Detection of Polycystic Ovary Syndrome

Milestone: Final Project Report

Group 2

Student 1: Fathima Salim

Student 2: Pratiksha Pradhan

Telephone No. Student 1: 8573471625

Telephone No. Student 2: 8573761450

Email ID Student 1: salim.f@northeastern.edu

Email ID Student 2: pradhan.pra@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: 

Signature of Student 2: 

Submission Date: 21st April, 2023

Table of Contents

Introduction

Problem Setting

Polycystic Ovary Syndrome (PCOS) is a syndrome that affects female reproductive systems by causing hormonal issues, which leads to irregular or absent menstrual cycles. The ovaries develop follicles, which prevent them from functioning properly and releasing the eggs on time. Some of the symptoms include excessive weight gain, acne, and insulin resistance. In extreme cases, this puts women at risk for type II diabetes and can even lead to infertility.

According to the U.S. Department of Health and Human Services, between 5% and 10% of women of childbearing age have PCOS. Most women realize they are suffering from PCOS when they have trouble conceiving, even though the syndrome might have started much before. Even though PCOS is prevalent and treatable, it usually goes undiagnosed.

Problem Definition

The idea behind our project is to determine the best classification model and predictors for accurately detecting PCOS at an early stage, so that women are aware of this issue and are able to take appropriate corrective measures. We would like to observe which attributes impact the diagnosis of PCOS in females.

Data Sources

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

The data will be taken from the dataset titled Polycystic Ovary Syndrome (PCOS) on Kaggle.

Data Description

The dataset consists of 541 rows, which indicate the details of 541 patients from 10 hospitals across Kerala, India. The number of columns is 44, including S.No., Patient File No., and the response variable PCOS (Y/N). The remaining columns are 41 features that are used to diagnose whether the patient has PCOS or not. We have discrete numeric variables (age), continuous numeric variables (weight), nominal categorical variables (blood group), and binary numeric (weight gain).

- For any Yes/No attributes, Yes=1 and No=0.

- Blood groups are indicated as follows: A+=11, A-=12, B+=13, B-=14, O+=15, O-=16, AB+=17, AB-=18

Since our dataset has only 541 rows and 41 features, we will explore methods to deal with such datasets, including dimension reduction techniques, to reduce the number of features to avoid overfitting.

Data Exploration, Visualization, Processing & Dimension Reduction

Outlier Analysis

On checking the output of the describe() function, we notice that columns such as Weight (kg), Cycle length (days), Marriage Status (yrs), I__beta-HCG(mIU/mL), FSH (mIU/mL), LH (mIU/mL), TSH (mIU/L), AMH(ng/mL), PRL(ng/mL), Vit D3 (ng/mL), PRG(ng/mL), RBS(mg/dl), Follicle No. (L), Follicle No. (R), Endometrium (mm) have outliers. We visualize some of these outliers now.

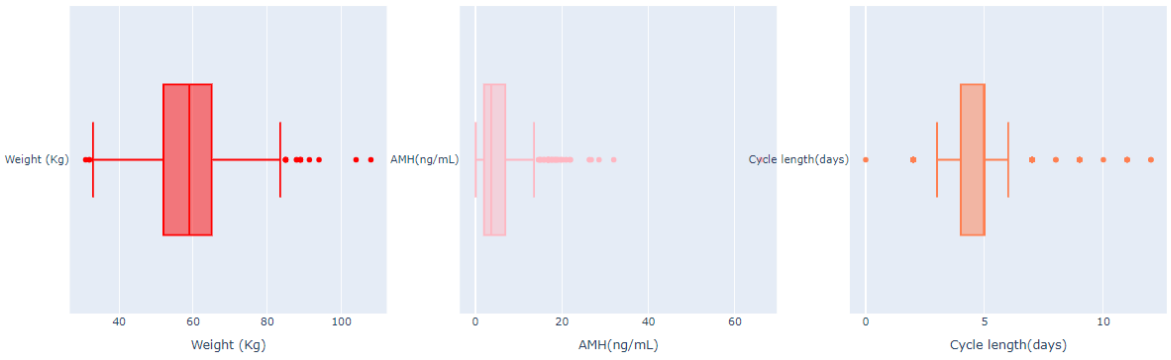

Figure 1: Outlier analysis using boxplots

Handling Missing Values

Using the info() function, we check which columns have how many null values. We make a plot of this using the missingno library.

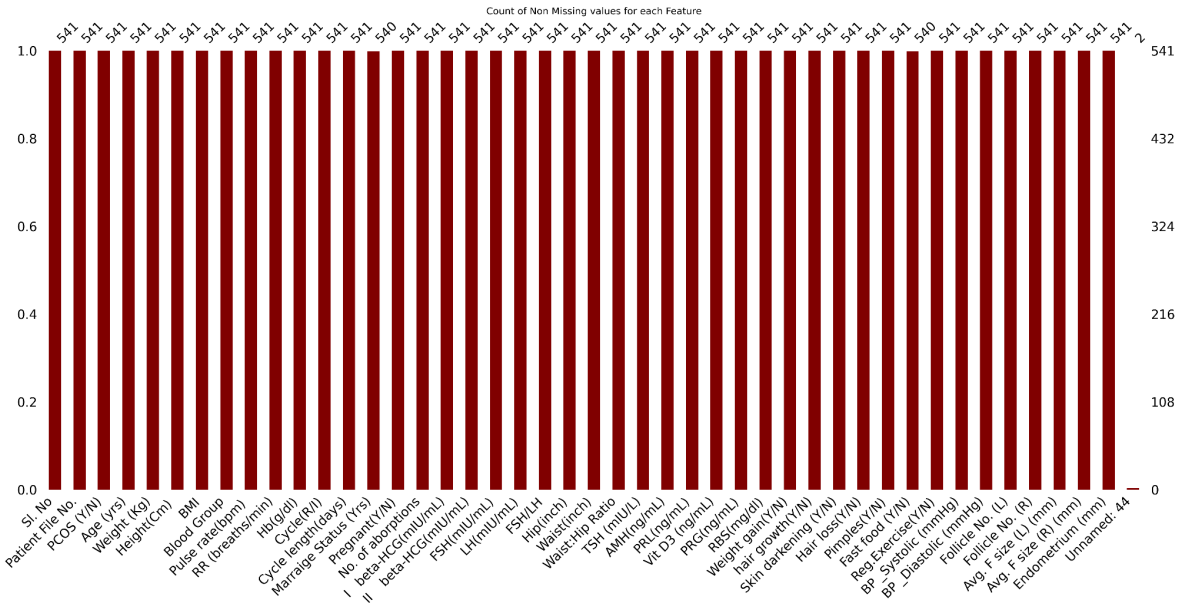

Figure 2: Missing values visualization

Next, we drop some redundant columns: ‘Unnamed: 44’, ‘Patient File No.’, ‘Sl No.’. We convert the ‘AMH(ng/mL)’ and ‘II beta-HCG(mIU/mL)’ columns from object to numeric.

