Multi-Modal Predictors for Cardiovascular Disease Risk and Outcomes

Project Deliverable 1
Author: Angel Morenu
University of Florida, M.S. in Applied Data Science
EEE 6778: Applied Machine Learning II (Fall 2025)

1. Problem Context and Project Summary

Nearly one in three deaths globally is attributable to cardiovascular disease (CVD), which continues to be the major cause of death. Even though survival rates have increased due to medical advancements, most risk-prediction algorithms' limited scope still prevents early detection and aggressive intervention. Conventional techniques frequently only use one data modality—physiological, clinical, or demographic—without combining their complementing advantages. The goal of this research is to create a multi-modal, hybrid machine learning framework that predicts cardiovascular risk and outcomes by combining physiological, clinical, and tabular data. In the end, the model will help clinicians make preventive decisions and empower patients through wearable technology by combining demographic and lifestyle variables, hospital admissions data, and ECG signals to enable more accurate, interpretable, and real-time prediction of CVD risk.

2. Dataset

This project integrates three publicly available datasets hosted on Kaggle, representing distinct but complementary modalities:

1. Cardiovascular Diseases Dataset (https://www.kaggle.com/datasets/mexwell/cardiovascular-diseases) – Tabular data containing demographic and lifestyle indicators such as age, gender, blood pressure, cholesterol, and BMI. This dataset is ideal for developing baseline classifiers and understanding clinical risk factors.
2. Hospital Admissions Data (https://www.kaggle.com/datasets/ashishsahani/hospital-admissions-data) – Structured clinical data with admission records, diagnoses, and treatment codes, used for longitudinal analysis of patient outcomes.
3. PTB-XL ECG Dataset (https://www.kaggle.com/datasets/khyeh0719/ptb-xl-dataset-reformatted) – A large collection of 12-lead ECG signals annotated with rhythm and diagnostic statements. This time-series data provides high-resolution physiological insight crucial for detecting abnormalities.

Data formats include CSV for tabular data and WFDB for ECG signals. Ethical considerations include ensuring de-identification of patient records, avoiding re-identification risks, and handling class imbalance responsibly to prevent biased model performance.

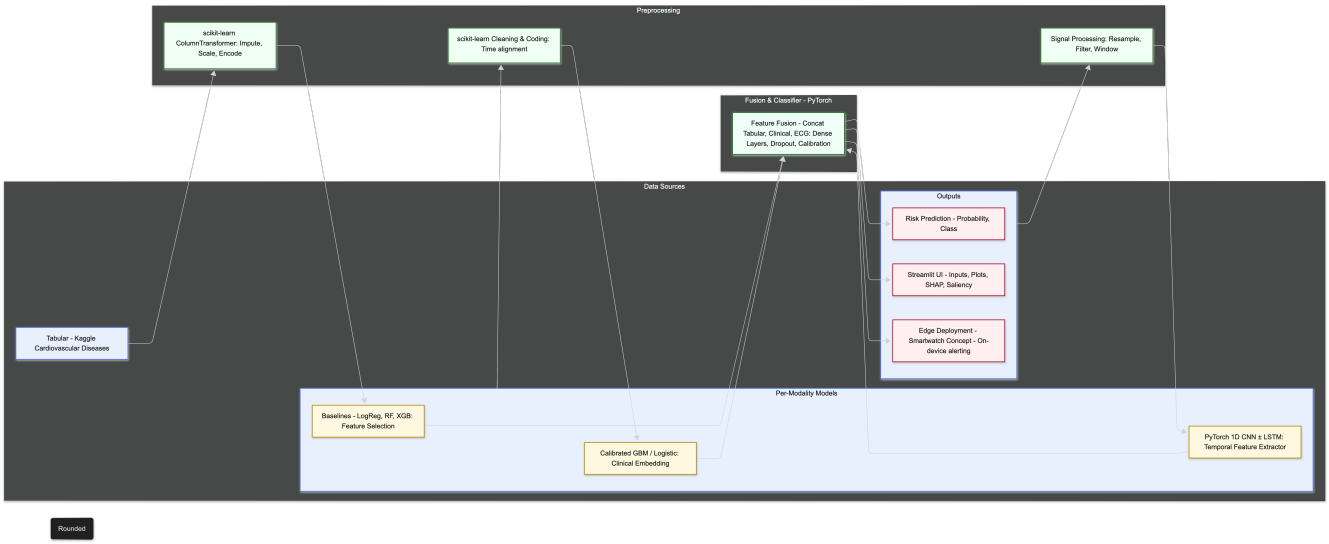
3. Planned Architecture

The system will employ a hybrid architecture that leverages both scikit-learn and PyTorch frameworks. The design follows a multi-stream data pipeline, where each modality is processed separately before feature fusion in a unified deep learning model.

• Tabular Stream (scikit-learn): Preprocessing with ColumnTransformer, feature scaling, and baseline modeling using Random Forests or Gradient Boosting.
• ECG Stream (PyTorch): 1D Convolutional Neural Network (CNN) for temporal feature extraction from ECG signals, followed by flattening into embeddings.
• Fusion Module (PyTorch): Concatenation of tabular and ECG embeddings into a fully connected fusion layer, followed by dense layers and a softmax classifier.
• Output: Binary or probabilistic prediction of CVD risk.

Model evaluation will rely on stratified 5-fold cross-validation using ROC-AUC, F1-score, and calibration metrics.

