Group 3 Final Report:
Heart Disease Prediction using Machine Learning

ZhiChong Lin, ZiDi Yao, Ke Ma
yaoz25@mcmaster.ca, mak11@mcmaster.ca, lin281@mcmaster.ca

1 Introduction

This project aims to use ML method to predict the likelihood of heart disease based on eleven clinical and demographic features. Heart disease continues to be a major global health concern, and early detection is crucial for reducing severe outcomes. As outlined in our project proposal, the goal of this work is to develop a reliable classification model that can identify high-risk patients using routinely collected medical measurements.

In this progress report, we will discuss the steps completed so far, including dataset preprocessing, feature encoding, and the development of a Chi-Square + PCA feature-engineering pipeline. We also present initial results from models such as SVM, Logistic Regression, and Random Forest, and outline feedback from the TA along with our planned next steps.

Building on these prior studies, our project uses a supervised learning pipeline based on Principal Component Analysis (PCA) and an RBF-kernel SVM classifier. We preprocess the data with a column transformer that standardizes numerical features and applies one-hot encoding to categorical features, then apply PCA and keep the number of components according to the Kaiser criterion [6]. The resulting low-dimensional representations are fed into an RBF-SVM, and the full pipeline is evaluated on the Kaggle Heart Failure Prediction dataset [4] using stratified 5-fold cross validation with standard metrics (accuracy, precision, recall, F1-score, and ROC–AUC).

2 Related Work

Our progress in this project builds directly on the foundation built in Milestone01, where we considered both the clinical motivation and the technical approaches. In that proposal, we discussed the limitations of traditional diagnostic methods and highlighted the importance of ML models that can identify multi-feature interactions such as age, cholesterol level, chest pain type, and ECG-related measurements, which observations align with prior work showing that heart disease prediction is well suited to ML due to its multivariate and non-linear nature.

As we talked about in M1, several existing studies have explored this predictive task using classical and modern machine learning techniques. Logistic regression remains a common baseline method, as demonstrated by Awan [3], who achieved approximately 85% accuracy on the Kaggle using minimal feature engineering. More advanced pipelines integrate supervised feature selection and dimensionality reduction. The Chi-Square + PCA framework proposed by Gárate-Escamila et al. [5] achieved up to 99% accuracy on multiple UCI heart disease datasets, motivating our decision in Milestone 1 to incorporate both Chi-Square filtering and PCA into our own preprocessing pipeline.

Moreover, foundational statistic work such as Kaiser factor analysis criterion [6] provides justification for retaining only principal components with eigenvalues greater than one, a rule we apply in our dimensionality

reduction stage. Other work has examined optimization strategies for medical classification models, including scalable L1-regularized training [2] and multi-task predictive structure learning [1], which highlight broader approaches to improving generalization when datasets are small—one of the challenges noted in our proposal.

3 Dataset and Preprocessing

This section will introduce the raw dataset we used, and how we clean and process it.

3.1 Dataset Description

The dataset we used in this project is the *Heart Failure Prediction Dataset* published by Fedesoriano on Kaggle [4]. It contains **918 patient observations** and **12 attributes**, including 11 clinical predictor variables and one binary target label indicating the presence of heart disease. This dataset was designed to support research on early detection of cardiovascular risks, particularly heart failure, which remains one of the leading causes of global mortality.

The dataset includes a mixture of demographic features (Age, Sex), physiological measurements (like RestingBP, Cholesterol, MaxHR), and exercise-induced ECG-related metrics (ExerciseAngina, Oldpeak, ST_Slope). Those attributes show common risk factors used in medical diagnostics for cardiovascular disease and have been widely adopted in machine learning models for clinical prediction tasks.

A complete list of raw features and their corresponding descriptions is provided in Table 2.

Feature	**Description**
Age	Age of patient (years)
Sex	Biological sex (M/F)
ChestPainType	Chest pain type (ATA, NAP, ASY, TA)
RestingBP	Resting blood pressure (mm Hg)
Cholesterol	Serum cholesterol (mg/dL)
FastingBS	Fasting blood sugar (0/1)
RestingECG	Resting ECG results (Normal, ST, LVH)
MaxHR	Maximum heart rate achieved
ExerciseAngina	Exercise-induced angina (Y/N)
Oldpeak	ST depression value induced by exercise
ST_Slope	Slope of ST segment (Up, Flat, Down)
HeartDisease	Target label (1 = disease, 0 = healthy)

Table 1: Raw dataset features.

