Predicting Heart Disease Using the UCI Heart Disease Dataset

DASC 5231 -- Machine Learning | Group 1 | Instructor: Dr. Zhu, L.

James Freeman Sara Ray Tim McGowan

Abstract

This paper uses the UCI Heart Disease Dataset to predict whether a patient is likely to have heart disease. Four models were tested: Logistic Regression, Random Forest, Support Vector Machine, and K-Nearest Neighbors. Each model was tuned using cross-validation, while the test set was kept separate until the final evaluation. SVM performed the best overall, with a test recall of 0.9412 and a ROC-AUC of 0.9091. In practical terms, it correctly identified about 94% of patients with heart disease. Additional checks, including calibration, learning, ROC, and ablation analyses, helped confirm how the model was performing and whether the main design choices were reasonable.

1. Introduction

Heart disease is one of the leading causes of mortality worldwide. Early and accurate detection is critical for enabling timely intervention and improving patient outcomes. Machine learning can help identify patterns in patient data that may be useful for early heart disease screening.

This project frames the problem as binary classification. The target variable -- originally coded 0 to 4 for disease severity -- was binarized so that any severity above zero is labeled Disease (1) and zero is No Disease (0). Four models were trained and evaluated with recall as the primary metric, reflecting the high cost of a missed diagnosis in a clinical screening context.

2. Dataset

The UCI Heart Disease Dataset was sourced from Kaggle (Karimsony, 2020). It contains 920 patient records across 16 attributes drawn from four clinical sites: Basel & Zurich, Switzerland, Budapest, Hungary, and Cleveland & Long Beach VA Medical Centers. This study uses the 14 core features most referenced in prior research, including age, sex, chest pain type, resting blood pressure, cholesterol, and maximum heart rate.

2.1 Class Distribution

The dataset shows a modest imbalance with approximately 55% Disease and 45% No Disease. Stratified sampling was applied during the train-test split to preserve this ratio across both sets, ensuring consistent class representation throughout training and evaluation.

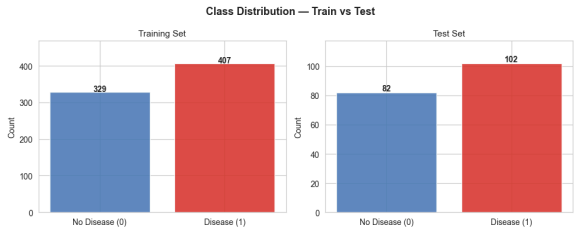

Figure 1. Class Distribution -- Train vs Test sets showing consistent class ratio preserved by stratified sampling.

Figure 1 confirms the stratification worked as intended. Training (329 vs 407) and test (82 vs 102) sets mirror the same ~55/45 Disease/No Disease ratio -- ensuring neither set is skewed toward one class during training or evaluation.

3. Data Preparation

3.1 Train-Test Split

The data was split 80/20 with stratification before any preprocessing was applied -- producing 736 training samples and 184 test samples. Splitting first is critical: it prevents test set information from influencing preprocessing decisions, which would constitute data leakage.

3.2 Missing Value Imputation

Numerical features (trestbps, chol, thalch, oldpeak) were imputed using the training set median. Categorical features (fbs, restecg, exang, slope, ca, thal) were imputed using the training set mode. All imputation statistics were derived from training data only -- preventing leakage.

It is worth noting that ca and thal each contained more than 50% missing values. Mode imputation may over-represent the most common category, and interpretations involving these features should be treated with appropriate caution. With further research, features *slope, ca,* and *thal* are more specialized tests that are not administered to all patients. As a result, the missingness is likely not random and may reflect clinical decision-making, which could lead to bias in the model.

3.3 One-Hot Encoding

All categorical columns were converted to numeric using one-hot encoding with drop_first=True to prevent multicollinearity. The training and test sets were aligned using pandas .align() to ensure identical column structure, filling any unseen test categories with zero.

4. Feature Selection and Analysis

Feature selection was approached using two independent methods -- Pearson correlation and Random Forest importance -- then cross-validated against each other. An agreement between the two methods gives the strongest confidence in a feature's predictive value. Disagreement flags features worth examining more carefully before committing to a model.

