Group X Progress 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 progress report summarizes the work completed so far for our heart disease prediction project. Our focus in this stage is to perform initial data exploration, clean and preprocess the dataset, and evaluate a set of baseline machine learning models. We document the steps taken to encode the raw clinical features, apply feature selection and dimensionality reduction techniques, and train several models using cross-validation. The report also talks about the feedback received from the TA and outlines the adjustments we plan to do in the next phase.

2 Related Work

This section summarizes the most relevant previous work. If no identical problem exists, describe the most similar tasks such as: – Medical risk prediction – Heart disease datasets – Classic ML models like logistic regression / SVM in healthcare Cite at least five references (use custom.bib). Length: 0.25–0.5 pages.

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 used in this project is the *Heart Failure Prediction Dataset* published by Fedesoriano on Kaggle [2]. 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 1.

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.

3.2 Target 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.

3.3 Feature Preprocessing

The feature matrix X was constructed by removing the target column and retaining the remaining 11 raw 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:** $\texttt{M} \to 1,\ \texttt{F} \to 0$
- **ExerciseAngina:** $\texttt{Y} \to 1,\ \texttt{N} \to 0$
- **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:

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

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

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

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

3.4 Final Processed Dataset

After applying the above processing , the final processed feature matrix contains:

918 samples × **11 fully numeric features**.

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

This processed dataset serves as the input to the feature selection (Chi-square) and dimensionality reduction (PCA) procedures described in the next section.

4 Model Inputs (Features)

During our data preprocessing phase, we utilize `scikit-learn` built-in function `pipeline` to transform our data before feeding into Machine Learning Model The dataset has both numerical and categorical features,

in the previous part we mentioned that for each numerical part we standardized them, while for categorical part we used one-hot encoding to transform them into vectors. In result, we have total 22 features after

preprocessing, including 6 numerical features and 16 categorical features "(One-Hot Encoding)". We then

went ahead to further transform our data by using both features selection and dimensionality reduction techniques. Which is so called Chi-PCA method [3].

- **Chi-Square:** Since in Mathematically, the Chi-square statistic is defined as:

  $$\chi^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

  , where O_{ij} is the observed frequency and E_{ij} is the expected frequency.

  Therefore, It is expecting a positive value for frequency, so we use `MinMaxScaler` to scale all numerical features to [0,1] range. So that all numerical features and categorical features are non-negative.

  Then we have a Hyperparameters 'k' to select top k features with highest Chi-square statistic with respect to the target label. In here we set k=9, as we found that 9 is the best parameter after conducting grid-search.

- **PCA:** After feature selection, we then applied Principal Component Analysis (PCA) to reduce the dimensionality of the selected features. PCA works by identifying the directions (principal components) in which the data varies the most, and projecting the data onto these directions.

  To determine the number of principal components to retain, we adopt the **Kaiser criterion** [4]. This rule suggests keeping only components with eigenvalues greater than 1.0.

After doing a experiments of calculating eigenvalues for every single increase of principal components, we found that the first five components have eigenvalues greater than 1.0. Thus, we decided to retain five principal components for our final feature representation.

Hence, each patient sample is represented as a compact (data, 5) feature vector summarizing the most informative physiological and categorical characteristics. This final feature set is then used as input to our machine learning model.

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.

To evaluate our implementation, we compared SVM–RBF against two baseline models:

- **Logistic Regression:** This baseline model was chosen from kaggle [1], where the author didn't use features selection or dimensionality reduction, and he was able to achieve 85% accuracy.

- **Random Forest:** From this paper [3], the author has 98% accuracy by using Random Forest with Chi-PCA method. However, he used a 74 features and around 1000 datapoint of heartdiease dataset from UCL.

With SVM-RBF, we evaluate it's accuracy by using cross-validation, and we was only able to achieve 86% of accuracy.

We then use Random Forest and Logistic Regression as our model, Randomforest was able to achieve 87% accuracy, while Logistic Regression was able to achieve 85% accuracy.

