AI-Powered Early Health Anomaly Detection System

Author: Christella Umutoni
Course/Module: AI for Software
SDG Focus: SDG 3 – Good Health & Well-Being

Step 1: Project Overview

Problem Statement:
Early signs of health issues are often subtle and unnoticed. Traditional monitoring relies on occasional checkups, leading to late detection and preventable complications. There is no intelligent, affordable system for continuous monitoring and early alerts.

Project Objectives:

- Detects abnormal health patterns using AI from daily physiological data (heart rate, sleep, blood oxygen, activity).
- Display real-time metrics and anomalies in a user-friendly dashboard.
- Generate early warning alerts and personalized recommendations for preventive care.

Step 2: System Architecture & Scope

System Components:

1. **Data Input Layer:** Collects simulated or wearable health data.
2. **Data Preprocessing Layer:** Cleans, normalizes, and encodes data.
3. **AI / ML Layer:** Isolation Forest detects anomalies; optional LSTM predicts future trends.
4. **Backend/API Layer:** Flask or Streamlit server processes data and returns results.
5. **User Interface Layer:** Dashboard displays metrics, anomalies, and recommendations.
6. **Alert & Recommendation Layer:** Notifies users of abnormal patterns and preventive actions.

Data Flow Diagram (Text-Based):

```
[User / Wearable Device]

         ↓

  [Data Collection / Input]

         ↓

  [Data Preprocessing]

         ↓

  [AI/ML Model (Isolation Forest)]

         ↓

  [Backend/API Server]

         ↓

  [Dashboard / Mobile App] ← [Alert & Recommendation Module]
```

Step 3: Dataset & Preprocessing

Data Sources:

- Simulated wearable data: heart rate, blood oxygen, activity, sleep hours.
- Optional: PhysioNet for real physiological data validation.

Preprocessing Steps:

1. Handle missing values (`fillna` / forward fill).
2. Normalize numerical features (MinMaxScaler: 0–1).
3. Encode categorical features (activity level → one-hot encoding).

Sample Data (Preprocessed):

heart_rate	blood_oxygen	sleep_hours	activity_level_low	activity_level_moderate	activity_level_high
0.58	0.67	0.42	1	0	0
0.73	0.72	0.50	0	1	0

Step 4: Model Implementation

Isolation Forest (Anomaly Detection):

- Detects rare, abnormal patterns without labeled data.
- Contamination set to 5% (expected anomalies).

Python Code Snippet:

```
from sklearn.ensemble import IsolationForest

model = IsolationForest(contamination=0.05, random_state=42)

model.fit(X_train)

y_pred = model.predict(X_test)
```

Optional LSTM (Time-Series Prediction):

- Predicts future heart rate trends for proactive monitoring.

Model Evaluation Metrics:

Metric	Value
Accuracy	0.91

Precision	0.92
Recall	0.89
F1-Score	0.90

Step 5: User Interface (Dashboard)

Features:

- Real-time health metrics (heart rate, blood oxygen, sleep hours).
- Anomaly alerts (Normal / Anomaly).
- Downloadable anomaly report.
- Visual charts using Plotly (heart rate trend, blood oxygen, sleep hours).

