

Q1: Cardiovascular Disease Prediction Models

i. Logistic Regression vs Decision Tree

Logistic Regression Structure: Linear model that applies sigmoid function to weighted sum of features Predictions: Calculates probability using: $P(CVD) = 1/(1 + e^{-(\beta_0 + \beta_1x_1 + \dots + \beta_nx_n)})$ Interpretability: Highly interpretable - coefficients show direct feature impact; positive coefficient means feature increases CVD risk

Decision Tree Structure: Hierarchical tree with nodes splitting on feature thresholds (e.g., "cholesterol > 240?") Predictions: Follows decision rules from root to leaf; leaf node determines class probability Interpretability: Very intuitive - provides if-then rules humans naturally understand, though deep trees become complex

Key difference: Logistic regression assumes linear decision boundary; decision trees capture non-linear patterns through sequential splits.

ii. SVM Model Development Steps

a. Data Preprocessing (critical for SVM): Standardization: Scale all features to mean=0, std=1 (e.g., StandardScaler) - essential because SVM is distance-based Handle missing values, encode categorical variables (smoking status) Address class imbalance if present

b. Kernel Selection: Start with linear kernel for baseline Try RBF (Radial Basis Function) kernel for non-linear relationships Use cross-validation to compare kernel performance

c. Hyperparameter Tuning: Optimize C (regularization) and gamma (for RBF) using GridSearchCV Use stratified k-fold cross-validation

d. Evaluation Metrics: Primary: AUC-ROC (handles class imbalance, shows discrimination ability) Secondary: Precision, Recall, F1-score (understand false positives vs false negatives in medical context) Sensitivity (recall) particularly important - missing CVD cases is costly Calibration curve (predicted probabilities match actual risk)

iii. Model Enhancement Strategies

Strategy to enhance SVM without changing algorithm: Use RBF (Gaussian) kernel instead of linear kernel. RBF can map data to infinite-dimensional space, capturing complex non-linear relationships between risk factors (e.g., interaction between age, cholesterol, and blood pressure). Tune gamma parameter to control decision boundary complexity.

Why choose XGBoost instead:

Feature interactions: XGBoost automatically captures complex interactions (e.g., "high blood pressure AND smoking AND age>50" combined risk) without manual feature engineering. It also handles mixed feature types naturally, provides built-in feature importance rankings for clinical insights, and typically achieves superior performance on tabular healthcare data with proper tuning.

Q2: A health research team is analyzing patient data to predict the onset of hypertension (high blood pressure). The dataset includes features such as age, weight, sodium intake, stress levels,genetic markers, and physical activity.

a) K-NN vs Logistic Regression

K-Nearest Neighbors (K-NN):

- Learning approach: Non-parametric, instance-based learning - stores all training data, makes no assumptions about data distribution Prediction method: Classifies based on majority vote of K nearest neighbors in feature space
- Key assumptions: Similar patients (nearby in feature space) have similar outcomes; assumes locality matters
- Feature scale sensitivity: Highly sensitive - unscaled features dominate distance calculations (e.g., weight in kg vs stress on 1-10 scale). Standardization mandatory.

Logistic Regression:

- Learning approach: Parametric model - learns fixed coefficients that define linear decision boundary Prediction method: Applies sigmoid to linear combination: $P(\text{hypertension}) = \sigma(\beta_0 + \beta_1 \cdot \text{age} + \beta_2 \cdot \text{weight} + \dots)$
- Key assumptions: Linear relationship between log-odds and features; feature independence
- Feature scale sensitivity: Less sensitive - coefficients adjust to feature scales, but standardization still recommended for regularization and coefficient interpretation

Key contrast: K-NN memorizes patterns locally; Logistic Regression generalizes globally with learned parameters.

b) Random Forest Model Building Process

Steps:

1. Data Preparation: Handle missing values, encode categorical variables (genetic markers), standardize if desired (not required for trees) Train-Validation-Test Split (e.g., 60%-20%-20%):
2. Training set: Build forest of decision trees Validation set: Tune hyperparameters (n_estimators, max_depth, min_samples_split) via cross-validation
3. Test set: Final unbiased performance evaluation after all tuning complete Purpose: Prevents overfitting and data leakage; ensures model generalizes to unseen patients
4. Model Training: Build multiple decision trees on bootstrapped samples with random feature subsets at each split
5. Ensemble Performance Improvement:

Variance reduction: Individual trees overfit differently; averaging predictions reduces overall variance Bias-variance balance: Combines diverse weak learners into strong predictor Robust predictions: Outliers/noise affect only subset of trees

Feature Importance Value in Healthcare: Identifies which factors most strongly predict hypertension (e.g., "sodium intake and age are top predictors"). This provides: a. Clinical insights for targeted interventions b. Explainability for doctors to trust and understand model decisions c. Resource allocation - focus monitoring on high-impact modifiable risk factors (sodium, physical activity)

c) High Accuracy, Low Recall Problem

Practical Meaning: The model correctly classifies most patients overall (high accuracy) but misses many hypertension cases (low recall/sensitivity). For doctors, this means:

i. Dangerous false negatives: Many at-risk patients labeled as "no hypertension" won't receive preventive care ii. Model may be biased toward majority class (healthy patients) iii. Creates false sense of security - missed diagnoses lead to untreated hypertension complications

Technique to Improve Recall:

i. Adjust classification threshold: Lower the probability threshold from default 0.5 to ~0.3-0.4. This classifies more patients as "positive" (hypertension risk), increasing recall by catching more true cases at the cost of some false positives. In healthcare, conservative approach preferred - better to flag healthy patients for follow-up than miss actual disease cases. ii. Alternatively: Use class weights in training to penalize false negatives more heavily, making the model prioritize identifying positive cases.

-END-