SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE

NAAC ACCREDITED ‘A’ GRADE
COLLEGE(ISO 9001:2008
CERTIFIED INSTITUTION)
NERUL, NAVI MUMBAI – 400706

PROJECT REPORT ON

POLYCYSTIC OVARY SYNDROME(PCOS) DETECTION

SUBMITTED BY

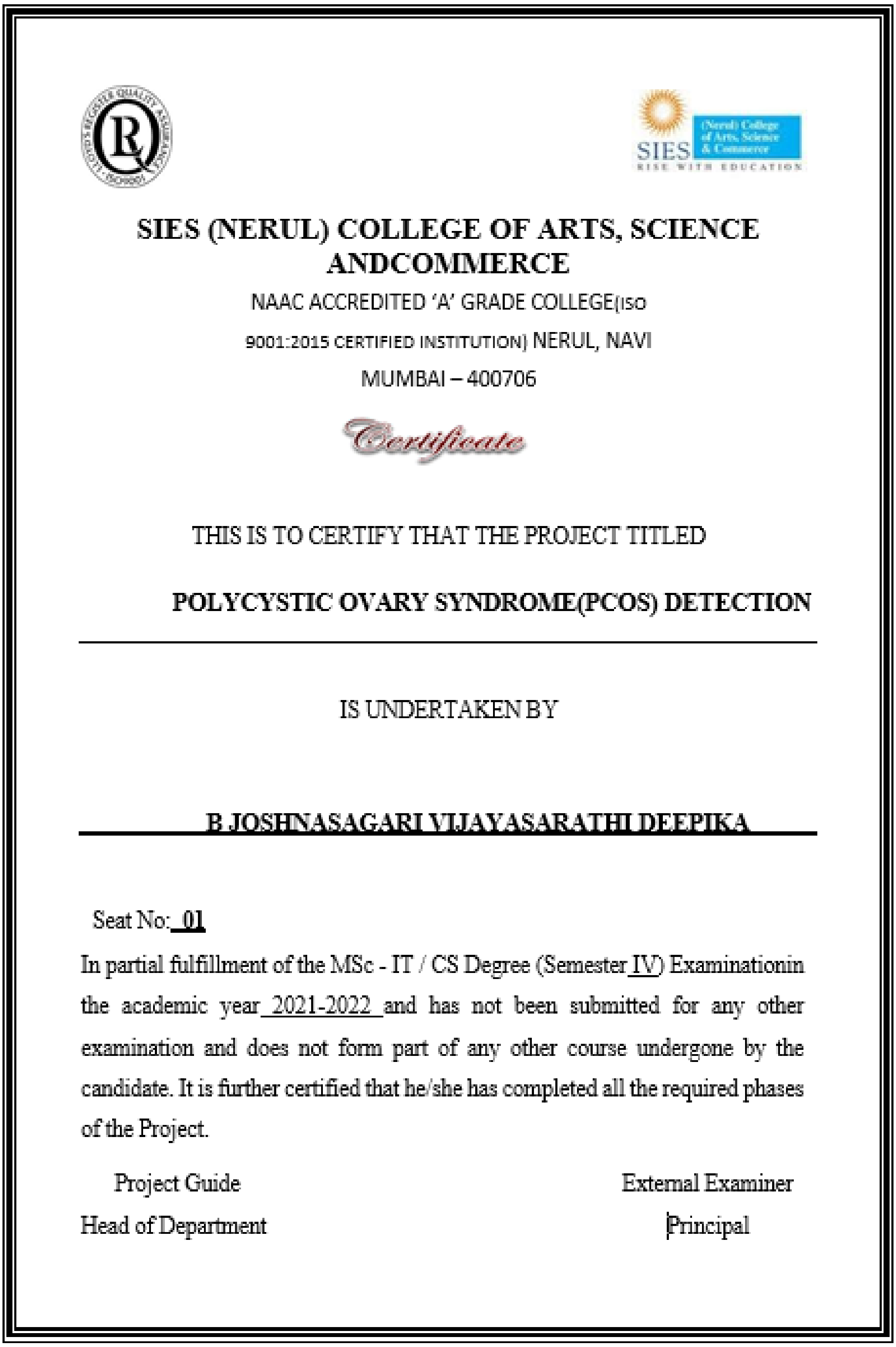
B JOSHNASAGARI VIJAYASARATHI DEEPIKA

UNDER THE GUIDANCE OF

ASST. PROF. MANASVI SHARMA

SUBMITTED IN THE PARTIAL FULLFILMENT FOR THE DEGREE OF

MSc. COMPUTER SCIENCE
SEMESTER – IV, 2021 - 2022



SIES (NERUL) COLLEGE OF ARTS, SCIENCE ANDCOMMERCE

NAAC ACCREDITED 'A' GRADE COLLEGE(ISO 9001:2015 CERTIFIED INSTITUTION) NERUL, NAVI MUMBAI – 400706

Certificate

THIS IS TO CERTIFY THAT THE PROJECT TITLED

POLYCYSTIC OVARY SYNDROME(PCOS) DETECTION

IS UNDERTAKEN BY

B JOSHNASAGARI VIJAYASARATHI DEEPIKA

Seat No: **01**

In partial fulfillment of the MSc - IT / CS Degree (Semester IV) Examinationin the academic year 2021-2022 and has not been submitted for any other examination and does not form part of any other course undergone by the candidate. It is further certified that he/she has completed all the required phases of the Project.

Project Guide	External Examiner
Head of Department	Principal

ACKNOWLEDGMENT

I extend my heartfelt gratitude and thanks to Asst. Professor Manasvi Sharma for providing me excellent guidance to work on this project and for their understanding and assistance by providing all the necessary information needed for my project topic. I would also like to acknowledge all the staffs for providing a helping hand to us in times of queries & problems. The project is a result of the efforts of all the peoples who are associated with the project directly or indirectly, who helped me to work to complete the project within the specified time frame. They motivated me in the project and gave a feedback on it to improve my adroitness.

Thanks to all my teachers, who were a part of the project in numerous ways and for the help and inspiration they extended to me and for providing the needed motivation.

With all Respects & Gratitude, I would like to thanks to all the people, who have helped for the development of the Project.

B JOSHNASAGARI VIJAYASARATHI DEEPIKA

MSc. Computer Science (Part-II)

SIES (Nerul) College of Arts, Science, and Commerce.

PROJECT ON

“POLYCYSTIC OVARY SYNDROME(PCOS) DETECTION”

ABSTRACT:

Polycystic Ovary Syndrome (PCOS) is a medical condition which causes hormonal disorder in women in their childbearing years. The hormonal imbalance leads to a delayed or even absent menstrual cycle.

Women with PCOS majorly suffer from excessive weight gain, facial hair growth, acne, hair loss, skin darkening and irregular periods leading to infertility in rare cases.

The existing methodologies and treatments are insuffcient for early-stage detection and prediction. To deal with this problem, we propose a system which can help in early detection and prediction of PCOS treatment from an optimal and minimal set of parameters.

Finally, the test dataset is used for performing the feature extraction process and the results are met with 85% accuracy using performance factors.

Thus the Objective of the Project is to Predict /See that the Patient having PCOS or not.

INTRODUCTION:

PCOS is a menstrual problem which is faced by many womens in teenagers or middle aged women. PCOS is a condition that affects a women's hormone levels.

Women's with PCOS produce higher then normal amount of male hormones.This hormone imbalance causes their body to skip the menstrual cycle/periods and makes it very difficult for them to get pregnant or concive. PCOS also causes hair growth on face and body which results in baldness.It can contribute to long term health problems such as diabetes and heart dieases.

PCOS affets a women's ovaries,the reproductive organs that produce estrogen and progestrogen hormones to regulate the menstrual cycle.The Ovaries also produce a small amount of male hormones called Androgens. Symptoms:Irregular Periods,Facial Growth,Growth of Acne,Weight Gain,Women has to face a lot of difficulties like sleep disorder and regular health issues.

The idea behind building this Project is that PCOS is a Problem in most of the women and they are not aware about it and if they are aware but they feel uncomfortable to check by the doctor(Gynaecologist) as most of them think that they have to go through lot of tests but in reality it is not true as early dignosis and treatment can reduce the risk of long term health issues.

PCOS affect 1 out of every 10 women.To eradicate the serious problem I have built this project to predict and help the women to acknowledge their condition weather they are suffering from PCOS or not.This Platform takes input from user then predicts the result. PCOS doesn't just affect a selected group of individuals.Nearly 6 to 12% of women have PCOS.

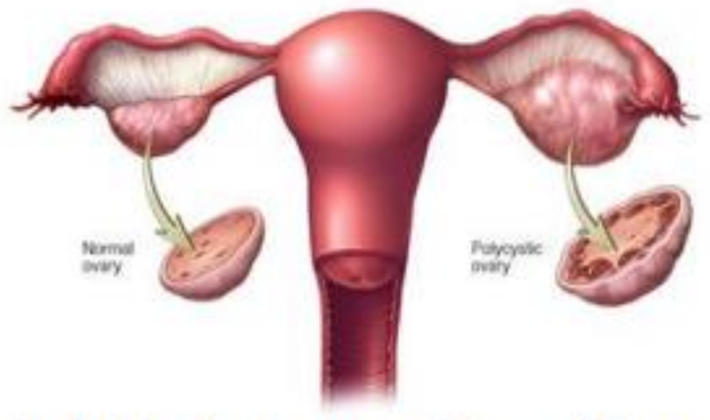
PCOS affects women of all backgrounds, and sizes. It affects models and athletes to populations, 16-year-old girls to 45-year-old women. Women with cysts who are infertile and experience physical side effects.Heavy menstrual bleeding and other irregularities, could be caused by a range of conditions, such as uterine fibroids, bleeding disorders,in addition to PCOS.

Women with PCOS are at a higher risk for insulin resistance, high blood sugar, obesity, high cholesterol, high blood pressure,Diabetes, and other cardiovascular diseases.One of the biggest issues with PCOS is that no one woman with PCOS looks or even experiences the same things as another.

There are different phases they are:Menstruation,Follicular Phase,Ovulation,Luteal Phase.

The term PCOS was first defined by Michael Leventhal and Irving Stein as a Triple term of 'Obesity', 'Hirsutism', and 'Amenorrhea'. In 1935 they realized the relationships between reproductive disorders and obesity. Hence it is also called as the 'Hyperandrogenic Anovulation' (HA) or 'Stein-Leventhal Syndrome. It is the most popular disorder of the endocrine system in the ovaries, affecting approximately 2-8% of women giving birth worldwide. Now it is also called "Syndrome O".

PCOS is diagnosed by gynecologists and good and better understanding required analyzing the disorder with a bunch of symptoms. The calculation of PCOS prevalence depends on the parameters used to determine this syndrome. Because the PCOS symptoms emerged and correlate with normal puberty progression features may not be examined in the early stages. This may be due to the inability to identify the young girl’s disorder.



Poly Cystic Ovary vs Normal Ovary.

IMPLEMENTATION DETAILS:

Data Collection:

Existing Data:

I have downloaded the data from the kaggle. The dataset is available in csv format.

Data set link:

https://www.kaggle.com/prasoonkottarathil/polycystic-ovary-syndrome-pcos

The Dataset is about different attributes such as Weight,Height,Pulse Rate(bpm), RR (breaths/min), Hb(g/dl), Cycle(R/I), Cycle length(days), I beta-HCG(mIU/mL), II beta-HCG(mIU/mL), PCOS(Y/N),FSH(mIU/mL), LH(mIU/mL), Follicle No. (L), Follicle No. (R), Avg. F size (L) (mm), Avg. F size (R) (mm), Endometrium (mm) etc.

Real Time Data:

I have collected real time data using Google Form.In this as soon as the results are available, you will be able to see them right away. You can also use charts and graphs to quickly summarise survey results.

I've prepared 15 questions that will help me determine how people think about the PCOS and what are their difficulties facing PCOS and summarize them accordingly.

The Questions are as follows:

1. What is Your age?
2. Have you ever been Pregnant or had Abortions?
3. Do you Exercise Regularly?
4. How much is your weight and height?
5. Do you eat junk-food on daily basis?
6. Do you Sleep well?
7. Do you have major facial growth?
8. Are you facing with Hair loss?
9. Are you facing problem with weight gain or weight loss?
10. Do you get too upset or depressed/stressed during your monthly cycle?
11. When did you get your first Menstrual Cycle?

12. What are/were the challenges did you face with PCOS?
13. Are you taking any treatment for PCOS?
14. Are you facing problem with irregular periods and weight loss/gain and acne?
15. Are you diagnosed with any other health issues when you were/are diagnosed with PCOS?

System Requirement:

Google Colab:

I used Google Colab for this project to write and execute arbitrary python code through the browser.Colab is hosted notebook service that requires no setup to use , while providing free access to computing resources including GPUs.

You can search Colab notebooks using Google Drive. Clicking on the Colab logo at the top left of the notebook view will show all notebooks in Drive. You can also search for notebooks that you have opened recently using File > Open notebook.

Python:

Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python has been built with extraordinary Python libraries that are used in Big Data for solving problems that are as follow:-

NumPy

Seaborn

Pandas

Sklearn

Matplotlib etc.

Algorithm:

I have used the classification algorithms in this project.

Decision Tree:

Decision tree is a supervised learning algorithm that is perfect for classification problems. As my dataset contains a high number of categorical values, I used decision tree to divide the data into leaf and nodes to predict the outcome.

Random Forest:

Random forest algorithm can be used for both classifications and regression task. It provides higher accuracy through cross validation. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

Logistic Regression:

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

SVM:

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

KNN:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

XGBRF & CatBoost Classifier:

CatBoost has the flexibility of giving indices of categorical columns so that it can be encoded as one-hot encoding using one_hot_max_size.

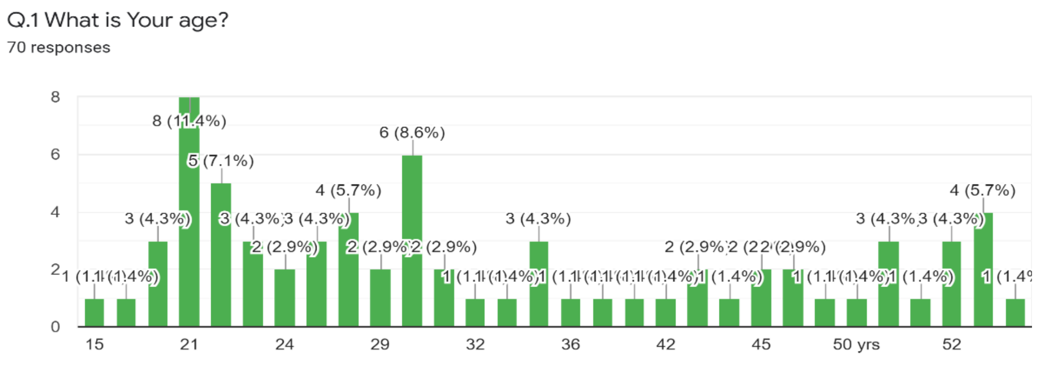
Unlike CatBoost, XGBoost cannot handle categorical features by itself, it only accepts numerical values similar to Random Forest. Therefore one has to perform various encodings like label encoding, mean encoding or one-hot encoding before supplying categorical data to XGBoost.

