Anomaly Detection in ECG Data:

Problem Statement: Develop an anomaly detection model to identify irregular heartbeats or arrhythmias in electrocardiogram (ECG) data. Early detection of cardiac anomalies can be crucial for timely intervention and improved patient outcomes.

Dataset: You can use the MIT-BIH Arrhythmia Dataset available on PhysioNet (<https://physionet.org/content/mitdb/1.0.0/>). This dataset contains 48 ECG recordings from 47 subjects, each containing two leads and annotated beat-by-beat by cardiologists. The dataset includes various types of arrhythmias and normal sinus rhythm.

Steps to approach the project:

1. Data Exploration: Load and visualize the ECG data to understand the different types of heartbeats and their characteristics. Familiarize yourself with the annotations provided by cardiologists.
2. Preprocessing: Preprocess the ECG data by filtering out noise, normalizing the signals, and segmenting the data into individual heartbeats. Ensure that the data is properly aligned and segmented for feature extraction.
3. Feature Extraction: Extract relevant features from the segmented ECG data that characterize the shape, amplitude, and duration of the heartbeats. You can use techniques like wavelet analysis, principal component analysis (PCA), or deep learning-based feature extraction.
4. Anomaly Detection Techniques: Experiment with different anomaly detection techniques to identify unusual heartbeats in the ECG data. Some methods to consider include Isolation Forest, One-Class SVM, or Local Outlier Factor. You can also explore deep learning techniques like autoencoders for unsupervised anomaly detection.
5. Model Evaluation: Evaluate your models using appropriate metrics, such as precision, recall, or F1-score. Perform cross-validation using different segments of the dataset to ensure the robustness of your model.
6. Web Application: Develop a web application where users can upload ECG data, and your model will analyze the signals and detect potential arrhythmias. Use Flask or Django for web application development.
7. Deployment: Deploy your web application on a cloud platform like Heroku, Google Cloud, or AWS to make it accessible to users.

Document your work in a Jupyter Notebook or a GitHub repository, detailing each step, and include a link to the deployed web application on your resume. This project demonstrates your ability to work with real-world medical data and apply machine learning techniques to a problem with significant clinical impact.