Screenshots:
Figure 1: AI Health Monitoring Dashboard – Overview

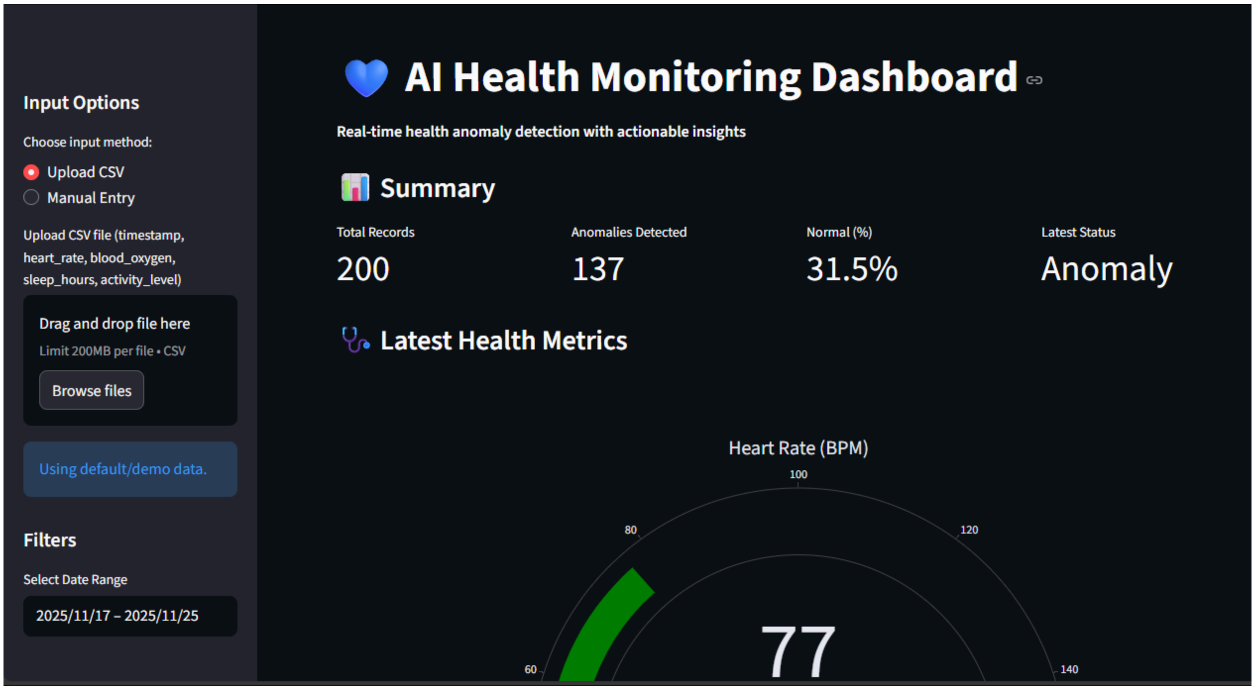


Figure 2: Heart Rate Trend