4.1 Correlation of All Features with Target

The chart below shows the Pearson correlation coefficient between every feature and the heart disease target variable, sorted by absolute strength. Red bars indicate a positive relationship (feature increases alongside disease risk), while blue bars indicate a negative one.

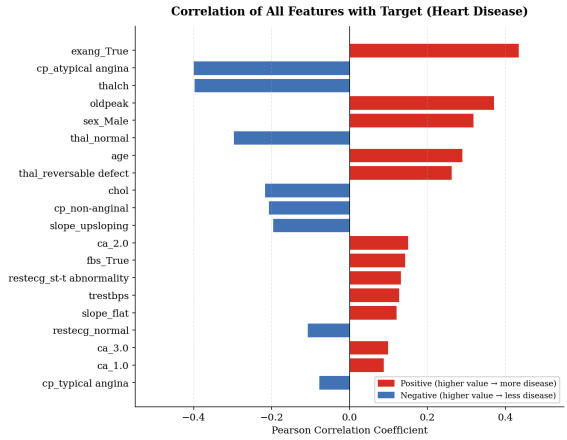

Figure 2. Pearson correlation of all features with the target variable. Sorted by absolute value -- strongest relationships appear at the top.

Key insight: exang_True leads with the strongest positive correlation (+0.43), meaning exercise-induced angina is the single most linearly predictive feature. thalch and cp_atypical angina are strong negative predictors -- higher maximum heart rate and atypical chest pain patterns are associated with lower disease risk.

4.2 Top 10 Feature Importances -- Random Forest

The Random Forest classifier ranked features by how much each one reduced prediction error across all decision trees. Unlike correlation, this method captures non-linear relationships and feature interactions -- giving a more complete picture of what the model actually learns.

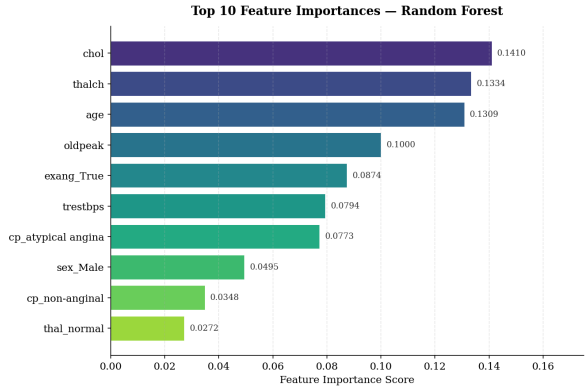

Figure 3. Top 10 features ranked by Random Forest importance score. Each bar represents the average impurity reduction across all trees.

Key insight: chol ranks #1 in Random Forest importance despite showing only weak negative correlation with the target. This is not a contradiction -- it suggests cholesterol has a complex, non-linear relationship with heart disease that trees can exploit but linear correlation cannot detect.

4.3 Interpreting the Two Methods Together

The chart below directly compares normalized scores from both methods for the seven most discussed features -- split into those where the methods agree and those where they diverge.

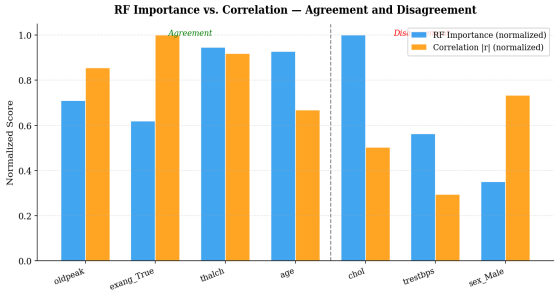

Figure 4. Normalized RF importance vs. absolute Pearson correlation for key features. Left of the divider: strong agreement. Right: meaningful disagreement between methods.

Where both methods agree -- oldpeak, exang_True, thalch, and age – gave us confidence that these are genuinely important predictors. Where they disagree -- chol, trestbps, and sex_Male -- the Random Forest is picking up on patterns that operate beyond a simple linear relationship.

This difference matters when choosing which model to use. Logistic Regression, being a linear model, will align more closely with the correlation results. Non-linear models like Random Forest, SVM with an RBF kernel, and KNN are better positioned to leverage the full set of top 10 features -- particularly chol and trestbps. Utilizing this result, the top 10 features were selected as a balance between model simplicity and predictive performance, which is later supported by the ablation study results.