Originally, the dataset contains:

$$\textbf{918 samples} \quad \times \quad \textbf{11 features}.$$

3.2 Feature Preprocessing

The feature matrix X was constructed by removing the target column and retaining the remaining 18 input attributes. Since the dataset includes both numerical and categorical variables, several preprocessing steps were required to convert all values into a machine-learning- ready numeric form.

Binary Encoding Three features, namely Sex, ExerciseAngina, and FastingBS contain only two possible values and were mapped directly to 0/1 following our preprocessing script:

- **Sex:** `M` $\rightarrow 1$, `F` $\rightarrow 0$
- **ExerciseAngina:** `Y` $\rightarrow 1$, `N` $\rightarrow 0$
- **FastingBS:** preserved as integer `0/1`

Ordinal Mapping of Multi-Class Features Three categorical features contain more than two categories. In the stored processed dataset (`X_encoded.csv`), they were converted to integer codes according to predefined mappings:

$$\text{ChestPainType: } \{\texttt{ATA}, \texttt{NAP}, \texttt{ASY}, \texttt{TA}\} \rightarrow \{0, 1, 2, 3\},$$

$$\text{RestingECG: } \{\texttt{Normal}, \texttt{ST}, \texttt{LVH}\} \rightarrow \{0, 1, 2\},$$

$$\text{ST\_Slope: } \{\texttt{Up}, \texttt{Flat}, \texttt{Down}\} \rightarrow \{0, 1, 2\}.$$

These mappings avoid string-based ambiguity and ensure that all feature columns are numeric at the preprocessing stage.

In contrast from progress report, we adopted feedback from TA to change our way of presenting One-Hot Encoding, instead of present them in one single column to split them across the columns for every single category.

Therefore, after spliting them into one-hot encoding, the final dataset shape changes to 918 x 18.

Feature	Description
Age	Age of patient (years)
Sex	Biological sex (M/F)
ChestPainType_0	ATA
ChestPainType_1	NAP
ChestPainType_2	ASY
ChestPainType_3	TA
RestingBP	Resting blood pressure (mm Hg)
Cholesterol	Serum cholesterol (mg/dL)
FastingBS	Fasting blood sugar (0/1)
RestingECG_0	Normal
RestingECG_1	ST
RestingECG_2	LVH
MaxHR	Maximum heart rate achieved
ExerciseAngina	Exercise-induced angina (Y/N)
Oldpeak	ST depression value induced by exercise
ST_Slope_0	Up
ST_Slope_1	Flat
ST_Slope_2	Down
HeartDisease	Target label (1 = disease, 0 = healthy)

Table 2: One-Hot Encoding Table.

3.3 Target Label Extraction

The target label `HeartDisease` is a binary indicator representing whether a patient shows signs of heart disease. We extract this column and converted it to integer form using:

$$y = \texttt{df['HeartDisease'].astype(int).}$$

The resulting is a one-dimensional vector , which was saved as `processed/y.csv` for all downstream training and evaluation.

The cleaned dataset was saved to `processed/X_encoded.csv`, and the corresponding feature names were exported to `processed/feature_names.txt` for reproducibility.

4 Model Inputs (Features)

During our data preprocessing phase, we use `scikit-learn` `Pipeline` and `ColumnTransformer` to transform the raw attributes before training. Numerical features (Age, RestingBP, Cholesterol, MaxHR, Oldpeak, FastingBS) are imputed with the median and standardized, while categorical features (Sex, ChestPainType, RestingECG, ExerciseAngina, ST_Slope) are converted into one-hot vectors. After this step each patient is represented by an 18-dimensional feature vector, matching the one-hot layout described in Table 2.

Motivated by our progress report and TA feedback, in the final system we use PCA as our only feature-engineering method, instead of the Chi-square + PCA combination proposed in [5].

- **PCA.** We apply Principal Component Analysis (PCA) to the standardized and one-hot encoded features to reduce dimensionality. PCA finds orthogonal directions (principal components) that capture the largest variance in the data and projects each sample onto these directions. To choose the number of components we adopt the Kaiser criterion [6], keeping only components with eigenvalues greater than 1. In our experiments the first five components satisfy this rule, so we retain five principal components for the final representation.

Hence, each patient sample is represented as a compact 5-dimensional feature vector summarizing the most informative physiological and categorical characteristics, which is then used as input to our machine learning models.

5 Model Implementation

We have implemented a supervised learning pipeline for binary classification of heart disease presence. Our main model is a **Support Vector Machine (SVM)** with a **Radial Basis Function (RBF)** kernel, implemented using the `scikit-learn` library. This kernel choice allows the decision boundary to be nonlinear, which is important given the heterogeneous mixture of categorical and numerical medical features in the dataset.

