PCOSAware: Leveraging Machine Learning for Symptom-Driven Early Awareness of Polycystic Ovary Syndrome

A Research Paper Based on a Final-Year Project

Hamid Shaikh

Department of Computer Science
Viva College , Virar

June 2025

Mail : ahskh384@gmail.com

Github - Hamidshk3084

Table of Contents

1. Abstract 2

2. Literature Review 2

3. Dataset and Feature Engineering 3

4. Methodology 4

5. Results and Evaluation 6

6. Web Application Overview 8

7. Conclusion 12

8. Future Possibilities 12

References 13

Section 1: Abstract

Polycystic Ovary Syndrome (PCOS) is a common hormonal disorder that affects many women, yet it often goes undiagnosed due to a lack of awareness or limited access to medical testing. This project aims to use machine learning not as a tool for medical diagnosis, but as a way to raise awareness among women who might not know they're at risk.

Instead of relying on clinical test results, our model works on simple and easily available information like age, weight, height, menstrual cycle patterns, pulse rate, and lifestyle habits. The dataset used was sourced from Kaggle and preprocessed to focus only on non-invasive inputs. We used various models including Logistic Regression, XGBoost, and Random Forest, with Random Forest giving the best results with an accuracy of around 89%.

This project also includes a user-friendly web application built with Flask. It allows users to input their symptoms and receive an estimated PCOS risk, encouraging them to seek medical attention if needed.

Overall, this research highlights how machine learning can help spread awareness about PCOS, supporting early self-screening in an accessible way, and paving the way for more personalized preventive healthcare tools."

Section 2: Literature Review

Polycystic Ovary Syndrome (PCOS) has become one of the most common health issues faced by women today, especially those in their reproductive years. According to studies, 1 in 10 women may have PCOS, and many remain undiagnosed. Over the years, researchers have tried to use machine learning (ML) to predict PCOS early — but most of these studies depend heavily on medical test reports like blood hormone levels, insulin levels, or ultrasound scans.

For example, many academic papers use features like LH/FSH ratio, insulin resistance, and anti-Müllerian hormone (AMH) levels. While these features do improve prediction accuracy, they require proper lab tests — which aren't always accessible or affordable for every woman, especially in rural or underserved areas.

Recent efforts have tried to make PCOS prediction more approachable by using basic health and lifestyle information. Some researchers have begun experimenting with features like BMI, irregular periods, weight gain, and acne. These approaches are useful for creating awareness and offering an early risk estimate, even before a clinical visit.

What makes our project different is that it focuses only on **non-invasive and easily available features** — things a woman already knows about herself. No lab reports are needed. We use cycle patterns, body measurements, and daily habits as inputs, and combine them with machine learning to build an awareness-based prediction system. This makes our approach more practical and user-friendly, especially for people who are just starting to learn about PCOS.

📄 Section 3: Dataset and Feature Engineering

For this project, we used a publicly available PCOS dataset from **Kaggle**, which contains medical records and symptom data of women diagnosed with or without PCOS. The dataset originally included both clinical (lab-test-based) and non-clinical features, but since our aim was to make the prediction process simple and accessible, we focused only on the **non-invasive inputs**.

🔹 Features Selected from the Dataset

The model uses the following **raw user inputs**:

- **Age** (in years)
- **Height** (cm)
- **Weight** (kg)
- **Waist circumference** (inches)
- **Hip circumference** (inches)
- **Pulse rate** (BPM)
- **Menstrual cycle length** (days)
- **Cycle regularity** (1 for irregular, 0 for regular)
- **Marriage status** (in years)

It also includes **yes/no symptom inputs**, where the user selects 1 for “yes” and 0 for “no”:

- Skin darkening
- Hair growth
- Weight gain
- Pimples
- Fast food consumption
- Regular exercise
- Hair loss

🧠 Auto-Computed Features

In addition to the raw inputs, we also generated a few **calculated values** in the backend to give more meaning to the symptoms:

Derived Feature	Description
BMI	Calculated as weight / (height in meters)^2, used to assess body composition
Waist-to-Weight Ratio	Used as a basic indicator of abdominal weight distribution

Derived Feature	Description
Cycle Score	Gives extra weight to risky cycle lengths (less than 21 or more than 35 days) and irregular cycles
Androgen Indicator	If the user has pimples or excessive hair growth (hirsutism), this is marked as 1
FastFood-WeightGain Flag	Combines fast food consumption and weight gain symptoms
Symptom Severity Score	Adds up all symptom-related yes/no inputs to represent total symptom load

These engineered features helped the model identify deeper patterns that may not be obvious from raw values alone, especially when medical test results are missing.

