATMENT

Polycystic ovary syndrome (PCOS) is a condition that affects a woman's hormone levels.

SYMPTOMS


Missed periods, irregular periods


Large ovaries


Excess body hair


Weight gain


Acne or oily skin


Male-pattern baldness


Infertility

CAUSES

The exact cause of PCOS isn't known.

DIAGNOSIS


Physical exam


Ultrasound


Blood tests

TREATMENT


Medications


Birth control pills


Surgery

HOME REMEDIES


Eating a healthy diet


Regular physical activity


Maintaining a healthy weight


Avoid smoking

PROJECT DESCRIPTION

●The team has used different machine learning algorithms like K-nearest neighbour (KNN), decision tree and SVM with different kernel functions to predict PCOS from the identification of new genes.
●It has used machine learning algorithms like Logistic Regression (LR) and Random Classifier to develop an automated system that will act as an assisted tool for the doctor for saving considerable time in examining the patients and hence reducing the delay in diagnosing the risk of PCOS by using metabolic and clinical factors in a feature vector.

In the modern era, there are several new technologies available to diagnose PCOS and one of them is Machine learning algorithms because they are exposed to new data. These algorithms learn from past experiences to produce reliable and repeatable decisions. In this article, Machine learning algorithms are used to identify the important features to diagnose PCOS.

TECHNICAL SPECIFICATIONS

Machine Learning Algorithms like K-nearest neighbour (KNN), decision tree and SVM with different kernel functions are used in this project to predict PCOS from the identification of new genes.
Other ML Algorithms like Logistic Regression (LR) and Random Classifier are implemented to develop an automated system that will act as an assisted tool for the doctor for saving considerable time in examining the patients and hence reducing the delay in diagnosing the risk of PCOS by using metabolic and clinical factors in a feature vector.

Here's how we will do it:

- The first step to building our model is to import our libraries and datasets into our Google Colab notebook.
- pandas: The most popular python library that is used for data manipulation and analysis. In this project, it is primarily useful for dataframe manipulation.
- NumPy: A python library that provides support for large, multi-dimensional arrays and matrices, and has high-level mathematical functions to help operate on and manipulate these arrays.
- matplotlib.pyplot and seaborn: Used for data visualization.

- We can start the project by making sure we have installed the latest version of seaborn, which will be used for data visualization.

Then, we can proceed on to the architecture model of our project.

Below is the table for the Literature Survey which is used for distilling information on thesis and research papers mentioned below to scrutinize the problem.

LITERATURE SURVEY

A literature survey is a piece of discursive prose, not a list describing or summarizing one piece of literature after another. It is an iterative process, assessing and distilling information. One of the key purposes of the literature survey is to investigate a problem that no one else has addressed.

Authors & Year Reference	Title Study	Journal/ Conference	Concept & Analysis	Limitations
C. Gopalkrishnan & M. Iyapparaja	Detection of PCOS from Ultrasound Images of Ovaries	International Journal of Recent Technology and Engineering September 2019	Description & Mechanism of PCOS, Ultrasound & Ultrasonography	It is difficult to implement the process of detecting PCOS from ultrasound image ovaries in accurate manner. New hybrid techniques have to develop for its improvement.
B. Bhanu	Follicle Diagnosis in PCOS	IJRTC 1995	Genetic Algorithm (GA)	Computational requirements are varied by the number of parameters
T. Chiang and Y. Q. Zhang	Delay in Period Cycle Measure	IJRTC 1997	Quadratic-Rate distortion method	Detection of minimum is failed
G. Vasavi & Dr. S. Jyothi	PCOS Detection Using Various Machine Learning Methods	Adv Research Journal in Dynamical & Control Systems, 05- Special Issue, July 2017	K-Nearest Neighbour Technique Based Classification of Biomedical Objects	Only few techniques have presented effective performance. Although the implementation of such techniques has proffered good quality outcomes but a great number of tasks are still to

				be enhanced in future.
Anita Raj & Remya George	Prediction System of PCOS using ML	Tencon 2019- IEEE Region 10 Conference	Classification of PCOS with the feature set transformed with Principal component Analysis (PCA)	Still difficult to diagnose or detect but give 90% of the best results.
Usman	Texture Features & Artificial Neural Network for Follicle Detection	JARDCS 2011	Automatic characterization of ovaries amidst complete female cycle	Necessity for enhancing on the denoising methodology as medical images is fraught with impair noise