Varies reason can be introduces in here, such as different dataset, the quality of dataset, or minor changes in the preprocessing phase that effect the data values meaning.

In future iterations, we plan to explore a **neural network architecture** (e.g., a multi-layer perceptron) to capture more complex feature interactions and potentially improve generalization performance. We also intent to change the detail in our preprocessing phase, such as using different order, or different preprocessing tools to present the data in a better way.

6 Evaluation Strategy and Results

6.1 Evaluation Method

Explain why you used stratified K-fold cross validation (e.g., small dataset size, need for robust evaluation).

6.2 Metrics

Explain why accuracy, precision, recall, F1, and AUC-ROC are important in medical diagnosis.

6.3 Results

Insert your figures:

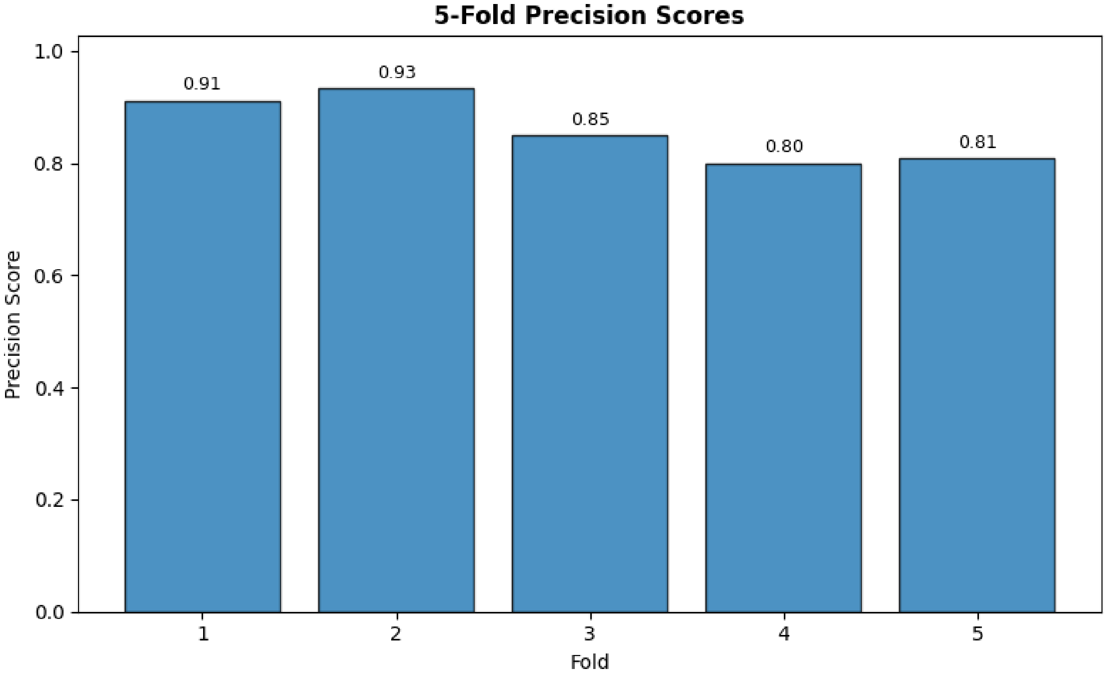
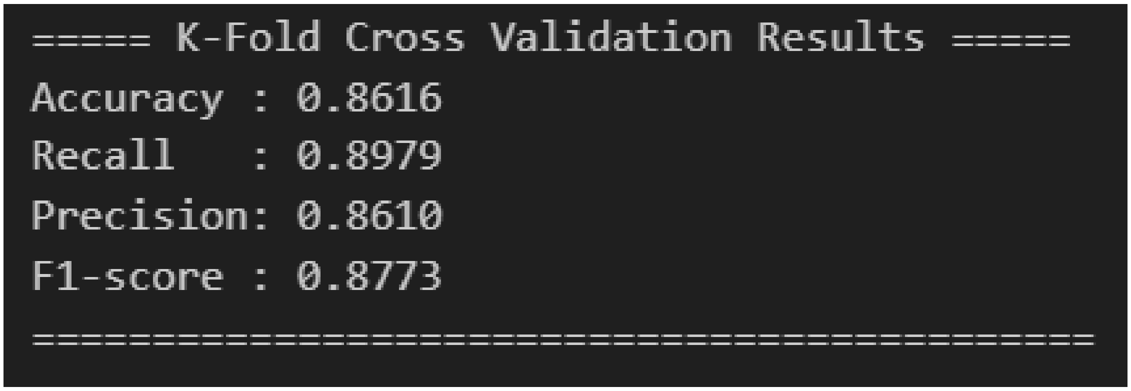