📄 Section 4: Methodology

This section outlines the step-by-step process followed to build the PCOS awareness prediction model — from data cleaning and feature selection to model training and evaluation.

🔹 1. Data Cleaning and Preprocessing

The dataset was sourced from Kaggle and contained both clinical (medical test-based) and non-clinical features. Since the aim was to create a tool that works without lab tests, only non-invasive inputs were retained.

Key preprocessing steps:

- **Removed unused features** like blood group and clinical hormone levels.
- **Missing values** were handled using:

```
X.fillna(X.mean(), inplace=True)
```

- **Target column** (PCOS (Y/N)) was separated from features.

🔹 2. Feature Selection Pipeline

To make the model more efficient and focus only on the most useful inputs, the following steps were used:

Step	Method	Purpose
Step 1	VarianceThreshold	Removes features with very low variance (constant values)

Step	Method	Purpose
Step 2	SelectKBest(f_classif)	Picks the top 20 statistically important features
Step 3	StandardScaler	Standardizes the data for better model performance

This entire pipeline helps eliminate noise and focuses on meaningful patterns in the data.

🔹 3. Class Imbalance Handling

The dataset had fewer PCOS cases compared to non-PCOS cases, which could cause the model to become biased. To solve this, we applied **SMOTE (Synthetic Minority Over-sampling Technique)**:

```
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_scaled, y)
```

SMOTE creates synthetic examples of the minority class, helping the model learn both classes equally well.

🔹 4. Train-Test Split

The data was then split into training and testing sets with an 80:20 ratio:

```
X_train, X_test, y_train, y_test = train_test_split(
    X_resampled, y_resampled, test_size=0.2, random_state=42
)
```

This allows us to evaluate how well the model performs on unseen data.

📄 Section 5: Results and Evaluation

After preprocessing and training, the models were evaluated using accuracy, precision, recall, F1 score, and confusion matrix. This helped measure how well the system predicted PCOS vs non-PCOS cases.

🔹 Model Evaluation Metrics

The three models used were:

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	82.5%	82%	85%	83%
XGBoost	87.3%	90%	83%	86%
Random Forest	**89.0%**	**90%**	**88%**	**89%**

📊 As seen in the performance chart and confusion matrix, **Random Forest** outperformed the others with the most balanced and accurate results.

🔹 Confusion Matrix (Random Forest)

	Predicted: No PCOS	Predicted: PCOS
Actual: No PCOS	64	7
Actual: PCOS	9	66

This shows that:

- 64 women without PCOS were correctly identified.
- 66 women with PCOS were correctly flagged.
- Very few misclassifications occurred, proving the model is both **sensitive (recall)** and **specific (precision)**.

🔹 Hyperparameter Tuning

To get the best possible results, we used **GridSearchCV** for model optimization:

✅ Random Forest Tuning

```
'n_estimators': [100, 200]
'max_depth': [None, 10, 20]
'min_samples_split': [2, 5]
'min_samples_leaf': [1, 2]
```

✅ XGBoost Tuning

```
'n_estimators': [100, 200]
'learning_rate': [0.01, 0.1]
'max_depth': [3, 5]
'subsample': [0.8, 1]
'colsample_bytree': [0.8, 1]
```

✅ Logistic Regression

Used with balanced class weights:

```
LogisticRegression(class_weight='balanced')
```

Each model was trained and tested **individually**. Final results confirmed that **Random Forest** was the most reliable for real-world usage.