DESIGN APPROACH AND DETAILS

In [ ]:
```
!pip install seaborn --upgrade
```

```
Requirement already up-to-date: seaborn in /usr/local/lib/python3.7/dist-packages (0.11.1)
Requirement already satisfied, skipping upgrade: pandas>=0.23 in /usr/local/lib/python3.7/dist-packages (from seaborn) (1.1.5)
Requirement already satisfied, skipping upgrade: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from seaborn) (1.19.5)
Requirement already satisfied, skipping upgrade: matplotlib>=2.2 in /usr/local/lib/python3.7/dist-packages (from seaborn) (3.2.2)
Requirement already satisfied, skipping upgrade: scipy>=1.0 in /usr/local/lib/python3.7/dist-packages (from seaborn) (1.4.1)
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.23->seaborn) (2018.9)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.23->seaborn) (2.8.1)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.
2->seaborn) (2.4.7)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied, skipping upgrade: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>=0.23->seaborn)
(1.15.0)
```

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.__version__
```

Out[ ]:
```
'0.11.1'
```

In [ ]:
```
from google.colab import files
uploaded = files.upload()
pcos = pd.read_csv('PCOS_no_infertility.csv')
pcos.head(15)
```

```
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving PCOS_no_infertility.csv to PCOS_no_infertility (1).csv
```

In [ ]:
```
from google.colab import files
uploaded = files.upload()
pcos = pd.read_csv('PCOS_no_infertility.csv')
pcos.head(15)
```

```
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving PCOS_no_infertility.csv to PCOS_no_infertility (1).csv
```

Out[ ]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13	Un
0	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height	BMI	Blood Group	Pulse rate(bpm)	RR (breaths/min)	Hb(g/dl)	Cycle(R/I)	Cycle length(days)	Marraige Status (Yrs)	Pregnant(Y/N)	No. of aborptions	HCG(n
1	0	28	44.6	152	19.3	15	78	22	10.48	2	5	7	0	0	
2	0	36	65	161.5	24.92116286	15	74	20	11.7	2	5	11	1	0	
3	1	33	68.8	165	25.27089073	11	72	18	11.8	2	5	10	1	0	
4	0	37	65	148	29.67494522	13	72	20	12	2	5	4	0	0	
5	0	25	52	161	20.06095444	11	72	18	10	2	5	1	1	0	
6	0	36	74.1	165	27.21763085	15	78	28	11.2	2	5	8	1	0	
7	0	34	64	156	26.29848784	11	72	18	10.9	2	5	2	0	0	
8	0	33	58.5	159	23.13990744	13	72	20	11	2	5	13	1	2	
9	0	32	40	158	16.02307323	11	72	18	11.8	2	5	8	0	1	
10	0	36	52	150	23.11111111	15	80	20	10	4	2	4	0	0	
11	0	20	71	163	26.72287252	15	80	20	10	2	5	4	1	2	
12	0	26	49	160	19.140625	13	72	20	9.5	2	5	3	0	1	
13	1	25	74	152	32.02906587	17	72	18	11.7	4	2	7	1	0	
14	0	38	50	152	21.64127424	13	74	20	12.1	2	5	15	0	0	

In [ ]:
```
df = pcos
```

In [ ]:
```
#Use first row of data as df header
new_header = df.iloc[0]
df = df[1:]
df.columns = new_header
```

In [ ]:
```
df
```

Out[ ]:

	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height	BMI	Blood Group	Pulse rate(bpm)	RR (breaths/min)	Hb(g/dl)	Cycle(R/I)	Cycle length(days)	Marraige Status (Yrs)	Pregnant(Y/N)	No. of aborptions	I beta-HCG(mIU/mL)	II HCG(mI
1	0	28	44.6	152	19.3	15	78	22	10.48	2	5	7	0	0	1.99	
2	0	36	65	161.5	24.92116286	15	74	20	11.7	2	5	11	1	0	60.8	
3	1	33	68.8	165	25.27089073	11	72	18	11.8	2	5	10	1	0	494.08	
4	0	37	65	148	29.67494522	13	72	20	12	2	5	4	0	0	1.99	
5	0	25	52	161	20.06095444	11	72	18	10	2	5	1	1	0	801.45	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
537	0	35	50	164.592	18.5	17	72	16	11	2	5	8	0	1	1.99	
538	0	30	63.2	158	25.3	15	72	18	10.8	2	5	4	1	1	80.13	
539	0	36	54	152	23.4	13	74	20	10.8	2	6	8	0	0	1.99	
540	0	27	50	150	22.2	15	74	20	12	4	2	2	0	0	292.92	
541	1	23	82	165	30.1	13	80	20	10.2	4	7	2	0	0	1.99	

541 rows × 42 columns

In [ ]:
```
#Checking for null values
df.isnull().sum()
```

In [ ]:
```
df
```

Out[ ]:

	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height	BMI	Blood Group	Pulse rate(bpm)	RR (breaths/min)	Hb(g/dl)	Cycle(R/I)	Cycle length(days)	Marraige Status (Yrs)	Pregnant(Y/N)	No. of aborptions	I beta-HCG(mIU/mL)	II HCG(mI
1	0	28	44.6	152	19.3	15	78	22	10.48	2	5	7	0	0	1.99	
2	0	36	65	161.5	24.92116286	15	74	20	11.7	2	5	11	1	0	60.8	
3	1	33	68.8	165	25.27089073	11	72	18	11.8	2	5	10	1	0	494.08	
4	0	37	65	148	29.67494522	13	72	20	12	2	5	4	0	0	1.99	
5	0	25	52	161	20.06095444	11	72	18	10	2	5	1	1	0	801.45	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
537	0	35	50	164.592	18.5	17	72	16	11	2	5	8	0	1	1.99	
538	0	30	63.2	158	25.3	15	72	18	10.8	2	5	4	1	1	80.13	
539	0	36	54	152	23.4	13	74	20	10.8	2	6	8	0	0	1.99	
540	0	27	50	150	22.2	15	74	20	12	4	2	2	0	0	292.92	
541	1	23	82	165	30.1	13	80	20	10.2	4	7	2	0	0	1.99	

539 rows × 42 columns

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 539 entries, 1 to 541
Data columns (total 42 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   PCOS (Y/N)              539 non-null    object
 1    Age (yrs)              539 non-null    object
 2   Weight (Kg)             539 non-null    object
 3   Height                  539 non-null    object
 4   BMI                     539 non-null    object
 5   Blood Group             539 non-null    object
 6   Pulse rate(bpm)         539 non-null    object
 7   RR (breaths/min)        539 non-null    object
 8   Hb(g/dl)                539 non-null    object
```