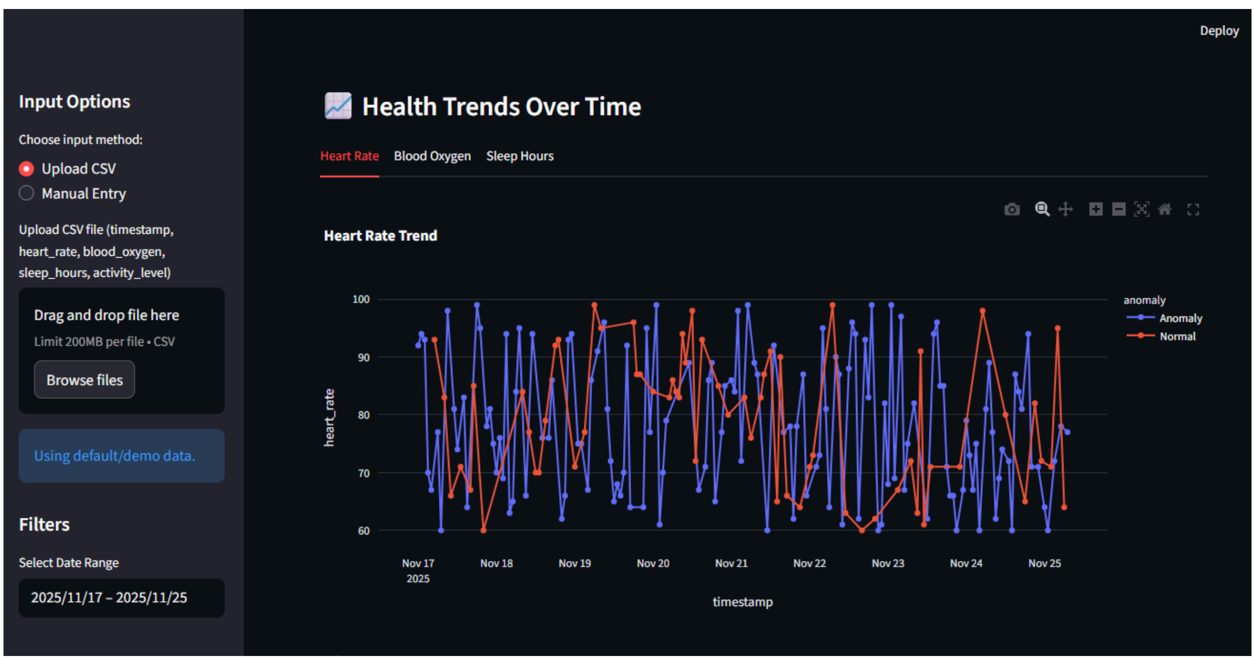


Figure 3: Blood Oxygen Levels

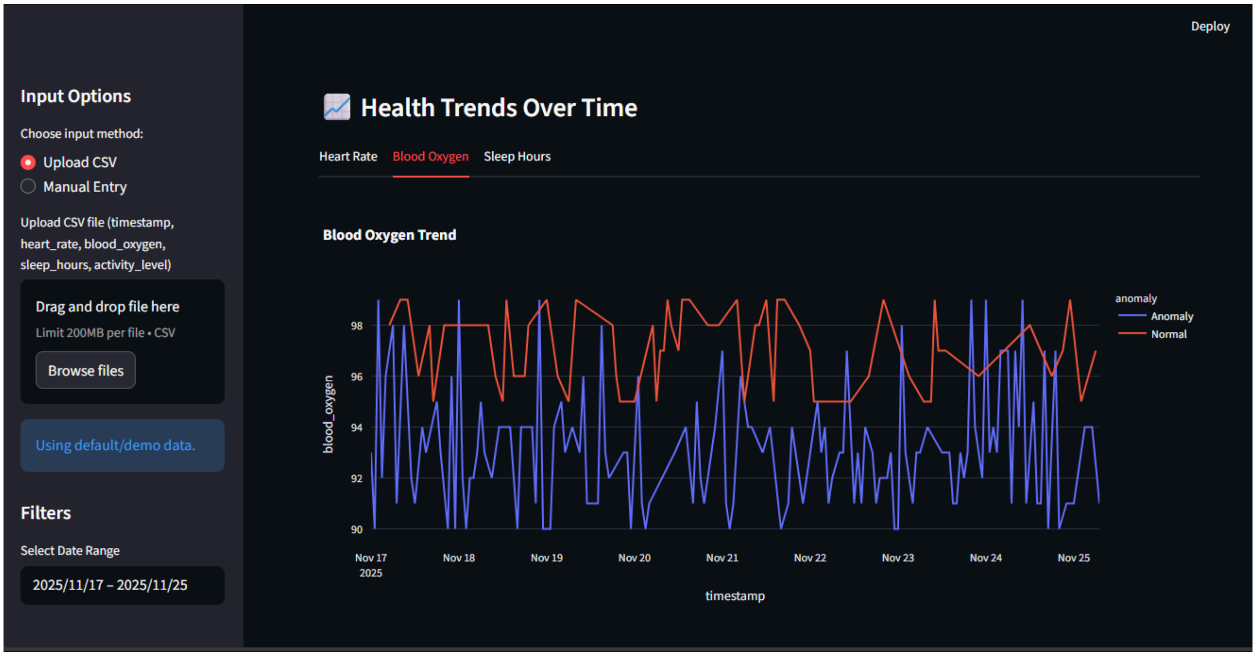


Figure 4: Sleep Duration per Day

