AI-Powered Health Monitoring System

Using AI to Detect Health Anomalies in Real-Time

Team Members:

- Thobeka Jali
- Sophia Nakhanu
- Jean Guya

Project Overview & SDG Alignment

Project Overview: We built an AI-powered system that monitors real-time health data (e.g., heart rate, blood oxygen levels) from wearable devices, detects anomalies, and provides early alerts.

SDG 3: Good Health & Well-being

Enables early detection of illness.

SDG 9: Industry, Innovation & Infrastructure

Promotes scalable, tech-based healthcare tools.

Problem Statement

- Many people wear smartwatches, but the data is underused.
- There's a need for systems that **detect abnormal health signs early** and **offer simple, useful feedback**.
- We aim to turn raw biometric data into actionable health insights using machine learning.

Data Collection

Source: Simulated dataset representing real-world data from wearable devices.

Metrics Collected:

- Heart rate
- Blood oxygen saturation (SpO_2)
- Body temperature *(optional extension)*

Sample Data (CSV):

Timestamp Heart Rate SpO_2 2023-10-01 10:00 78 bpm 97% 2023-10-01 10:01 110 bpm 88% ← anomaly

Model & Approach

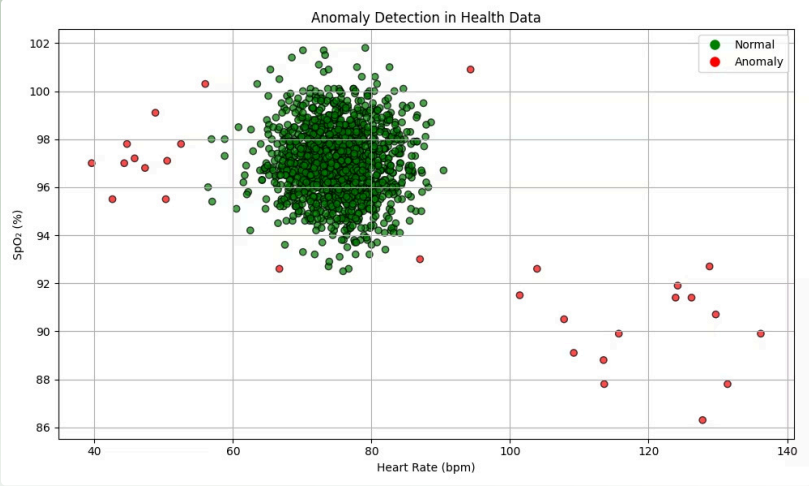
Algorithm: Isolation Forest (unsupervised anomaly detection)

Features Used:

- Heart rate
- Blood oxygen levels

Why Isolation Forest?

- Good for detecting outliers in time-series health data
- Works without needing large labeled datasets



Results

Evaluation Metrics:

- **Precision:** 0.897
- **Recall:** 0.867
- **F1 Score:** 0.881

Interpretation:

- High precision = very few false positives
- High recall = most anomalies are correctly flagged
- Balanced performance for real-world use

User Interface

Built with: Streamlit (Python)

Features:

- Real-time health metric display
- Clear anomaly alerts (e.g., high heart rate)
- Simple health advice

Demo Screenshot:

Testing & Validation

Unit Testing:

Checked each script for correct behavior (data loading, preprocessing, model inference)

Integration Testing:

Full flow tested from data → prediction → app display

User Testing:

Manual testing of the Streamlit interface with various inputs

Deployment

Local Deployment:

- App runs locally with streamlit run app/streamlit_app.py

Cloud-Ready:

- Code structured for deployment on platforms like:
  - Streamlit Cloud
  - Hugging Face Spaces
  - AWS / Azure

Challenges & Lessons Learned

Challenges Faced:

- Balancing model sensitivity vs false alerts
- Designing a simple but effective UI
- Simulating realistic health data

Lessons Learned:

- Unsupervised models are powerful with good preprocessing
- UI clarity is just as important as backend accuracy
- Modular code makes testing and deployment easier

Conclusion & Next Steps

Conclusion:

We built a working AI system that detects potential health issues using simulated biometric data and shows alerts through an easy-to-use web app.

Next Steps:

- Connect with real wearable APIs (Fitbit, Apple Health)
- Add more health signals (e.g., sleep, temperature)
- Deploy app to the cloud
- Partner with healthcare startups for field testing