In [ ]:
```
# Create a loop to convert all values into numeric
for column in df:

    # Select column contents by column
    # name using [] operator
    columnSeriesObj = df[column]
    df[column] = pd.to_numeric(df[column], errors='coerce')
```

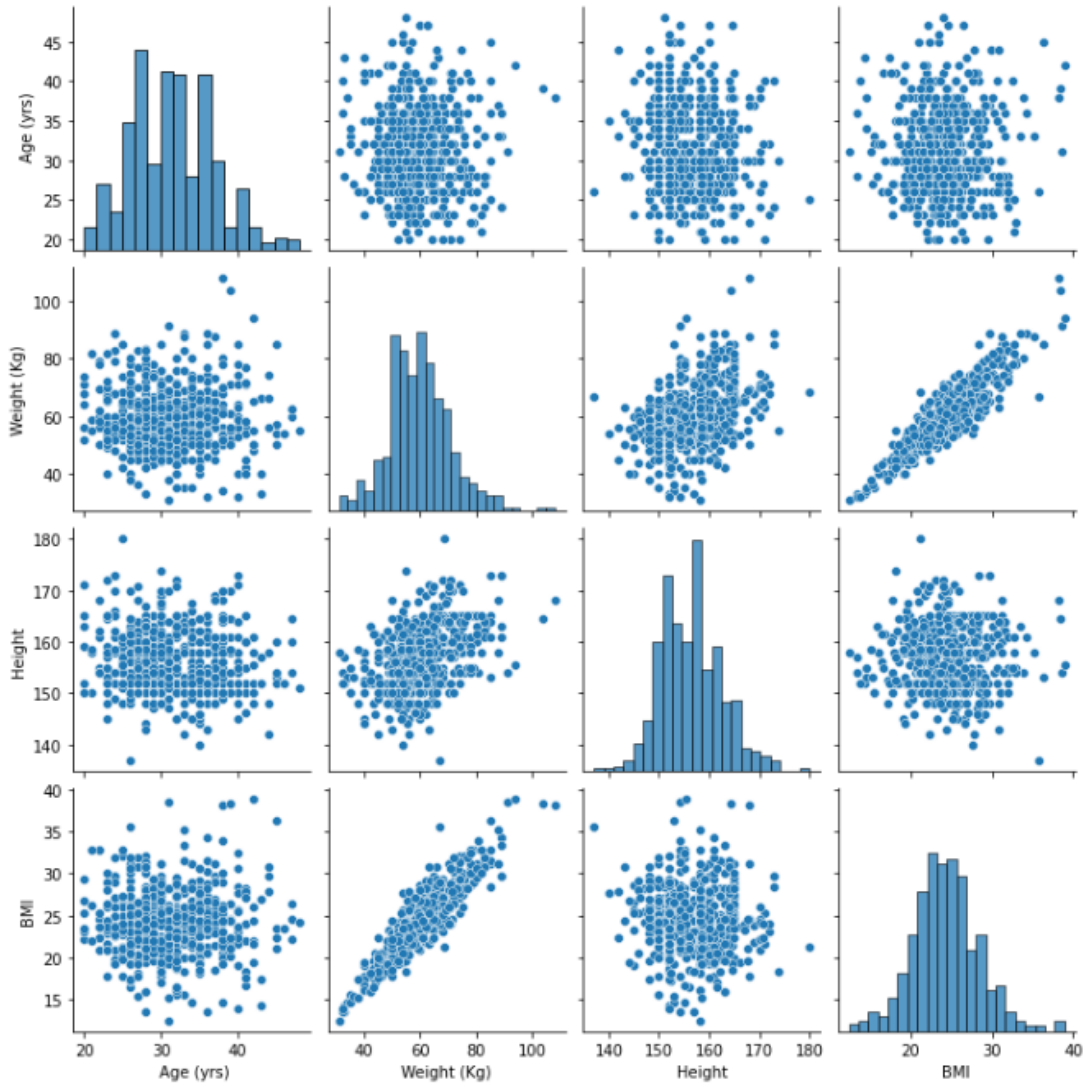
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
```

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 539 entries, 1 to 541
Data columns (total 42 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   PCOS (Y/N)              539 non-null    int64
 1    Age (yrs)              539 non-null    int64
 2   Weight (Kg)             539 non-null    float64
 3   Height                  539 non-null    float64
 4   BMI                     539 non-null    float64
 5   Blood Group             539 non-null    int64
 6   Pulse rate(bpm)         539 non-null    int64
 7   RR (breaths/min)        539 non-null    int64
 8   Hb(g/dl)                539 non-null    float64
 9   Cycle(R/I)              539 non-null    int64
 10  Cycle length(days)      539 non-null    int64
 11  Marraige Status (Yrs)   539 non-null    float64
 12  Pregnant(Y/N)           539 non-null    int64