Loss Function:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b))$$

where C is the penalty parameter controlling the trade-off between the margin size and misclassification tolerance, and $\phi(\cdot)$ denotes the nonlinear mapping induced by the **RBF Kernel:**

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

The model was optimized using the `libsvm` implementation, which employs a coordinate descent solver with kernel caching for efficiency.

With SVM-RBF, we evaluate it's accuracy by using cross-validation, and we was only able to achieve 87% of accuracy.

6 Process

In our project proposal we planned to follow the Chi-square + PCA + RBF–SVM framework from Gárate-Escamila et al. [5]. In the early stage of the project (before the progress report), we implemented a first version of this pipeline: nominal features such as `ChestPainType`, `RestingECG`, and `ST_Slope` were encoded as integers (e.g., ATA $\rightarrow$ 0, NAP $\rightarrow$ 1, ASY $\rightarrow$ 2), and we evaluated an RBF–SVM using a simple 80%/20% train–test split. This gave us an initial accuracy of about 85%, which confirmed that the approach was reasonable but left room for improvement.

In the progress report, the TA gave us two concrete suggestions: (1) replace label encoding of multi-class categorical variables with one-hot encoding, and (2) experiment with a different models like a neural network. After that feedback we revised our preprocessing: we switched to one-hot encoding (either via `pandas.get_dummies` or `OneHotEncoder` in a `ColumnTransformer`), which expanded the feature space to 18 columns and removed the artificial ordering implied by the integer codes.

Our original plan to strictly reproduce a full Chi-square + PCA pipeline changed slightly as we test more with the Kaggle dataset. Because this dataset has only 11 original features and relatively clean structure, we found that PCA alone already captures most of the variance, and chi-square filtering did not noticeably increase performance. So, we simplified the feature engineering stage to focus on PCA and foces our comparison on three models: Neural Network (as a simple baseline), RBF–SVM (our main model), and LightGBM + SHAP (The method found online). Overall, we followed the spirit of our original plan (PCA + SVM with additional baselines), but refined the details of preprocessing and evaluation in response to TA feedback and empirical results.

7 Evaluation Strategy and Results (Error Anaysis) and Baseline

In turns of method evalution, we first split the dataset into training and validaion. Training dataset with 80% of the data, while validation dataset with 20% of the data.

7.1 Evaluation Method

In our evaluation stage, we use stratified K-fold cross-validation. The reason for this choice is that our dataset contains only 918 patient records, and using a fixed train/validation/test split may lead to overfitting due to the small number of data points.

7.2 Metrics

SVM with RBF Kernel Model Performance :

Same as progress report, we have in total six metrics to evaluate our model performance. The metrics used in this project include **accuracy, recall, precision, F1-score, and AUC–ROC and Confusion Matrix**. Our rationale for choosing these metrics is as follows. First, in disease prediction, precision (closely related to Type II error) is especially important. Hospitals cannot afford to misclassify a patient with the disease as healthy. Therefore, minimizing false negatives is crucial. In addition, we use the AUC–ROC curve because it is an effective way to evaluate the overall performance of a binary classifier.