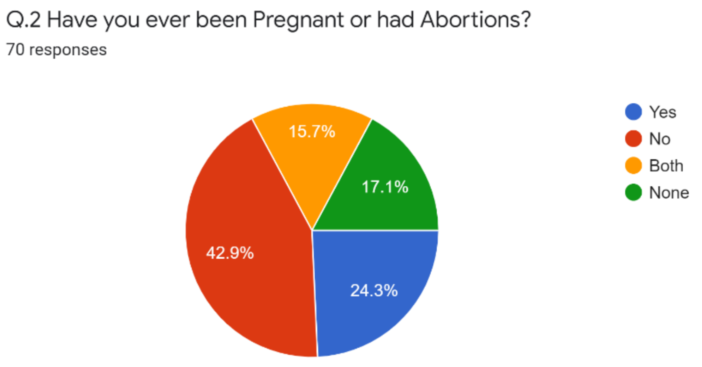
EXPERIMENTAL SET UP AND RESULTS:

Existing Data:

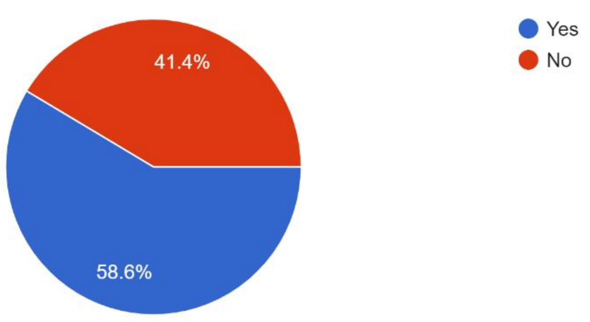
The Dataset has 43 attributes that contains information of Patient File No, PCOS(Y/N), Age (yrs),Weight(kg),Height(Cm),BMI,Blood group,Pulse Rate(bpm), RR (breaths/min), Hb(g/dl), Cycle(R/I), Cycle length(days), Marraige Status (Yrs), Pregnant(Y/N), No. of aborptions, I beta-HCG(mIU/mL), II beta-HCG(mIU/mL), FSH(mIU/mL), LH(mIU/mL), Hip(inch),Waist(inch), Waist/Hip Ratio, TSH (mIU/L), AMH(ng/mL), PRL(ng/mL), Vit D3 (ng/mL), PRG(ng/mL), RBS(mg/dl), Weight gain(Y/N), hair growth(Y/N), Skin darkening (Y/N), Hair loss(Y/N), Pimples(Y/N), Fast food (Y/N), Reg.Exercise(Y/N), BP_Systolic (mmHg), BP_Diastolic (mmHg),Follicle No. (L), Follicle No. (R), Avg. F size (L) (mm), Avg. F size (R) (mm), Endometrium (mm).

Real Data:

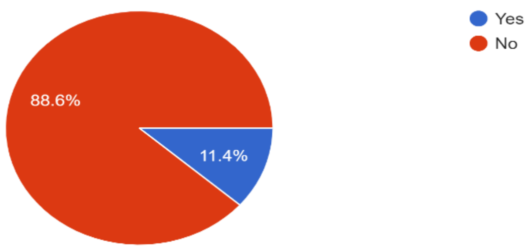




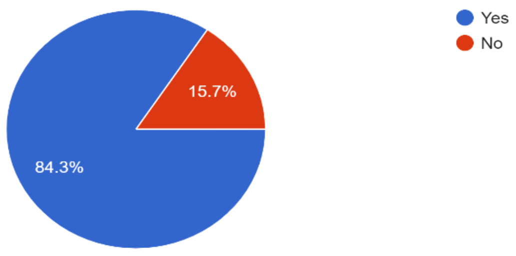
Q.3 Do you Exercise Regularly?
70 responses



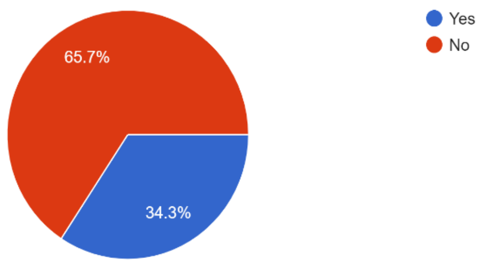
Q.5 Do you eat junk-food on daily basis?
70 responses

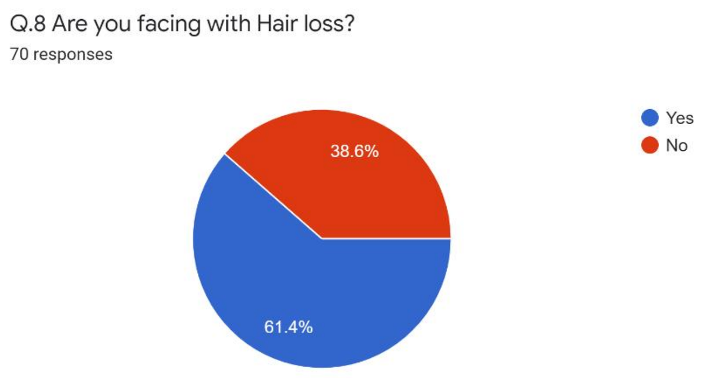


Q.6 Do you Sleep well?
70 responses



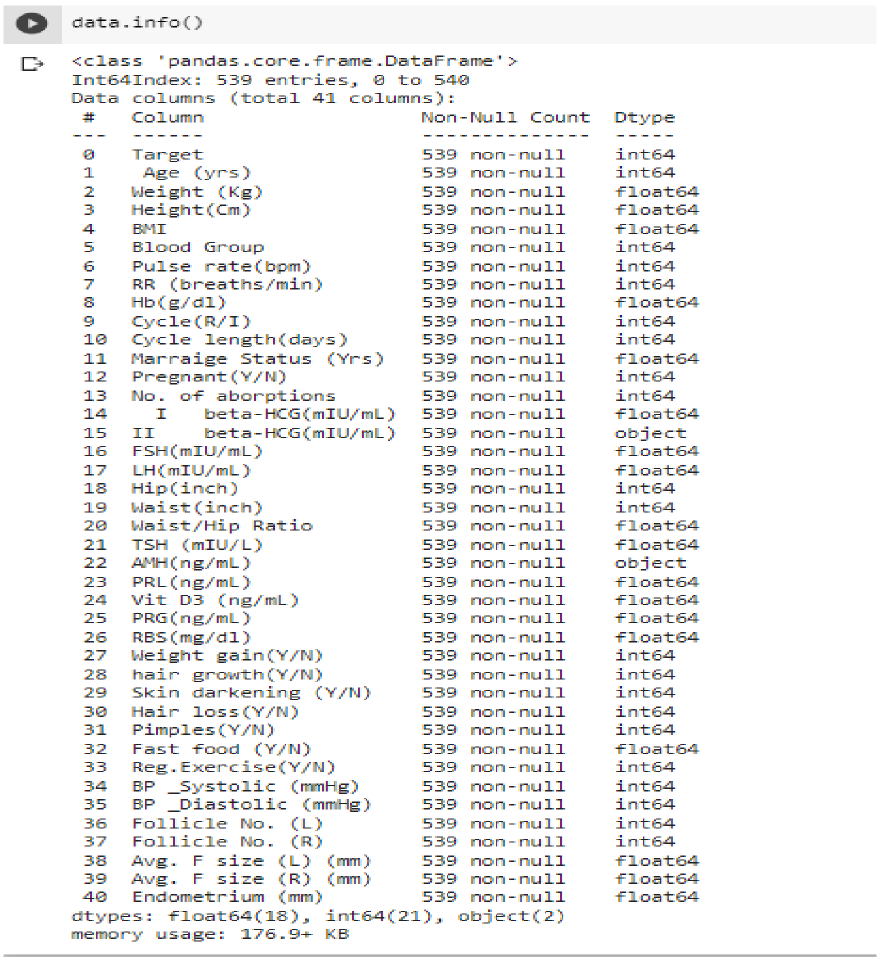
Q.7 Do you have major facial growth?
70 responses





From this form I have collected the answers in terms of binary and categorical values .The binary values are Yes/NO where people will select yes or no on other hand the Categorical values are for the question ‘When did you get your first Menstrual Cycle?’, ‘What are/were the challenges did you face with PCOS?’, ‘What is Your age?’

Exisiting Data:

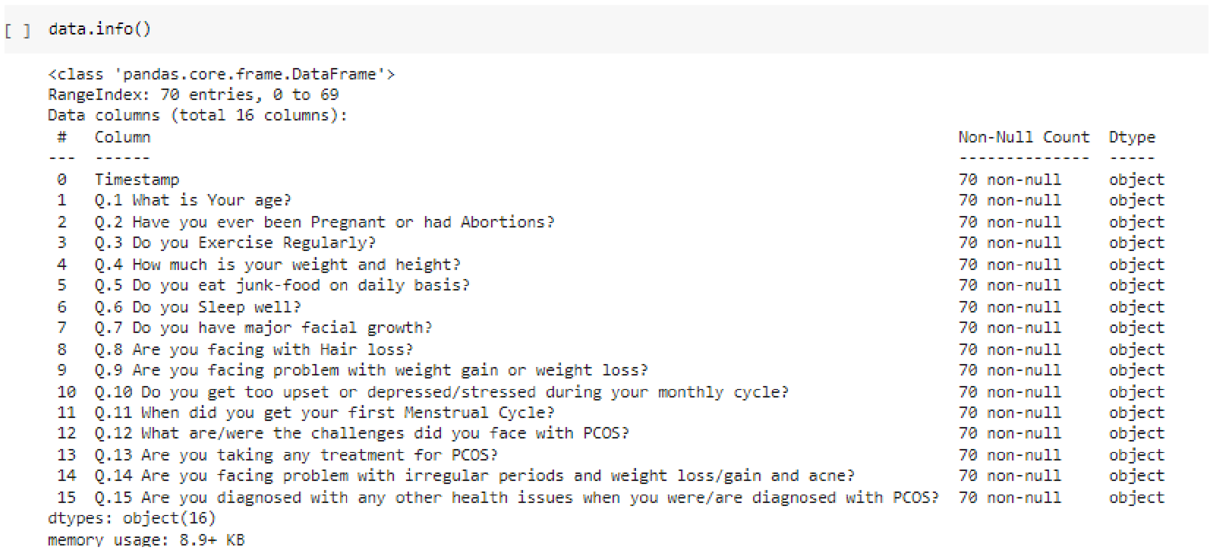


```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 539 entries, 0 to 540
Data columns (total 41 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Target                  539 non-null    int64
 1    Age (yrs)              539 non-null    int64
 2   Weight (Kg)             539 non-null    float64
 3   Height(Cm)              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)  539 non-null    object
 16  FSH(mIU/mL)             539 non-null    float64
 17  LH(mIU/mL)              539 non-null    float64
 18  Hip(inch)               539 non-null    int64
 19  Waist(inch)             539 non-null    int64
 20  Waist/Hip Ratio         539 non-null    float64
 21  TSH (mIU/L)             539 non-null    float64
 22  AMH(ng/mL)              539 non-null    object
 23  PRL(ng/mL)              539 non-null    float64
 24  Vit D3 (ng/mL)          539 non-null    float64
 25  PRG(ng/mL)              539 non-null    float64
 26  RBS(mg/dl)              539 non-null    float64
 27  Weight gain(Y/N)        539 non-null    int64
 28  hair growth(Y/N)        539 non-null    int64
 29  Skin darkening (Y/N)    539 non-null    int64
 30  Hair loss(Y/N)          539 non-null    int64
 31  Pimples(Y/N)            539 non-null    int64
 32  Fast food (Y/N)         539 non-null    float64
 33  Reg.Exercise(Y/N)       539 non-null    int64
 34  BP _Systolic (mmHg)     539 non-null    int64
 35  BP _Diastolic (mmHg)    539 non-null    int64
 36  Follicle No. (L)        539 non-null    int64
 37  Follicle No. (R)        539 non-null    int64
 38  Avg. F size (L) (mm)    539 non-null    float64
 39  Avg. F size (R) (mm)    539 non-null    float64
 40  Endometrium (mm)        539 non-null    float64
dtypes: float64(18), int64(21), object(2)
memory usage: 176.9+ KB
```