Rank	Feature	RF Importance	Correlation |r|	Agreement
1	chol	0.1410	0.218	Disagree
2	thalch	0.1334	0.399	Agree
3	age	0.1309	0.290	Agree
4	oldpeak	0.1000	0.371	Agree
5	exang_True	0.0874	0.434	Agree
6	trestbps	0.0794	0.128	Disagree
7	cp_atypical angina	0.0773	0.401	Agree
8	sex_Male	0.0495	0.318	Partial
9	cp_non-anginal	0.0348	0.208	Agree
10	thal_normal	0.0272	0.298	Agree

Table 1. Side-by-side comparison of RF importance and Pearson correlation for the top 10 features.

5. Modeling Approach

All four models were wrapped in a scikit-learn Pipeline combining StandardScaler with the classifier. This ensures scaling is applied within each cross-validation fold -- preventing the scaler from seeing validation data. Hyperparameters were tuned via GridSearchCV with 5-fold cross-validation scored on ROC-AUC. The test set was not consulted at any stage of model selection.

5.1 Logistic Regression (Baseline)

Logistic Regression was chosen as the baseline for its interpretability. Hyperparameters tuned: C (0.001, 0.01, 0.1, 1, 10, 100) and penalty (L1, L2). Total: 12 combinations x 5 folds = 60 training runs.

5.2 Random Forest

Random Forest combines multiple decision trees to capture non-linear relationships. Hyperparameters tuned: n_estimators (100, 200), max_depth (None, 5, 10), min_samples_split (2, 5), min_samples_leaf (1, 2). Total: 120 training runs.

5.3 Support Vector Machine

SVM maximizes the margin between classes, using RBF kernel for non-linear boundaries. probability=True was set to enable ROC-AUC scoring via Platt scaling. Total: 9 combinations x 5 folds = 45 training runs.

5.4 K-Nearest Neighbors

KNN classifies by proximity to training points. Odd k values prevent tied votes. Hyperparameters tuned: n_neighbors (3,5,7,9,11), weights (uniform, distance), metric (euclidean, manhattan). Total: 20 combinations x 5 folds = 100 runs.

6. Evaluation Strategy

Model selection followed a strict two-level methodology. Within each model, the best hyperparameters were found via 5-fold CV on training data only. Across models, comparison was based entirely on training-side CV metrics -- the test set was never consulted during selection. The winning model was retrained on all 736 training samples, then evaluated on the held-out test set exactly once.

Recall for the Disease class was selected as the primary metric. In a clinical screening context, a missed diagnosis (false negative) carries a far greater cost than a false positive. Therefore, ROC-AUC, F1-score, and accuracy were included as supporting metrics to provide a more complete evaluation.

7. Results

7.1 Cross-Validation Model Comparison

Model	CV Recall	CV ROC-AUC	CV F1	CV Accuracy
SVM (Selected)	0.8943	0.8768	0.8418	0.8138
Random Forest	0.8721	0.8785	0.8424	0.8193
KNN	0.8351	0.8639	0.8295	0.8111
Logistic Reg.	0.8279	0.8773	0.8247	0.8057

Table 2. CV results -- training data only. Sorted by primary metric (Recall).

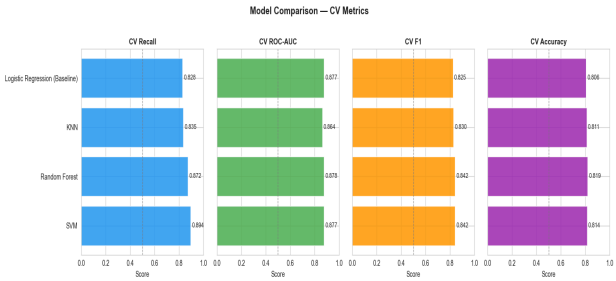

Figure 5. Model Comparison -- CV Metrics across all four models. Sorted by recall (primary metric). Y-axis labels shown on the leftmost panel only.