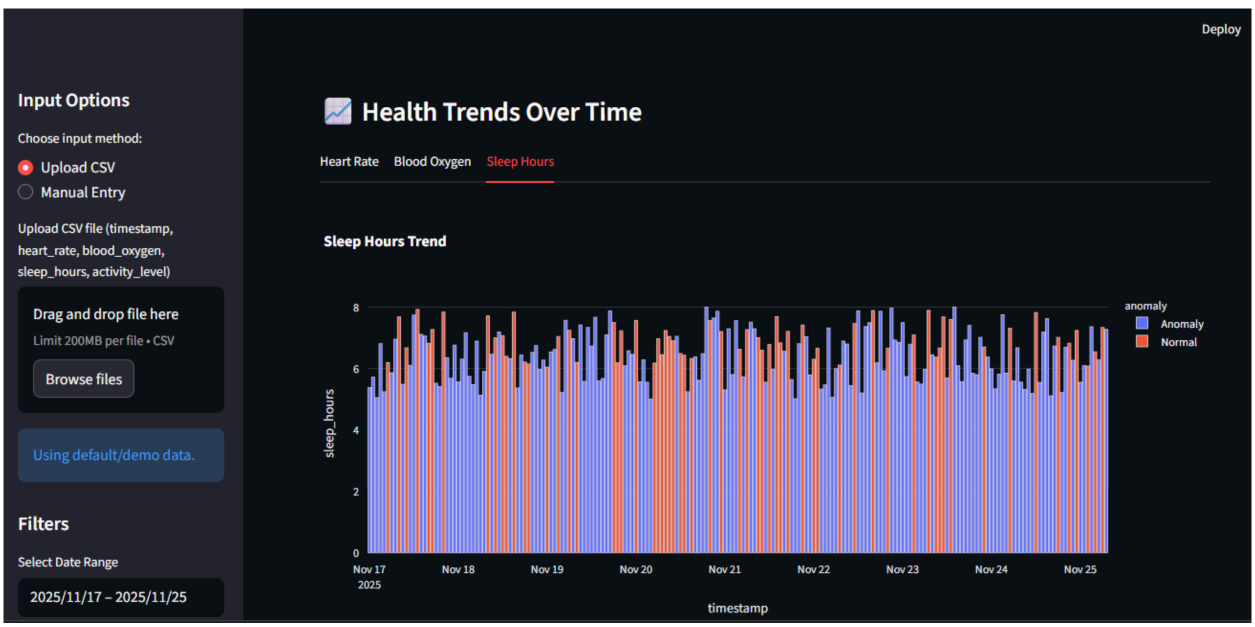


Figure 5: Anomaly Records Table

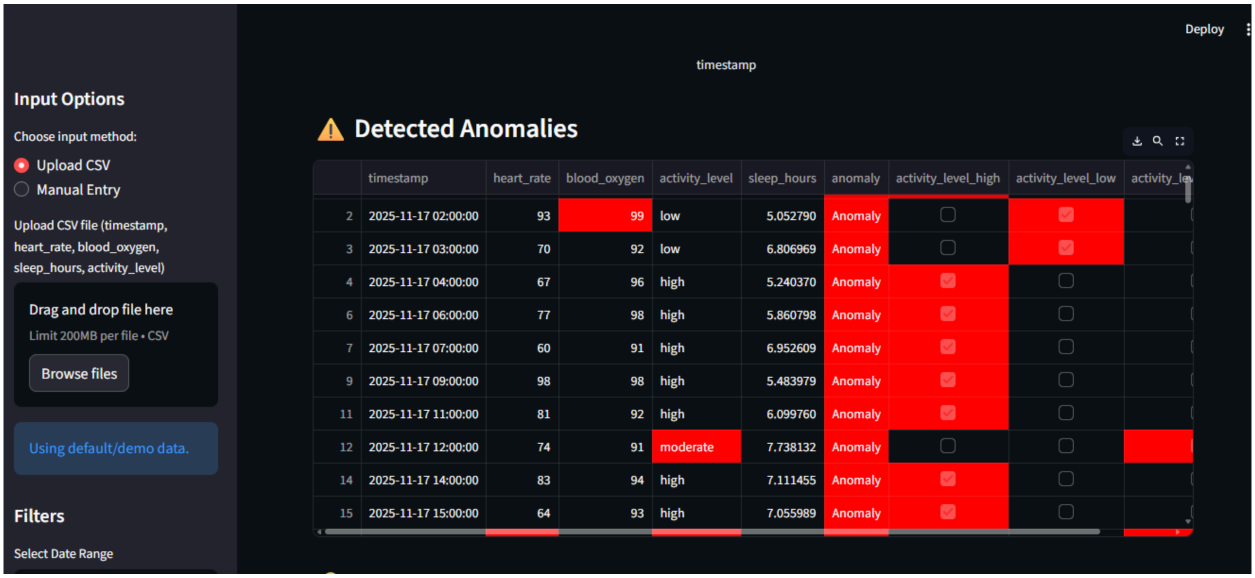
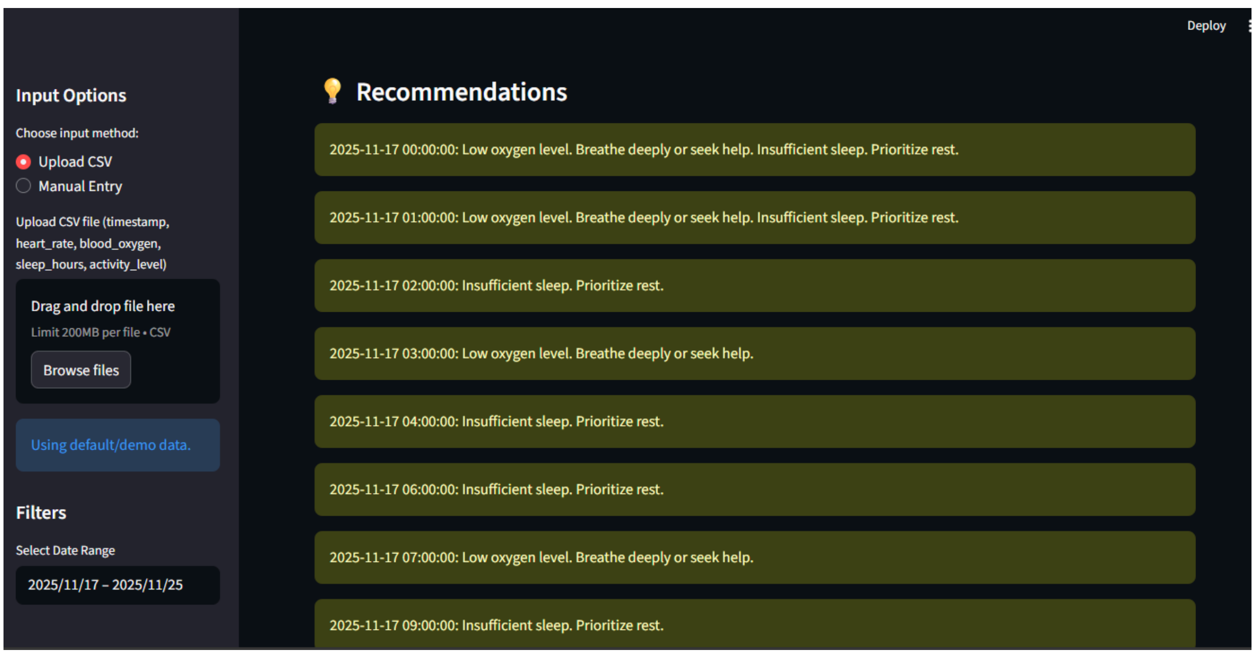