The Dataset Consists of 541rows and 44columns

Real Time Data:



```
[ ]  data.info()
```

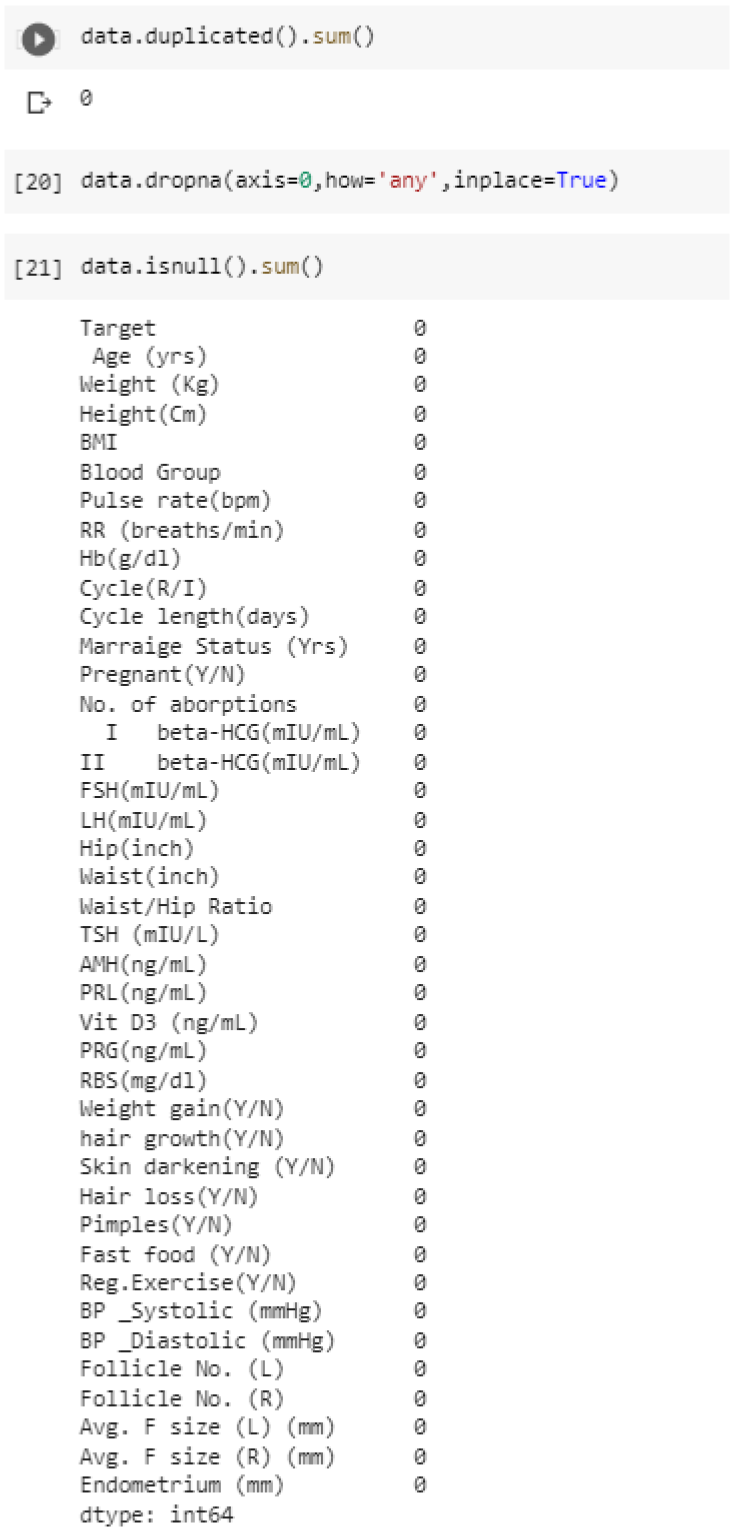
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 16 columns):
 #   Column                                                                          Non-Null Count  Dtype 
---  ------                                                                          --------------  ----- 
 0   Timestamp                                                                       70 non-null     object
 1   Q.1 What is Your age?                                                           70 non-null     object
 2   Q.2 Have you ever been Pregnant or had Abortions?                               70 non-null     object
 3   Q.3 Do you Exercise Regularly?                                                  70 non-null     object
 4   Q.4 How much is your weight and height?                                         70 non-null     object
 5   Q.5 Do you eat junk-food on daily basis?                                        70 non-null     object
 6   Q.6 Do you Sleep well?                                                          70 non-null     object
 7   Q.7 Do you have major facial growth?                                            70 non-null     object
 8   Q.8 Are you facing with Hair loss?                                              70 non-null     object
 9   Q.9 Are you facing problem with weight gain or weight loss?                     70 non-null     object
 10  Q.10 Do you get too upset or depressed/stressed during your monthly cycle?      70 non-null     object
 11  Q.11 When did you get your first Menstrual Cycle?                               70 non-null     object
 12  Q.12 What are/were the challenges did you face with PCOS?                       70 non-null     object
 13  Q.13 Are you taking any treatment for PCOS?                                     70 non-null     object
 14  Q.14 Are you facing problem with irregular periods and weight loss/gain and acne?   70 non-null     object
 15  Q.15 Are you diagnosed with any other health issues when you were/are diagnosed with PCOS?  70 non-null     object
dtypes: object(16)
memory usage: 8.9+ KB
```

The Dataset Consists of 70rows and 16columns

DATA CLEANING:

Existing Data:

Checking for Duplicte and Null Values:



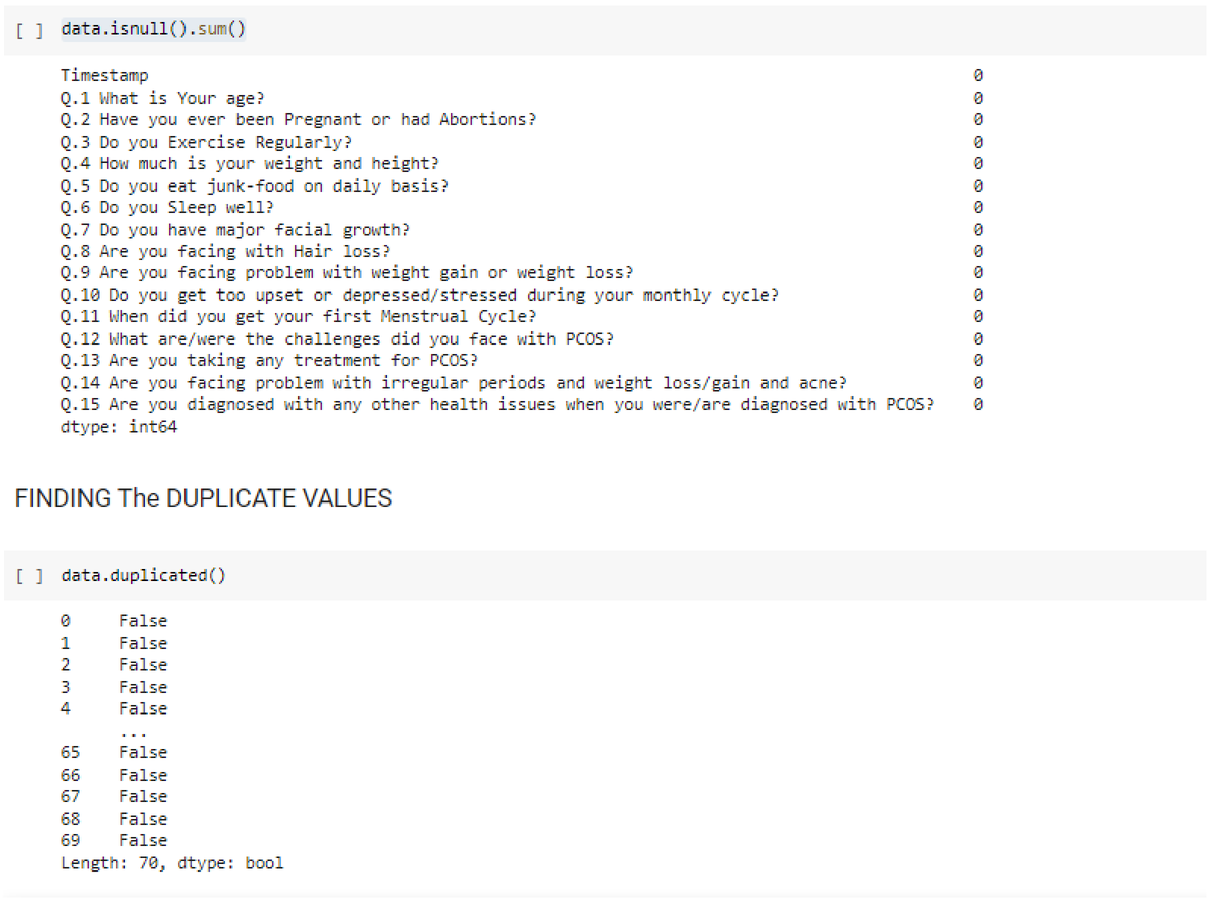
```
data.duplicated().sum()

0
```

```
[20] data.dropna(axis=0,how='any',inplace=True)
```

```
[21] data.isnull().sum()

Target                    0
 Age (yrs)                0
Weight (Kg)               0
Height(Cm)                0
BMI                       0
Blood Group               0
Pulse rate(bpm)           0
RR (breaths/min)          0
Hb(g/dl)                  0
Cycle(R/I)                0
Cycle length(days)        0
Marraige Status (Yrs)     0
Pregnant(Y/N)             0
No. of aborptions         0
  I   beta-HCG(mIU/mL)    0
II    beta-HCG(mIU/mL)    0
FSH(mIU/mL)               0
LH(mIU/mL)                0
Hip(inch)                 0
Waist(inch)               0
Waist/Hip Ratio           0
TSH (mIU/L)               0
AMH(ng/mL)                0
PRL(ng/mL)                0
Vit D3 (ng/mL)            0
PRG(ng/mL)                0
RBS(mg/dl)                0
Weight gain(Y/N)          0
hair growth(Y/N)          0
Skin darkening (Y/N)      0
Hair loss(Y/N)            0
Pimples(Y/N)              0
Fast food (Y/N)           0
Reg.Exercise(Y/N)         0
BP _Systolic (mmHg)       0
BP _Diastolic (mmHg)      0
Follicle No. (L)          0
Follicle No. (R)          0
Avg. F size (L) (mm)      0
Avg. F size (R) (mm)      0
Endometrium (mm)          0
dtype: int64
```

In this dataset, there are no Missing values and Duplicate values.

Real Time Data:
Checking for Duplicte and Null Values:



```
[ ] data.isnull().sum()
```

```
Timestamp                                                                                0
Q.1 What is Your age?                                                                    0
Q.2 Have you ever been Pregnant or had Abortions?                                        0
Q.3 Do you Exercise Regularly?                                                           0
Q.4 How much is your weight and height?                                                  0
Q.5 Do you eat junk-food on daily basis?                                                 0
Q.6 Do you Sleep well?                                                                   0
Q.7 Do you have major facial growth?                                                     0
Q.8 Are you facing with Hair loss?                                                       0
Q.9 Are you facing problem with weight gain or weight loss?                              0
Q.10 Do you get too upset or depressed/stressed during your monthly cycle?               0
Q.11 When did you get your first Menstrual Cycle?                                        0
Q.12 What are/were the challenges did you face with PCOS?                                0
Q.13 Are you taking any treatment for PCOS?                                              0
Q.14 Are you facing problem with irregular periods and weight loss/gain and acne?        0
Q.15 Are you diagnosed with any other health issues when you were/are diagnosed with PCOS?    0
dtype: int64
```

FINDING The DUPLICATE VALUES

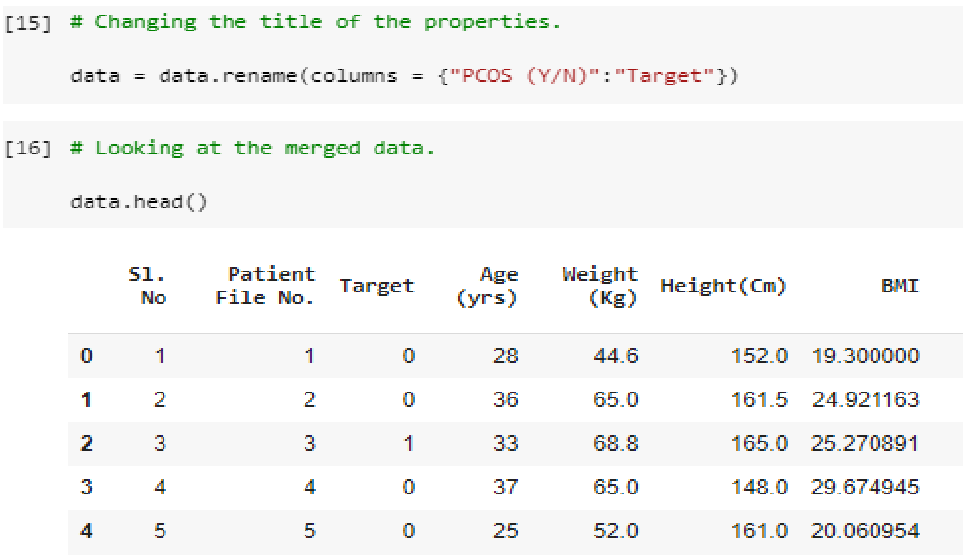
```
[ ] data.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
       ...  
65     False
66     False
67     False
68     False
69     False
Length: 70, dtype: bool
```

In this dataset, there are no Missing values and Duplicate values.

Renaming the Columns:

Existing Data:

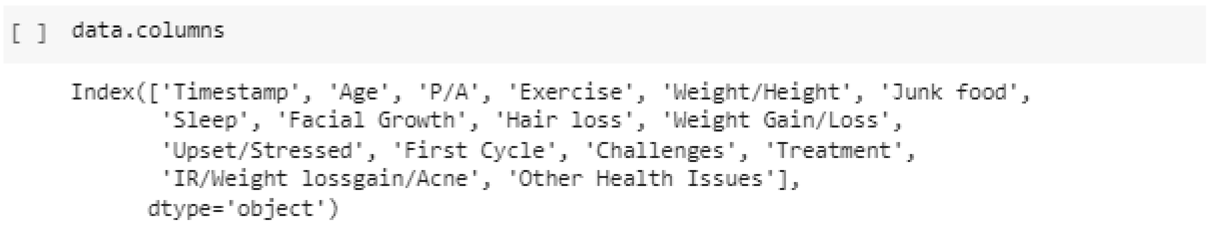


```
[15] # Changing the title of the properties.

     data = data.rename(columns = {"PCOS (Y/N)":"Target"})
```

```
[16] # Looking at the merged data.

     data.head()
```

	Sl. No	Patient File No.	Target	Age (yrs)	Weight (Kg)	Height(Cm)	BMI
0	1	1	0	28	44.6	152.0	19.300000
1	2	2	0	36	65.0	161.5	24.921163
2	3	3	1	33	68.8	165.0	25.270891
3	4	4	0	37	65.0	148.0	29.674945
4	5	5	0	25	52.0	161.0	20.060954

Real Time Data:

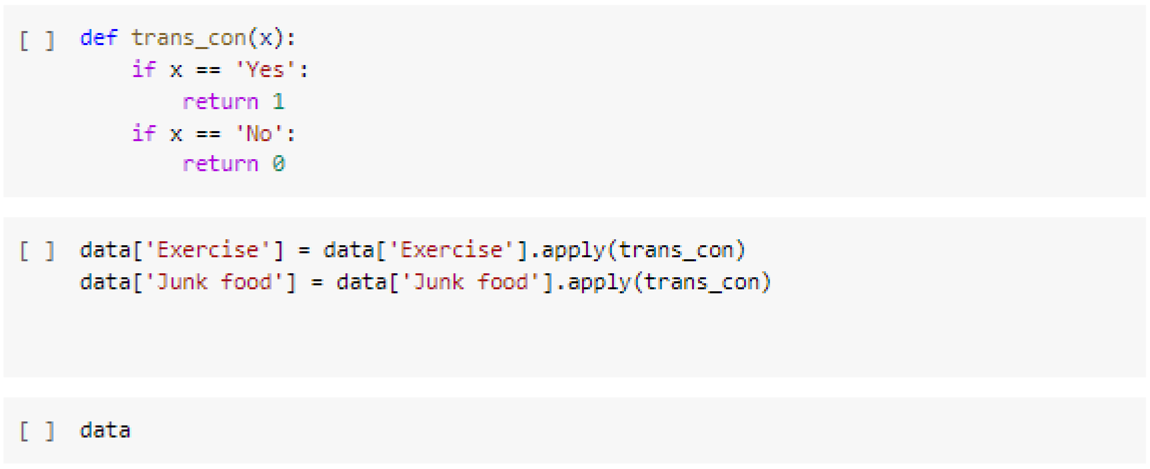