Figure 1: 5-fold precision scores.



```
===== K-Fold Cross Validation Results =====
Accuracy : 0.8616
Recall   : 0.8979
Precision: 0.8610
F1-score : 0.8773
============================================
```

Figure 2: Summary metrics across folds.

7 Feedback and Future Plans

Summarize TA feedback and your improvements:

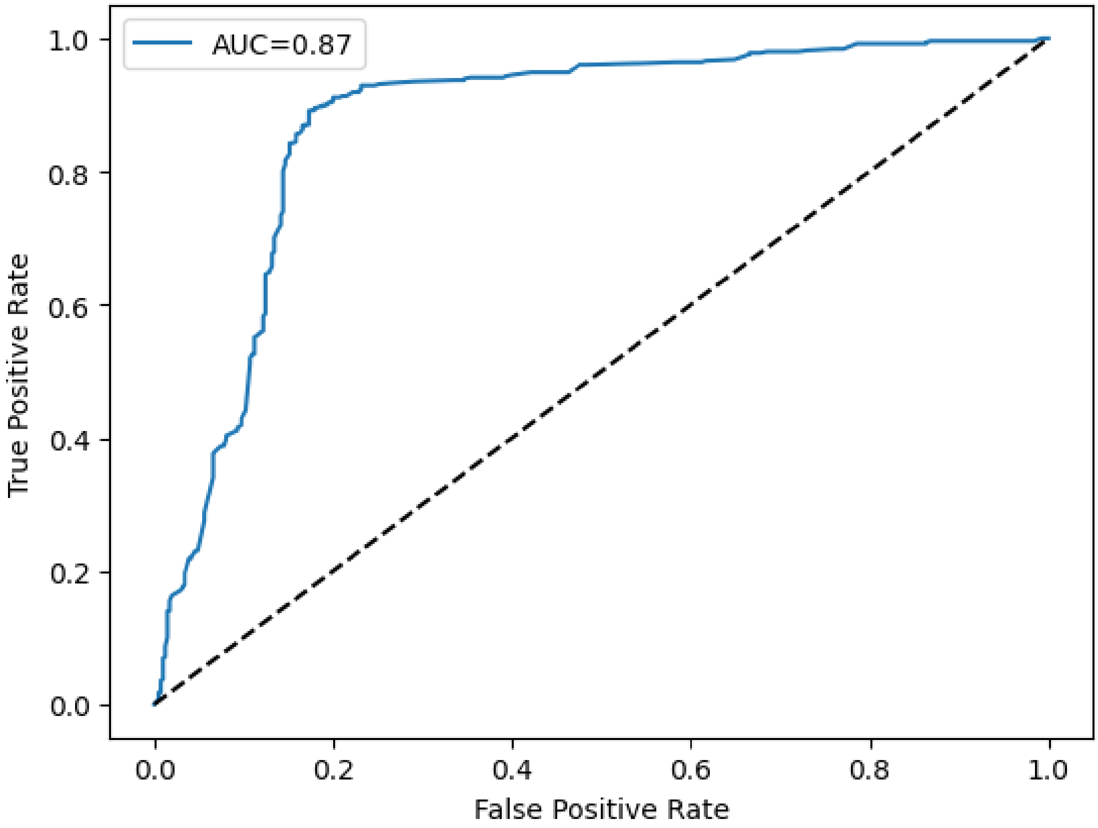


Figure 3: AUC–ROC curves.