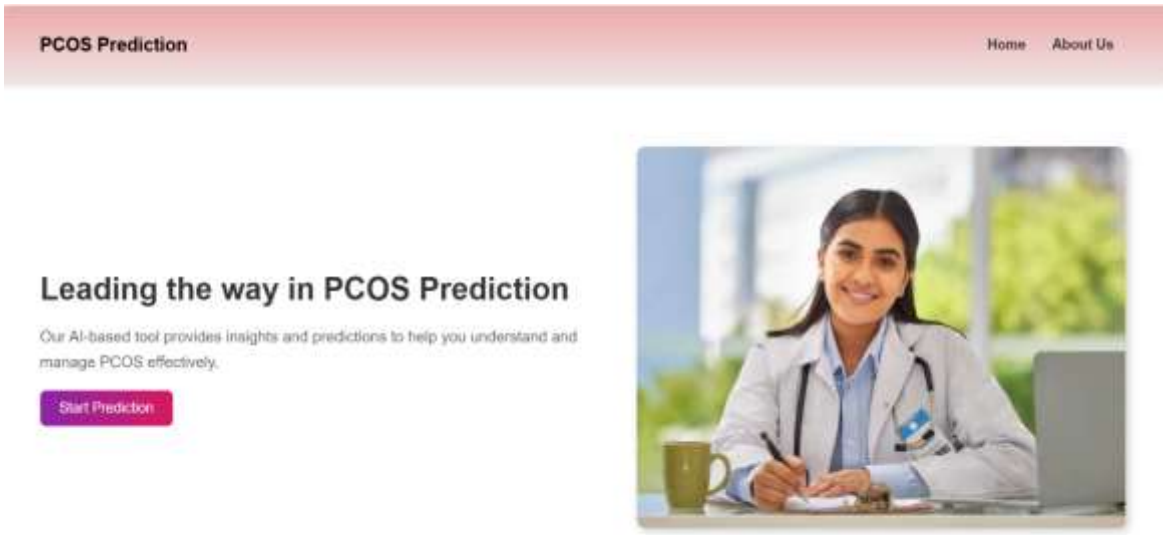
📄 Section 6: Web Application Overview

Overview

The project includes a user-friendly web application developed using Flask. It is designed to raise awareness about PCOS through an accessible and educational interface.