Finally, we impute the missing values in the columns ‘Marriage Status (Yrs)’, ‘Fast food (Y/N)’, ‘AMH(ng/mL)’, and ‘II beta-HCG(mIU/mL)’. There is only one missing value in each of these columns, so we impute it with the median values of the columns respectively as the mean is sensitive to outliers and we have those in our dataset.

Feature selection/Dimension Reduction

We drop redundant columns such as Sl.No in the data cleaning process. We calculate the correlation matrix and plot it as a heatmap. We drop ‘FSH/LH’ since it has a correlation value of 0.95 or greater with ‘FSH’. We also drop ‘Waist (inch)’ and ‘Hip (inch)’ as we already have a column with their ratios. We further reduce the dimensions as we implement each model by considering their feature importance scores.

The final correlation matrix is as below:

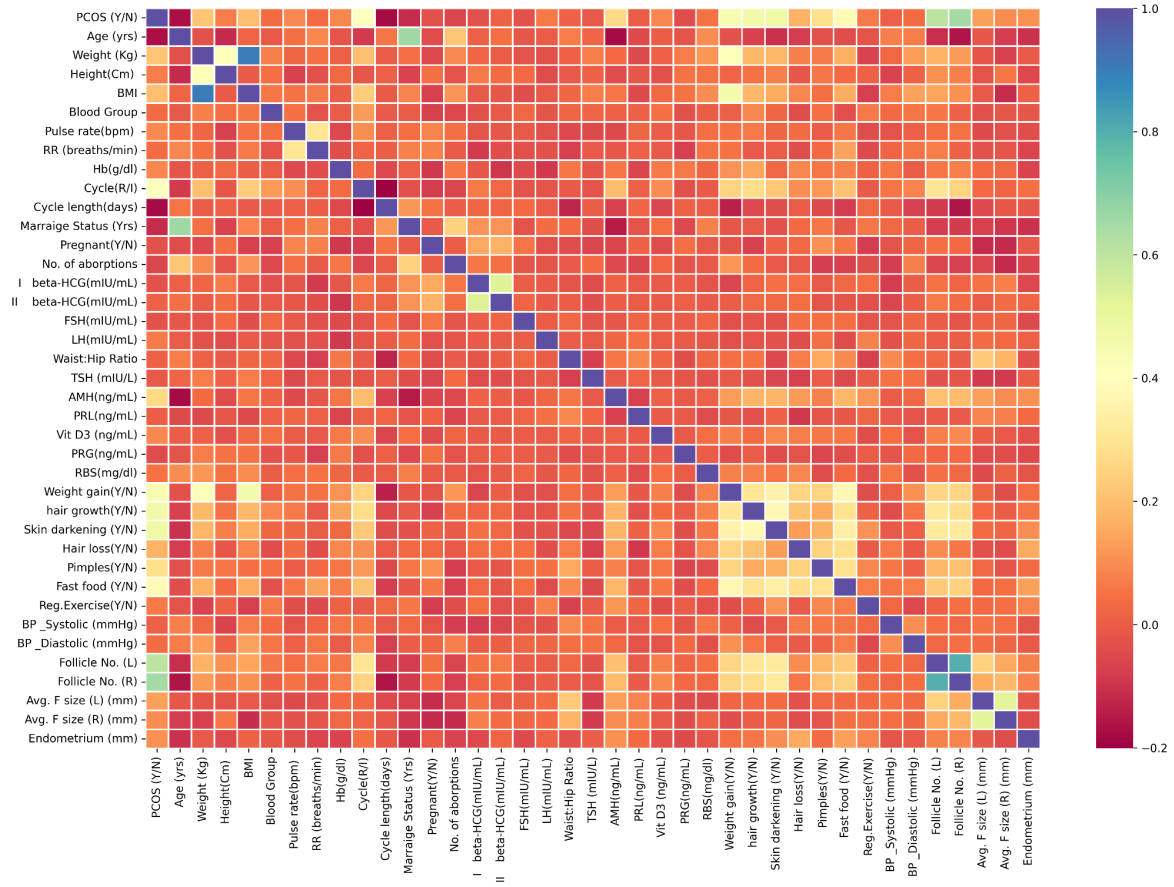

Figure 3: Correlation heatmap

We extracted the correlation value list of every feature with PCOS(Y/N). We can see the Follicle values have the highest correlation with the target variable.

PCOS (Y/N)	1.000000
Follicle No. (R)	0.648327
Follicle No. (L)	0.603346
Skin darkening (Y/N)	0.475733
hair growth(Y/N)	0.464667
Weight gain(Y/N)	0.441047
Cycle(R/I)	0.401644
Fast food (Y/N)	0.376183
Pimples(Y/N)	0.286077
AMH(ng/mL)	0.264141
Weight (Kg)	0.211938
BMI	0.200176
Hair loss(Y/N)	0.172879
Avg. F size (L) (mm)	0.132992
Endometrium (mm)	0.106648
Avg. F size (R) (mm)	0.097690
Pulse rate(bpm)	0.091821
Hb(g/dl)	0.087170
Vit D3 (ng/mL)	0.085494
Height(Cm)	0.068254
Reg.Exercise(Y/N)	0.065337

Figure 4: Correlation values list with respect to target variable in descending order

Data Visualization

Based on the above correlations of each variable with the target variable, we make plots of those variables with correlation values > 0.4. We view the changes in these variables across the ages of the patients. We also create a scatterplot matrix to view the correlation between a few of the variables.