- Replace label encoding with one-hot encoding
- Consider switching from SVM to neural networks for performance gains
- Create a new GitHub branch for experiments

Team Contributions

Describe what each team member worked on.

References

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

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

[3] 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.

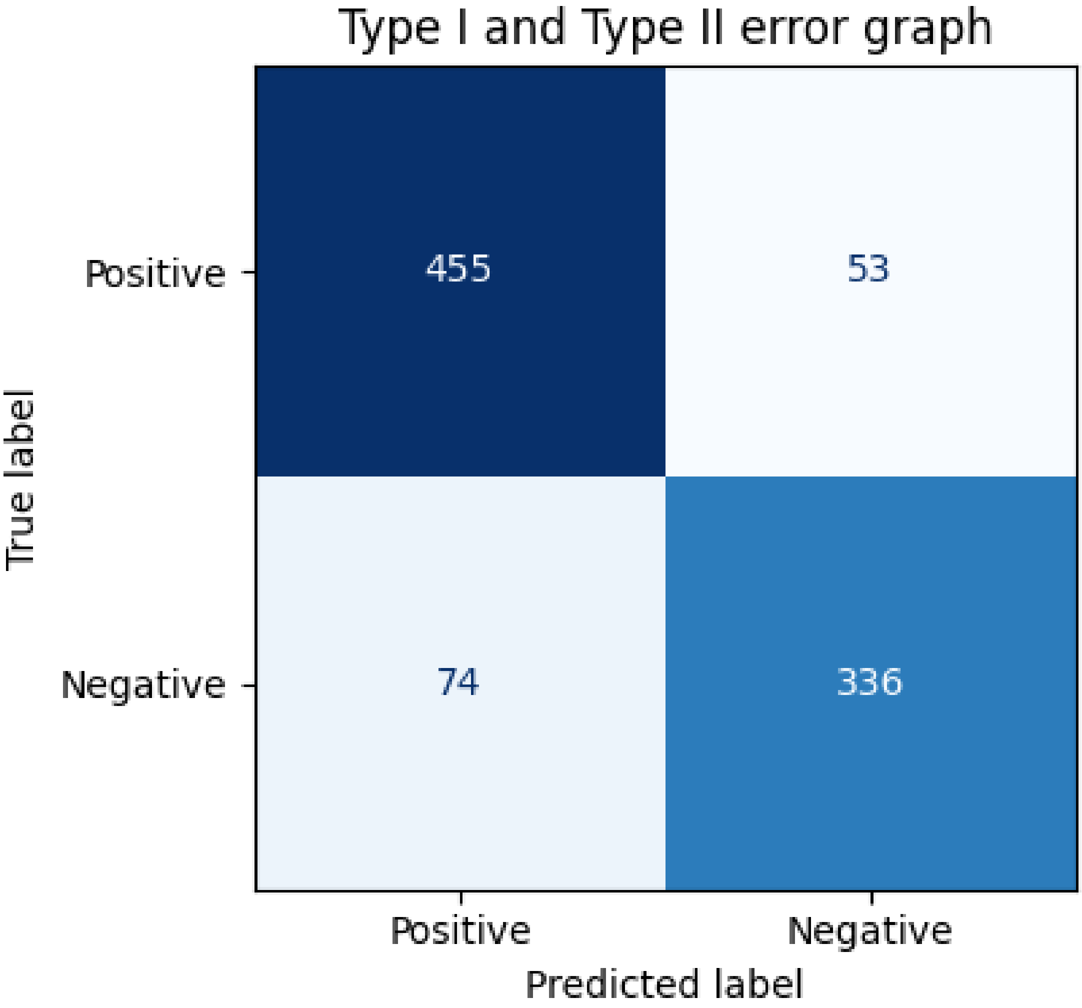


Figure 4: Confusion matrices.

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