**Project: Building an Intelligent Health Monitoring & Heart Attack Prediction System Using AWS EMR, SageMaker, Lambda, Athena, and SNS**

AKASHKUMAR DINESHKUMAR

MS. Information Technology, Arizona State University

IFT 512: Advanced Big Data Analytics/AI

Durgesh Sharma

Nov 23rd, 2025

**Project: Building an Intelligent Health Monitoring & Heart Attack Prediction System Using AWS EMR, SageMaker, Lambda, Athena, and SNS**

**Project Description:**

The Heart Health Alert project illustrates a complete, end-to-end data pipeline that detects potential heart-related risks using patient vitals and machine learning. The system first generates synthetic health data to simulate realistic signals for patient monitoring. Following this, it is processed in a distributed environment using Spark and AWS EMR for computing seven-day averages for each patient. In Phase 3, the cleaned and aggregated dataset is used to train an XGBoost model in AWS SageMaker. The learned model will be made available as a real-time inference endpoint. Finally, Phase 4 embeds that endpoint into a Lambda function, assessing fresh vital signs for increased danger and informing Amazon SNS when a patient appears to be at heightened risk.Together, these phases reproduce how a healthcare organization might build an automated early-warning system using cloud technologies.

**Project Workflow:**

The project flows through a practical pipeline. Phase 1 generates, for 20 patients, synthetic vitals to simulate continuous health monitoring. Phase 2 links these vitals with past health data and calculates weekly averages by using AWS EMR with Spark. Phase 3 trains an XGBoost model with the merged dataset and uses it as an endpoint for inference. Phase 4 uses real-time predictions from the model to send alerts via AWS Lambda. This general workflow shows how data flows from generation through processing into model training and automated clinical alerting.

**Phase 1 Script:**

1. What is the main purpose of generating simulated\_vitals.csv in Phase 1, and how does this file mimic real patient monitoring data? The simulated\_vitals.csv file provides realistic daily vitals that resemble wearable or hospital monitoring data. It emulates real conditions to generate heart rate, blood pressure, and sleeping values within normal human ranges. This ensures that downstream processing and model training operate with data that behaves like true physiological measurements rather than artificial noise.
2. Why does the script assign random but realistic ranges for heart rate, blood pressure, and sleep hours? The script contains realistic ranges to introduce human variation but also avoid numbers that are medically implausible. This helps ensure that the patterns seen by Spark processing and the machine learning model are like those of real patient behaviors, but also helps maintain the relevance of the synthetic data for further analysis and predictions, making them more dependable and practical.
3. How many days of vitals are simulated per patient, and why is this important for calculating weekly averages in Phase 2? Each patient is simulated for seven days to create a full week of vitals. This is important because Phase 2 calculates weekly averages for heart rate, blood pressure, and sleep-all of which require complete seven-day data to provide stable, consistent, and clinically relevant aggregated features used during model training.

**Phase 2 Script:**

1. Describe how the Spark script aggregates the seven-day vitals for each Patient ID. It then groups the dataset by Patient ID and calculates the weekly average of heart rate, systolic and diastolic blood pressure, and sleep hours. This turns noisy daily data into higher-level health indicators that are clean and more representative of each patient's normal condition; thus, preparing the dataset for a merge with historical records.
2. The historical dataset (heart\_attack\_prediction\_dataset.csv) contains many more IDs than the simulated file. What join strategy (left, right, inner) is used, and how does it determine which patients appear in the final dataset? This is an inner join; only Patient IDs present in both the historical dataset and the simulated vitals appear in the final combined file. Only 20 patients have newly generated vitals, so these are the only IDs that survive the join and move forward into the model-training phase.
3. Explain how the combined “Blood Pressure” column is created and why it must later be split again in Phase 3. Spark temporarily combines the systolic and diastolic into one string for convenience. However, machine-learning models need numeric inputs, so Phase 3 subsequently splits this into separate numeric systolic and diastolic columns to ensure XGBoost gets the structured and correctly formatted features.
4. Why are we dropping – old vitals and risk column. These weekly averages replace the previous vitals, and having both would be redundant or contradictory data, so they are removed. The current risk column is removed to avoid data leakage; this ensures that the model will be trained only on real features and will not disclose any information from already generated outcome labels.

**Phase 3 Script:**

1. During preprocessing, what steps are performed to make the data compatible with XGBoost training? It then does some preprocessing, cleaning the data, handling missing values, encoding categorical fields, splitting the blood pressure column, and converting everything to a numeric format. The final dataset is saved in LibSVM format, which XGBoost will need. These steps help make sure that the inputs are consistent and avoid errors in model optimization and training.
2. What is saved in feature\_list.txt, and how is this file later used by Lambda in Phase 4? The feature\_list.txt file stores the exact ordering of features used during model training. Lambda loads this list when processing new vitals and arranges the incoming values in the same order as expected by the XGBoost endpoint. This alignment will ensure prediction accuracy by avoiding mismatched or incorrectly ordered inputs.
3. What metric (AUC) is used to evaluate the XGBoost model, and what does it indicate about model performance? AUC or Area Under the ROC Curve is a measure of how well the model differentiates high-risk from low-risk patients. A high value for AUC shows better discrimination ability and more reliable predictions. It is commonly used in medical ML due to the fact that it evaluates performance across different thresholds rather than at a single point.
4. How does the endpoint\_name generated in Phase 3 link directly to the inference step in the Lambda script? SageMaker creates an endpoint\_name uniquely when the model is deployed. The Lambda function refers to this exact name in its call to submit new vitals for real-time prediction. This tightly couples the deployment step with the alerting logic, ensuring that Lambda calls the proper active model endpoint.

**Phase 4:**

1. How do Phases 1 – 4 together represent a complete data-driven pipeline from data generation to real-time clinical alerts? Taken together, the four phases represent a realistic healthcare analytics pipeline. Phase 1 generates the vitals, Phase 2 aggregates them, Phase 3 trains and deploys a model, and Phase 4 uses the model to evaluate new data and send alerts. Together, they simulate automated end-to-end clinical decision support.
2. If model performance degrades over time, which phases would need to be revisited or retrained, and why? Phases 1, 2 and 3 need to be revisited. New vitals need to be generated or collected, aggregated again and used to retrain the model. Model drift occurs as patient behavior or population health changes over time, hence training data and model should be refreshed periodically.
3. In the current setup of the Heart Health Alert project, will alerts be generated only for the 20 simulated ids, if tour answer is "yes" then explain" why" ? Yes, alerts are only generated for the 20 simulated IDs because they are the only patients producing new vitals that reach the Lambda function. The alert system depends on incoming data, and since historical patients have no active vitals flowing through the pipeline, they never trigger real-time predictions or alerts.

**Figure 1**

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 2**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 3**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 4**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 5**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 6**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 7**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 8**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Video Link:**

Zoom Meeting: [Link](https://asu.zoom.us/rec/share/Qd8vRxW4W5_VYVq45XDCxGvbbFUF4NcwSJ02N7cDZhXCU9XAVqOd3q0XduEiqQ1g.1W7Xn-exapRrMOkF)

Passcode: akash3@3