AI INNOVATION CHALLENGE

NeoHealthAI - A Comprehensive Healthcare Analytics Platform for Women's Health Monitoring and Menstrual Phase Prediction using Machine Learning

A Project Report

Submitted by:

Zeel Panchal (Team Leader) – AI/ML Engineer

Bansari Patel – AI/ML Engineer

Chirag Mandani – Full Stack Developer

Vishal Jani – Full Stack Developer

Members of

TEAM NEOHEALTH

ABSTRACT

Women's health tracking has historically been fragmented, relying heavily on manual logging methods that lack predictive capabilities and fail to integrate holistic lifestyle data. NeoHealth AI addresses this gap by providing a unified, production-grade healthcare analytics platform. The system leverages advanced Machine Learning (XGBoost) to predict menstrual cycle phases (Follicular, Ovulation, Luteal, Menstrual) based on complex hormonal patterns and daily physiological metrics.

The platform integrates a secure, modern web interface for logging daily health data—including hormonal levels (LH, Estrogen, PdG), sleep quality, stress levels, and symptoms. By processing this data through a robust preprocessing pipeline and an inference engine, NeoHealth AI provides real- time, data-driven health insights. This report details the design, implementation, and performance of the system, demonstrating its potential to empower users with actionable wellness intelligence.

CHAPTER 1: INTRODUCTION

1.1. Problem Definition

Traditional menstrual tracking applications often function as simple calendars, ignoring the complex interplay between hormones and lifestyle factors. Users struggle to correlate their symptoms (e.g., fatigue, mood swings) with their physiological state, leading to a lack of actionable health awareness. There is a critical need for an intelligent system that uses historical and real-time data to predict health states accurately.

1.2. Project Objectives

- **Predictive Analytics:** To develop a Machine Learning model capable of accurately classifying menstrual phases using hormonal and vital sign data.
- **Holistic Health Monitoring:** To create a dashboard that visualizes trends in key biomarkers, including Luteinizing Hormone (LH), Estrogen, and Pregnanediol Glucuronide (PdG).
- **User-Centric Interface:** To build a responsive, intuitive web application that simplifies daily data logging and interpretation.
- **Data-Driven Insights:** To provide personalized feedback based on predicted physiological states efficiently.

CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

2.1. System Architecture

The NeoHealth AI system follows a modern Client-Server architecture:

- **Frontend Layer (Client):** Built with React.js and Vite, this layer handles user interaction, data visualization (using Recharts), and secure authentication. It communicates with the backend via RESTful APIs.
- **Backend Layer (Server):** Developed using Flask (Python), this layer manages API requests, validates data using Marshmallow schemas, and orchestrates the Machine Learning pipeline.
- **Data Layer:** Stores user profiles and health logs in a relational database (PostgreSQL / SQLite) managed via SQLAlchemy.
- **ML Inference Engine:** A dedicated module that loads pre-trained XGBoost models and Scalers to generate real-time predictions for incoming user data.

2.2. Proposed Workflow

1. **User Authentication:** Secure login using JWT (JSON Web Tokens).
2. **Data Logging:** Users input daily stats (Hormones, Sleep, Stress, Symptoms).
3. **Data Processing:** The backend cleans and scales the data, generating "lag features" (previous days' data) to capture temporal dependencies.
4. **Prediction:** The processed features are fed into the XGBoost classifier to determine the current cycle phase.
5. **Visualization:** Results are returned to the dashboard, displaying the current phase and trend graphs.

CHAPTER 3: IMPLEMENTATION AND TECHNOLOGY STACK

3.1. Technology Stack

- **Programming Languages:** Python (Backend/ML), JavaScript (Frontend)
- **Frontend Frameworks:** React 19, Vite, TailwindCSS, Framer Motion
- **Backend Framework:** Flask
- **Machine Learning Libraries:** Scikit-Learn, XGBoost, Pandas, NumPy, Joblib
- **Database:** PostgreSQL / SQLite
- **Tools:** Git, Postman, VS Code

3.2. Machine Learning Methodology

Algorithm: XGBoost (Extreme Gradient Boosting) Classifier was selected for its high performance on structured, tabular data.

Feature Engineering:

- Hormonal Markers: LH, Estrogen, PdG.
- Vitals: Resting Heart Rate, Deep Sleep Minutes, Daily Steps.
- Symptom Scoring: Aggregated scores for cramps, fatigue, and mood.
- Temporal Features: Lag features (e.g., lh_prev1, stress_prev2) were created to model the sequential nature of the menstrual cycle.

Training Process:

- Data was split into training and testing sets.
- Label Encoding was used for the target variable (Cycle Phases).
- Standard Scaling was applied to normalize numerical inputs.
- Hyperparameter tuning was performed to optimize learning_rate, max_depth, and n_estimators.

CHAPTER 4: RESULTS AND DISCUSSION

4.1. Model Performance

The trained XGBoost model demonstrated robust performance in classifying the four key menstrual phases:

- Follicular Phase
- Ovulation (Fertility Peak)
- Luteal Phase
- Menstrual Phase

Evaluated using Multi-class Log Loss (mlogloss) and Accuracy metrics, the model successfully outperformed baseline algorithms (such as Random Forest) by effectively utilizing the temporal lag features.

4.2. Application Outputs

- **Dashboard:** Successfully renders real-time graphs showing the correlation between stress levels and hormonal fluctuations.
- **Prediction Widget:** Instantly updates the "Current Phase" status card upon entry of new daily logs.
- **Responsiveness:** The application maintains a smooth 60fps UI on both desktop and mobile devices.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1. Conclusion

NeoHealth AI successfully bridges the gap between medical data and user accessibility. By integrating clinical-grade machine learning with a consumer-friendly interface, the project provides a valuable tool for women's health observability. The system meets all primary objectives of accurate prediction, secure logging, and interactive visualization.

5.2. Future Enhancements

- **Wearable Integration:** Direct API synchronization with Apple Health, Fitbit, and Oura Ring to automate data entry.
- **Mobile Application:** Development of a native mobile app using React Native for push notifications.
- **LLM Integration:** Implementation of a specialized Health Chatbot using Large Language Models to answer user queries securely.
- **Community Module:** Adding anonymous forums for peer support and community engagement.

REFERENCES

1. XGBoost Documentation - https://xgboost.readthedocs.io/
2. React.js Official Documentation - https://react.dev/
3. Flask API Development - https://flask.palletsprojects.com/
4. Menstrual Cycle Physiology - National Institutes of Health (NIH) Resources.