 13  No. of aborptions       539 non-null    int64
 14    I   beta-HCG(mIU/mL)  539 non-null    float64
 15  II    beta-HCG(mIU/mL)  538 non-null    float64
 16  FSH(mIU/mL)             539 non-null    float64
 17  LH(mIU/mL)              539 non-null    float64
 18  FSH/LH                  539 non-null    float64
 19  Hip(inch)               539 non-null    int64
 20  Waist(inch)             539 non-null    int64
 21  Waist:Hip Ratio         539 non-null    float64
 22  TSH (mIU/L)             539 non-null    float64
 23  AMH(ng/mL)              538 non-null    float64
 24  PRL(ng/mL)              539 non-null    float64
 25  Vit D3 (ng/mL)          539 non-null    float64
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x7ff7b0e0b910>

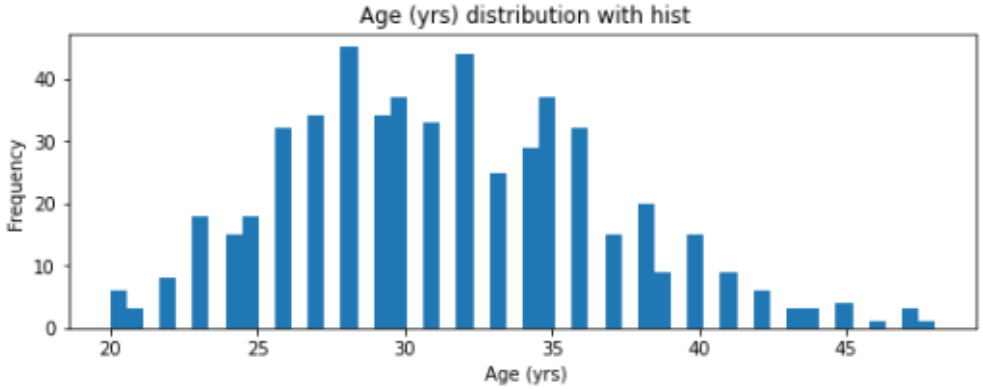


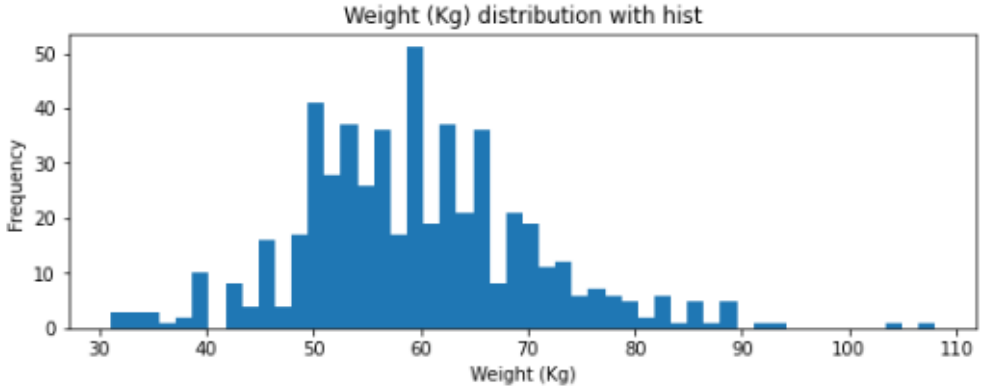
In [ ]:

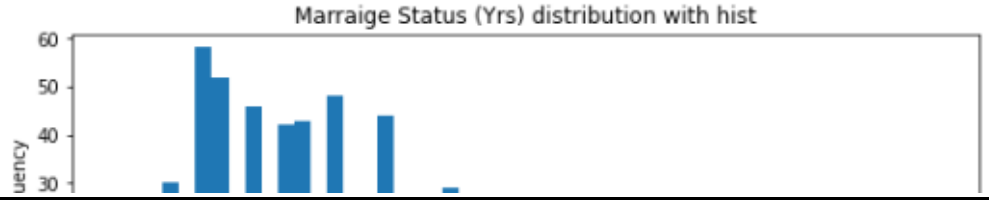
```
#Plot histogram
def plot_hist(variable):
    plt.figure(figsize = (9,3))
    plt.hist(df[variable], bins = 50)
    plt.xlabel(variable)
    plt.ylabel("Frequency")
    plt.title("{} distribution with hist".format(variable))
    plt.show()
```

In [ ]:

```
numericVar = [" Age (yrs)", "Weight (Kg)","Marraige Status (Yrs)"]
for n in numericVar:
    plot_hist(n)
```







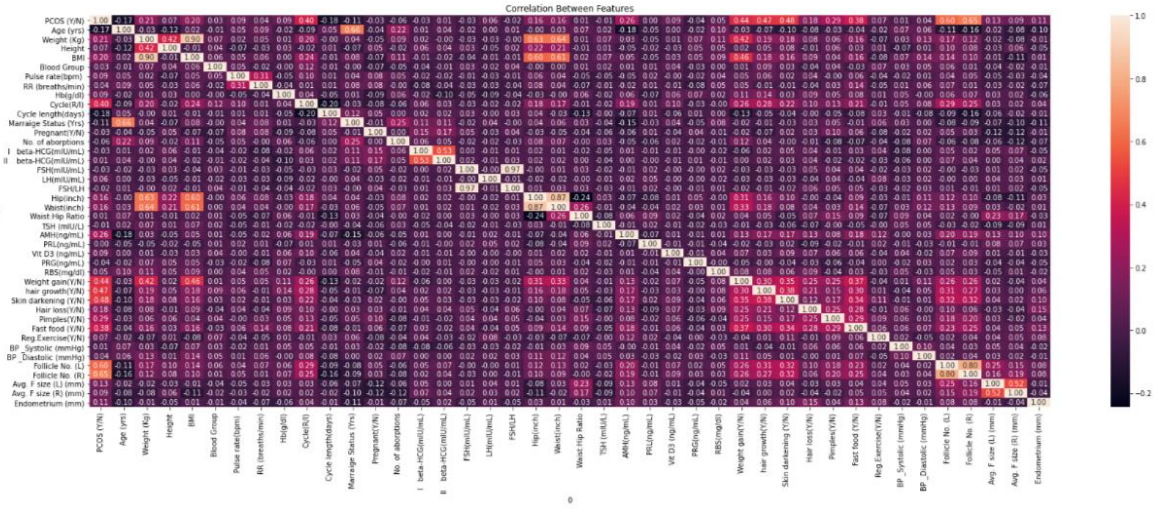
In [ ]:
```
df.corr()
```

Out[ ]:

0	PCOS (Y/N)	Age (yrs)	Weight (Kg)	Height	BMI	Blood Group	Pulse rate(bpm)	RR (breaths/min)	Hb(g/dl)	Cycle(R/I)	Cycle length(days)	Marraige Status (Yrs)	Pregnant(Y/N)	No. aborptio
PCOS (Y/N)	1.000000	-0.172436	0.210280	0.066361	0.198638	0.034244	0.092437	0.038031	0.086934	0.402312	-0.183781	-0.113701	-0.026586	-0.0567
Age (yrs)	-0.172436	1.000000	-0.031582	-0.121124	0.019868	-0.012655	0.045910	0.089509	-0.022466	-0.086540	0.052985	0.661834	-0.044235	0.2211
Weight (Kg)	0.210280	-0.031582	1.000000	0.420005	0.901719	0.071601	0.019995	0.045048	0.008979	0.201862	-0.004477	0.043238	-0.051148	0.0936
Height	0.066361	-0.121124	0.420005	1.000000	-0.006869	0.039981	-0.073801	-0.031093	0.025084	-0.020910	0.007512	-0.066508	0.048157	-0.0247
BMI	0.198638	0.019868	0.901719	-0.006869	1.000000	0.061174	0.050277	0.064197	0.002533	0.235702	-0.007615	0.083298	-0.074875	0.1095
Blood Group	0.034244	-0.012655	0.071601	0.039981	0.061174	1.000000	0.047719	-0.023064	-0.002153	0.123744	-0.008485	-0.002247	-0.070714	-0.0538
Pulse rate(bpm)	0.092437	0.045910	0.019995	-0.073801	0.050277	0.047719	1.000000	0.305751	-0.052581	0.103183	0.006831	0.038505	0.081828	0.0456
RR (breaths/min)	0.038031	0.089509	0.045048	-0.031093	0.064197	-0.023064	0.305751	1.000000	-0.038149	0.012243	0.005788	0.079823	0.081720	-0.0048
Hb(g/dl)	0.086934	-0.022466	0.008979	0.025084	0.002533	-0.002153	-0.052581	-0.038149	1.000000	0.039702	-0.052484	0.006273	-0.094530	0.0595
Cycle(R/I)	0.402312	-0.086540	0.201862	-0.020910	0.235702	0.123744	0.103183	0.012243	0.039702	1.000000	-0.204648	-0.032283	-0.077656	-0.0552
Cycle length(days)	-0.183781	0.052985	-0.004477	0.007512	-0.007615	-0.008485	0.006831	0.005788	-0.052484	-0.204648	1.000000	0.117289	0.050248	0.0045
