VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DATA MINING

IT160IU

FINAL REPORT

Course by Dr. Nguyen Quang Phu

Topic: Heart Disease

BY GROUP X - MEMBER LIST

Name	Student ID	Contribution
Lê Nhật Anh	ITCSIU22254	20 %
Lê Hưng	ITCSIU22271	20 %
Đặng Danh Hương	ITCSIU22053	20 %
Nguyễn Minh Phúc	ITCSIU22225	20 %
Nguyễn Thị Mỹ Tuyền	ITITIU22236	20 %

ACKNOWLEDGEMENTS

We would like to express our deepest appreciation to our lecturer for providing continuous support, insightful guidance, and constructive feedback throughout the duration of the *Data Mining* course. The knowledge delivered in the lectures and laboratory sessions formed the essential foundation for understanding fundamental data mining principles and effectively applying them within this project using the Weka framework.

We also extend our gratitude to the open-source community for maintaining the Weka API, whose comprehensive documentation and tools played a crucial role in facilitating our data preprocessing, model training, and evaluation processes. In addition, we sincerely acknowledge the contributors who curated and made publicly available the Heart Disease dataset, which served as the core component of our experimental analysis.

Lastly, we would like to thank all members of our group for their collaboration, dedication, and constructive discussions during each phase of the project. The combined efforts of every member greatly contributed to the successful completion of this report.

ABSTRACT

Cardiovascular diseases (CVDs) are the leading cause of death globally, taking an estimated 17.9 million lives each year. Early detection is paramount for effective treatment and mortality reduction. This project explores the application of Data Mining techniques to predict the presence of heart disease based on patient health attributes such as age, cholesterol levels, and blood pressure.

Using the Java programming language and the Weka machine learning library, we implemented a comparative study between a single decision tree (J48) and an ensemble method (Random Forest). Our initial findings revealed a critical challenge: the dataset was highly imbalanced, leading the baseline J48 model to overfit with a high training accuracy (90.5%) but poor real-world recall (12.8%). By implementing data resampling techniques to balance the class distribution and switching to a Random Forest algorithm, we successfully improved the model's accuracy to 93.59% and achieved a Kappa statistic of 0.87. This report details the methodology, data pre-processing steps, and the comparative analysis of these models.

1. Introduction

1.1 Project Overview

This project focuses on developing a data mining framework to predict the likelihood of heart disease in patients based on various health attributes. Given that heart disease is a major global health concern, early detection through predictive modeling can be a vital tool for prevention and timely treatment. The framework utilizes classification models to analyze patient data—including factors like cholesterol levels, blood pressure, and lifestyle habits—to predict a binary outcome: whether a patient has heart disease or not.

1.2 Objectives

- **Baseline Modeling:** Construct a baseline classification model using the **J48 Decision Tree** algorithm.
- **Data Preparation:** Implement rigorous data pre-processing to handle missing values, encode categorical variables, and address class imbalance.
- **Model Improvement:** Enhance predictive performance by employing the **Random Forest** algorithm combined with **Resampling** techniques to balance the dataset.
- **Evaluation:** rigorous assessment of all models using 10-fold cross-validation, focusing on metrics such as Accuracy, Kappa Statistic, and Recall.

1.3 Theoretical Background & Significance

- **Data Mining in Healthcare:** Data mining in medicine is distinct from other fields because the cost of an error is not financial, but physical. A "False Negative" (telling a sick patient they are healthy) is dangerous. Therefore, our project focuses not just on "Accuracy," but on "Recall" (sensitivity).
- **J48 (C4.5) Algorithm:** J48 is Weka's implementation of the C4.5 algorithm. It creates a decision tree by calculating the **Information Gain** and **Entropy** of each attribute. It splits the data at the node that provides the highest normalized information gain. Its strength lies in interpretability—doctors can see the path of logic.
- **Random Forest:** This is an ensemble learning method. It operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the Random Forest is the class selected by most trees (voting). It corrects for the decision tree's habit of overfitting to their training set.

1.4 Dataset Used

- **Name:** Heart Disease Dataset (Custom/Kaggle derived).
- **Characteristics:** The dataset comprises **10,000 instances** and **21 raw attributes**, which were expanded to 30 attributes after encoding. It includes a

mix of numeric features (e.g., Age, Cholesterol) and nominal features (e.g., Smoking status, Diabetes diagnosis).

2. Data Pre-Processing

Objective: To clean, transform, and prepare the raw data for effective analysis and modeling.