```
[ ]  data.columns

     Index(['Timestamp', 'Age', 'P/A', 'Exercise', 'Weight/Height', 'Junk food',
            'Sleep', 'Facial Growth', 'Hair loss', 'Weight Gain/Loss',
            'Upset/Stressed', 'First Cycle', 'Challenges', 'Treatment',
            'IR/Weight lossgain/Acne', 'Other Health Issues'],
           dtype='object')
```

Conversion of data from object to integer

I have used Decision Tree Algorithm for my prediction where it work on numeric data therefore I have converted my binary data in to numeric values i.e Yes = 1 and No = 0 .

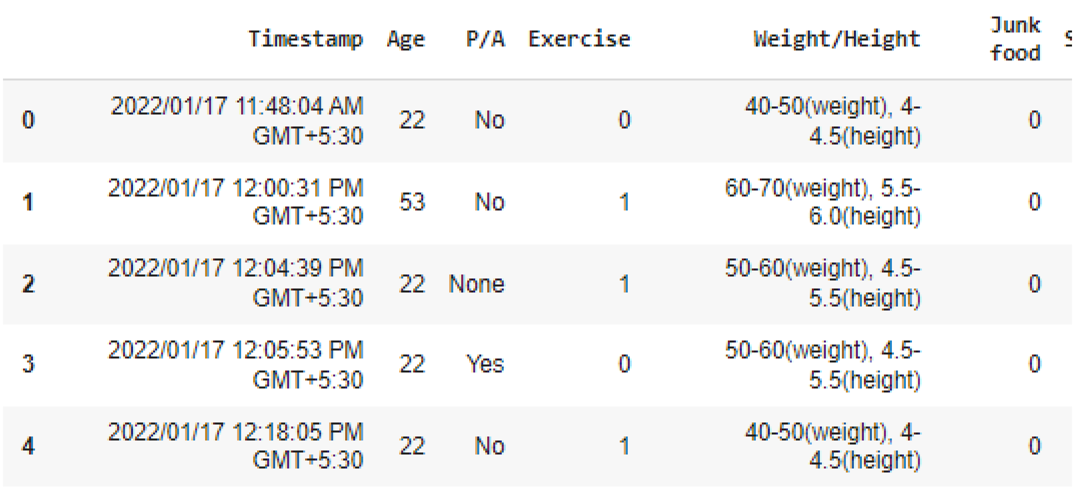
Real Time Data:



```
[ ]  def trans_con(x):
         if x == 'Yes':
             return 1
         if x == 'No':
             return 0
```

```
[ ]  data['Exercise'] = data['Exercise'].apply(trans_con)
     data['Junk food'] = data['Junk food'].apply(trans_con)
```

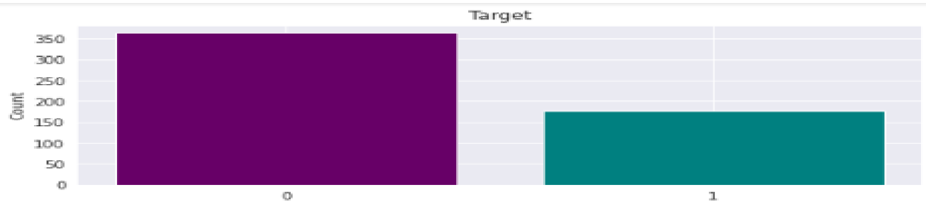
```
[ ]  data
```



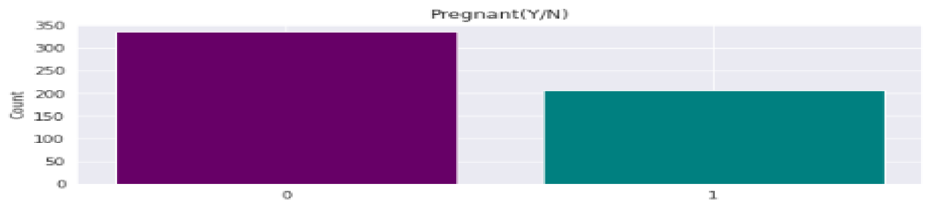
	Timestamp	Age	P/A	Exercise	Weight/Height	Junk food	S
0	2022/01/17 11:48:04 AM GMT+5:30	22	No	0	40-50(weight), 4-4.5(height)	0	
1	2022/01/17 12:00:31 PM GMT+5:30	53	No	1	60-70(weight), 5.5-6.0(height)	0	
2	2022/01/17 12:04:39 PM GMT+5:30	22	None	1	50-60(weight), 4.5-5.5(height)	0	
3	2022/01/17 12:05:53 PM GMT+5:30	22	Yes	0	50-60(weight), 4.5-5.5(height)	0	
4	2022/01/17 12:18:05 PM GMT+5:30	22	No	1	40-50(weight), 4-4.5(height)	0	

DATA VISUALIZATION:

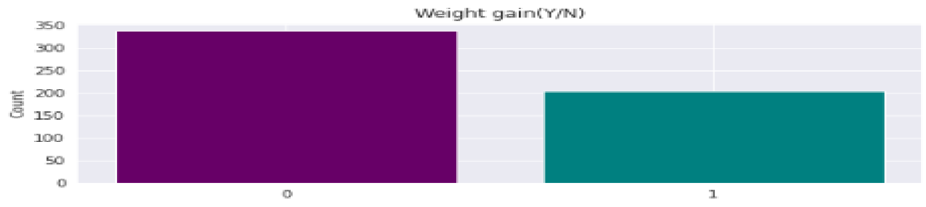
Existing Data:



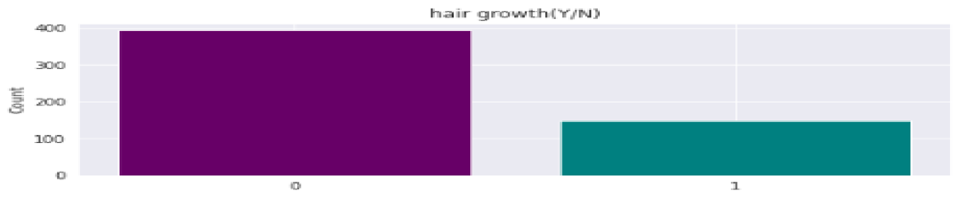
```
Target:
 0    364
1    177
Name: Target, dtype: int64
```



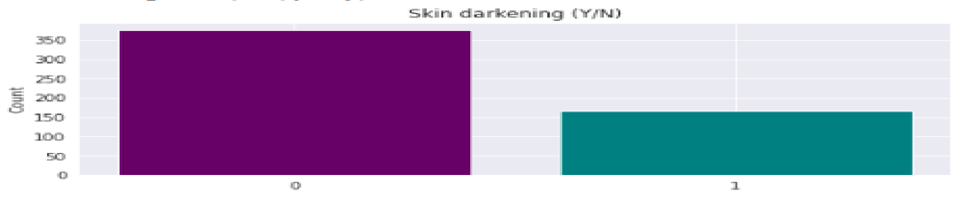
```
Pregnant(Y/N):
 0    335
1    206
Name: Pregnant(Y/N), dtype: int64
```



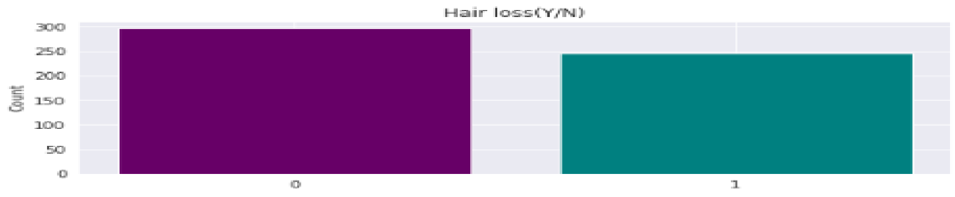
```
Weight gain(Y/N):
 0    337
1    204
Name: Weight gain(Y/N), dtype: int64
```



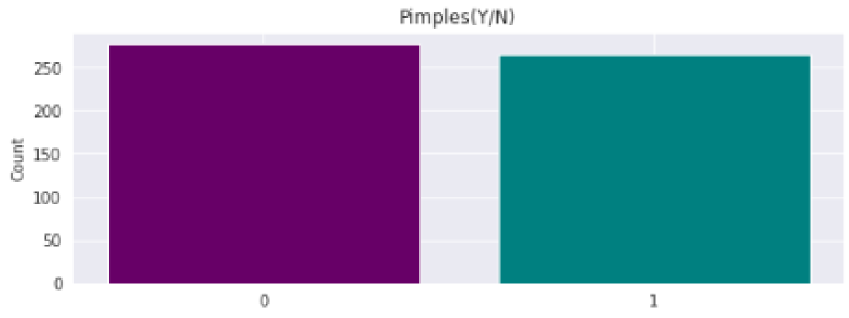
```
hair growth(Y/N):
 0    393
1    148
Name: hair growth(Y/N), dtype: int64
```



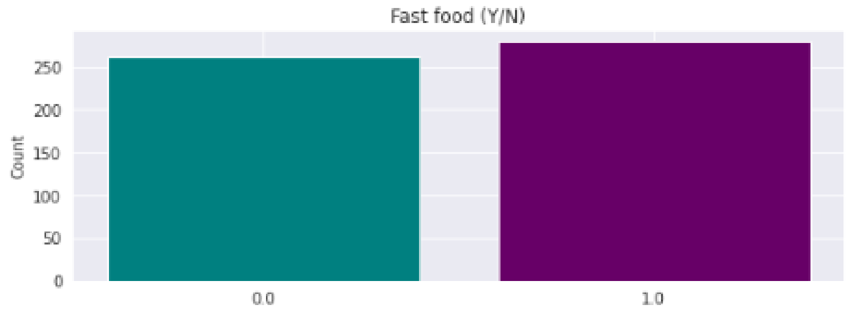
```
Skin darkening (Y/N):
 0    375
1    166
Name: Skin darkening (Y/N), dtype: int64
```



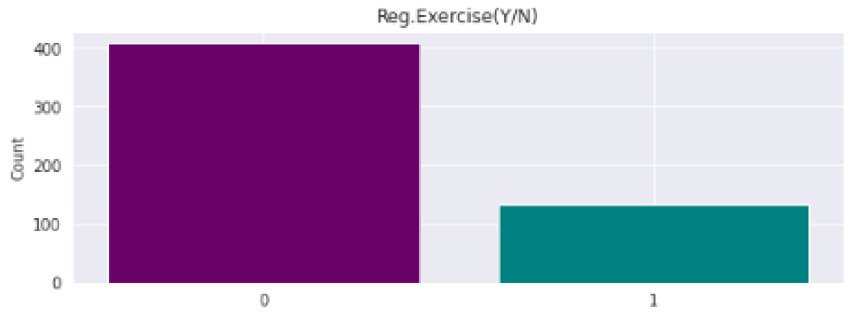
```
Hair loss(Y/N):
 0    296
1    245
Name: Hair loss(Y/N), dtype: int64
```



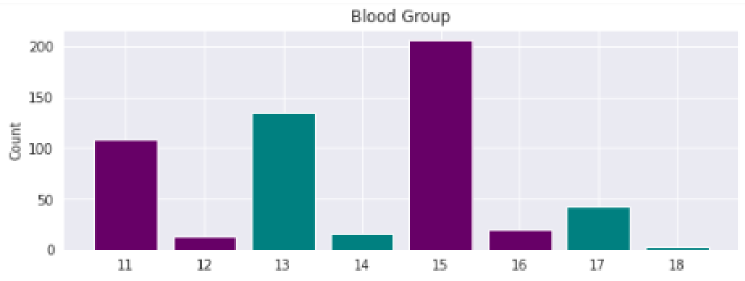
```
Pimples(Y/N):
 0    276
1    265
Name: Pimples(Y/N), dtype: int64
```



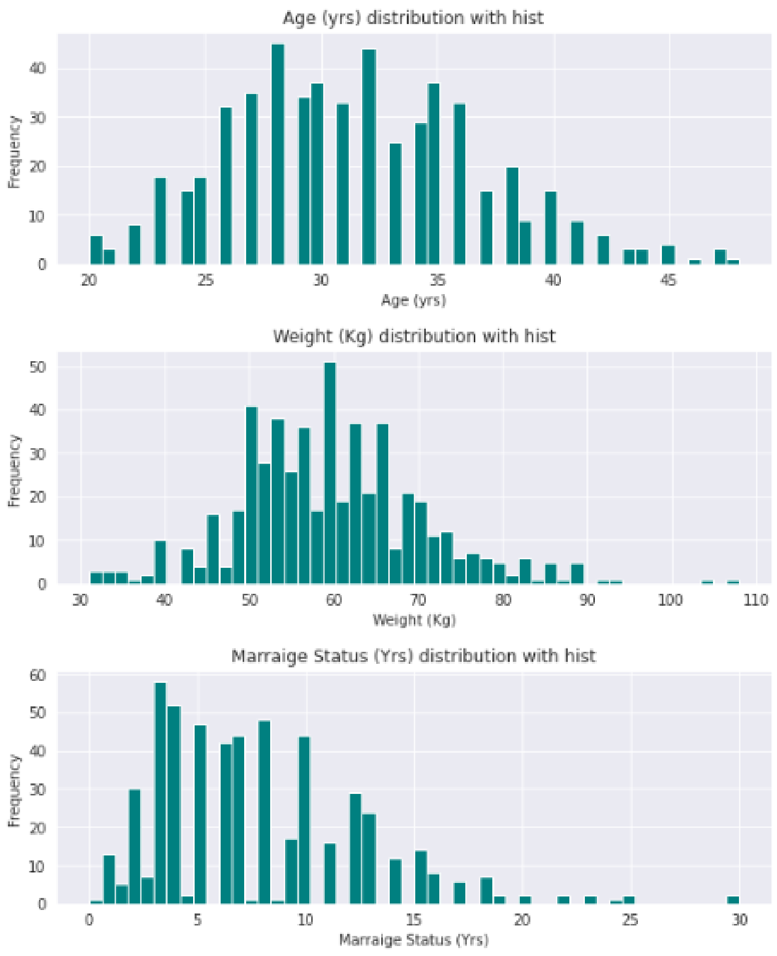
```
Fast food (Y/N):
 1.0    279
0.0    262
Name: Fast food (Y/N), dtype: int64
```



```
Reg.Exercise(Y/N):
 0    407
1    134
Name: Reg.Exercise(Y/N), dtype: int64
```

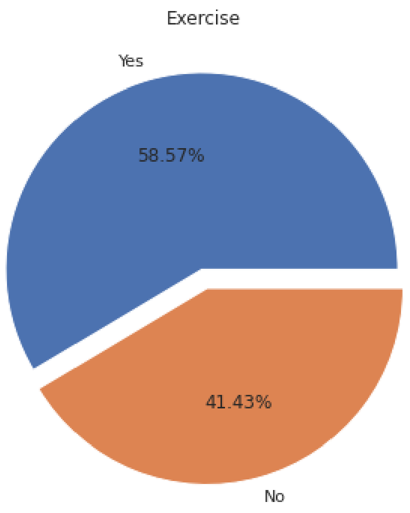