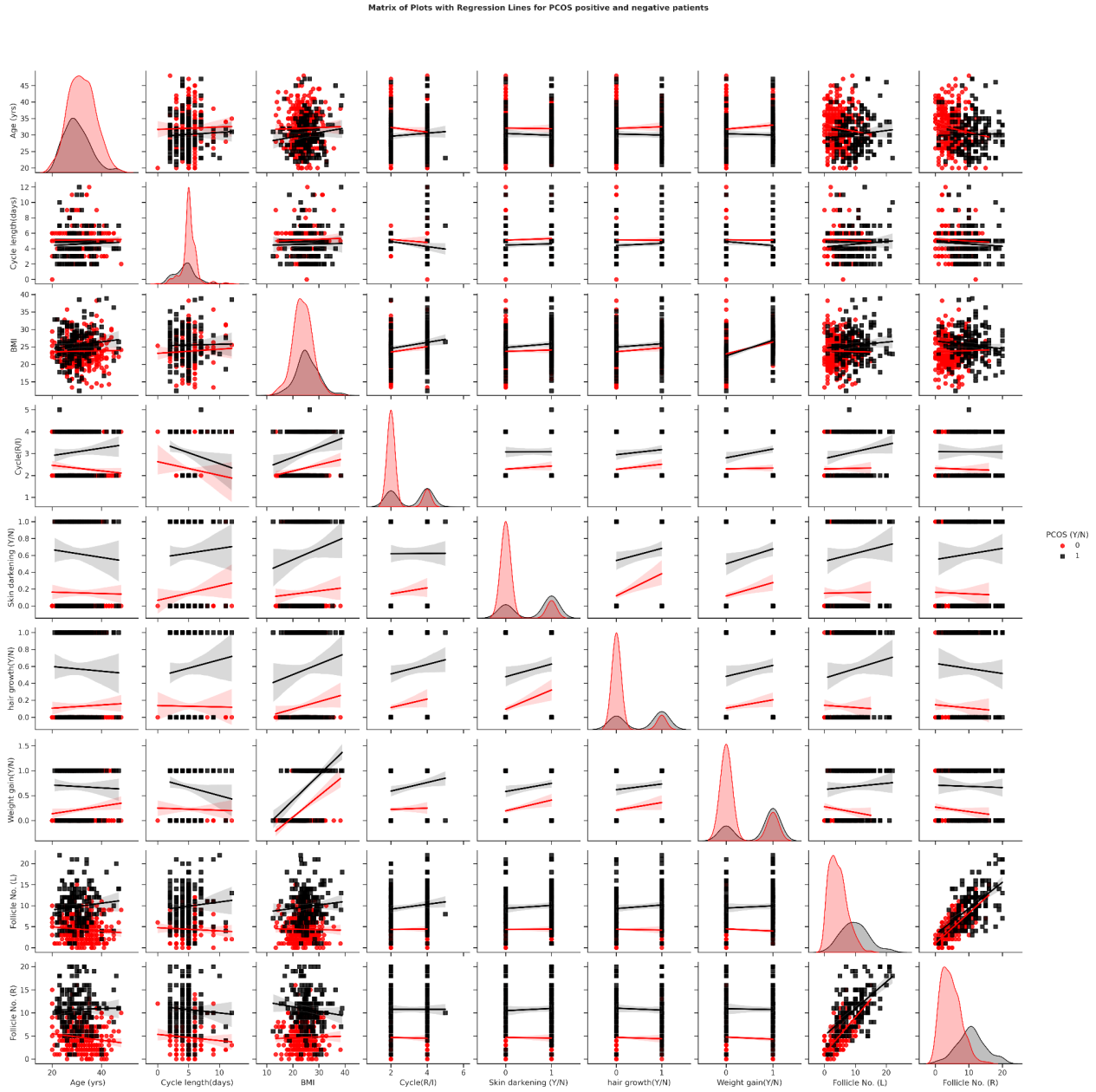

Figure 5: Scatterplot matrix for some of the variables

A few observations:

- We can see that BMI is consistent for people without PCOS as age increases but for people with PCOS BMI increases with age.
- For the number of days of the menstrual cycle it's consistent for people without PCOS while for people with PCOS the number of days is comparatively less but it increases with age.
- For people with PCOS, skin darkening is more prominent at a younger age and reduces with age. For people without PCOS, skin darkening is not very prominent and there is not much difference as they age.
- As for hair growth, there is more hair growth among people with PCOS, especially at a younger age, and this decreases as the women age. For people without PCOS, hair growth slightly increases with age but is much lesser than in people with PCOS.

Follicles are underdeveloped sacs in which eggs develop. Since the left and right follicles have the highest correlation value wrt to the target variable after plotting the same, we create a plot of follicle values. We can see that follicle values are comparatively higher for people with PCOS for both left and right.

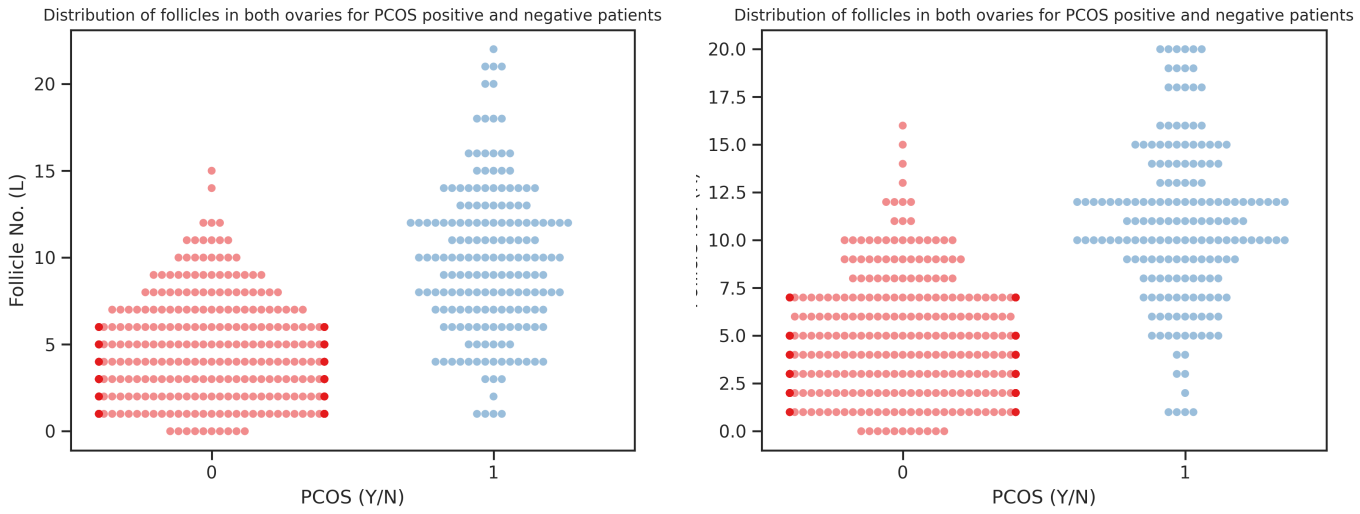

Figure 6: (a) Follicle No. (L) distribution across the classes (b) Follicle (R) distribution across the classes

Data Mining Tasks

Oversampling

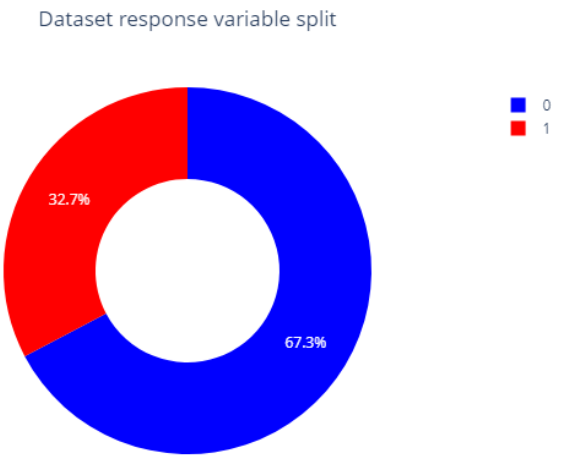

Figure 7: Dataset split visualization

We can see that there is an imbalance in the dataset and our class of interest which is 1 comprises only 32% of the dataset.