Figure 5 shows these results side by side across all four metrics. The CV Recall panel gives the clearest comparison: SVM has the highest bar, and this is where the differences between the models stand out the most. The ROC-AUC and F1 panels show that Random Forest was a strong second-place model, while Logistic Regression still performed well as the simplest baseline in the group.

SVM performed very well overall, ranking first in recall, tying for first in F1, and finishing second in both ROC-AUC and accuracy. All four models performed similarly, which suggests that the selected features were useful predictors no matter which model was used.

7.2 Final Test Set Evaluation

SVM was retrained on all 736 training samples using the best hyperparameters: kernel=rbf, C=0.1, gamma=scale. The held-out test set was used for the first and only time.

Metric	CV Estimate	Final Test	Change
Recall	0.8943	0.9412	+0.0469
ROC-AUC	0.8768	0.9091	+0.0323
Accuracy	0.8138	0.8424	+0.0286

Table 3. CV estimate vs. final test performance -- SVM.

The final model outperformed its CV estimates across all three metrics. Retraining on the full training set provided approximately 25% more data than any individual CV fold, contributing to the improvement and suggesting the model generalizes well with no strong evidence of overfitting. This is further supported by the close alignment between cross-validation and test performance, indicating that the model generalizes well to unseen data.

7.3 Confusion Matrix

	Predicted: No Disease	Predicted: Disease
Actual: No Disease	TN = 59	FP = 23
Actual: Disease	FN = 6	TP = 96

Table 4. Confusion matrix -- SVM final model on held-out test set.

Of 102 patients who actually had heart disease, 96 were correctly identified -- a recall of 0.9412. Only 6 patients slipped through undetected. The 23 false positives represent healthy patients flagged for further evaluation -- an acceptable tradeoff in screening.

8. Bonus Analyses

8.1 Class Imbalance and Fairness

The 55/45 split was addressed using stratify=y at the split stage. At this imbalance level, resampling techniques such as SMOTE were not required -- confirmed by the final Disease recall of 0.9412.

See Figure 1

8.2 Calibration Curve

The calibration curve suggests that the SVM's predicted probabilities are fairly reliable. SVC with probability=True applies Platt scaling internally. Points near the diagonal suggest that the model's confidence scores are reasonably close to what we would expect -- meaningful in healthcare, where

clinicians may use probability estimates to prioritize follow-up care.

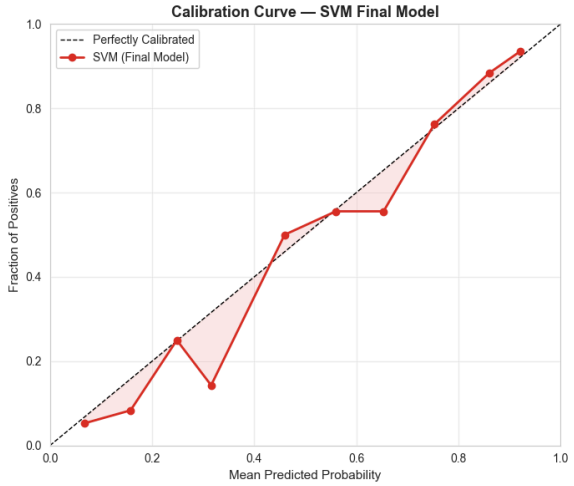

Figure 6. Calibration Curve -- SVM Final Model. The dashed diagonal represents perfect calibration. Points close to the line indicate trustworthy probability estimates.

Looking at Figure 6, the SVM curve tracks reasonably close to the diagonal across most of the probability range. There is some waviness in the 0.2-0.4 range -- a known limitation of Platt scaling on smaller datasets. Overall, the probability estimates seem useful, though they should still be interpreted with caution in a healthcare setting.

8.3 ROC Curve

The SVM achieved AUC = 0.9091, correctly ranking a Disease patient above a No Disease patient 90.9% of the time. The curve hugs the top-left corner, confirming strong discriminative ability well above the 0.50 random baseline.