```
Blood Group:
 15    206
13    135
11    108
17     42
16     19
14     16
12     13
18      2
Name: Blood Group, dtype: int64
```



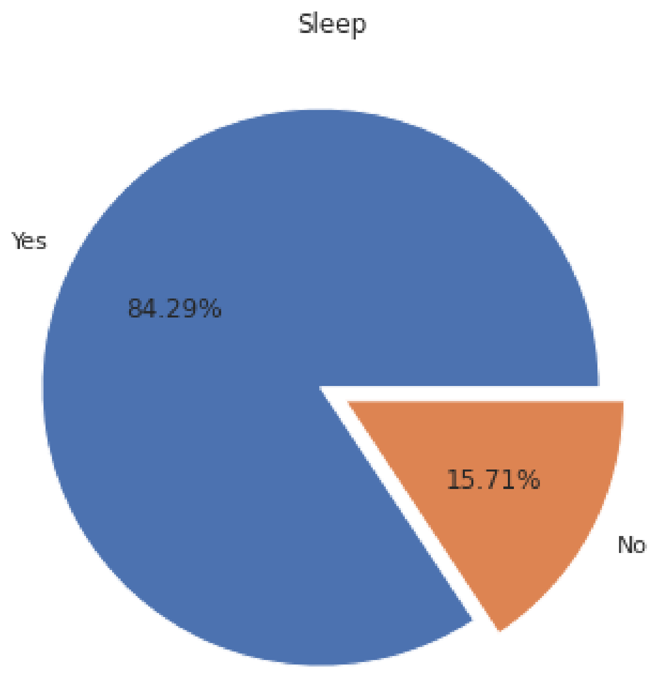
Real Time Data:

Count of people do regular exercise or not.



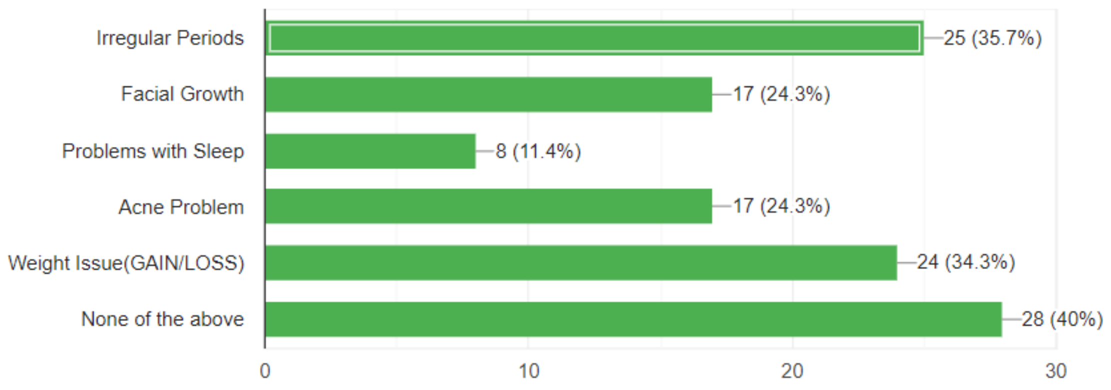
The interpretation of the graph is that 58.57% ie 41 people do regular exercise where 41.43% i.e 29 people does not do regular exercise.

Count of people having good sleep or not.



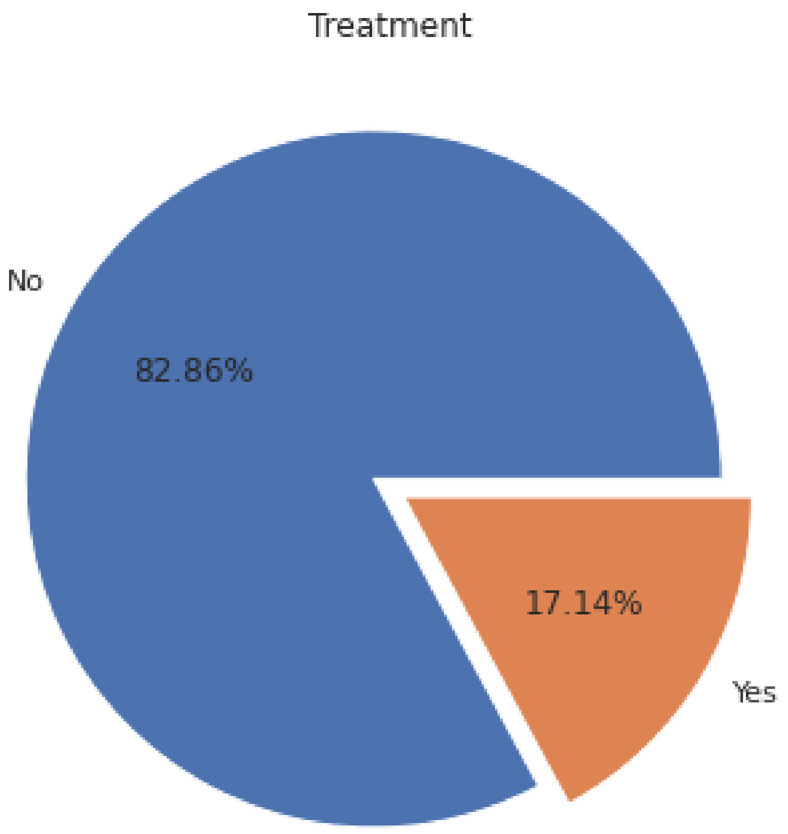
The interpretation of the graph is that 84.29% ie 59 people have good sleep where 15.71% i.e 11 people does not have good sleep.

Representing how people face different challenges in PCOS.



The interpretation of the graph is that **28** people does not have any issue where as **42** people do have problems such as Irregular Periods,Facial Growth,Problems with Sleep,Acne Problem and Weight Issue.

Count of People Undertaking Treatment

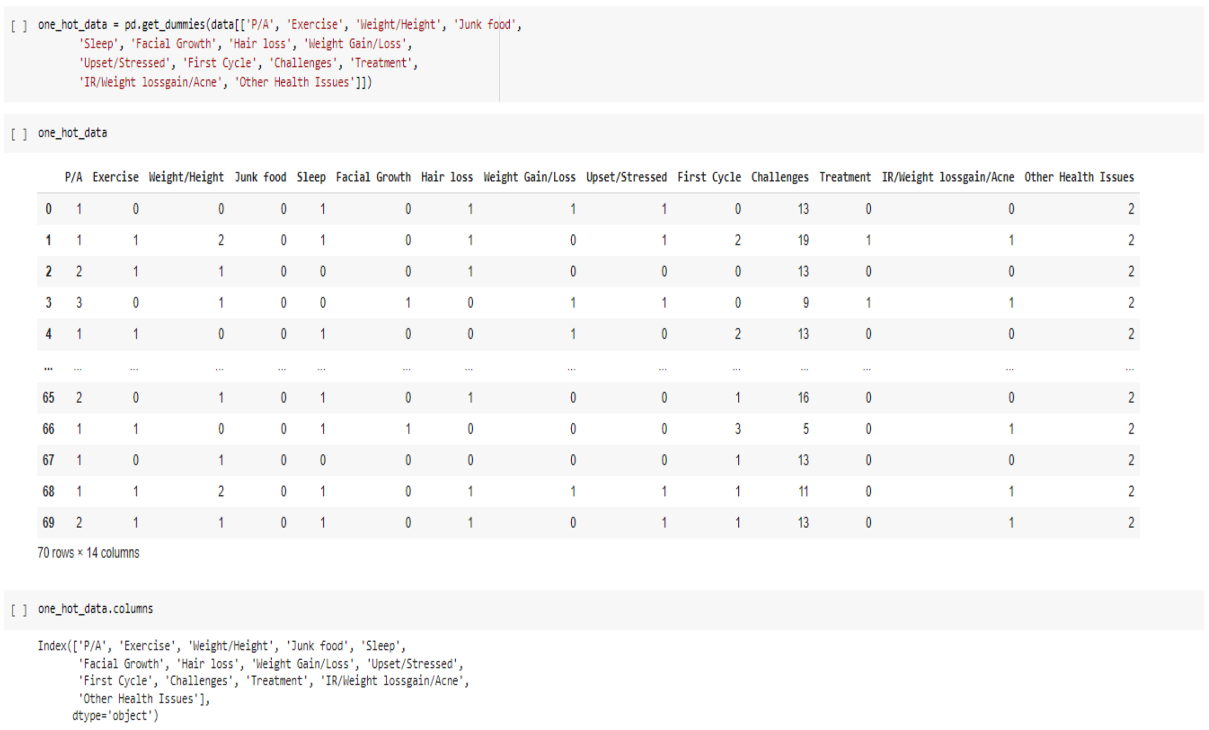


The interpretation of the graph is that 82.86% ie 58 people are not taking any treatment where 17.14% i.e 12 people are taking treatment for PCOS.

One-Hot Encoding:

A one hot encoding is a representation of categorical variables as binary vectors. So I have used the One Hot Encoding to convert theseattributes into the binary form using One Hot Encoding.

pd.get_dummies when applied to a column of categories where we have one category per observation will produce a new column (variable) for each uniquecategorical value. It will place a one in the column corresponding to the categorical value present for that observation. This is equivalent to one hot encoding.



```
[ ] one_hot_data = pd.get_dummies(data[['P/A', 'Exercise', 'Weight/Height', 'Junk food',
           'Sleep', 'Facial Growth', 'Hair loss', 'Weight Gain/Loss',
           'Upset/Stressed', 'First Cycle', 'Challenges', 'Treatment',
           'IR/Weight lossgain/Acne', 'Other Health Issues']])
```

```
[ ] one_hot_data
```

	P/A	Exercise	Weight/Height	Junk food	Sleep	Facial Growth	Hair loss	Weight Gain/Loss	Upset/Stressed	First Cycle	Challenges	Treatment	IR/Weight lossgain/Acne	Other Health Issues
0	1	0	0	0	1	0	1	1	1	0	13	0	0	2
1	1	1	2	0	1	0	1	0	1	2	19	1	1	2
2	2	1	1	0	0	0	1	0	0	0	13	0	0	2
3	3	0	1	0	0	1	0	1	1	0	9	1	1	2
4	1	1	0	0	1	0	0	1	0	2	13	0	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65	2	0	1	0	1	0	1	0	0	1	16	0	0	2
66	1	1	0	0	1	1	0	0	0	3	5	0	1	2
67	1	0	1	0	0	0	0	0	0	1	13	0	0	2
68	1	1	2	0	1	0	1	1	1	1	11	0	1	2
69	2	1	1	0	1	0	1	0	1	1	13	0	1	2

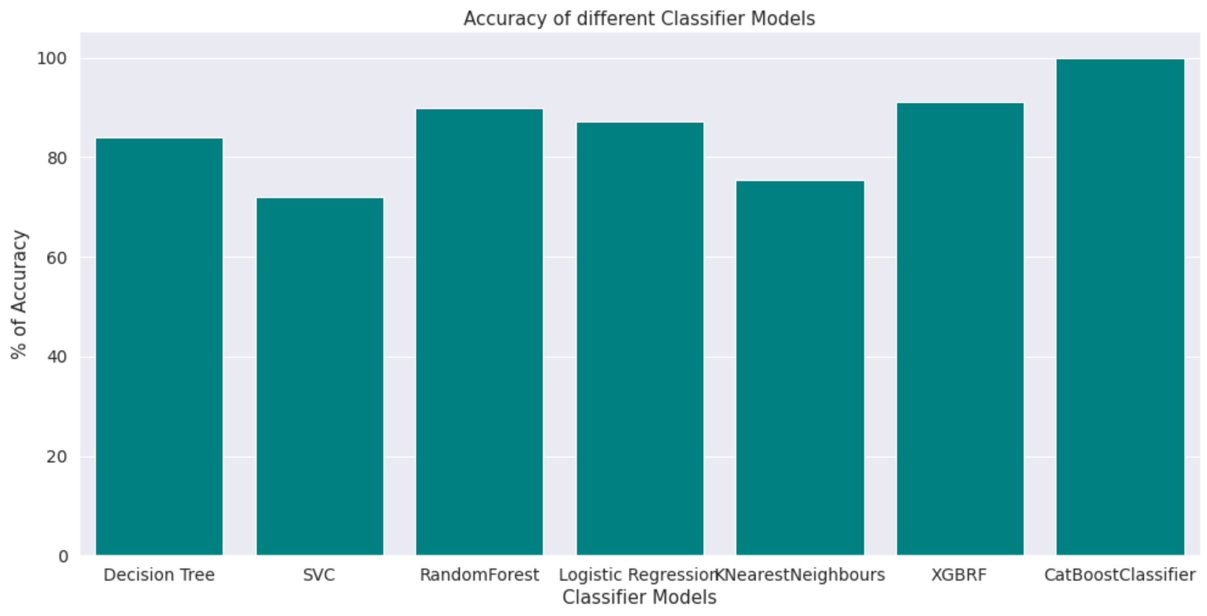
70 rows × 14 columns