Oversampling is a technique used in machine learning to address the problem of imbalanced datasets, where one class of the target variable is significantly underrepresented compared to the other class(es). In such cases, the model may become biased towards the majority class, leading to poor performance in predicting the minority class.

We used Synthetic Minority Oversampling Technique (SMOTE) to address the imbalance in our dataset. In SMOTE, we duplicate examples that are close in the feature space, drawing a line between the examples in the feature space and a new sample at a point along that line. In this way, we duplicate examples from the minority class prior to fitting a model.

Splitting the dataset

The hold-out method is used for data partitioning, such that the train and tests set were generated in the ratio of 70:30. We perform splitting before standardization and scaling to

prevent information leakage. Otherwise, knowing the distribution of the whole dataset might influence how outliers are detected and processed and how the model is parameterized.

Standardization and Scaling

Standardization and scaling can improve the performance of the machine learning model. If the features have different scales, the model may place undue importance on the feature with the larger scale, leading to inaccurate predictions. Standardization and scaling help to normalize the features, allowing the model to learn from all the features equally and make more accurate predictions.

We separate the categorical and numeric variables into two separate datasets, standardize only the numeric variables, and merge the datasets together. This is the dataset after performing standardization and scaling:

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)	Age (yrs)	Weight (Kg)	Height(Cm)	BMI	Blood Group	Pulse rate(bpm)
0	0	1	0	1	1.0	0	-1.321539	-0.786346	0.710457	-1.114995	0.682838	-0.351718
1	1	1	0	0	1.0	0	0.799819	2.164533	1.789416	1.352414	-1.020909	-0.076296
0	0	0	1	1	1.0	0	-0.357285	-0.677675	0.788993	-1.042992	0.682838	-0.351718
0	0	0	0	0	0.0	1	-0.164435	-1.505611	0.883853	-1.909059	-1.020909	-0.351718
0	0	0	0	0	0.0	0	-0.357285	-0.033724	0.440244	-0.218190	-0.452994	-0.351718
...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	0	0	0	0.0	0	-0.357285	0.058269	-0.954753	0.529096	-0.452994	0.199127
1	1	1	1	1	0.0	0	-0.550136	-0.033724	0.614618	-0.291370	0.682838	-0.351718
1	1	1	1	1	1.0	0	-0.164435	-0.484490	1.946840	-1.239384	-1.588825	-0.351718
0	0	0	1	1	0.0	1	-0.164435	0.334247	2.358364	-0.622331	0.682838	-0.351718
1	1	1	1	1	1.0	0	1.378372	0.361845	-1.303502	1.075819	-1.588825	1.300817

Figure 8: Standardized and scaled dataset

Model Exploration and Selection

1. Logistic Regression

Logistic Regression is a simple and elegant model used for classification tasks and works well with small datasets. Logistic regression is sensitive to outliers in the data and since our dataset contains significant outliers we did hyperparameter tuning to find the best parameters that fit the model well. We implemented grid search to find the best parameters.

The best parameters we got are:

- C=0.004832930238571752. This is the inverse of regularization strength. Smaller values signify stronger regularization.
- Penalty=L2 or Ridge Regularization

On fitting the best model, we get the following accuracies:

Accuracy on training set = 88.8%

Accuracy on testing set = 84.02%

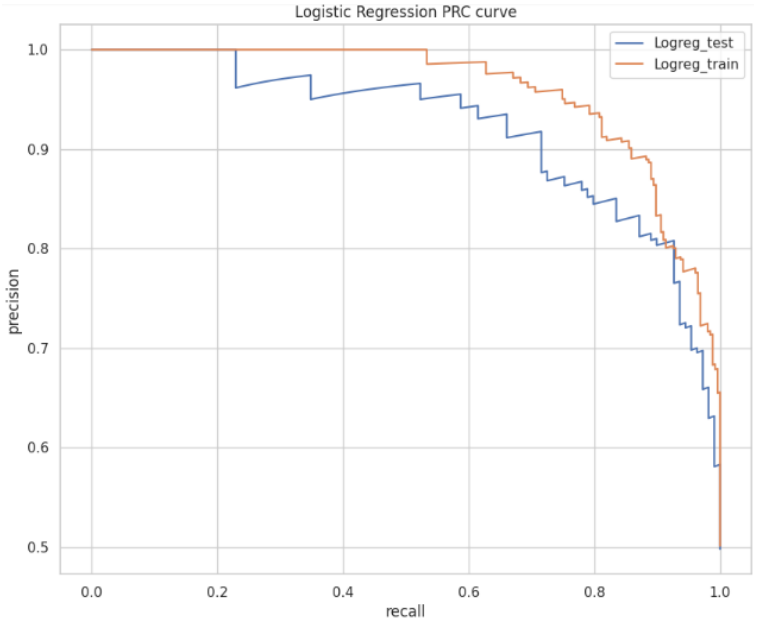

Figure 9: Precision-Recall curve for Logistic Regression model after hyperparameter tuning

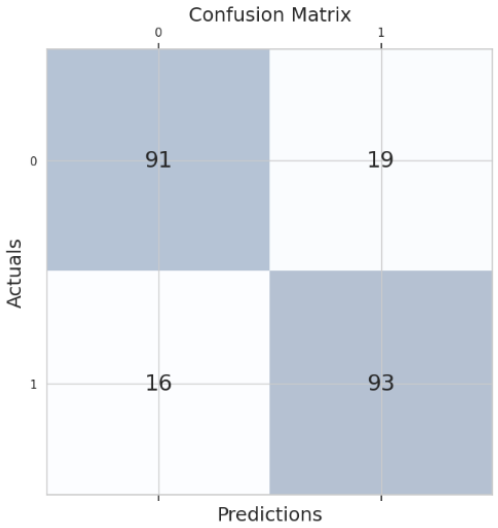

Figure 10: Confusion matrix for Logistic Regression model after hyperparameter tuning

Accuracy after performing cross-validation with k as 5 = 82.55%

We can see that the model is overfitting as there is a considerable difference between both the accuracies. We then check for feature importance. We use odds to calculate feature importance. If we increase any feature X_j by one unit, then the prediction will change e to the power of its weight/coefficient. If the coefficient is positive, the odds increase by that factor, else decrease. We can apply this rule to all weights to find the feature importance.

