Q&A Preparation Guide

Heart Disease Risk Assessment - CMPE 257
Prepared for presentation Q&A session (~2 minutes)

Quick Stats to Remember: Binary F1 = 85.3% (+13.7% vs target) | 3-Class F1 = 71.4% (above 55-65% published) | Improvement = +21.9% | Train samples = 736 | Test samples = 184 | Features = 18 (5 engineered)

Model Performance & Methodology

Q: Why did you choose a hierarchical approach instead of direct multi-class classification?

A: We chose hierarchical classification because it mirrors the clinical decision-making process. Doctors first determine if a patient has heart disease, then assess the severity. More importantly, binary classification is inherently easier - we achieved 85.3% F1 for binary versus only 71.41% for direct 3-class. By leveraging the strong binary classifier first, we achieve 21.9% improvement over the baseline. This approach also provides better probability calibration through Bayesian fusion of the two stages.

Q: Why is your severe class F1 so much lower than the other classes?

A: The severe class has only 135 samples total (27 in test), compared to 411 for no disease. Even with BorderlineSMOTE, there's inherently less signal to learn from. Additionally, the boundary between moderate and severe disease can be clinically ambiguous - the original classes 3 and 4 had significant overlap. Despite this, our model is conservative and tends to under-predict rather than over-predict severity, which is safer in a screening context. We can flag uncertain cases for specialist review.

Q: How did you handle the 66% missing data in your most predictive feature?

A: We used a three-pronged approach. First, KNN imputation with k=5 fills missing values based on similar patients. Second, we created missing indicator features (ca_missing, thal_missing) that capture the signal that these tests weren't ordered - often because the patient appeared healthy. Third, we preserved the original features so the imputed values can be weighted appropriately by the model. Interestingly, the missingness itself is informative - expensive tests are often skipped for low-risk patients.

Q: Why not use deep learning given the complexity of the problem?

A: Deep learning typically requires thousands to millions of samples to learn robust representations. Our dataset has only 920 samples. Traditional ML methods like SVM and Random Forest are actually better suited for this scale because they can learn effectively from limited data, are less prone to overfitting, and provide interpretable feature importance. Additionally, ensemble methods like Random Forest have proven very successful in tabular medical data.

Clinical Deployment & Validation

Q: Limitations?

A:

1. Not enough samples in the severe category. Even after grouping classes 3 and 4 together, we only have 135 severe patients total (27 in test set). Machine learning needs more

examples to learn patterns. This shows in the 48% F1-score for the Severe category compared to 83% for No Disease.
2. Missing data is another challenge. We handle 66% missing values with indicators and imputation. But we still lose the actual test results for most patients. This limits how much information the model can learn from. It might also create bias if missing data is related to things we did not measure.

Q: How would this system be validated before clinical deployment?

A: Clinical deployment would require multiple validation stages. First, external validation on independent datasets like Framingham or MIMIC-III to test generalizability. Second, prospective validation comparing model predictions against actual clinical outcomes over time. Third, a clinical trial comparing outcomes for patients screened with versus without our tool. We'd also need IRB approval and compliance with FDA software as medical device (SaMD) regulations. Our current results are promising for a screening tool, but not for diagnosis.

Q: What's the difference between a screening tool and a diagnostic tool?

A: A screening tool identifies patients who should receive further testing - it prioritizes sensitivity over specificity. A diagnostic tool provides a definitive answer. Our system is explicitly designed for screening and triage, not diagnosis. We recommend that all Mild and Severe predictions receive follow-up with clinical testing like echocardiography or coronary angiography. The medical disclaimer in our application makes this clear. The value is in identifying at-risk patients in resource-limited settings who might otherwise never get tested.

Q: How would you handle the liability if the model misclassifies a severe case as healthy?

A: This is a critical concern. First, we explicitly position this as a screening aid, not a replacement for clinical judgment - the final decision always rests with the healthcare provider. Second, our model has 87% recall for detecting any disease, meaning we miss relatively few sick patients. Third, our conservative predictions mean we're more likely to over-refer than under-refer. Fourth, proper clinical deployment would include clear documentation of model limitations, appropriate use cases, and integration into clinical workflows where physicians review all predictions. This is similar to how radiology AI is deployed - as decision support, not autonomous diagnosis.

Technical Implementation

Q: Why did you choose KNN with k=5 for missing value imputation? What does the k value affect?

A: K is the number of similar patients we look at to fill in missing values. When we have a patient with missing data, KNN finds the k most similar patients who DO have that data, then uses their values to fill in the gap.

For example, if a patient is missing their cholesterol value, KNN looks at the 5 most similar patients based on their other features - like age, blood pressure, chest pain type - and uses the average of their cholesterol values.

Five neighbors is a sweet spot. It's enough patients that we're not relying on just one potentially noisy example, but small enough that we're still finding truly similar patients. It's actually a very common default in machine learning - you'll see k=5 used in many applications.

Q: Why did you use StandardScaler for normalization?

A: StandardScaler makes all features have the same scale - mean of 0 and standard deviation of 1. This is crucial for our models, especially SVM.

Look at our features: age ranges from 29 to 77, blood pressure ranges from 94 to 200, but sex is just 0 or 1. If we don't scale, blood pressure, with its large range, would dominate the model's decisions just because the numbers are bigger, not because it's more important.