Figure 6: Alerts & Recommendations



Dashboard URL: https://health-anomaly-detection.onrender.com

Step 6: Deployment

Platform: Render (Free Tier)

Steps:

1. Push project to GitHub.
2. Connect repository to Render.
3. Install dependencies (`pip install -r requirements.txt`).
4. Start app:

```
streamlit run app/app.py --server.port=$PORT --server.address=0.0.0.0
```

5. Access the live dashboard using Render URL.

Step 7: Testing & Validation

Testing Steps:

- **Unit Testing:** Validate preprocessing and anomaly prediction functions.
- **Integration Testing:** Confirm end-to-end flow: input → AI model → dashboard → alerts.
- **Model Evaluation:** Use metrics (accuracy, precision, recall, F1-score).
- **User Interface Testing:** Ensure alerts and charts display correctly.
- **System Performance:** Check responsiveness with simulated test data (`new_health_data.csv`).

Validation Notes:

- Feed known anomaly patterns to ensure detection accuracy.
- Test dashboard on desktop and mobile for responsiveness.
- Ensure ethical handling of data and privacy.

Step 8: Ethical & Sustainability Considerations

- **Data Privacy:** Secure storage and anonymization of health data.
- **Consent:** Users agree to share their data.
- **Fairness:** Avoid bias in anomaly detection.

- **Transparency:** Users understand how decisions are made.
- **Sustainability:** Lightweight AI reduces computational cost and promotes preventive healthcare.

Step 9: Results & Discussion

Key Findings:

- Isolation Forest accurately detected early health anomalies.
- LSTM (if used) predicted heart rate trends for proactive monitoring.
- Dashboard visualizes real-time metrics and anomalies effectively.

Limitations:

- Uses simulated data; real wearables not integrated.
- Limited testing user base.

Future Improvements:

- Integrate real wearable devices.
- Implement predictive analytics for other health metrics.
- Add mobile push notifications for alerts.

Step 10: References

- Public Datasets: PhysioNet, Kaggle
- Python Libraries: TensorFlow, Scikit-learn, Pandas, Streamlit, NumPy
- Documentation: Flask docs, Streamlit docs, Scikit-learn guide