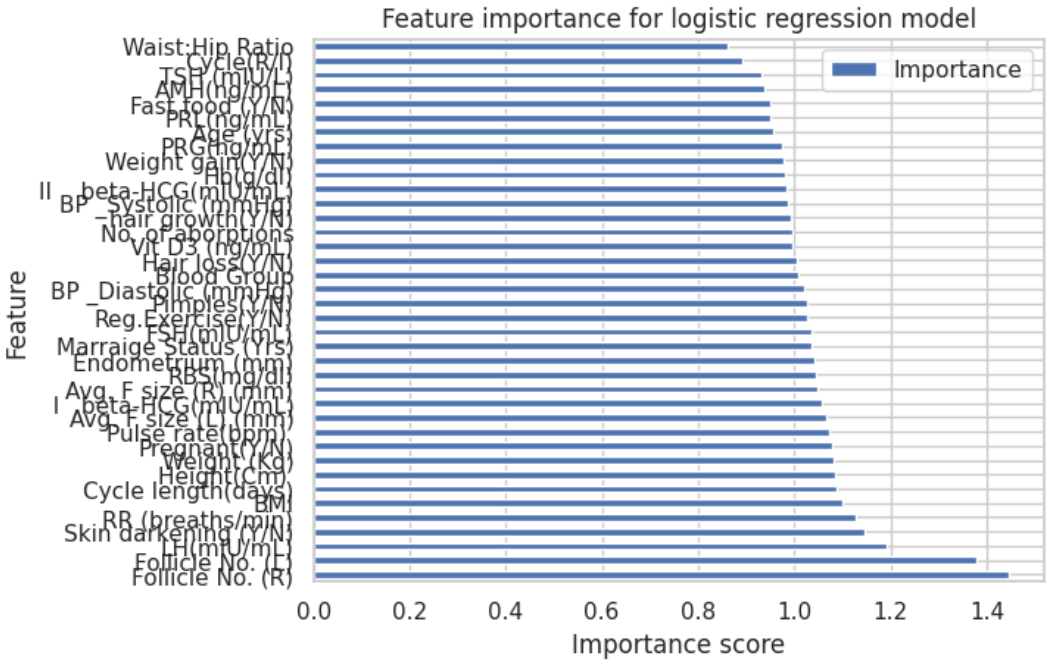

Figure 11: Feature importance barplot for Logistic Regression model

We use recursive feature elimination with cross-validation (RFECV) to find the optimal number of features. These are the optimal features: 'Pregnant(Y/N)', 'Weight gain(Y/N)', 'hair growth(Y/N)', 'Skin darkening (Y/N)', 'Pimples(Y/N)', 'Fast food (Y/N)', 'Reg.Exercise(Y/N)', ' Age (yrs)', 'Weight (Kg)', 'Height(Cm) ', 'BMI', 'Pulse rate(bpm) ', 'RR (breaths/min)', 'Hb(g/dl)', 'Cycle(R/I)', 'Cycle length(days)', 'Marraige Status (Yrs)', 'No. of aborptions', ' I beta-HCG(mIU/mL)', 'FSH(mIU/mL)', 'LH(mIU/mL)', 'Waist:Hip Ratio', 'AMH(ng/mL)', 'Vit D3 (ng/mL)', 'PRG(ng/mL)', 'RBS(mg/dl)', 'BP _Systolic (mmHg)', 'BP _Diastolic (mmHg)', 'Follicle No. (L)', 'Follicle No. (R)', 'Avg. F size (L) (mm)', 'Avg. F size (R) (mm)', 'Endometrium (mm)'.

On fitting the model with these attributes, we get the following accuracies:

Accuracy on training set = 88.8%

Accuracy on testing set = 84.01%

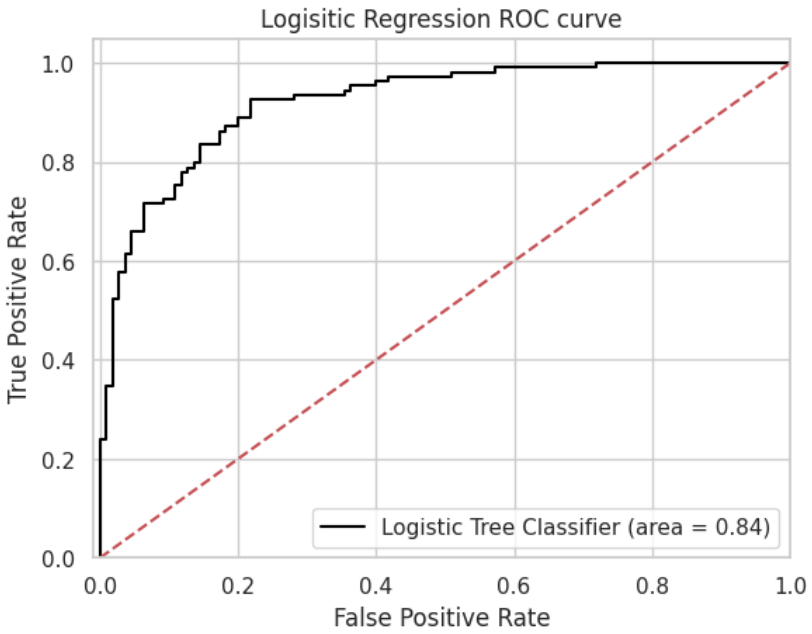

Figure 12: ROC curve for Logistic Regression model

2. Naive Bayes

Gaussian Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on the Bayes theorem and assumes that the features are independent and have a Gaussian distribution.

Advantages: Can perform reasonably well even with small amounts of training data and can handle datasets with a large number of features or attributes.

Disadvantages: Assumes that all features are independent of each other, and that the features are normally distributed or follow a specific distribution.

Accuracies before hyper parameter tuning:

Accuracy on training set = 87.03%

Accuracy on test set = 83.11%

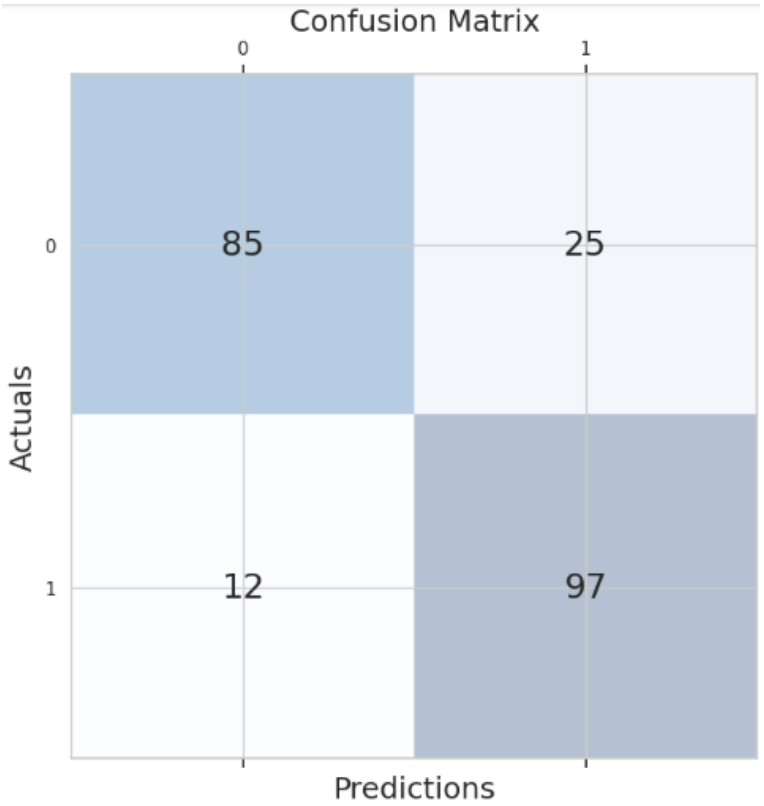

Figure 13: Confusion matrix for Naive Bayes model before hyperparameter tuning

After implementing hyperparameter tuning and performing feature selection using SelectKBest(), the best parameters we get are:

- var_smoothing = 1e-09. It is the portion of the largest variance of all features that is added to variances for calculation stability.