Figure 1 below illustrates the proposed system architecture and data flow from preprocessing to deployment.



4. User Interface Plan

A Streamlit-based interactive interface will serve as the user-facing component. The interface will allow clinicians or users to upload ECG data or input demographic parameters to receive instant predictions. Outputs will include predicted risk level (Low/Moderate/High), confidence scores, and visualizations of ECG signal attention maps. This design promotes interpretability by showing which features most influenced the prediction. A future iteration envisions deployment on a conceptual wearable ECG device, where the model operates on edge hardware to provide continuous monitoring and alerts.

5. Innovation and Anticipated Challenges

Within the framework of Applied Machine Learning II, this project presents a number of innovative and pedagogically significant developments.
In order to integrate diverse biomedical data—demographic, clinical, and physiological—into a single deep-learning framework, it first uses a multi-modal architecture. The majority of

cardiovascular-risk models only use tabular indicators, but this system combines structured and sequential modalities using a hybrid pipeline that combines PyTorch and scikit-learn. This allows each library to focus on its strengths, with PyTorch handling flexible representation learning and feature fusion and scikit-learn handling interpretable preprocessing and classical baselines.
Second, the study uses Edge AI deployment to investigate the possibility of running real-time inference directly on wearable devices with minimal power consumption, like smartwatches with ECG capabilities. This demonstrates how deep learning advancements can be translated into easily accessible, preventive healthcare solutions, balancing scientific rigor with societal relevance.
Finally, the addition of transparent variables—saliency maps for ECG signals and SHAP for tabular variables—improves model transparency, promoting clinical interpretability and ethical AI principles.

Three principal technical challenges are anticipated:

Data heterogeneity and alignment.
Each modality has distinct formats and sampling frequencies. To mitigate integration errors, standardized preprocessing pipelines and feature normalization will be implemented, with metadata logged for reproducibility.

Overfitting and limited generalization.
The fusion network may overfit small ECG subsets or class-imbalanced outcomes. Regularization (dropout, L2), early stopping, and cross-validation will be applied, along with synthetic augmentation and class weighting.

Interpretability versus complexity.
Deep architectures risk becoming opaque to clinicians. To balance accuracy and trust, SHAP and attention-based visualizations will be incorporated into the Streamlit interface so users can inspect which features drive predictions.

6. Implementation Timeline

Week (Dates)	Focus	Key Tasks	Outputs / Deliverables	Risks & Mitigations
(Oct 20–Oct 26)	Data audit & repo setup	Verify schemas, leakage, class balance; finalize target labels; stand up repo structure; create `environment.yml`; download small data samples	`setup.ipynb` runs; data dictionary; initial EDA plots; repo link live	Data gaps → log issues; freeze a "minimum viable subset" for rapid iteration
(Oct 27–Nov 2)	Preprocessing pipelines	Implement `src/preprocess.py` for tabular/clinical; ColumnTransformer + impute/scale/OHE; export train/val/test NPZ/CSV	Saved arrays in `data/processed/`; `artifacts/tabular_transformer.joblib`; README updated	Mismatch in columns → add column map & schema checks; save `*_meta.json`
(Nov 3–Nov 9)	Baselines + calibration	Train LR/RF/XGB on tabular; stratified CV; compute ROC-AUC, PR-AUC, F1, Brier; probability calibration (Platt/Isotonic)	`results/` metrics & curves; baseline report table; confusion matrices	Class imbalance → class weights & PR-AUC reporting
(Nov 10–Nov 16)	ECG pipeline (DL)	Implement 1D-CNN (±LSTM) in `src/model.py`; create ECG loaders; basic augmentation; early stopping	Saved ECG checkpoints; learning curves; example saliency/Grad-CAM on beats	Overfitting → dropout/L2, reduce capacity, early stop
(Nov 17–Nov 23)	Feature fusion	Concatenate tabular embedding + ECG embedding; train FusionNet; ablations (tabular-only vs ECG-only vs fused)	Fusion metrics table; ablation chart; model card draft	Feature scale drift → normalize/standardize consistently; freeze transformer
(Nov 24–Nov 30)	Explainability & robustness	SHAP on tabular; saliency on ECG; noise/missingness stress tests; subgroup fairness slices	Explainability figures; robustness table; fairness slice report	Holiday week time loss → pre-script SHAP/plots; limit scope to top features
(Dec 1–Dec 7)	Streamlit UI & polish	Wire UI inputs; model inference; probability bars; upload ECG; display SHAP/saliency; error handling	`streamlit run ui/app.py` demo; screenshots in `docs/`; short demo video (optional)	Dependency issues → pin env; provide CPU fallback

(Dec 8–Dec 11)	QA, documentation, hand-off	Reproduce results from clean clone; finalize README; finalize blueprint PDF; tag release	Final repo tag `v1.0`; zipped `results/`; submission on Canvas	Repro failures → add quickstart + seed; include sample data

7. Responsible AI Reflection

The architectural layout of this project is based on the concepts of responsible AI. By assessing model bias across demographic groupings, fairness will be addressed, and it will be made sure that risk forecasts do not prejudice any community. Open-source documentation and interpretable visuals will preserve transparency. By maximizing the effectiveness of model training and utilizing lightweight architectures for edge deployment, environmental sustainability will be taken into account. The overall goal of this study is to strike a balance between ethical responsibility and predictive power.