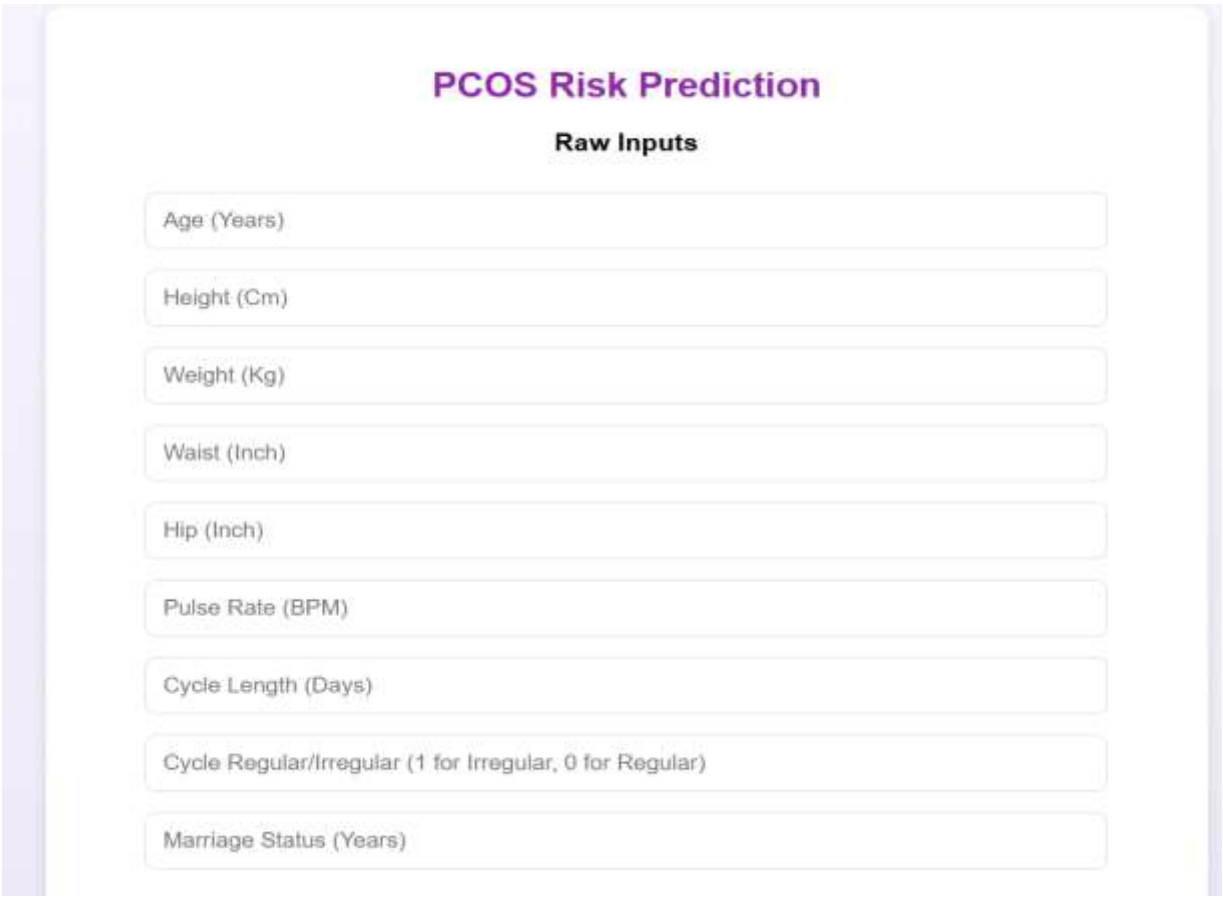
6.1. Application Pages

1. **Introduction Page**
    - Serves as the landing page
    - Provides a concise explanation of the app's purpose

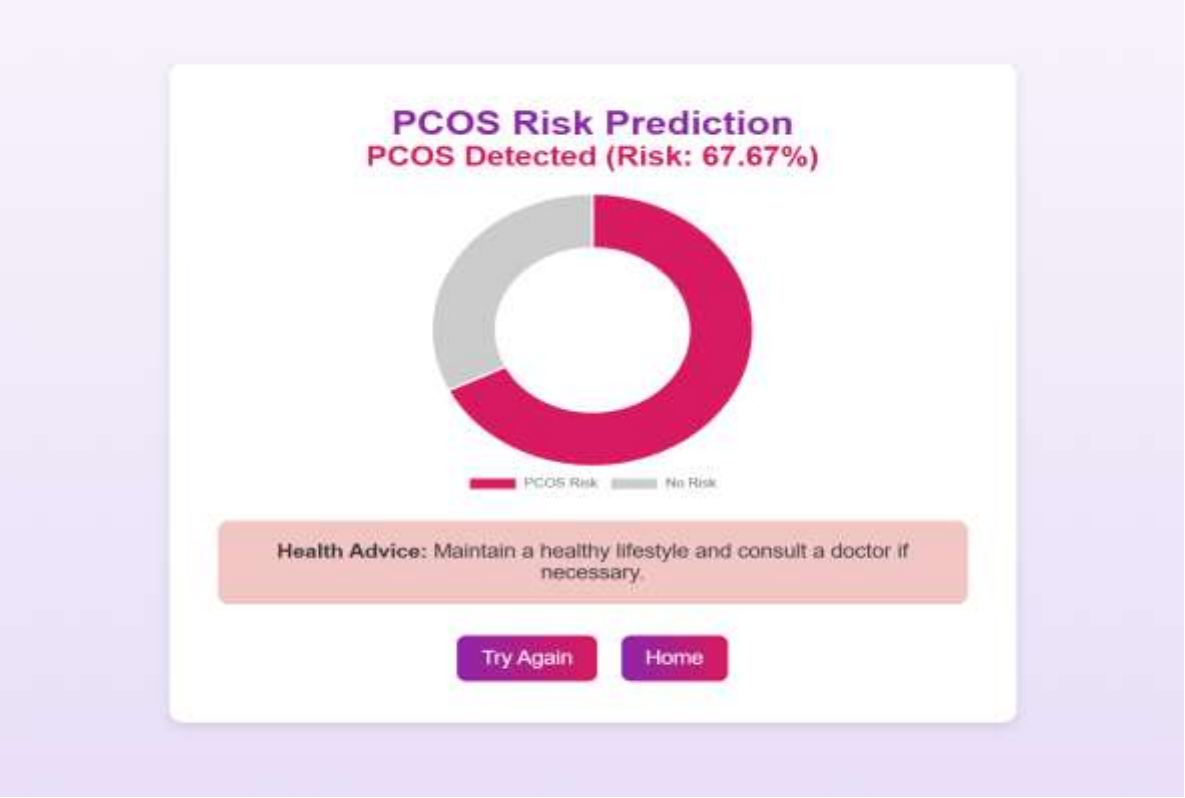


    - Includes a “College Project” disclaimer and a brief “About Us” section introducing the developer or team

2. **Input Page**
    - Users enter non-clinical data, including:
        - Personal details: Age, Height, Weight, Waist, Hip, Pulse, Menstrual Cycle Length and Regularity, Marriage Status

3. 

        - Symptom indicators: Skin Darkening, Hair Growth, Weight Gain, Pimples, Fast-Food Consumption, Regular Exercise, Hair Loss
        - Designed to be simple and intuitive

4. **Result Page**
    - Displays the model's estimated PCOS risk percentage
    - Offers personalized health advice encouraging consultation with professionals or lifestyle changes
5. 

6.2. Technical Workflow

1. **User Flow**
    - Navigation path: Introduction → Input → Result
2. **Backend Processing**
    - Reads and validates user inputs
    - Computes derived features (BMI, Cycle Score, Androgen Indicator, Symptom Severity, etc.)
    - Constructs a feature vector and feeds it to the trained **Random Forest** model (loaded via joblib)
    - Generates risk probability and returns the result
3. **Output Rendering**
    - Risk displayed as a percentage
    - Advice is dynamically generated based on the risk level
    - Emphasizes awareness and encourages medical consultation

6.3. Design & Accessibility

- Uses **Bootstrap CSS** for a clean, responsive layout
- Tailored for users with **no medical or technical background**
- Clearly states that the app is for raising awareness, not medical diagnosis
- Contains educational content and disclaimers to ensure responsible use

Conclusion

As a final-year college student, I developed this web-based PCOS awareness tool with two main goals in mind: **education** and **empowerment**. Powered by a carefully trained Random Forest model (89% accuracy, 0.89 F1-score), the system offers an early *awareness check* rather than a medical diagnosis. It relies solely on user-provided, non-invasive inputs—such as age, BMI, menstrual cycle information, and lifestyle habits—to estimate the likelihood of PCOS.

By steering clear of costly lab tests, this project makes health insights more **approachable**, especially for young women who may not be aware they're at risk. The web interface—broken down into Introduction, Input, and Result pages—ensures accessibility and clarity, guiding users through the experience like a conversation. The tool ends each session with a clear suggestion:" If your risk appears high, consider consulting a healthcare provider".

It's designed not only to inform but also to prompt timely, actionable steps.

Future Possibilities

1. **Broader & More Diverse Data Collection**
    - Incorporate data from volunteers across different age groups, regions, and cultures to strengthen the model's generalizability and sensitivity to diverse backgrounds.
2. **Enhanced Feature Set with Intelligent Health Indicators**
    - Introduce features like BMI category assessment, waist-to-hip ratio interpretation, basal metabolic rate estimates, and even stress levels to enhance prediction capabilities.
3. **Personalized Insight Engine**
    - Build a recommendation module that can provide users with tailored tips, such as personalized exercise plans, dietary suggestions, and lifestyle adjustments based on their risk profile.
4. **Language Localization & Cultural Adaptation**
    - Launch multilingual support—especially in Marathi, Hindi, and other regional languages—to increase accessibility and cultural relevance across India and beyond.
5. **Online Deployment with User Feedback Loop**
    - Host the application on platforms like Heroku, AWS, or Azure, and collect real-world user feedback to refine the interface, improve model accuracy, and track engagement and outcomes.
6. **Integration into Academic and Community Outreach**
    - Share the tool at college symposiums, local health camps, and women's wellness programs to measure its educational impact and drive improvements through community-led feedback.
7. **Towards a Preventive Healthcare Ecosystem**
    - Connect the tool with telemedicine services, diet/nutrition platforms, or mental wellness apps to provide users a holistic, preventive care experience.

References

1. Rotterdam ESHRE/ASRM-Sponsored PCOS Consensus Workshop Group. “Revised 2003 consensus on diagnostic criteria and long-term health risks related to polycystic ovary syndrome.” *Fertility and Sterility* 81.1 (2004): 19–25.
2. Teede, Helena J., et al. “Recommendations from the international evidence-based guideline for the assessment and management of polycystic ovary syndrome.” *Human Reproduction* 33.9 (2018): 1602–1618.
3. World Health Organization. “Polycystic Ovary Syndrome (PCOS) Factsheet.” World Health Organization, 2023. Available at: https://www.who.int/news-room/fact-sheets/detail/polycystic-ovary-syndrome.
4. Dataset Source: Kaggle.com – Polycystic Ovary Syndrome (PCOS) Dataset. Available at: Polycystic ovary syndrome (PCOS)