Marraige Status (Yrs)	-0.113701	0.661834	0.043238	-0.066508	0.083298	-0.002247	0.038505	0.079823	0.006273	-0.032283	0.117289	1.000000	-0.007629	0.2461
Pregnant(Y/N)	-0.026586	-0.044235	-0.051148	0.048157	-0.074875	-0.070714	0.081828	0.081720	-0.094530	-0.077656	0.050248	-0.007629	1.000000	0.0020
No. of aborptions	-0.056741	0.221169	0.093623	-0.024775	0.109532	-0.053838	0.045672	-0.004838	0.059510	-0.055237	0.004583	0.246122	0.002085	1.0000
I beta-HCG(mIU/mL)	-0.027820	0.007928	0.015838	0.062524	-0.010223	-0.035430	-0.020625	-0.084756	-0.017163	0.064315	0.020100	0.111390	0.150637	0.0577
II beta-HCG(mIU/mL)	0.013413	0.042787	-0.000924	0.036754	-0.015905	-0.011020	-0.016383	-0.038889	-0.095008	0.028944	0.018823	0.112525	0.170652	0.0467
FSH(mIU/mL)	-0.030300	-0.017804	-0.025759	0.031094	-0.040789	0.028154	-0.013149	-0.032302	-0.047566	-0.025800	0.029796	-0.023549	0.052763	-0.0182
LH(mIU/mL)	0.064136	0.000479	-0.029878	-0.045381	-0.013384	-0.019520	-0.032415	-0.031081	-0.089329	-0.020987	-0.001587	0.035410	-0.034381	-0.0189
FSH/LH	-0.018535	0.012367	-0.004925	0.021976	-0.012138	0.036223	-0.013089	-0.043382	-0.039832	-0.016318	0.025847	-0.003001	0.044657	-0.0264
Hip(inch)	0.161700	-0.003565	0.633911	0.216065	0.596729	-0.001977	0.062698	0.075094	-0.025845	0.175864	0.039425	0.037672	-0.032181	0.0789
Waist(inch)	0.162143	0.033705	0.639154	0.209676	0.607394	0.003285	0.038037	0.039064	0.001472	0.168932	0.026419	0.056751	0.048564	0.0738

PROJECT DEMONSTRATION

In [ ]:
```
#Plot correlation matrix
corr_matrix= df.corr()
plt.subplots(figsize=(30,10))
sns.heatmap(corr_matrix, annot = True, fmt = ".2f");
plt.title("Correlation Between Features")
plt.show()
```



In [ ]:
```
#Split the dataset into 70% training data and 30% testing data
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3 , random_state = 0)
```

In [ ]:
```
#Scale the data (Feature Scaling)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