7.3 Results

The figures are listed in the last page, please check them out

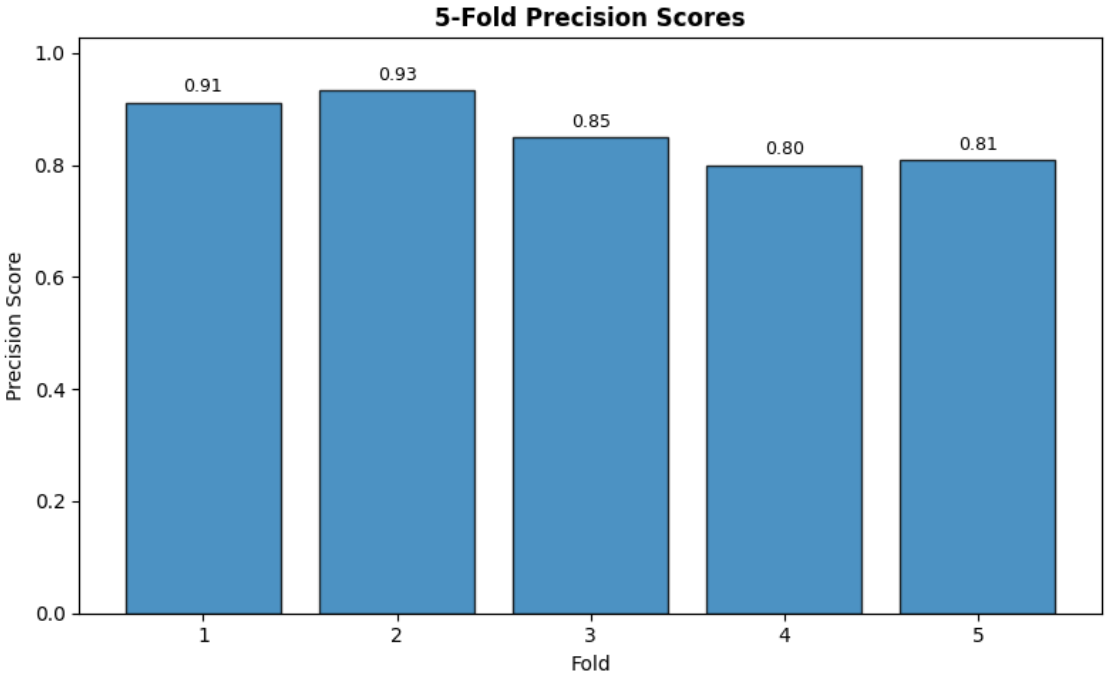

Figure 1: 5-fold precision scores.

```
===== K-Fold Cross Validation Results =====
Accuracy : 0.8627
Recall   : 0.9001
Precision: 0.8610
F1-score : 0.8785
============================================
```

Figure 2: Summary metrics across folds.

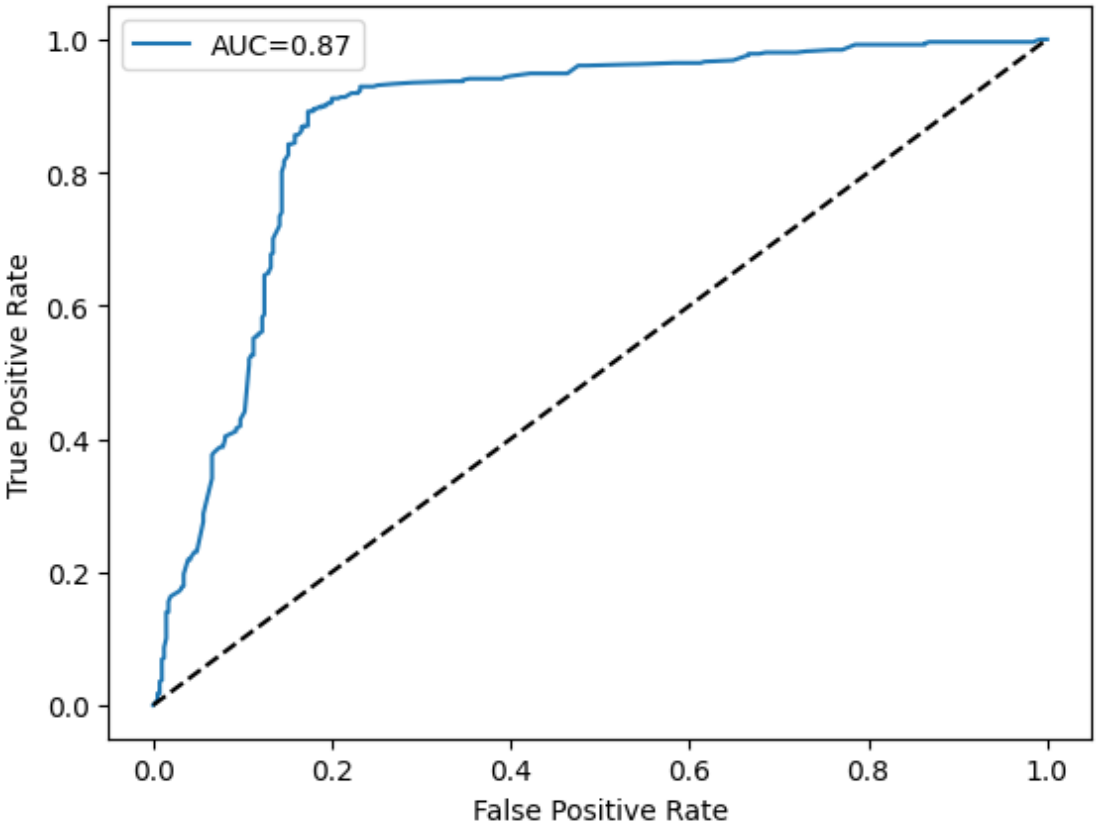

Figure 3: AUC–ROC curves.

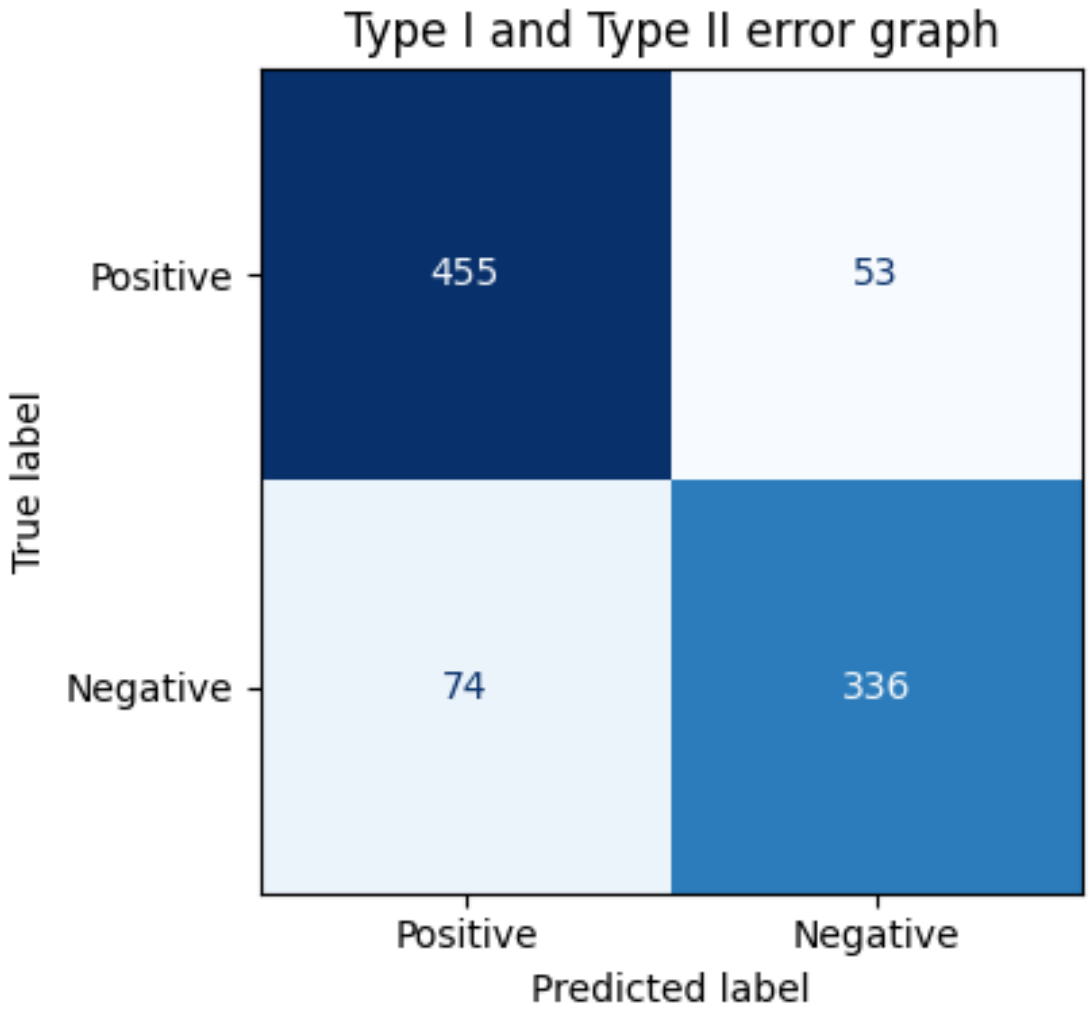

Figure 4: Confusion matrices.

Team Contributions

Jeffrey Lin:
ZiDi Yao:
Ke Ma:

References

[1] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, December 2005.

[2] Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, 2007.

[3] M.Usman Aslam Awan. Heart disease prediction using logistic regression. Kaggle Notebook, 2020. Online; accessed: 2025-11-11.

[4] Fedesoriano. Heart failure prediction dataset. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction, 2020. Accessed 2025-11-14.

[5] Anna Karen Gárate-Escamila, Amir Hajjam El Hassani, and Emmanuel Andrès. Classification models for heart disease prediction using feature selection and pca. *Informatics in Medicine Unlocked*, 19:100330, 2020.

[6] Henry F. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20(1):141–151, 1960. Original work published 1960.