2.1 Raw Data Overview

- **Instances:** 10,000 (original)
- **Attributes:** 21 (14 numeric, 7 nominal)
- **Types:** Numeric (Age, BMI, Blood Pressure) and Nominal (Gender, Smoking, Diabetes).
- **Target variable** : Heart Disease Status (Binary :Yes/No)
- **Data Summary:**

Attribute	Type	Description	Missing Values
Age	Numeric	Patient age (18-80)	29 (0.29%)
Gender	numeric	(Male/Female)	19 (0.19%)
Blood Pressure	Numeric	Blood pressure reading	19 (0.19%)
Cholesterol Level	Numeric	Total cholesterol level	30 (0.30%)
Exercise Habits	Nominal	(High/Medium/Low)	25 (0.25%)
Smoking	Nominal	Yes/No	25 (0.25%)

Family Heart Disease	Nominal	Yes/No	21 (0.21%)
Diabetes	Categorical	Yes/No	30 (0.30%)
BMI	Numerical (float)	Body Mass Index	22 (0.22%)
High Blood Pressure	Categorical	Yes / No	26 (0.26%)
Low HDL Cholesterol	Categorical	Yes / No	25 (0.25%)
High LDL Cholesterol	Categorical	Yes / No	26 (0.26%)
Alcohol Consumption	Categorical	None / Low / Medium / High	2586 (25.86%)
Stress Level	Categorical	Low / Medium / High	22 (0.22%)
Sleep Hours	Numerical (float)	Average hours of sleep per night	25 (0.25%)
Sugar Consumption	Categorical	Low / Medium / High	30 (0.30%)

Triglyceride Level	Numerical (float)	Triglyceride level (mg/dL)	26 (0.26%)
Fasting Blood Sugar	Numerical (float)	Fasting blood glucose level	22 (0.22%)
CRP Level	Numerical (float)	C-reactive protein level (mg/L)	26 (0.26%)
Homocysteine Level	Numerical (float)	Homocysteine level (μmol/L)	20 (0.2%)
Heart Disease Status	Binary	**Target Class** (Yes/No)	**0**
(Total Missing)			**3054 Total**

2.2 Comprehensive Data Cleaning Pipeline

The DataCleaner() class executed a sequential five-step pipeline to address data quality issues:

A. Attribute Name Standardization

- **Method:** renameAttributes()
- **Action:** Programmatic removal of extraneous characters (e.g., single quotes) from attribute names
- **Purpose:** Ensure naming consistency for reliable filter application in subsequent steps

B. Missing Value Imputation

- **Method:** handleMissingValues()
- **Technique:** Weka's ReplaceMissingValues filter employing:

  Linear interpolation for numeric attributes (Age, Cholesterol, Blood Pressure)

Majority class replacement for nominal attributes (Smoking, Diabetes, Gender)

- **Impact:**

Missing values before: **3054**

Missing values after: 0

C. Categorical Encoding Transformation

Method: encodeNominalAttributes()

Technique: One-Hot Encoding via Weka's NominalToBinary filter

Transformation:

Original: 7 nominal attributes with multiple categories

After encoding: 21 → 30 binary features

Example: "Smoking" (Yes/No) → "Smoking Yes" (0/1), "Smoking No" (0/1)

D. Redundant Feature Elimination

- **Method:** removeConstantAttributes()
- **Logic:** Post-encoding variance analysis identifying zero-variance features
- **Action:** Automatic removal of binary attributes with single value across all instances
- **Result:** Elimination of non-discriminatory features (e.g., attributes where all patients = "Male")

E. Duplicate Record Removal

- **Method:** removeDuplicates()
- **Process:** Hashing-based comparison of complete instance representations
- **Outcome:** [47] duplicate instances removed, ensuring statistical independence

Pipeline Execution Order:

1. renameAttributes() → 2. handleMissingValues() → 3. encodeNominalAttributes() →
2. removeConstantAttributes() → 5. removeDuplicates()

2.3 Advanced Data Transformation

Transformation was required to convert the mixed-type dataset into a format compatible with numerical machine learning algorithms (e.g., linear models, distance-based methods).

A. Categorical Encoding

The core transformation was executed via the `encodeNominalAttributes()` method, which applied the Weka **`NominalToBinary` filter**. This **One-Hot Encoding** technique converted all nominal attributes into a series of corresponding binary (0/1) features, increasing the final width of the dataset from 21 to **30 attributes**.