For example, in SVM, the algorithm calculates distances between data points. If blood pressure varies by 100 units and sex varies by 1 unit, blood pressure will completely dominate those distance calculations. StandardScaler fixes this by putting everything on the same scale.

Q: How does BorderlineSMOTE differ from regular SMOTE, and why did you use it?

A: Regular SMOTE generates synthetic samples randomly throughout the minority class feature space. BorderlineSMOTE focuses on samples near the decision boundary - these 'borderline' samples are the most informative for classification. It uses k-nearest neighbors to identify minority samples that have majority class neighbors, then synthesizes new samples only around these boundary cases. This is more effective for our imbalanced medical data because it reinforces the decision boundary rather than cluttering the interior of class regions.

Q: How do you handle categorical variables in your preprocessing?

A: We use label encoding for categorical features like chest pain type, ECG results, and thallium test results. This preserves ordinal relationships where they exist - for example, chest pain types have a natural severity ordering. For the SVM, we apply StandardScaler after encoding to normalize feature ranges. For Random Forest, encoding is sufficient since tree-based methods don't require scaling. We save all encoders and scalers in preprocessing_artifacts.pkl to ensure consistent transformation for new patients.

Feature Engineering & Interpretability

Q: Can you walk me through the engineered features you created?

A:

1. **age_group.** Instead of just using age as a number like 45 or 67, we put people into groups: young, middle-aged, senior, and elderly. We used the World Health Organization's standard categories. Why? Because heart disease risk doesn't go up smoothly with age. It jumps at certain points. A 45-year-old and a 55-year-old have similar ages, but doctors treat them very differently for heart disease risk. This helps the model see those jumps.
2. **blood pressure category.** The American Heart Association has specific cutoff points: normal is under 120, elevated is 120 to 129, Stage 1 high blood pressure is 130 to 139, and Stage 2 is 140 and above. The difference between 119 and 121 might seem small, but medically, it means you crossed from normal to elevated. We wanted the model to see these important thresholds that doctors use every day.
3. **cholesterol category.** Same idea - we made three groups: good is under 200, borderline is 200 to 239, and high is 240 and up. These are the standard medical cutoffs. Again, it's about helping the model understand the thresholds that matter clinically.
4. **heart rate reserve.** This one is a simple calculation: we take 220, subtract the person's age, then subtract their maximum heart rate during exercise. This number tells us how much extra capacity their heart has. Lower numbers mean better fitness. Cardiologists use this all the time to check cardiovascular health.
5. **cardiovascular risk score.** We combined several things into one score from 0 to 10: how old they are, how high their blood pressure is above 120, how high their cholesterol is above 200, if they have diabetes, and if exercise causes chest pain. Heart disease is never just one thing - it's usually many factors together. This score captures that.

Q: How did you come up with the cv_risk_score formula?

A: The cv_risk_score is a composite of established cardiovascular risk factors: age (normalized by 100), resting blood pressure (normalized by 200), cholesterol (normalized by 300), and ST

depression oldpeak (normalized by 10). These normalization factors bring each component to roughly the same scale. The formula is inspired by clinical risk calculators like the Framingham Risk Score, but simplified for our available features. The fact that it ranks #3 in feature importance validates that combining domain knowledge with data improves predictions.

Q: Why not use more sophisticated interpretability methods like SHAP?

A: SHAP values are definitely on our roadmap for future work. Currently, we use Random Forest feature importance, which provides aggregate importance scores but doesn't explain individual predictions. SHAP would allow us to show which features pushed a specific patient toward each severity level. This is crucial for clinical adoption - doctors need to understand why a model made a prediction to trust it and act on it.

Q: How confident are you in the feature importance rankings?

A: Random Forest feature importance based on Gini impurity is generally reliable but can be biased toward high-cardinality features. We validated our rankings by checking correlation with target (ca: 0.52, oldpeak: 0.44, matching our top features), by trying feature ablation (removing top features significantly hurts performance), and by comparing with published clinical knowledge (vessels blocked and ST depression are known predictors). For production, we'd use permutation importance or SHAP for more robust rankings.

Data & Generalizability

Q: How generalizable is your model to different populations?

A: Our dataset includes patients from four hospitals across the US and Europe, which provides some diversity. However, 79% of patients are male, and the age range is 29-77. Performance might differ for women, younger adults, or elderly patients. Before deployment in a new population, we'd recommend validation on local data. The advantage of using basic clinical features (not genetics or imaging) is that they're universally available, making the model more deployable in resource-limited settings.

Q: The dataset is from the 1980s - is it still relevant?

A: Great question. While diagnostic technology has advanced, the underlying pathophysiology of heart disease hasn't changed. The features we use - age, blood pressure, cholesterol, chest pain patterns, exercise tolerance - remain the core clinical assessment today. What has changed is treatment options and patient awareness. The value of our model is for initial screening in settings without advanced diagnostics, where these basic assessments are still the starting point. For a production system, we'd want to validate on more recent data.

Q: Why didn't you use larger, more recent datasets?

A: The UCI dataset is well-curated, publicly available, and has been extensively studied, making it ideal for an academic project with reproducible benchmarks. Larger datasets like MIMIC or UK Biobank require data use agreements and significant preprocessing. For a course project, UCI gave us meaningful challenges (missing data, imbalance) while being tractable. In a production setting, we would absolutely use larger, more recent, and more diverse datasets. We mention external validation on Framingham and MIMIC in our future work.