Accuracies after fitting the best model:

Accuracy on training set = 86.05%

Accuracy on test set = 85.84%

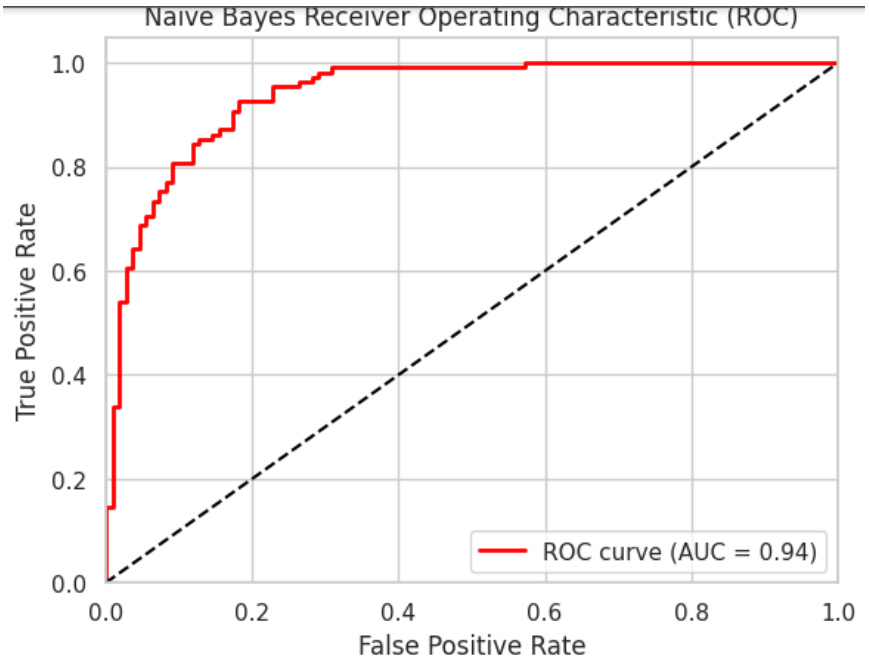

Figure 14: ROC curve for Naive Bayes model

Accuracy after performing cross-validation with k as 5 = 86.63%

3. Decision Trees

A decision tree is a type of supervised machine learning algorithm used for classification and regression analysis. It is a tree-like model that breaks down a dataset into smaller and smaller subsets based on the values of the features, ultimately leading to a prediction or decision. Since we have less data, outliers can have a huge impact on the dataset. So we try out decision trees since they are robust to outliers and feature selection is intrinsic, hence the number of predictors used will also reduce, which is what we want to avoid overfitting.

After implementing hyperparameter tuning, the best parameters we get are:

- Criterion = entropy. This specifies the impurity criterion to be used.
- Max_depth = 25. This specifies the maximum depth of the tree.
- Max_features = 20. The number of features to consider while splitting. This is where the feature selection is intrinsic.

Accuracies after fitting the best model:

Accuracy on training set = 100%

Accuracy on test set = 84.93%

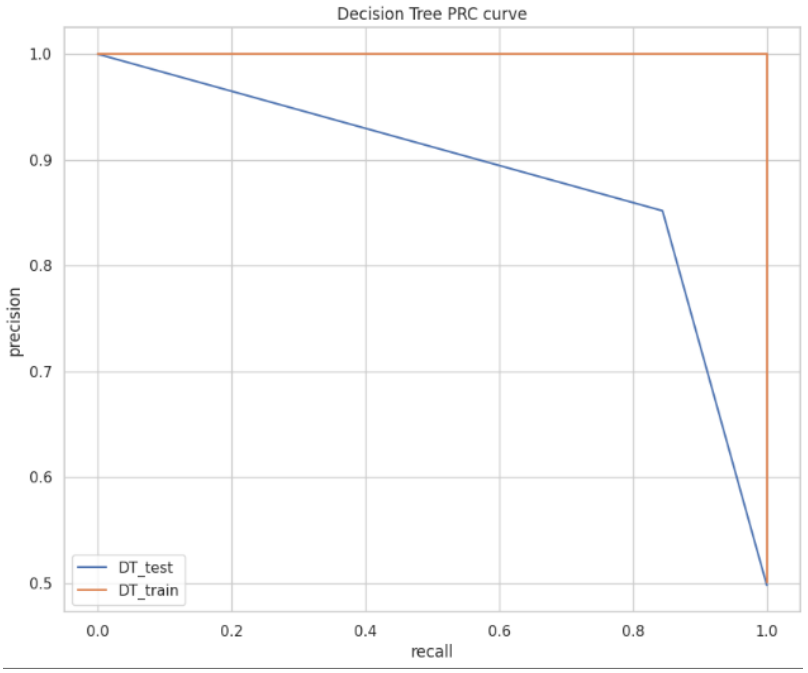

Figure 15:Precision-Recall curve for Decision Tree Classifier

We can get the feature importance scores by using the feature_importances_ attribute of the classifier.

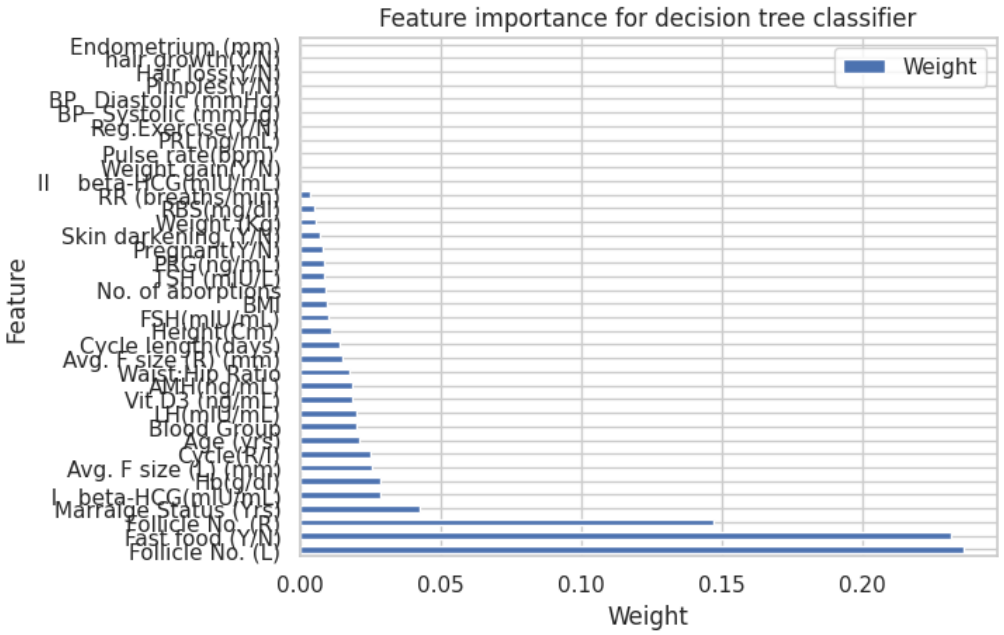

Figure 16: Feature importance barplot for Decision Tree Classifier

Since we know that we should choose 20 features based on the above hyper parameter tuning, instead of using RFECC, we use Recursive Feature Elimination (RFE) and specify the number of features to be 20.