In [ ]:
```
# Create a function for the models
def models(X_train, Y_train):
  #Logistic Regression
  from sklearn.linear_model import LogisticRegression
  log = LogisticRegression(random_state = 0)
  log.fit(X_train, Y_train)

  #Random Forest Classifier
  from sklearn.ensemble import RandomForestClassifier
  forest = RandomForestClassifier(n_estimators = 50, criterion = 'entropy', random_state = 0)
  forest.fit(X_train, Y_train)

  #Print the models' accuracy on the training data
  print('Logistic Regression Training Accuracy:', log.score(X_train, Y_train))
  print('Random Forest Classifier:', forest.score(X_train, Y_train))

  return log, forest
```

In [ ]:
```
model = models(X_train, Y_train)
```

```
Logistic Regression Training Accuracy: 0.92
Random Forest Classifier: 1.0
```

SUMMARY

The purpose of this project was to build a PCOS detecting Analyser. So, through our analysis and research, we've successfully made this possible to have built our project which would benefit when put to use. There's no test to definitively diagnose PCOS. The doctor is likely to start with a discussion of the patient's medical history, including the menstrual periods and weight changes. A physical exam will include checking for signs of excess hair growth, insulin resistance and acne. Awareness helps the public understand the above-mentioned symptoms which are not something to be ignored and getting it checked is necessary. Therefore, there's a need to have a device which would properly detect PCOS with accurate results. Hence, this analyser is designed which would serve for it as it gives 85-90% accuracy. We got to learn a lot of applications of Artificial Intelligence with great experiences.

REFERENCES

1. **https://github.com/Aryadubey14/PCOS/blob/main/PCOSDiagnosisModel.ipynb**
   (Entire project and model link)
2. **www.googleimages.com**
3. **Diagnosis and Treatment of Polycystic Ovary Syndrome By Tracy Williams, MD and Rami Mortada, MD.**
4. **https://youtu.be/ErqD1BC5-v8**
5. **https://www.trp.org.in/issues/detection-of-polycystic-ovarian-syndrome-a-literature-survey**
6. **A Detailed Research on Detection of Polycystic Ovary Syndrome from Ultrasound Images of Ovaries C. Gopalakrishnan, M. Iyapparaja**
7. **i-HOPE: Detection And Prediction System For Polycystic Ovary Syndrome (PCOS) Using Machine Learning Techniques**
   Publisher: IEEE