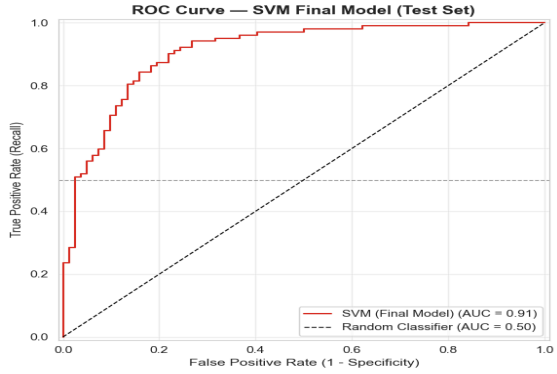

Figure 7. ROC Curve — SVM Final Model on held-out test set. AUC = 0.91, well above the 0.50 random classifier baseline.

Figure 7 makes this visual. The red curve shoots steeply upward in the low false positive rate region — meaning the model catches most Disease cases while flagging very few healthy patients as at-risk. This pattern is useful in a screening setting because the model identifies many disease cases early while keeping false positives relatively low.

8.4 Learning Curve

Training and validation recall curves converged at a high score with a small gap -- indicating the model generalizes well and is not significantly overfitting.

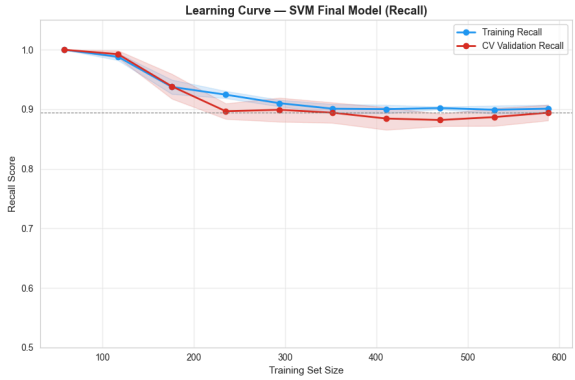

Figure 8. Learning Curve — SVM Final Model scored on Recall. Shaded bands show one standard deviation across CV folds. Curves converge cleanly, confirming no significant overfitting.

Figure 8 shows an important pattern. Both training and validation recall start near 1.0 with very few samples, which is expected since a small training set is easy to memorize. As training size increases, training recall drops toward ~0.90 while validation recall stabilizes in the same range. Since the two curves come together around 0.90, the model does not appear to be overfitting too badly. The final test recall of 0.9412 lands above this plateau — consistent with giving the model the full 736-sample training set rather than a CV fold.

8.5 Ablation Study

SVM trained on all 20 features achieved CV recall of 0.9286 vs. 0.8943 for the top 10 -- a difference of 0.0343. The top 10 model trades a small amount of recall for a simpler, more interpretable pipeline. The ablation study helps show what was gained and lost by using only the top 10 features.

9. Limitations and Future Work

Binarizing the target removes information about disease severity. Features ca and thal had over 50% missing values and were imputed with the mode, which may introduce bias. Feature selection was performed before cross-validation rather than within each fold, which may introduce slight optimistic bias in performance estimates. Future work could explore

multiple imputation strategies, test additional architectures, and investigate fairness metrics across demographic subgroups within the dataset.

10. Conclusion

This project demonstrated that machine learning can effectively predict heart disease risk using clinical features with reasonably strong performance. Among the four models tested, SVM performed the best overall and achieved a final recall of 0.9412 on the held-out test set. This means the model correctly identified most patients with heart disease. In a real-world screening context, this is especially important, since failing to detect at-risk patients can have serious consequences.

Beyond model performance, this project highlights the importance of using a careful workflow. Splitting the data before preprocessing, tuning models with cross-validation, and saving the test set for the final evaluation helped reduce the risk of data leakage. Although the model performed well, the results should still be viewed with caution because some features had many missing values, and the target variable was simplified into a binary outcome. Future work could test other imputation methods, including disease severity, and examine model fairness across different patient groups.

Overall, this project shows that machine learning can be a valuable tool for early heart disease screening, with potential for clinical decision-making and improving patient outcomes when applied carefully and responsibly.

References

Karimsony, R. (2020). Heart Disease Data. Kaggle. kaggle.com/datasets/redwankarimsony/heart-disease-data

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

Detrano, R., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. American Journal of Cardiology, 64(5), 304-310.