Accuracies after fitting the best model with the top 20 features:
Accuracy on training set = 100%
Accuracy on test set = 85.84%

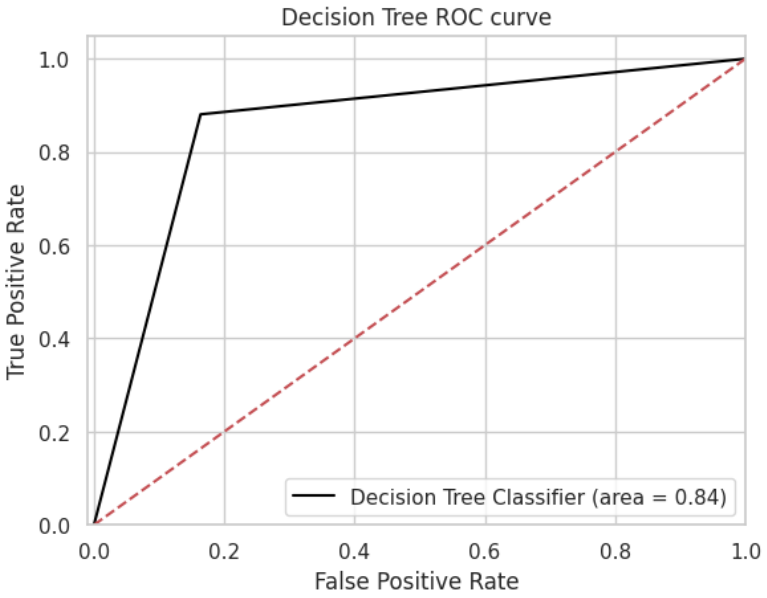

Figure 17: ROC curve for Decision Tree Classifier

We see that the model is still overfitting, as seen by the difference between the training and testing accuracies. A method to fix this is to use ensemble methods. Hence, we move on to Random Forest.

4. Random Forest

Random Forest is a popular machine learning algorithm used for classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make more accurate predictions than a single decision tree. Each decision tree is trained on a bootstrap sample, which is a random subset of the training data with replacement, and a random subset of the features. The final prediction is made by combining the predictions of all the trees in the forest. Combining results from several models can give us more accurate predictions and

it will have improved generalizability. This process helps to reduce overfitting, which is a common problem with decision trees.

After implementing hyperparameter tuning, the best parameters we get are:

- Max_depth = 20
- Max_features = 3
- Min_samples_leaf = 2. Minimum number of samples required to be at the leaf node.
- Min_samples_split = 8. Minimum number of samples required to split an internal node.
- N_estimators = 50. The number of trees in the forest.

Accuracies after fitting the best model:

Accuracy on training set = 99.41%

Accuracy on test set = 90.41%

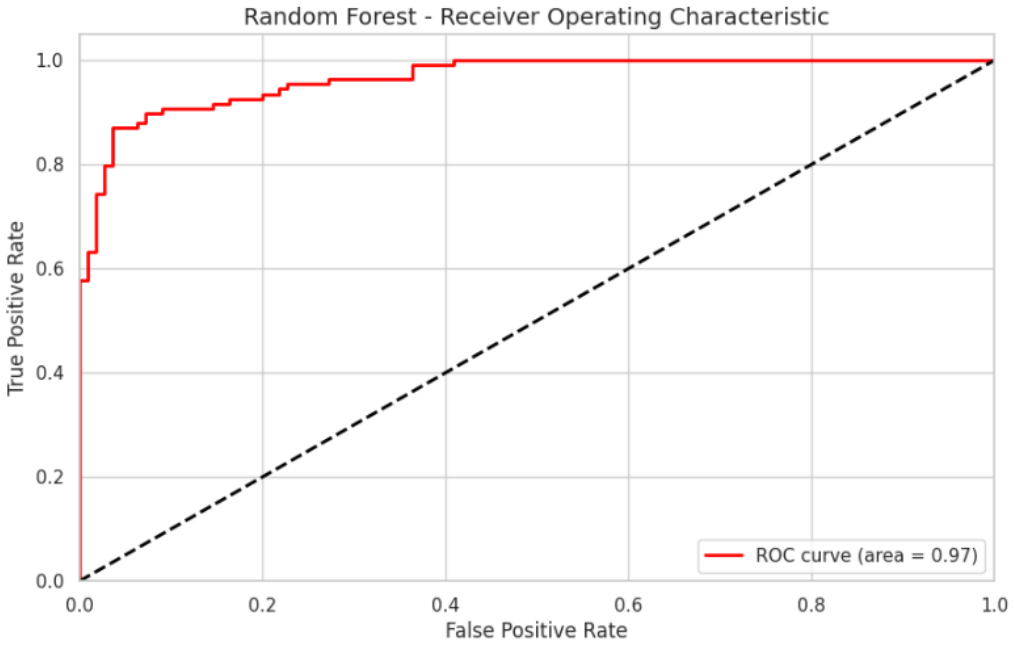

Figure 18: ROC curve for Random Forest model

5. XGBoost

XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It is an implementation of gradient boosting algorithms that uses multiple weak learners (decision trees) to model a predictive

function. It is widely used in various applications such as classification, regression, and ranking problems.

Advantages: High accuracy, handles different kinds of data, wide range of hyperparameters for customization, provides feature importance scores.

Disadvantages: Highly complex, potential overfitting, require significant computational resources.