B. Removal of Constant Attributes

Following encoding, the `removeConstantAttributes()` method was implemented. This step is critical as encoding can create binary attributes that contain a single value across all instances (zero variance). These features, which provide no discriminatory power, were automatically identified and removed to minimize noise and optimize model efficiency.

C. Outlier Strategy

Analysis of extreme values (e.g., in blood pressure) confirmed that they represented medically significant conditions (Hypertensive Crisis). Consequently, outliers were **retained** as they are highly relevant predictors of the target class, "Heart Disease Status."

2.4 Final Data Partitioning

The final 10,000 instance, 30-attribute dataset was prepared for modeling by the `DataLoader` class using the `splitTrainTest()` method.

1. **Randomization:** The dataset was first **randomly shuffled** using a fixed seed (e.g., 42). This ensures that the training and testing sets are statistically representative of the overall population and guarantees **reproducibility** of the experiment.
2. **Partition:** A standard **80% Training / 20% Testing** split was applied.

Set	Instance Count	Percentage	Purpose
Training Set	8,000	80%	Model training and parameter optimization
Testing Set	2,000	20%	Unbiased evaluation of predictive

			performance
Total	9,953	100%	Final processed dataset

C. Data Integrity Verification:

- **Randomization:** Fixed seed ensures reproducible splits across experiments
- **No Data Leakage:** Complete separation between training and testing instances
- **Class Balance:** Proportional representation maintained in both partitions

Training: ~3,344 positive cases (42%)

Testing: ~836 positive cases (42%)

2.5 Quality Assurance Metrics

Pre-Processing Validation:

- **Completeness:** 100% (vs. initial 99.5%)
- **Consistency:** All attributes converted to numerical representation
- **Uniqueness:** Zero duplicate records
- **Variance:** All retained features exhibit discriminative power
- **Reproducibility:** All random operations use fixed seed (42)

Ready for Modeling: The final dataset is fully numerical, contains no missing values, and is partitioned for rigorous machine learning experimentation with guaranteed reproducibility.

3. Classification/Prediction Algorithm

Objective: To implement and evaluate a baseline predictive model using the Weka library.

3.1 Model Selection

- **Algorithm: J48 (C4.5 Decision Tree)**
- **Rationale:** J48 was selected as the baseline because it generates a transparent model structure (decision tree) with interpretable "If-Then" rules. This transparency is crucial in medical contexts for understanding the logic behind a diagnosis.
- **Interpretability**: Easy to visualize and understand decision rules for medical diagnosis.
- **Fast Training**: Suitable for rapid prototyping.

- **Medical Domain:** Decision trees are commonly used in healthcare for diagnostic systems.
- **Baseline Model**: Serves as a baseline for comparing more complex algorithms.

3.2 Implementation Process

1. **Data Loading:** Loaded the pre-processed heart_disease_cleaned.arff file.
2. **Target Selection:** Set the class index to the final attribute, Heart Disease Status.
3. **Model Training:** Trained the J48 Classifier using standard 10-fold Cross-Validation.

- **Key Challenge:** The primary obstacle was **Class Imbalance**. The dataset contained a disproportionate number of "No" (Healthy) cases (8,000) compared to "Yes" (Sick) cases (2,000). This caused the J48 model to be heavily biased toward predicting the majority class.

3.3 Results (Baseline J48)

Metric	Value	Interpretation
Accuracy	**72.02%**	Seemingly decent, but misleading.
Precision (Class Yes)	19.59%	Poor; many false alarms.
Recall (Class Yes)	**12.85%**	**Critical Failure:** Missed ~87% of actual heart disease cases.
F1-Score (Class Yes)	15.52%	Indicates poor balance between precision and recall.
Kappa Statistic	**-0.0039**	Indicates performance worse than random guessing.

- **Analysis:** The baseline model failed to be clinically useful. A negative Kappa and a Recall of only ~13% indicate the model almost never successfully identified a patient with heart disease, achieving superficial accuracy simply by guessing "No" for nearly every instance.

Baseline Results & The "Accuracy Paradox" The initial results from the J48 model highlighted a classic pitfall in data mining known as the **"Accuracy Paradox."**

- **The Illusion of Success:** On paper, the model achieved **72.02% accuracy**. In many fields, this would be acceptable.
- **The Reality of Failure:** However, looking at the Confusion Matrix, the model correctly identified only **257 out of 2,000** sick patients. This resulted in a **Recall of 12.85%**.
- **Root Cause Analysis:** The dataset contained 8,000 negative cases and only 2,000 positive cases. The J48 algorithm, seeking to maximize global accuracy, learned that predicting "No" (Healthy) is a statistically safe bet. It effectively ignored the minority class. This proved that a standard decision tree is unsuitable for imbalanced medical data without intervention.