```
[ ] one_hot_data.columns

Index(['P/A', 'Exercise', 'Weight/Height', 'Junk food', 'Sleep',
       'Facial Growth', 'Hair loss', 'Weight Gain/Loss', 'Upset/Stressed',
       'First Cycle', 'Challenges', 'Treatment', 'IR/Weight lossgain/Acne',
       'Other Health Issues'],
      dtype='object')
```

ANALYSIS OF THE RESULT:

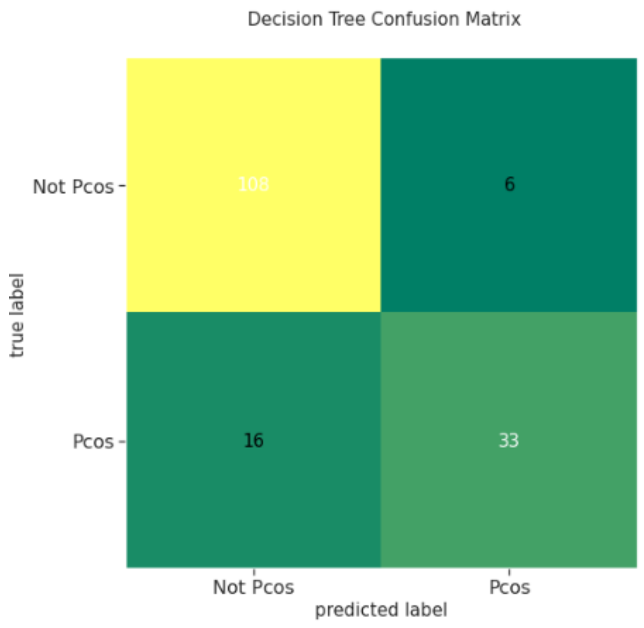
Algorithms:

Exsiting Data:



Confusion Matrix of Different Classifiers:

Decision Tree:



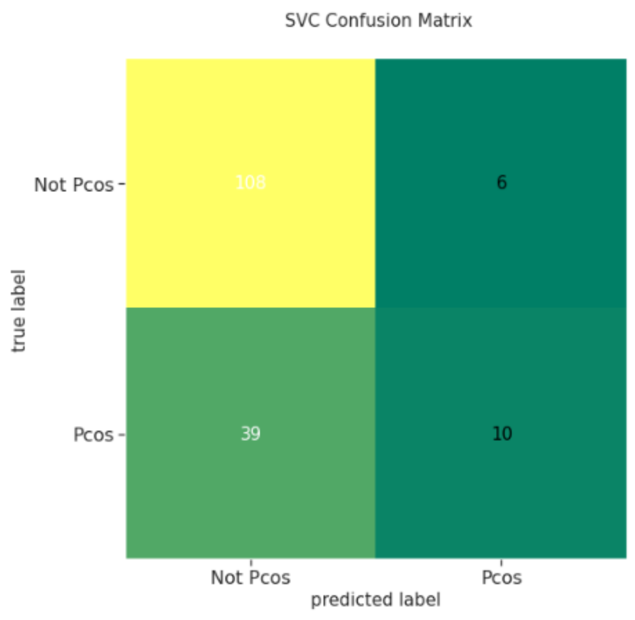
True positive – 33% people have PCOS

True Negative-108% people does not have PCOS

False Positive-6% is we are saying they have PCOS but they do not have

False Negative-16% is we are saying they does not have PCOS but they have

SVC:



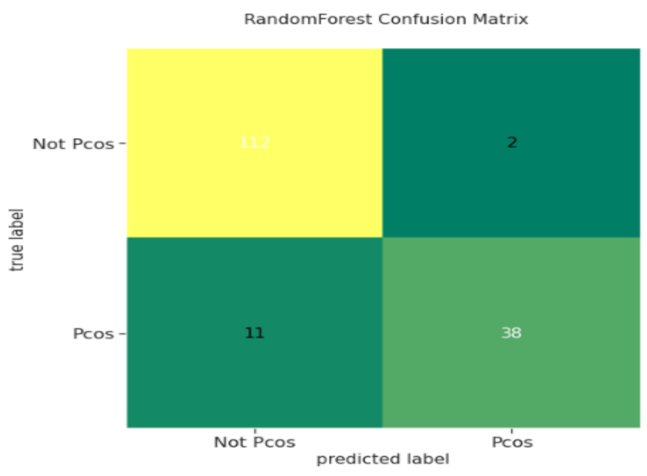
True positive – 10% people have PCOS

True Negative-108% people does not have PCOS

False Positive-6% is we are saying they have PCOS but they do not have

False Negative-39% is we are saying they does not have PCOS but they have

Random Forest:



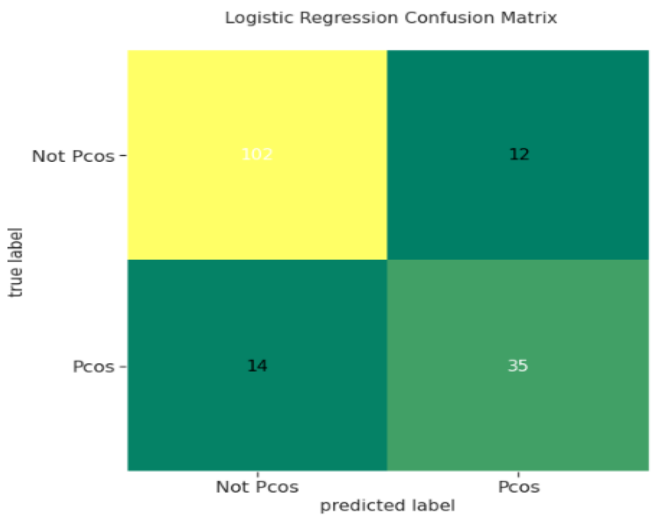
True positive – 38% people have PCOS

True Negative-112% people does not have PCOS

False Positive-2% is we are saying they have PCOS but they do not have

False Negative-11% is we are saying they does not have PCOS but they have

Logistic Regression:

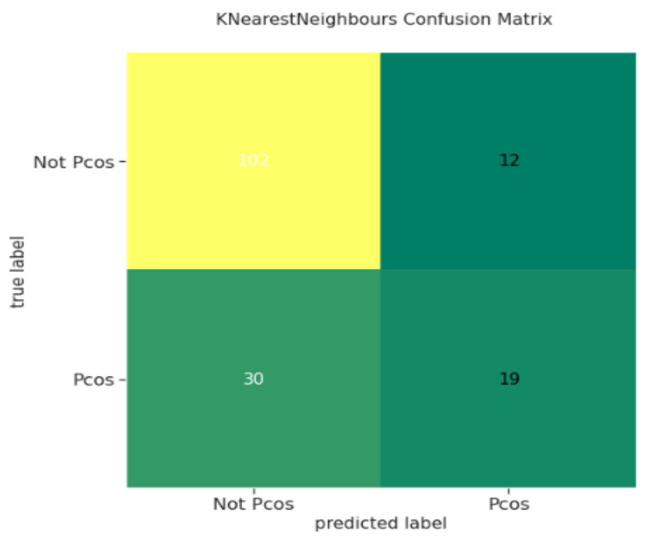


True positive – 35% people have PCOS

True Negative-102% people does not have PCOS

False Positive-12% is we are saying they have PCOS but they do not have
False Negative-14% is we are saying they does not have PCOS but they have

KNN:



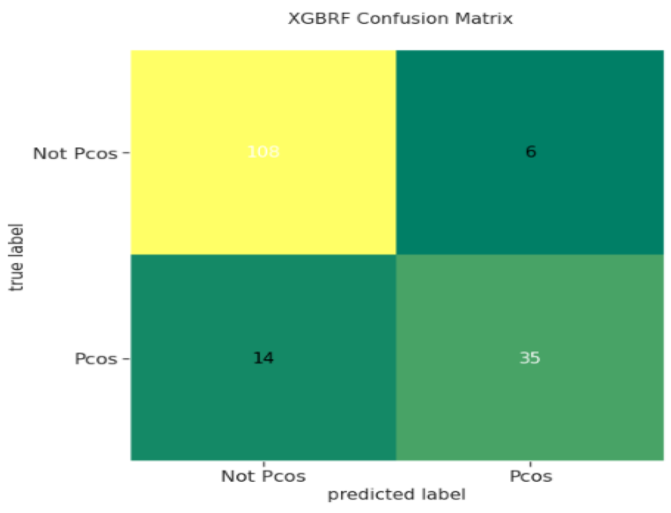
True positive – 19% people have PCOS

True Negative-102% people does not have PCOS

False Positive-12% is we are saying they have PCOS but they do not have

False Negative-30% is we are saying they does not have PCOS but they have

XGBRF:



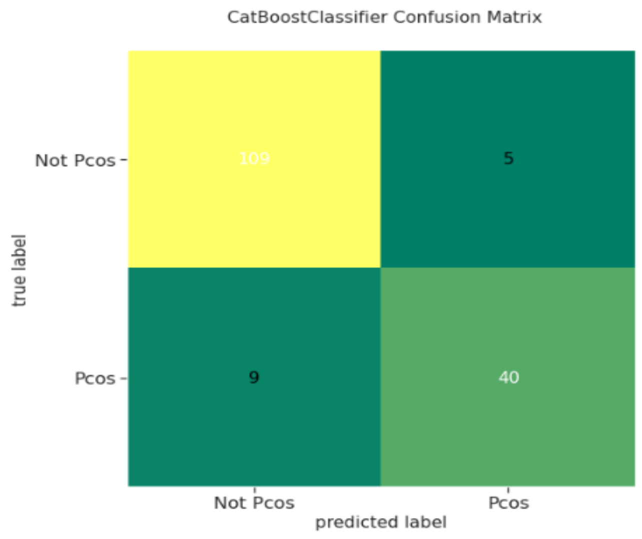
True positive – 35% people have PCOS

True Negative-108% people does not have PCOS

False Positive-6% is we are saying they have PCOS but they do not have

False Negative-14% is we are saying they does not have PCOS but they have

CatBoost:



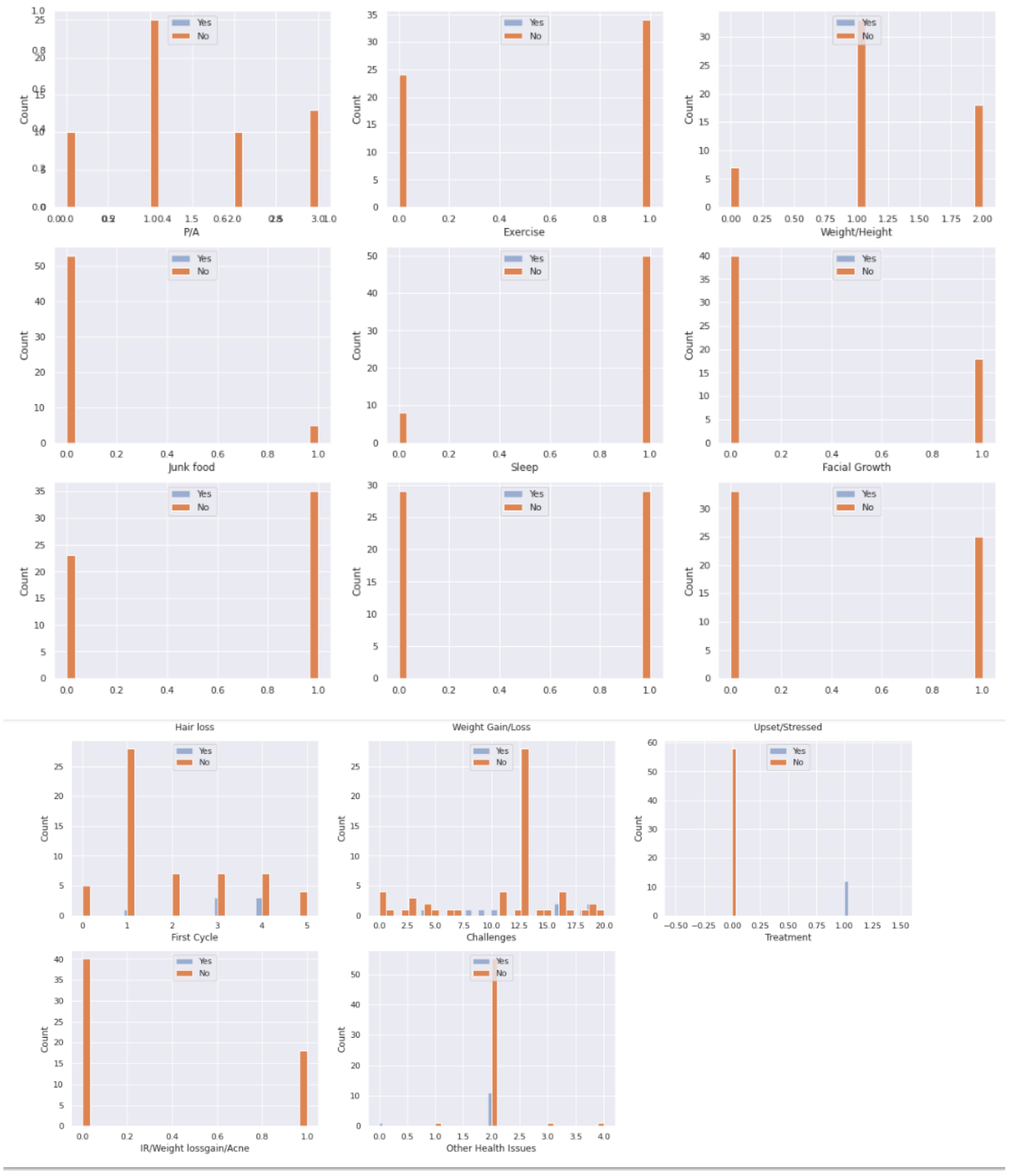
True positive – 40% people have PCOS

True Negative-109% people does not have PCOS

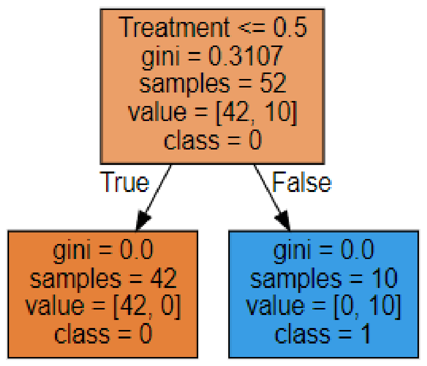
False Positive-5% is we are saying they have PCOS but they do not have

False Negative-9% is we are saying they does not have PCOS but they have

Real Time Data:

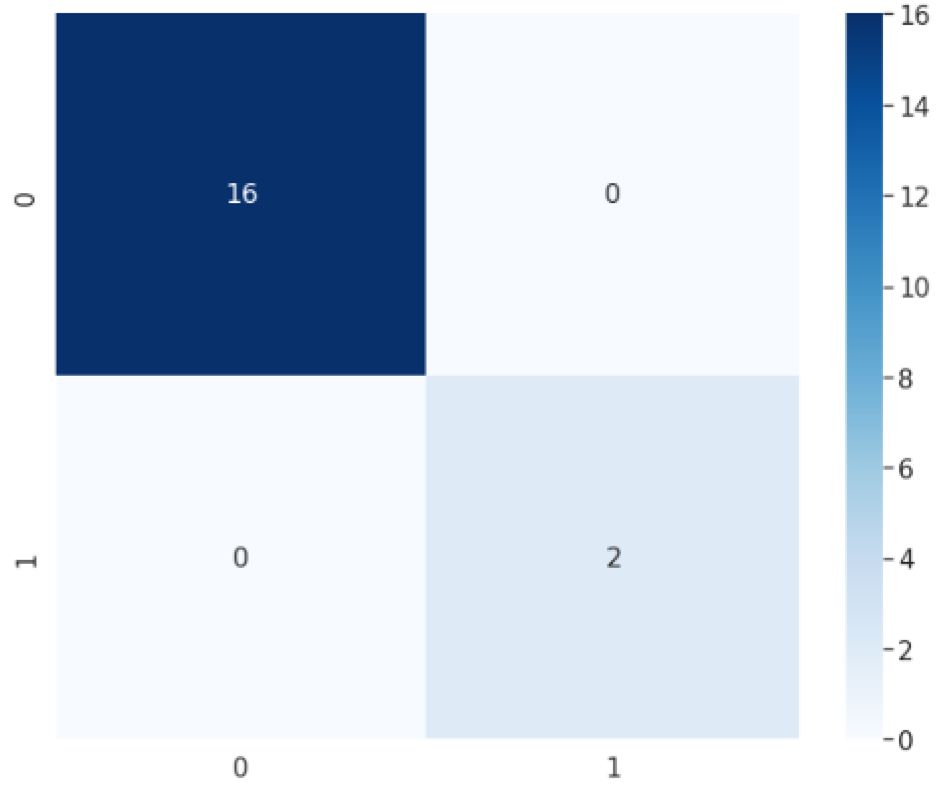


Showing the Tree Diagram of the Variable “Treatment”:



Classification Report & Confusion Matrix:

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      1.00      1.00         2

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18
```



True positive – 2% people have PCOS

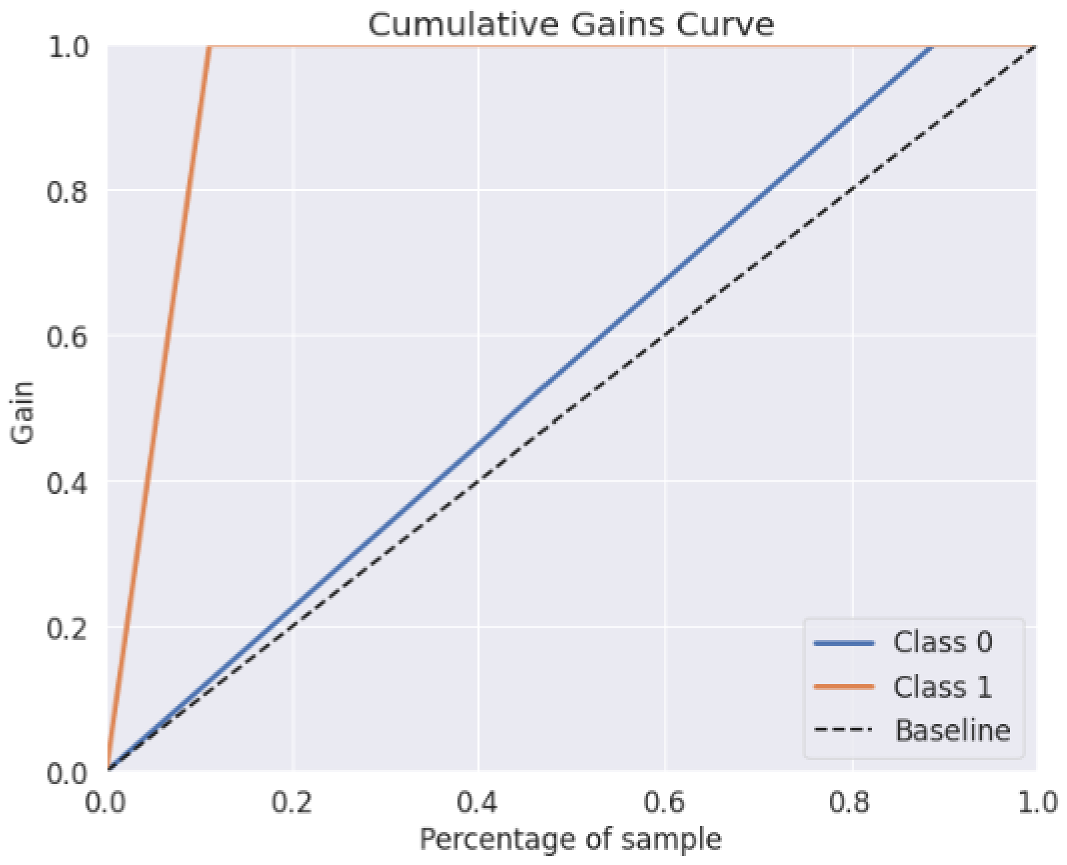
True Negative-16% people does not have PCOS

False Positive-0% is we are saying they have PCOS but they do not have

False Negative-0% is we are saying they does not have PCOS but they have

Gain Chart:

```
<Figure size 504x504 with 0 Axes>
```


Cumulative Gains Curve
Gain
Percentage of sample
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Class 0
Class 1
Baseline

CONCLUSION:

PCOS is one of the most common disorders affecting women of reproductive age.It is extremely important to treat these patients early to help them deal with the emotional stress.

For many women with this syndrome,improving infertility is a primary goal of therapy.It is something that can be treated at early stage by a primary health expert.However,maintaining a healthy and active lifestyle may decrease the chance of developing this condition and its associated problems.

Although not well understood, insulin resistance seems to underlie many of the clinical manifestations of PCOS. Experts can assess and manage many of the presenting complaints and lifestyle issues, such as menstrual disorders, hirsutism, and obesity, which are associated with PCOS.

FUTURE ENHANCEMENT:

- We can recommend the best gynecologist to the person using Content Based Filtering.
- Further we can do the analysis of persons behavior and that will help in speedy recovery.
- We can also do the analysis on the basis of symptoms of the person after sometime to the treatment.

PROGRAM CODE:

Existing Data:

Importing the Libaries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
pip install catboost
from collections import Counter
from mlxtend.plotting import plot_confusion_matrix
from sklearn.model_selection import train_test_split, StratifiedKFold, Gri
dSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
import xgboost
import lightgbm
from catboost import CatBoostClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
```

Loading the Data:

```
from google.colab import drive
drive.mount('/content/drive')
cd drive/MyDrive/Project Dataset/
!ls
inf = pd.read_csv('PCOS_infertility.csv')
woinf = pd.read_excel('PCOS_data_without_infertility.xlsx', sheet_name=
 'Full_new')
```

Details of the Data:

```
data.info(verbose = True, null_counts = False)
                    OR
data.info()
```

Columns Name:

```
inf.columns
woinf.columns
```

Drop,Merge,Rename the Columns:

```
# Merging the two files as per patient file no.
data = pd.merge(woinf,inf, on='Patient File No.', suffixes={'','_wo'},how='
left')

#Dropping the repeated features after merging.
data =data.drop(['Unnamed: 43', 'Sl. No_wo', 'PCOS (Y/N)_wo', '  I   beta-
HCG(mIU/mL)_wo','II    beta-
HCG(mIU/mL)_wo', 'AMH(ng/mL)_wo'], axis=1)

# Changing the title of the properties.
data = data.rename(columns = {"PCOS (Y/N)":"Target"})


# Dropping unnecessary features.
data = data.drop(["Sl. No","Patient File No."],axis = 1)
```

Checking Null Values:

```
data.dropna(axis=0,how='any',inplace=True)
data.isnull().sum()
```

Checking for Duplicated Values:

```
data.duplicated().sum()
```

Total Count:

```
data.describe()
```

Datatypes of Columns:

```
data["AMH(ng/mL)"].head()
data["II    beta-HCG(mIU/mL)"].head()
```

Dealing with Categoral and Missing Values:

```
data["AMH(ng/mL)"] = pd.to_numeric(data["AMH(ng/mL)"], errors='coe
rce')
data["II    beta-HCG(mIU/mL)"] = pd.to_numeric(data["II    beta-
HCG(mIU/mL)"], errors='coerce')

data['Marraige Status (Yrs)'].fillna(data['Marraige Status (Yrs)'].median(),i
nplace=True)
data['II    beta-HCG(mIU/mL)'].fillna(data['II    beta-
HCG(mIU/mL)'].median(),inplace=True)
data['AMH(ng/mL)'].fillna(data['AMH(ng/mL)'].median(),inplace=True)
data['Fast food (Y/N)'].fillna(data['Fast food (Y/N)'].median(),inplace=Tru
e)
```

Data Visualization &EDA:

```
#Categoral Variable
def bar_plot(variable):
   """
    input: variable example : Target
    output: bar plot & value count
    """
   # Get feature
   var = data[variable]
   # Count number of categorical variable(value/sample)
   varValue = var.value_counts()
   # Visualize
   plt.figure(figsize = (9,3))
```

```
    plt.bar(varValue.index,varValue,color=colors)
    plt.xticks(varValue.index,varValue.index.values)
    plt.ylabel("Count")
    plt.title(variable)
    plt.show()
    print("{}: \n {}".format(variable,varValue))

category = ["Target", "Pregnant(Y/N)", "Weight gain(Y/N)", "hair growth(
Y/N)", "Skin darkening (Y/N)", "Hair loss(Y/N)", "Pimples(Y/N)", "Fast f
ood (Y/N)", "Reg.Exercise(Y/N)", "Blood Group"]
for c in category:
    bar_plot(c)


#Numerical Variable
def plot_hist(variable):
    plt.figure(figsize = (9,3))
    plt.hist(data[variable], bins = 50,color=colors[1])
    plt.xlabel(variable)
    plt.ylabel("Frequency")
    plt.title("{} distribution with hist".format(variable))
    plt.show()

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

# Examaning a correlation matrix of all the features.
corrmat = data.corr()
plt.subplots(figsize=(18,18))
sns.heatmap(corrmat,cmap="Set3", square=True);
# How all the features correlate with the PCOS.
corrmat['Target'].sort_values(ascending=False)
# Having a look at features bearing significant correlation.

plt.figure(figsize=(12,12))
k = 12 #number of variables with positive for heatmap
l = 3 #number of variables with negative for heatmap
cols_p = corrmat.nlargest(k,'Target')['Target'].index
cols_n = corrmat.nsmallest(l, 'Target')['Target'].index
```

```
cols = cols_p.append(cols_n)

cm = np.corrcoef(data[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True,cmap="Set3", annot=True, square=True,
 fmt='.2f', annot_kws={'size': 10},
                 yticklabels=cols.values, xticklabels=cols.values)
plt.show()
data.corr()
#Patterns Of Length Of The Menstrual Cycle
# Length of menstrual phase in PCOS vs normal

fig=sns.lmplot(data=data,x="Age (yrs)",y="Cycle length(days)", hue="Tar
get",palette=colors)
plt.show(fig)
#Patterns Of BMI

# Pattern of weight gain (BMI) over years in PCOS and Normal.

fig= sns.lmplot(data =data,x="Age (yrs)",y="BMI", hue="Target", palette
= colors )
plt.show(fig)
'''Patterns Of Irregularity In Mensuration
Apparently in the feature "Cycle(R/I)" value:
4 indicates irregular menstrual cycle
2 indicates a regular menstrual cycle'''
# Cycle IR wrt age
sns.lmplot(data =data,x="Age (yrs)",y="Cycle(R/I)", hue="Target",palette
=colors)
plt.show()

#Number Of Follicles
# Distribution of follicles in both ovaries.
sns.lmplot(data =data,x='Follicle No. (R)',y='Follicle No. (L)', hue="Targe
t",palette=colors)
plt.show()
# Exploring the above observation with the help of Boxplot
color = ["pink", "green"]
features = ["Follicle No. (L)","Follicle No. (R)"]
for i in features:
    sns.swarmplot(x=data["Target"], y=data[i], color="black", alpha=0.5 )
    sns.boxenplot(x=data["Target"], y=data[i], palette=color)
    plt.show()

#Some Miscellaneous EDA
features = ["Age (yrs)","Weight (Kg)", "BMI", "Hb(g/dl)", "Cycle length(
days)","Endometrium (mm)" ]
for i in features:
    sns.swarmplot(x=data["Target"], y=data[i], color="black", alpha=0.5 )
    sns.boxenplot(x=data["Target"], y=data[i], palette=color)
    plt.show()
```

Algorithms:

```
#DATA MODELING
#Train - Test Split
#Assiging the features (X)and target(y).
X= data.drop(labels = ["Target"],axis = 1)
y=data.Target

#Splitting the data into test and training sets.
X_train,X_test, y_train, y_test = train_test_split(X,y, test_size=0.3)

print("X_train",len(X_train))
print("X_test",len(X_test))
print("y_train",len(y_train))
print("y_test",len(y_test))

#Simple Logistic Regression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
acc_log_train = round(logreg.score(X_train, y_train)*100,2)
acc_log_test = round(logreg.score(X_test,y_test)*100,2)
print("Training Accuracy: % {}".format(acc_log_train))
print("Testing Accuracy: % {}".format(acc_log_test))

'''Grid Search
Here, taking 5 Machine Learning classifiers and evaluating the mean accur
acy of each of them by stratified cross validation.
```

```
Decision Tree
SVM
Random Forest
KNN
Logistic Regression'''
random_state = 42
classifier = [DecisionTreeClassifier(random_state = random_state),
             SVC(random_state = random_state),
             RandomForestClassifier(random_state = random_state),
             LogisticRegression(random_state = random_state),
             KNeighborsClassifier()]
# Decision Tree
dt_param_grid = {"min_samples_split" : range(10,500,20),
                "max_depth": range(1,20,2)}
# SVM
svc_param_grid = {"kernel" : ["rbf"],
                 "gamma": [0.001, 0.01, 0.1, 1],
                 "C": [1,10,50,100,200,300,1000]}
# Random Forest
rf_param_grid = {"max_features": ['auto', 'sqrt', 'log2'],
                "n_estimators":[300,500],
                "criterion":["gini"],
                'max_depth' : [4,5,6,7,8,9,10,12],}
# Logistic Regression
logreg_param_grid = {"C":np.logspace(-3,3,7),
                    "penalty": ["l1","l2"]}
# KNN
knn_param_grid = {"n_neighbors": np.linspace(1,19,10, dtype = int).tolist(
),
                 "weights": ["uniform","distance"],
                 "metric":["euclidean","manhattan"]}
classifier_param = [dt_param_grid,
                   svc_param_grid,
                   rf_param_grid,
                   logreg_param_grid,
                   knn_param_grid]


cv_result = []
best_estimators = []
```

```
for i in range(len(classifier)):
    clf = GridSearchCV(classifier[i], param_grid=classifier_param[i], cv =
StratifiedKFold(n_splits = 10),
                        scoring = "accuracy", n_jobs = -1,verbose = 1)
    clf.fit(X_train,y_train)
    cv_result.append(round(clf.best_score_*100,2))
    best_estimators.append(clf.best_estimator_)
    print(cv_result[i])

best_estimators

dt = best_estimators[0]
svm = best_estimators[1]
rf = best_estimators[2]
lr = best_estimators[3]
knn = best_estimators[4]

#XGBRF and CatBoost
# XGBRF Classifier
xgb_clf = xgboost.XGBRFClassifier(max_depth=3, random_state=random
_state)
xgb_clf.fit(X_train,y_train)
acc_xgb_clf_train = round(xgb_clf.score(X_train, y_train)*100,2)
acc_xgb_clf_test = round(xgb_clf.score(X_test,y_test)*100,2)
cv_result.append(acc_xgb_clf_train)
print("Training Accuracy: % {}".format(acc_xgb_clf_train))
print("Testing Accuracy: % {}".format(acc_xgb_clf_test))
# CatBoost Classifier
cat_clf = CatBoostClassifier()
cat_clf.fit(X_train,y_train)
acc_cat_clf_train = round(cat_clf.score(X_train, y_train)*100,2)
acc_cat_clf_test = round(cat_clf.score(X_test,y_test)*100,2)
cv_result.append(acc_cat_clf_train)
print("Training Accuracy: % {}".format(acc_cat_clf_train))
print("Testing Accuracy: % {}".format(acc_cat_clf_test))
```

Result:

```
#Showing the model, accuracy and confusion matrix.
model_list = ['Decision Tree','SVC','RandomForest','Logistic Regression','
KNearestNeighbours','XGBRF','CatBoostClassifier']

fg = sns.factorplot(x = model_list, y = cv_result, size= 7, aspect=2 ,color=
colors[1], saturation=5,kind='bar', data=data)
plt.title('Accuracy of different Classifier Models')
plt.xlabel('Classifier Models')
plt.ylabel('% of Accuracy')
plt.show()

# Plotly Bar Chart:
import plotly.graph_objects as go
trace1 = go.Bar(
                x = model_list,
                y = cv_result,
                marker = dict(color = 'rgb(32, 155, 110)',
                              line=dict(color='rgb(0,0,0)',width=1.5)))
layout = go.Layout(title = 'Accuracy of different Classifier Models' , xaxis
= dict(title = 'Classifier Models'), yaxis = dict(title = '% of Accuracy'))
fig = go.Figure(data = [trace1], layout = layout)
fig.show()

model = [dt,svm,rf,lr,knn,xgb_clf,cat_clf]
predictions = []

for i in model:
    predictions.append(i.predict(X_test))
for j in range(7):
    cm = confusion_matrix(y_test, predictions[j])
    plot_confusion_matrix(cm, figsize=(12,8), hide_ticks=True, cmap=plt.c
m.summer)
    plt.title(" {} Confusion Matrix".format(model_list[j]))
    plt.xticks(range(2), ["Not Pcos","Pcos"], fontsize=16)
    plt.yticks(range(2), ["Not Pcos","Pcos"], fontsize=16)
    plt.show()
```

Real Time Data:

Importing the Libaries:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import graphviz
from sklearn import svm
from sklearn.tree import export_graphviz
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import recall_score,precision_score
from sklearn.metrics import classification_report, confusion_matrix,accura
cy_score,roc_curve, auc
%matplotlib inline
```

Loading the Data:

```
from google.colab import drive
drive.mount('/content/drive')
cd drive/MyDrive/Project Dataset/
!ls
data = pd.read_csv('Project Survey.csv')
```

Details of the Data:

```
data.info()
```

No.of Rows & Columns In Dataset:

```
data.shape
```

Columns Name:

```
data.columns
```

Datatype of Columns:

```
data.dtypes
```

Checking Null Values:

```
data.isnull().sum()
```

Finding the Duplicate Values:

```
data.duplicated()
```

Total Count:

```
data.describe()
```

Renaming the Columns:

```
data.rename(columns = {'Q.1 What is Your age?':'Age',
                    'Q.2 Have you ever been Pregnant or had Abortions?':'P/A',
                    'Q.3 Do you Exercise Regularly?':'Exercise',
                    'Q.4 How much is your weight and height?':'Weight/Height',
                    'Q.5 Do you eat junk-food on daily basis?':'Junk food',
                    'Q.6 Do you Sleep well?':'Sleep',
                    'Q.7 Do you have major facial growth?':'Facial Growth',
                    'Q.8 Are you facing with Hair loss?':'Hair loss',
                    'Q.9 Are you facing problem with weight gain or weight loss?':'W
eight Gain/Loss',
                    'Q.10 Do you get too upset or depressed/stressed during your mon
thly cycle?':'Upset/Stressed',
                    'Q.11 When did you get your first Menstrual Cycle?':'First Cycle',
                    'Q.12 What are/were the challenges did you face with PCOS?':'Ch
allenges',
                    'Q.13 Are you taking any treatment for PCOS? ':'Treatment',
                    'Q.14 Are you facing problem with irregular periods and weight l
oss/gain and acne?':'IR/Weight lossgain/Acne',
                    'Q.15 Are you diagnosed with any other health issues when you w
ere/are diagnosed with PCOS?':'Other Health Issues'},inplace = True)

data.columns
```

Count of Values From Each Column:

```
data['Age'].value_counts()
data['P/A'].value_counts()
data['Exercise'].value_counts()
data['Weight/Height'].value_counts()
data['Junk food'].value_counts()
data['Sleep'].value_counts()
data['Facial Growth'].value_counts()
data['Hair loss'].value_counts()
data['Weight Gain/Loss'].value_counts()
data['Upset/Stressed'].value_counts()
data['First Cycle'].value_counts()
data['Challenges'].value_counts()
data['Treatment'].value_counts()
data['IR/Weight lossgain/Acne'].value_counts()
data['Other Health Issues'].value_counts()
```

DataVisualization:

```
sns.set_theme(style='darkgrid')
```

Analysis:

Age:

```
data['Age'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        figsize=(20,15))
plt.axis('off')
plt.title("Age")
plt.show()
               OR

data['Age'].value_counts().plot(kind='barh', figsize=(10, 8))
plt.xlabel("Scale", labelpad=14)
plt.ylabel("Group of Different Age", labelpad=14)
plt.xlim(0, 10)
plt.title("Age", y=1.02);
```

P/A:

```
data['P/A'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        figsize=(8, 6))
plt.axis('off')
plt.title("P/A")
plt.show()
```

Exercise:

```
data['Exercise'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Exercise")
plt.show()
```

Weight/Height:

```
data['Weight/Height'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        figsize=(8, 6))
plt.axis('off')
plt.title("Weight/Height")
plt.show()
```

Junk food:

```
data['Junk food'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Junk food")
plt.show()
```

Sleep:

```
data['Sleep'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Sleep")
plt.show()
```

Facial Growth:

```
data['Facial Growth'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Facial Growth")
plt.show()
```

Hair loss:

```
data['Hair loss'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Hair loss")
plt.show()
```

Weight Gain/Loss:

```
data['Weight Gain/Loss'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Weight Gain/Loss")
plt.show()
```

Upset/Stressed:

```
data['Upset/Stressed'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Upset/Stressed")
plt.show()
```

First Cycle:

```
data['First Cycle'].value_counts().plot(kind ='pie',autopct='%1.2f%%',
        figsize=(8, 6))
plt.axis('off')
plt.title("First Cycle")
plt.show()
```

Challenges:

```
data['Challenges'].value_counts().plot(kind='pie',autopct='%1.2f%%',
        figsize=(25,20))
plt.axis('off')
plt.title("Challenges")
plt.show()
         OR
data[ 'Challenges'].value_counts().plot(kind='barh', figsize=(10, 8))
plt.xlabel("Scale", labelpad=14)
plt.ylabel("Reasons", labelpad=14)
plt.xlim(0, 30)
plt.title("Challenges", y=1.02);
```

Treatment:

```
data['Treatment'].value_counts().plot(kind='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("Treatment")
plt.show()
```

IR/Weight lossgain/Acne:

```
data['IR/Weight lossgain/Acne'].value_counts().plot(kind='pie',autopct='%1.2f%%',
        explode=[.05,.05],figsize=(8, 6))
plt.axis('off')
plt.title("IR/Weight lossgain/Acne")
plt.show()
```

Other Health Issues:

```
data['Other Health Issues'].value_counts().plot(kind='pie',autopct='%1.2f%%',
      figsize=(20,15))
plt.axis('off')
plt.title("Other Health Issues")
plt.show()

plt.hist([data['Exercise'],data['Junk food'],data['IR/Weight lossgain/Acne'],
data['Facial Growth'],data['Sleep'],data['Treatment']], color=['blue', 'orange
','red','green','purple','yellow'])
```

```
plt.title('To know how patients evaluated the factors towards PCOS')
plt.ylabel('Different Age')
plt.xlabel('scale')
plt.legend(['Exercise','Junk food','IR/Weight lossgain/Acne','Facial Growth
','Sleep','Treatment'],loc=9)
plt.ylim(15, 150)
plt.show()
plt.rcParams["figure.figsize"] = (10, 8)
```

<u>To Identify patients perception towards PCOS:</u>

```
diff = pd.crosstab(data['Treatment'],data['Challenges'])
diff

plt.figure(figsize=(15, 9))
plt.xlabel('Q.13 Are you taking any treatment for PCOS? [Traetment]',)
plt.ylabel('Q.14 Are you facing problem with irregular periods and weight
loss/gain and acne? [IR/Weight lossgain/Acne]')
plt.title('Patients perception towards PCOS')
sns.heatmap(diff, annot = True, cmap = 'flare', fmt = 'd', cbar_kws={"orien
tation": "horizontal"});
```

<u>Converting the Data to Integer:</u>

Function for Converting Binary Data:

```
def trans_con(x):
   if x == 'Yes':
      return 1
   if x == 'No':
      return 0
data['Exercise'] = data['Exercise'].apply(trans_con)
data['Junk food'] = data['Junk food'].apply(trans_con)
data['Sleep'] = data['Sleep'].apply(trans_con)
data['Facial Growth'] = data['Facial Growth'].apply(trans_con)
data['Hair loss'] = data['Hair loss'].apply(trans_con)
data['Weight Gain/Loss'] = data['Weight Gain/Loss'].apply(trans_con)
data['Upset/Stressed'] = data['Upset/Stressed'].apply(trans_con)
data['Treatment'] = data['Treatment'].apply(trans_con)
```

```
data['IR/Weight lossgain/Acne'] = data['IR/Weight lossgain/Acne'].apply(t
rans_con)
data
```

Function for Converting Categorical Data:

```
from sklearn.preprocessing import LabelEncoder
number = LabelEncoder()
data['P/A'] = number.fit_transform(data['P/A'].astype('str'))
data['Weight/Height'] = number.fit_transform(data['Weight/Height'].astype
('str'))
data['First Cycle'] = number.fit_transform(data['First Cycle'].astype('str'))
data['Other Health Issues'] = number.fit_transform(data['Other Health Issu
es'].astype('str'))
data['Challenges'] = number.fit_transform(data['Challenges'].astype('str'))
data
```

Correlation between Variables:

```
cols=['P/A', 'Exercise', 'Weight/Height', 'Junk food',
      'Sleep', 'Facial Growth', 'Hair loss', 'Weight Gain/Loss',
      'Upset/Stressed', 'First Cycle', 'Challenges', 'Treatment',
      'IR/Weight lossgain/Acne', 'Other Health Issues']

g = sns.heatmap(data[cols].corr(),annot=True,cmap='RdYlGn',linewidths=
0.2)
g.set_xticklabels(g.get_ymajorticklabels(), fontsize = 20)
g.set_yticklabels(g.get_ymajorticklabels(), fontsize = 18)
fig=plt.gcf()
fig.set_size_inches(15,10)
plt.show()
data.corr()
```

Conversion Using One Hot Encoding:

```
one_hot_data = pd.get_dummies(data[['P/A', 'Exercise', 'Weight/Height', 'J
unk food',
      'Sleep', 'Facial Growth', 'Hair loss', 'Weight Gain/Loss',
      'Upset/Stressed', 'First Cycle', 'Challenges', 'Treatment',
      'IR/Weight lossgain/Acne', 'Other Health Issues']])
```

```
one_hot_data
one_hot_data.columns
```

Algorithms:

```
fig, axes = plt.subplots(6, 3, figsize=(20, 30))

c = 1
for idx, (colName, ax)  in enumerate(list(zip(data.columns, axes.flatten()))
):
    features = ['P/A', 'Exercise', 'Weight/Height', 'Junk food', 'Sleep',
       'Facial Growth', 'Hair loss', 'Weight Gain/Loss', 'Upset/Stressed',
       'First Cycle', 'Challenges', 'Treatment', 'IR/Weight lossgain/Acne',
       'Other Health Issues',]
    if colName in features:

        pos = data[data['Treatment'] == 1][colName]
        neg = data[data['Treatment'] == 0][colName]

        ax = fig.add_subplot(6, 3, c)
        ax.set_xlabel(colName,fontsize=12)
        ax.set_ylabel('Count')
        ax.set_title="{} PCOS/Not PCOS".format(colName)


        pos.hist(alpha = 0.5, bins=30, label='Yes')

        ax = fig.add_subplot(6, 3, c)
        neg.hist(alpha = 1, bins=30, label='No')
        ax.legend(loc=9)
        c += 1
    else:
        [ax.set_visible(False) for ax in axes.flatten()[idx+1:]]

train, test = train_test_split(one_hot_data, test_size =0.25,random_state=1
0)
train.shape
test.shape
c1 = DecisionTreeClassifier(min_samples_split=10)
features = ['P/A', 'Exercise', 'Weight/Height', 'Junk food', 'Sleep',
      'Facial Growth', 'Hair loss', 'Weight Gain/Loss', 'Upset/Stressed',
      'First Cycle', 'Challenges', 'Treatment', 'IR/Weight lossgain/Acne',
      'Other Health Issues']

X_train = train[features]
y_train = train['Treatment']

X_test = test[features]
y_test = test['Treatment']

tree1 = c1.fit(X_train, y_train)

y_pred_t1 = c1.predict(X_train)

y_pred1 = c1.predict(X_test)

dot_data = tree.export_graphviz(tree1,feature_names = X_train.columns,cl
ass_names=["0","1"], filled=True, precision=4)
graph = graphviz.Source(dot_data, format="png")
graph

sat_train_score_1 = accuracy_score(y_train, y_pred_t1)*100

sat_test_score_1 = accuracy_score(y_test, y_pred1)*100
sat_test_score_1

print(classification_report(y_test, y_pred1))

cm1 = confusion_matrix(y_test, y_pred1)

df_cm1 = pd.DataFrame(cm1, range(2), range(2))
sns.set(font_scale=1.4)
sns.heatmap(df_cm1, annot=True, cmap='Blues', annot_kws={"size": 12},
 fmt='g') # font size
plt.show()

!pip install scikit-plot
```

```
import scikitplot as skplt
plt.figure(figsize=(7,7))
skplt.metrics.plot_cumulative_gain(y_test, model1_test)
plt.show()

rf1 = RandomForestClassifier(max_depth=4, n_estimators = 20)

rf1.fit(X_train,y_train)

rf1_model_pred = rf1.predict(X_test)

rf1_model_pred

sat_test_score_rf = accuracy_score(y_test, rf1_model_pred)*100
sat_test_score_rf

rd = print(classification_report(y_test, rf1_model_pred))
rf_cm = confusion_matrix(y_test, rf1_model_pred)
df_cm3 = pd.DataFrame(rf_cm, range(2), range(2))
sns.set(font_scale=1.4)
sns.heatmap(df_cm3, annot=True, cmap='Blues', annot_kws={"size": 16},
 fmt='g')
plt.show()
```

Plotting Accuracy Graph of All Algorithms:

```
modelacc = pd.DataFrame({"Decision Tree Accuracy ": [sat_test_score_1],
                         "Random Forest Accuracy": [sat_test_score_rf]})
modelacc.plot.bar(align='edge',figsize=(15,8),ec="black");
plt.xlabel('Scores', fontsize=20)
plt.ylabel('Models',fontsize=20)
plt.legend(fontsize=10.5);

modelacc = pd.DataFrame({"SVC Accuracy ": [sat_test_score_1],
                         "Logistic Regression Accuracy": [sat_test_score_rf]})
modelacc.plot.bar(align='edge',figsize=(15,8),ec="black");
plt.xlabel('Scores', fontsize=20)
plt.ylabel('Models',fontsize=20)
plt.legend(fontsize=10.5);
```

```
modelacc = pd.DataFrame({"KNearestNeighbours Accuracy ": [sat_test_score_1],
                         "XGBRF Accuracy": [sat_test_score_rf]})
modelacc.plot.bar(align='edge',figsize=(15,8),ec="black");
plt.xlabel('Scores', fontsize=20)
plt.ylabel('Models',fontsize=20)
plt.legend(fontsize=10.5);
```

```
modelacc = pd.DataFrame({"CatBoost Classifier Accuracy ": [sat_test_score_1],
                         "XGBRF Accuracy": [sat_test_score_rf]})
modelacc.plot.bar(align='edge',figsize=(15,8),ec="black");
plt.xlabel('Scores', fontsize=20)
plt.ylabel('Models',fontsize=20)
plt.legend(fontsize=10.5);
```