After implementing hyperparameter tuning, the best features we get are:

- Learning_rate = 0.2
- Min_samples_leaf = 0.4
- Min_samples_split = 1.0,
- N_estimators = 150

Accuracies after fitting the best model:

Accuracy on training set = 88.41%

Accuracy on test set = 85.84%

Accuracy after performing cross-validation with k as 5 = 85.65%

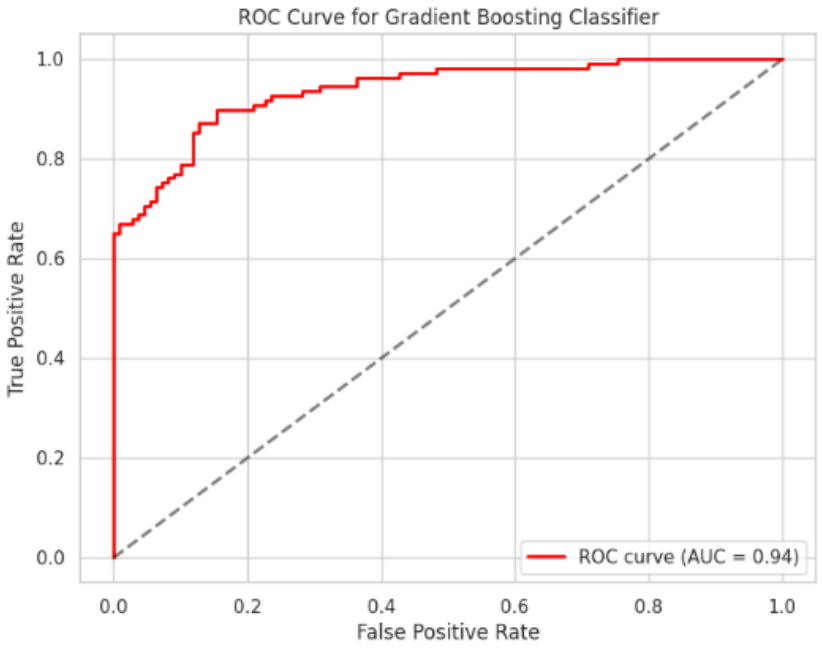

Figure 19: ROC curve for Gradient Boost Classifier

Model Comparison

Model	Training Accuracy	Testing Accuracy	Sensitivity	Precision	F1-Score	AUC
Logistic Regression	88.8%	84.01%	85%	83%	84%	0.84
Naive Bayes	**86.05%**	**85.84%**	87%	85%	86%	0.94
Decision Tree	99%	85.84%	88%	84%	86%	0.84
Random Forest	99.41%	90.41%	91%	90%	90%	0.97
Gradient Boost	**88.41%**	**85.84%**	**90%**	83%	86%	0.94

Table 1: Comparison of the different metrics for all the implemented models

Project Results and Challenges

Results

Both Naive Bayes and Gradient Boosting models seem to be performing reasonably well on PCOS dataset, with training and testing accuracies above 85% and F1 scores above 86%. However, there are some differences in the sensitivity and precision of the two models.

The Naive Bayes model has a slightly higher precision and a slightly lower sensitivity compared to the Gradient Boost model. This means that the Naive Bayes model is better at identifying true negatives (i.e., individuals without PCOS) but may miss some true positives (i.e., individuals with PCOS). On the other hand, the Gradient Boosting model has a higher sensitivity, which means it is better at identifying true positives, but may have a higher false positive rate.

Since the goal is to accurately identify individuals with PCOS (i.e. high sensitivity is important), the **Gradient Boost model is more suitable**. It is an ensemble technique and a very flexible model, which is a good option for a dataset like ours with a mix of categorical and numeric variables.

Overall, with the Gradient Boost model, we get high accuracy (85.84%), close to no overfitting (training accuracy and testing accuracy are pretty close), high sensitivity (90%, which is desired as we want to give more importance to our class of interest).

Based on the feature important scores Gradient Boost model generated, these are some of the most important features :

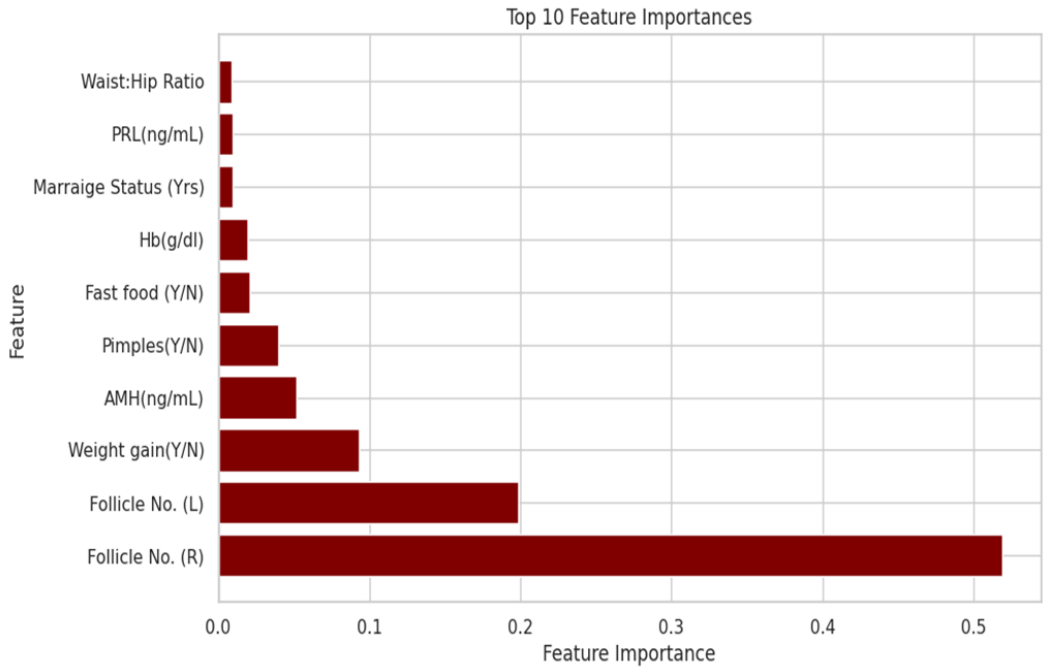

Figure 20: Feature importance barplot for Gradient Boost model

Challenges

1. Not enough data is readily available.
2. Our dataset has 541 records and 45 features, which makes it prone to overfitting.
3. We had to reduce the dimensions but since our dataset has a mix of numeric and categorical variables, we were unable to implement any one dimensionality reduction technique.
4. We had to implement other methods including grid search for hyper parameter tuning, which in some cases is not very computationally efficient and was time consuming.

Project Impact and Future Work

Impact

The diagnostic criteria in use for PCOS is very ambiguous and involves a lot of factors. The exact cause of PCOS is still a topic of research and is unknown. Our proposed model can be implemented in the healthcare industry for early screening based on patient information. The doctors can intimate such patients and assist them in making changes to their lifestyle and scheduling regular check-ups to prevent leading to issues like diabetes and reproductive problems which can arise if not taken care of properly when dealing with PCOS.

Future work

1. Since PCOS diagnosis depends on multiple factors, we could also use imaging techniques to analyze ultrasounds and the like and integrate these into our model to get more accurate results.
2. As PCOS is a chronic condition and evolves over time, models can be built to learn this evolution. Such models can also be used to understand how different medical treatments can affect PCOS related attributes.

References

1. https://www.healthline.com/health/polycystic-ovary-disease
2. https://www.kaggle.com/datasets/prasoonkottarathil/polycystic-ovary-syndrome-pcos
3. https://www.youtube.com/watch?v=Mc5iK0AtGNc&t=251s