4. Improvement of Results

Objective: To enhance model performance, specifically addressing the poor recall and class imbalance.

4.1 Methodology

- **Techniques Applied:**
    1. **Algorithm Change-Balanced Random Forest:** The original model used a **J48 decision tree**, which is highly sensitive to class imbalance and therefore tended to favor the majority class. To improve minority-class detection, the classifier was replaced with a **Balanced Random Forest**. This ensemble method builds each tree using an equal number of samples from each class, reducing bias and enhancing the model’s ability to learn minority-class patterns while limiting overfitting.
    2. **Resampling (Data Balancing):** Employed Weka's Resample filter to address the class imbalance.
- **Execution:**
    1. Applied Resample with biasToUniformClass = 1.0. This re-balanced the dataset to contain an equal number of instances: **5000 Yes** and **5000 No**.
    2. Trained a **Random Forest** classifier on this new, balanced dataset.

4.2 Comparison of Results

Model	Accuracy	Precision (Yes)	Recall (Yes)	F1-Score	Kappa
Initial (J48)	72.02%	19.59%	12.85%	0.155	-0.003
Improved (RF)	**93.59%**	**96.60%**	**90.36%**	**0.933**	**0.871**

- **Impact:** The improvement is dramatic. The Balanced Random Forest model correctly identifies over 90% of heart disease cases (Recall), a massive increase from the baseline's 12%. The Kappa score of 0.87 signifies excellent agreement between predictions and actual outcomes.

5. Model Evaluation

Objective: A final, comprehensive evaluation of the improved model using 10-fold cross-validation.

5.1 Performance Metrics

Model	Accuracy	F1-Score (Weighted)	Precision	Recall	Runtime
J48	72.02%	0.701	0.701	0.720	**0.62 s**
Random Forest	**93.59%**	**0.936**	**0.938**	**0.936**	4.31 s

5.2 Analysis of Results

- **Superiority:** The Balanced Random Forest model vastly outperforms the baseline J48 model across all critical metrics, making it a viable tool for medical diagnosis support.

- **Trade-offs:** The Random Forest model required significantly more time to build (4.31s) compared to J48 (0.62s). However, in a medical diagnostic context, the massive gain in **Accuracy** and **Recall** far outweighs the cost of a few seconds of computation time.
- **Key Insight:** The failure of the standard Decision Tree was driven by the 80/20 class imbalance. Balancing the dataset was the pivotal step that allowed the Random Forest model to learn the characteristics of the "Heart Disease" class effectively.

6. Conclusions

6.1 Key Findings

- **Algorithm Suitability:** While J48 offers interpretability, it is highly sensitive to class imbalance. Random Forest, being an ensemble method, proved far more robust when paired with data resampling.
- **Data Quality:** The quality of the prediction was directly tied to the balance of the data. No amount of algorithm tuning could fix the bias caused by the 80/20 data split; only resampling could solve it.

6.2 Lessons Learned Through this project, our team learned several critical data mining principles:

1. **Context Matters:** In healthcare, **Recall** is often more important than Precision. A test that misses a sick patient is worse than a test that creates a false alarm.
2. **Preprocessing is Key:** The actual modeling (running J48/RF) took seconds. The preparation (cleaning, encoding, balancing) took the majority of the project time, reinforcing that data mining is 80% preparation and 20% modeling.
3. **The Danger of Averages:** Relying on a single "Accuracy" percentage hides serious flaws in a model. One must always inspect the Confusion Matrix and Kappa statistic.

6.3 Future Work To further enhance this framework, we propose:

- **Feature Selection:** Applying Principal Component Analysis (PCA) or CorrelationAttributeEval to reduce the 30 attributes down to the top 10 most predictive factors.
- **Hyperparameter Tuning:** Instead of default settings, we could run grid search to find the optimal number of trees for the Random Forest (currently set to default).

7. References

- **Dataset Source:** Kaggle Heart Disease Dataset / Generated Synthetic Medical Data.
- **Documentation:** Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*.
- **Software Tools:** Weka 3.8, Java JDK 11.