Team Name:	Heart Protectors
Team Leader:	Thien Van Ky Nguyen
Team Members:	?, ?, ?, ?
Meeting Date/Time:	April 23, 2025
Attendees:	All team members

1 Project Topic / Service Scenario

Our project uses several machine learning models to predict heart failure risks from health data. Heart failure is a big health problem worldwide that affects millions of people and puts a huge strain on hospitals and healthcare systems. Current methods for detecting heart failure risk aren't great at catching problems early, which means patients often don't get help until it's too late.

We're building a tool that can spot people who might be at risk of heart failure by looking at their health information. This could help doctors take action sooner with patients who are at high risk, possibly preventing heart failure from happening in the first place and saving lives.

2 Selected Dataset

Dataset Information

For this project, we utilize a single, comprehensive heart disease dataset from Kaggle to build and validate our predictive models:

1. **Heart Disease UCI** (10,000 entries) – `https://www.kaggle.com/datasets/oktayrdeki/heart-disease`

Table 1: Heart Disease UCI Dataset Features

Categorical Features	**Numerical Features**
Gender	Age
Exercise Habits	Blood Pressure
Smoking	Cholesterol Level
Family Heart Disease	BMI
Diabetes	
High Blood Pressure	
Low HDL Cholesterol	
High LDL Cholesterol	
Stress Level	
Sleep Hours	
Sugar Consumption	
Triglyceride Level	
Fasting Blood Sugar	
CRP Level	
Homocysteine Level	

3 Models Implemented

The project implements the following machine learning models for heart failure risk prediction:

- **Neural Network (NN)**: A feed-forward multilayer perceptron trained on the UCI dataset with early stopping and dropout regularization.
- **Support Vector Machine (SVM)**: A kernelized SVM (linear and RBF kernels evaluated) optimized via grid search on the regularization parameter C and kernel bandwidth.

4 Problems and Challenges

Throughout the development of our heart failure risk prediction system, we encountered various technical and methodological challenges that required careful consideration and innovative solutions.

4.1 Data Quality and Preprocessing Challenges

Data Quality Issues Overview

Our analysis of the heart disease dataset revealed:

- Missing values in multiple columns (19-30 entries per column)
- Substantial missing data in 'Alcohol Consumption' (2,586 of 10,000 entries missing)
- Need for proper handling of both numerical and categorical features
- Potential outliers requiring detection and treatment

4.1.1 Missing Values

One of the most significant challenges we faced was handling missing values in our datasets. Our initial analysis of the heart disease dataset revealed varying degrees of missing data across

different features. Most notably, the 'Alcohol Consumption' feature had 2,586 missing values out of 10,000 entries (approximately 25.9%), necessitating its complete removal from the dataset to maintain overall data integrity. Other features had fewer missing values (ranging from 19 to 30 entries), but still required appropriate handling techniques.

To address this issue, we implemented a systematic approach:

- For numerical features (such as Age, Blood Pressure, Cholesterol Level, and BMI), we applied mean imputation to replace missing values with the average of each respective column.
- For categorical features (such as Gender, Exercise Habits, and Smoking status), we employed mode imputation, replacing missing values with the most frequently occurring category in each column.

This imputation strategy helped preserve the dataset's size while minimizing statistical bias, although we acknowledge that imputation can potentially introduce subtle distortions in the data distribution.

```
# 1. For numerical columns - impute with mean
numerical_columns = df.select_dtypes(include=[np.number]).columns
# Fill missing values with the mean of each column
df[numerical_columns] = df[numerical_columns].fillna(df[numerical_columns].mean
    ())

# 2. For categorical columns - impute with mode (most frequent value)
categorical_columns = df.select_dtypes(exclude=[np.number]).columns
# Fill missing values with the most frequent value (mode) for each column
for col in categorical_columns:
    df[col] = df[col].fillna(df[col].mode()[0])
```

4.1.2 Feature Encoding and Transformation

Another challenge was the appropriate encoding of categorical variables for model training. Our dataset contained multiple categorical features that needed transformation into a format suitable for machine learning algorithms. While our preprocessing script identified categorical features for potential one-hot encoding, ensuring that these transformations maintained the semantic meaning of each category without introducing unintended relationships was challenging.

Additionally, proper scaling of numerical features required careful consideration to prevent features with larger magnitude ranges (like Cholesterol Level) from dominating those with smaller ranges (like Age) during model training.

4.2 Methodological Challenges

4.2.1 Feature Selection

Determining the most predictive features for heart failure risk presented a significant challenge. Our initial approach included using ANOVA F-tests to identify statistically significant features,

but the effectiveness of this method varied across different subsets of our data. The interrelationships between various health indicators complicated the feature selection process, as some features that appeared individually insignificant could become important when considered in combination with others.

4.2.2 Model Selection and Optimization

Selecting the most appropriate machine learning models for heart failure prediction required balancing accuracy, interpretability, and computational efficiency. Our project implements Neural Network and Support Vector Machine models, each presenting unique challenges:

- ✔ **Neural Network (NN)**: Our deep learning approach required careful architecture design, including determining the optimal number of hidden layers, neurons per layer, and activation functions. Additionally, preventing overfitting through techniques like dropout and early stopping required extensive experimentation.
- ✔ **Support Vector Machine (SVM)**: While effective for high-dimensional classification, optimizing SVM required extensive hyperparameter tuning, particularly for the regularization parameter (C) and kernel selection (linear, polynomial, RBF). Finding the right balance between model complexity and generalization capability was challenging.

4.3 Implementation Challenges

4.3.1 Data Integration

Working with multiple datasets from different sources introduced integration challenges. Variations in feature naming conventions, units of measurement, and data collection methodologies across datasets required careful harmonization to create a unified and consistent training dataset.

4.3.2 Cross-Validation Strategy

Implementing an appropriate cross-validation strategy was crucial for reliable model evaluation but presented challenges due to potential class imbalance in the target variable (heart disease status). Ensuring that each fold maintained representative distributions of both positive and negative cases required specialized stratification techniques.

Insights and Solutions

Despite these challenges, our team implemented several effective solutions:

- a
- a
- a
- a

5 Each Member's Implementation and Evaluation

5.1 Member 1 (Name, graduate/undergraduate)

Neural Network Implementation

Model and Implementation details:
The Neural Network model was implemented with the following key components:

- a
- a
- a
- a
- a

Evaluation Results:

- Accuracy:
- Precision:
- Recall:
- F1 Score:
- AUC-ROC:

5.2 Member 2 (Name, graduate/undergraduate)

Support Vector Machine Implementation

Model and Implementation details:
The SVM implementation featured:

- a
- a
- a
- a
- a

Evaluation Results:

- Accuracy:
- Precision:
- Recall:
- F1 Score:
- AUC-ROC:

5.3 Member 3 (Name, graduate/undergraduate)

Data Preprocessing and Feature Engineering

Implementation details:
This work focused on:

- a
- a
- a
- a
- a

Evaluation Results:

- Accuracy:
- Precision:
- Recall:
- F1 Score:
- AUC-ROC:

5.4 Member 4 (Name, graduate/undergraduate)

Visualization and Model Evaluation

Implementation details:
This work focused on:

- a
- a
- a
- a
- a

Evaluation Results:

- Accuracy:
- Precision:
- Recall:
- F1 Score:
- AUC-ROC: