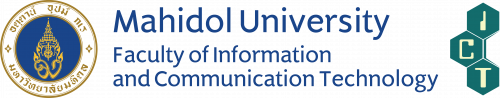
*Faculty of Information and Communication Technology Mahi dol University 2024* 

**ITCS227 – Introduction to Data Science Capstone Project Report**

***A Comparative Study of K-Nearest Neighbors and Logistic Regression for Heart Disease Prediction***

**Group Members**

| **Name** | **Student ID** |
| --- | --- |
| Mr. Kittipob Bumphen | 6688011 |
| Mr. Nonthapath Chaworanun | 6688018 |
| Miss Yu-Han Huang | 6688027 |
| Miss Dungwang Srisa-ard | 6688113 |
| Miss Xinyi Chen | 6688232 |

**A Comparative Study of K-Nearest Neighbors and Logistic Regression for Heart Disease Prediction**

**Abstract**

This report presents a comparative analysis of two machine learning models—K-Nearest Neighbors (KNN) and Logistic Regression—for predicting heart disease using the UCI Heart Disease dataset [2]. The main findings are as follows:

* Logistic Regression achieved better performance than KNN.
* Data preprocessing, including missing value handling and feature normalization, played an important role in improving model performance.
* The results highlight the potential of simple and interpretable machine learning models in supporting the early detection of heart disease in clinical practice.

**Practical Implication:** Logistic Regression can be considered a **cost-effective and reliable tool** for initial heart disease screening in healthcare settings.

The complete code and dataset preprocessing steps have been open-sourced on GitHub: https://github.com/XinyiChen232/DS-Project.

**Content**

[**1. Introduction** 4](#_heading=h.9hopodwrxb9x)

[**2. Methodology** 4](#_heading=h.mj4fo9es6506)

[**2.1** **Implementation Environment** 4](#_heading=h.93w9yk138t6u)

[**2.2** **Data acquisition** 4](#_heading=h.vrmejkjtuy19)

[**2.3** **Data preprocessing** 5](#_heading=h.3zzejlfhpag1)

[**2.4 Data Understanding** 5](#_heading=h.egbhqnbaj09z)

[**2.5** **Model Development** 9](#_heading=h.cgf82xpo9ees)

[**2.5.1 KNN** 9](#_heading=h.15aol1w91oy8)

[**2.4.2 Logistic Regression** 10](#_heading=h.hkwkqkt10738)

[**3. Result** 13](#_heading=h.ttx8f9jw1dbj)

[**3.1 Performance Comparison** 13](#_heading=h.3gg05d70lqt3)

[**3.2 Feature Importance via Logistic Regression** 13](#_heading=h.7hpdz5fzs66c)

[**3.3 Learning Curves** 13](#_heading=h.dy59jm5iee0o)

[**4. Conclusion** 14](#_heading=h.e7bn48ozqy9c)

[**5. Discussion** 14](#_heading=h.qmt8bn7eifiz)

[**6. Acknowledgement** 15](#_heading=h.mjhitwtb1sjj)

[**7. Reference** 15](#_heading=h.ktlhyvwfbh6)

1. **Introduction**

Cardiovascular diseases account for 32% of global deaths annually [1], creating an urgent need for accurate early detection tools. This report investigates machine learning approaches for heart disease prediction using the Cleveland dataset, which contains 13 clinical features from 297 patients [2]. The study compares two fundamentally different algorithms:

* **KNN**: A non-parametric instance-based learner
* **Logistic Regression**: A probabilistic linear classifier

Primary research questions:

1. Which model achieves better predictive performance?
2. What clinical features are most strongly associated with heart disease?
3. How can these findings inform clinical decision-making?
4. **Methodology**
   1. **Implementation Environment**

The experiments were conducted using Python 3.9.19 on Ubuntu 22.04 (via WSL) with the following libraries:

* NumPy, Pandas, Matplotlib, Seaborn for data handling and visualization
* Scikit-learn (Pedregosa et al., 2011) for model development and evaluation
* SciPy for statistical computations

All code was written and executed in Visual Studio Code.

* 1. **Data acquisition**

The dataset used in this study is sourced from the Heart Disease dataset in the UCI Machine Learning Repository, specifically utilizing the Cleveland database [2]. While the original database contains 76 attributes, this experiment focuses on 14 key variables, detailed in **Table 1** primarily used to distinguish between the presence (values 1-4) and absence (value 0) of heart disease. It should be noted that all personally identifiable information (such as patient names and social security numbers) has been anonymized and replaced with dummy values to protect patient privacy. Among the 14 variables included in the analysis, two features - "ca" and "thal" - contain missing values that require special handling during data preprocessing. The dataset is moderately sized with a total volume of 59.2KB, containing clinically significant indicator features while maintaining a reasonable data volume suitable for analytical computations.

| **Variable Name** | **Role** | **Type** | **Description** | **Units** | **Missing Values** |
| --- | --- | --- | --- | --- | --- |
| age | Feature | Integer |  | years | no |
| sex | Feature | Categorical |  |  | no |
| cp | Feature | Categorical |  |  | no |
| trestbps | Feature | Integer | resting blood pressure (on admission to the hospital) | mm Hg | no |
| chol | Feature | Integer | serum cholestoral | mg/dl | no |
| fbs | Feature | Categorical | fasting blood sugar > 120 mg/dl |  | no |
| restecg | Feature | Categorical |  |  | no |
| thalach | Feature | Integer | maximum heart rate achieved |  | no |
| exang | Feature | Categorical | exercise induced angina |  | no |
| oldpeak | Feature | Integer | ST depression induced by exercise relative to rest |  | no |
| slope | Feature | Categorical |  |  | no |
| ca | Feature | Integer | number of major vessels (0-3) colored by flourosopy |  | yes |
| thal | Feature | Categorical |  |  | yes |
| num | Target | Integer | diagnosis of heart disease |  | no |

**Table 1:** Heart Disease Prediction Variable Description Table

* 1. **Data preprocessing**

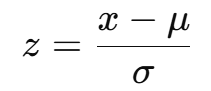
Although the dataset contains missing values in the variables ‘ca’ and ‘thal’, their proportions are relatively small (1.320132% and 0.660066%, respectively). Therefore, the missing entries were removed using the ‘dropna()’ method. Subsequently, data type conversion was performed to ensure that the variables ‘ca’ and ‘thal’ are represented as floating-point numbers, which is essential for numerical computations and compatibility with machine learning algorithms.

* 1. **Data Understanding**

To facilitate binary classification, we first convert the multi-class target variable—which represents different types of heart disease—into a binary format, distinguishing between the presence and absence of disease.

Before normalization, the statistical characteristics of key features in the dataset are summarized in **Table 2.1**, while their distributions are visualized in **Figure 1.1**. As shown, the features vary significantly in terms of scale, dispersion, and distribution shape. Such disparities can negatively impact the performance of many machine learning algorithms, particularly those sensitive to feature magnitude, such as logistic regression and K-nearest neighbors.

To fix this issue, we use the Standard Scaler to conduct feature normalization. This method transforms each feature *x* according to the formula:



where μ is the mean and σ is the standard deviation of the feature. This ensures that all features are rescaled to have zero mean and unit variance.

After standardization, the updated statistical summary is shown in **Table 2.2**, and the transformed feature distributions are visualized in **Figure 1.2**. The normalization process brings all features to a comparable scale, which offers multiple advantages: it prevents features with larger numerical ranges from dominating the model, improves training stability, and often accelerates convergence for algorithms that rely on gradient-based optimization.

To further explore feature relationships, we compute the correlation matrix and visualize it using a heatmap, as shown in **Figure 3**. This helps identify potential multicollinearity and provides valuable insights for feature selection and model development.

|  | **age** | **sex** | **cp** | **trestbps** | **chol** | **fbs** | **restecg** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 |
| **mean** | 54.54 | 0.68 | 3.16 | 131.69 | 247.35 | 0.14 | 1.00 |
| **std** | 9.05 | 0.47 | 0.96 | 17.76 | 52.00 | 0.35 | 0.99 |
| **min** | 29.00 | 0.00 | 1.00 | 94.00 | 126.00 | 0.00 | 0.00 |
| **25%** | 48.00 | 0.00 | 3.00 | 120.00 | 211.00 | 0.00 | 0.00 |
| **50%** | 56.00 | 1.00 | 3.00 | 130.00 | 243.00 | 0.00 | 1.00 |
| **75%** | 61.00 | 1.00 | 4.00 | 140.00 | 276.00 | 0.00 | 2.00 |
| **max** | 77.00 | 1.00 | 4.00 | 200.00 | 564.00 | 1.00 | 2.00 |

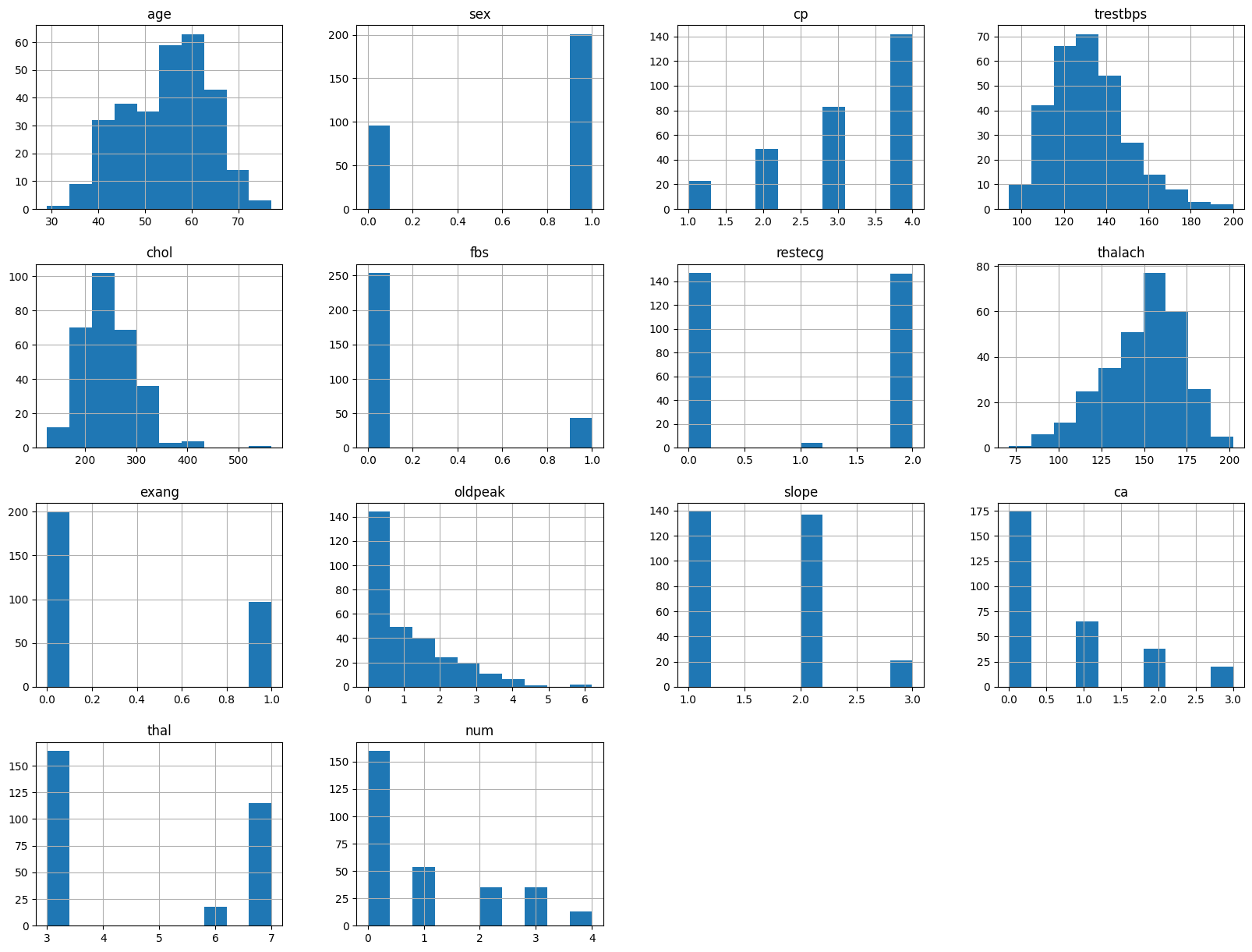
|  | **thalach** | **exang** | **oldpeak** | **slope** | **ca** | **thal** | **num** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 |
| **mean** | 149.60 | 0.33 | 1.06 | 1.60 | 0.68 | 4.73 | 0.46 |
| **std** | 22.94 | 0.47 | 1.17 | 0.62 | 0.94 | 1.94 | 0.50 |
| **min** | 71.00 | 0.00 | 0.00 | 1.00 | 0.00 | 3.00 | 0.00 |
| **25%** | 133.00 | 0.00 | 0.00 | 1.00 | 0.00 | 3.00 | 0.00 |
| **50%** | 153.00 | 0.00 | 0.80 | 2.00 | 0.00 | 3.00 | 0.00 |
| **75%** | 166.00 | 1.00 | 1.60 | 2.00 | 1.00 | 7.00 | 1.00 |
| **max** | 202.00 | 1.00 | 6.20 | 3.00 | 3.00 | 7.00 | 1.00 |

**Table 2.1:** Summary Statistics of Key Features (Before Normalization)

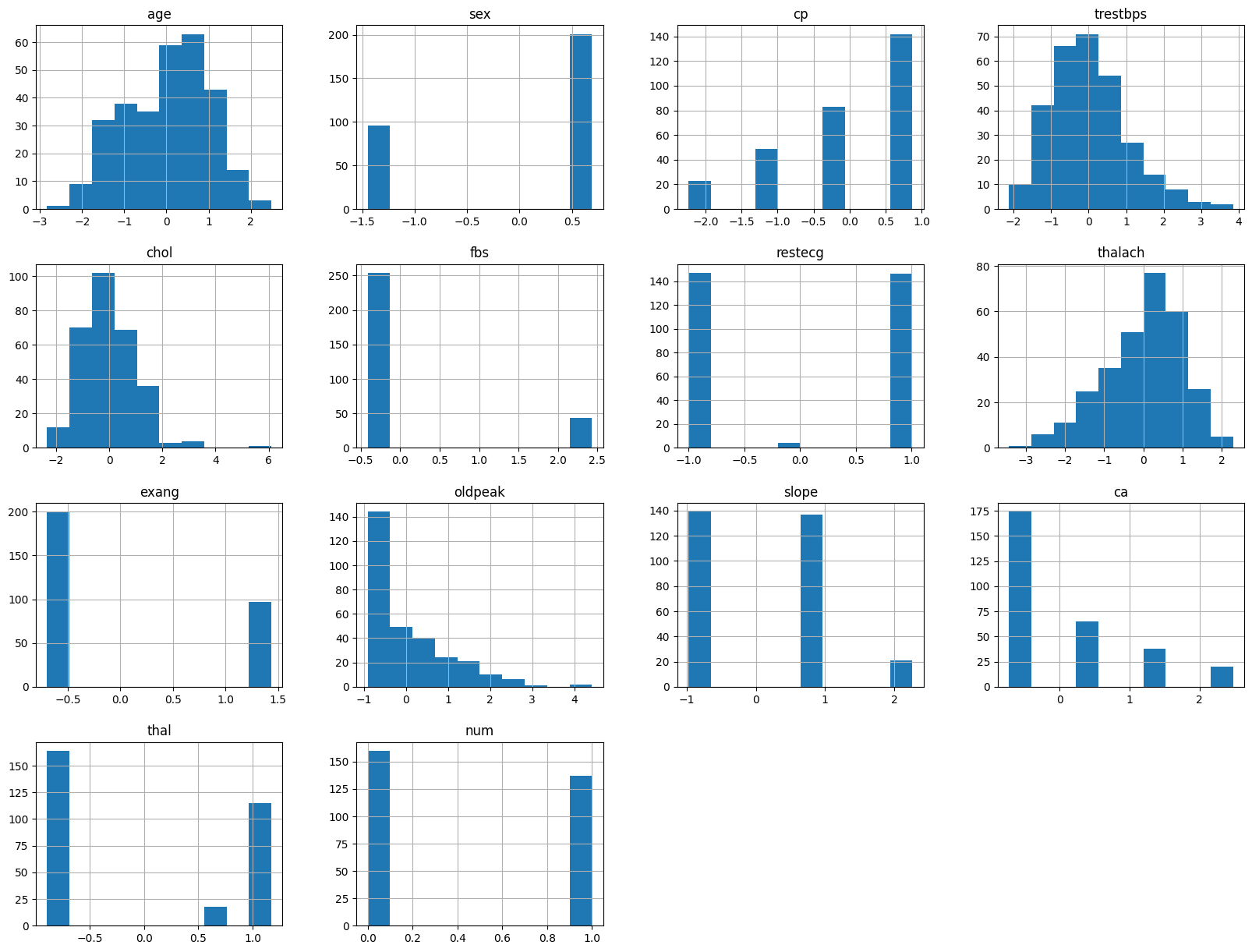
|  | **age** | **sex** | **cp** | **trestbps** | **chol** | **fbs** | **restecg** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 |
| **mean** | 0.00 | -0.00 | -0.00 | 0.00 | -0.00 | 0.00 | 0.00 |
| **std** | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| **min** | -2.83 | -1.45 | -2.24 | -2.13 | -2.34 | -0.41 | -1.00 |
| **25%** | -0.72 | -1.45 | -0.16 | -0.66 | -0.70 | -0.41 | -1.00 |
| **50%** | 0.16 | 0.69 | -0.16 | -0.10 | -0.08 | -0.41 | 0.00 |
| **75%** | 0.71 | 0.69 | 0.87 | 0.47 | 0.55 | -0.41 | 1.01 |
| **max** | 2.49 | 0.69 | 0.87 | 3.85 | 6.10 | 2.43 | 1.01 |

|  | **thalach** | **exang** | **oldpeak** | **slope** | **ca** | **thal** | **num** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 | 297.00 |
| **mean** | 0.00 | -0.00 | 0.00 | -0.00 | -0.00 | 0.00 | 0.46 |
| **std** | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 |
| **min** | -3.43 | -0.70 | -0.91 | -0.98 | -0.72 | -0.89 | 0.00 |
| **25%** | -0.72 | -0.70 | -0.91 | -0.98 | -0.72 | -0.89 | 0.00 |
| **50%** | 0.15 | -0.70 | -0.22 | 0.64 | -0.72 | -0.89 | 0.00 |
| **75%** | 0.72 | 1.44 | 0.47 | 0.64 | 0.34 | 1.17 | 1.00 |
| **max** | 2.29 | 1.44 | 4.42 | 2.26 | 2.48 | 1.17 | 1.00 |

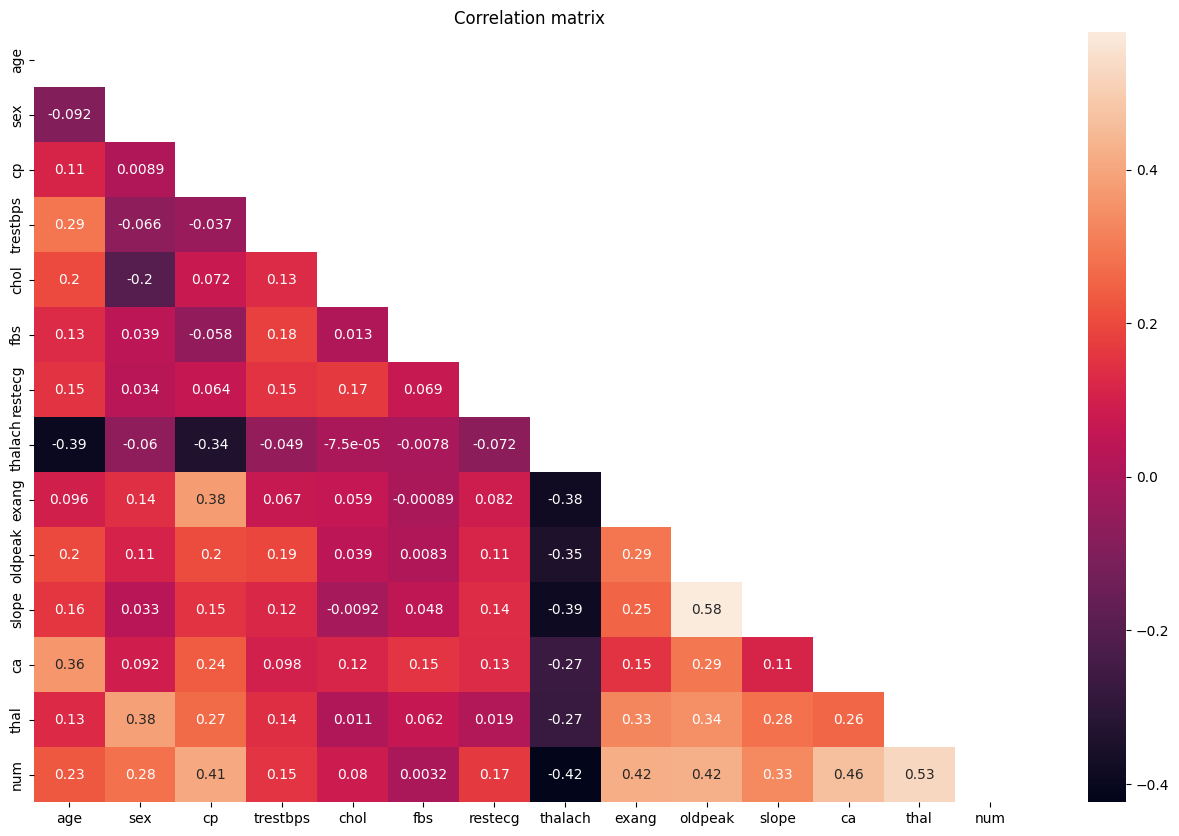
**Table 2.2:** Summary Statistics of Key Features (After Normalization)

****

**Figure 1.1:** Feature Distributions (Before Normalization)

****

**Figure 1.2:** Feature Distributions (After Normalization)

****

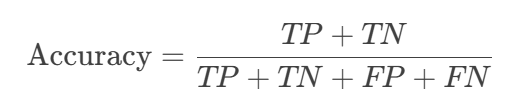
**Figure 3:** Correlation Matrix Heatmap of Features

* 1. **Model Development**

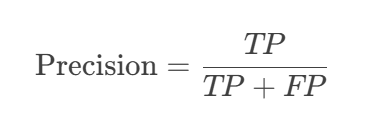
**Evaluation metrics**

In order to comprehensively assess the diagnostic performance of the proposed model in a clinical context, four commonly adopted evaluation metrics are employed: Accuracy, Precision, Recall, and F1-score. These metrics provide complementary insights, particularly crucial in the presence of class imbalance typically observed in medical datasets.

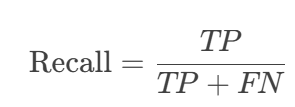
**Accuracy** reflects the overall proportion of correct predictions, including both true positives and true negatives. It offers a general view of model performance but may be less informative in imbalanced datasets.



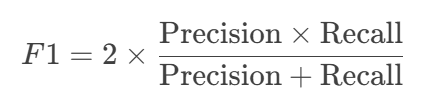
**Precision** measures the proportion of true positives among all predicted positives. High precision reduces false positives, helping to avoid unnecessary treatment and anxiety in healthy individuals.



**Recall** indicates the proportion of actual positive cases correctly identified. It is crucial for minimizing false negatives, particularly in detecting high-risk patients in critical conditions.



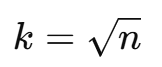
**F1-score** is the harmonic mean of precision and recall. It provides a balanced evaluation, especially valuable in imbalanced datasets, ensuring both accurate detection and clinical reliability.



* + 1. **KNN**

K-Nearest Neighbors (KNN) is a non-parametric, instance-based classification algorithm that determines the class of a test sample by identifying the *k* nearest neighbors in the feature space and applying a majority voting scheme. In this study, the KNN algorithm was systematically implemented for heart disease prediction.

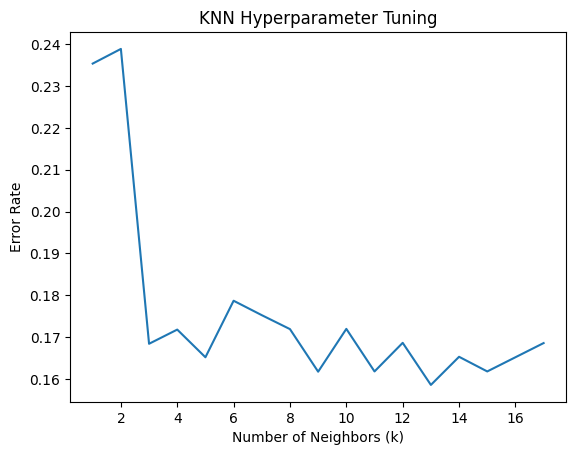
To determine the optimal number of neighbors *k*, we first use the following formula to calculate the threshold:

****

where n represents the total number of data points. This formula provides a heuristic estimate for *k*, which helps ensure that the chosen *k* is appropriately scaled with respect to the dataset size.

A grid search strategy was then applied over the range *k=1* to *k=17*. Using 5-fold cross-validation [3], the classification error for each value of *k* was evaluated, and *k=13* was selected as the optimal parameter due to its lowest validation error, as shown in **Figure 4**. This selection strikes a balance by avoiding overfitting from a small *k* and underfitting from a large *k*.

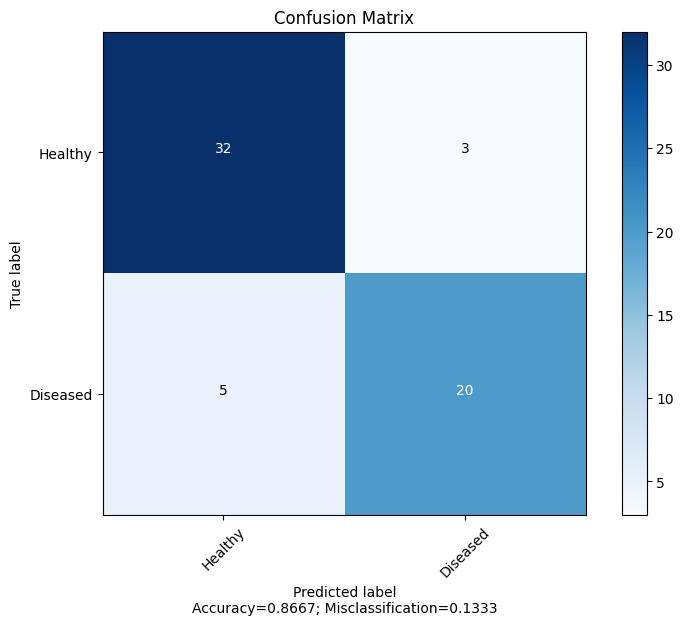
On the hold-out test set, the model demonstrated strong performance with an accuracy of 0.8667, a precision of 0.8696, a recall of 0.8000, and an F1 score of 0.8333, as shown in **Table 3**. The confusion matrix in **Figure 5** further confirms the model's robust predictive capability in real-world applications.



**Figure 4:** Error Rate for Different Values of k in K-Nearest Neighbors Hyperparameter Tuning

| **Metric** | **Value** |
| --- | --- |
| Accuracy | 0.8667 |
| Precision | 0.8696 |
| Recall | 0.8000 |
| F1 | 0.8333 |

**Table 3:** Performance Metrics for KNN Model on the Test Set



**Figure 5:** Confusion Matrix of KNN Model on the Test Set

**2.4.2 Logistic Regression**

Logistic Regression is a widely used statistical learning method for binary classification problems, which maps the linear combination of features to the interval (0,1) through the Sigmoid function to estimate the probability of a sample belonging to the positive class. Its core formula is:

where β represents the model coefficients, optimized through maximum likelihood estimation.

To validate the model’s performance, a stratified k-fold cross-validation (k=5) experimental design was adopted, with the training set consisting of 80% of the original data (n=237) and 20% as an independent test set (n=60).

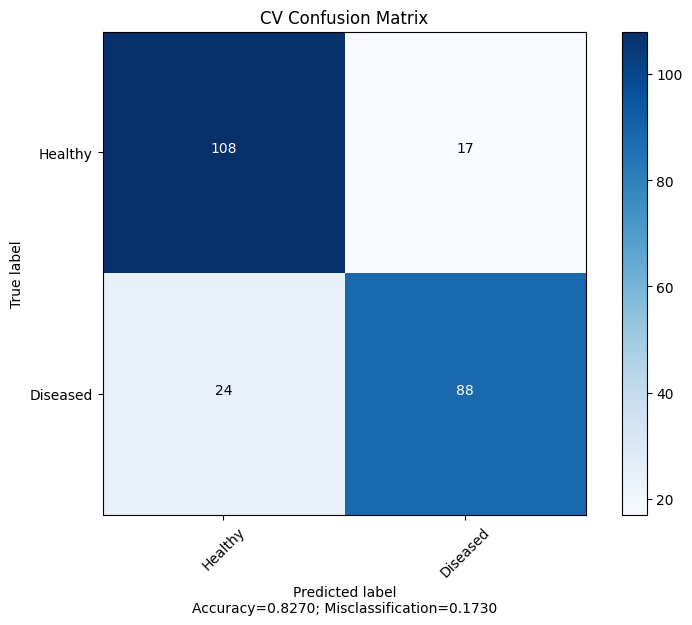
The training set was divided into five mutually exclusive subsets, with approximately 47 samples per fold. Four subsets (80%) were used for training, and one subset (20%) was used for validation. The results of the five validation rounds were aggregated to provide a robust evaluation.

The cross-validation [3] results showed that the average accuracy of the training set was 82.70%, demonstrating the model's overall predictive capability; the precision was 84.17%, reflecting the reliability of positive predictions; the recall was 78.66%, indicating the model's ability to capture true cases; and the F1 score was 81.13%, balancing precision and recall, as shown in **Table 5.1** and the confusion matrices in **Figure 6.1**.

On the independent test set, the model performed even better, with accuracy increasing to 90.00% (+7.3%), precision reaching 91.30%, false positive rate as low as 8.7%, recall maintaining at 84.00%, and F1 score at 87.50%, as shown in **Table 5.2** and the confusion matrices in **Figure 6.2**.

| **Metric** | **Value** |
| --- | --- |
| Average Accuracy | 0.8270 |
| Average Precision | 0.8417 |
| Average Recall | 0.7866 |
| Average F1 | 0.8113 |

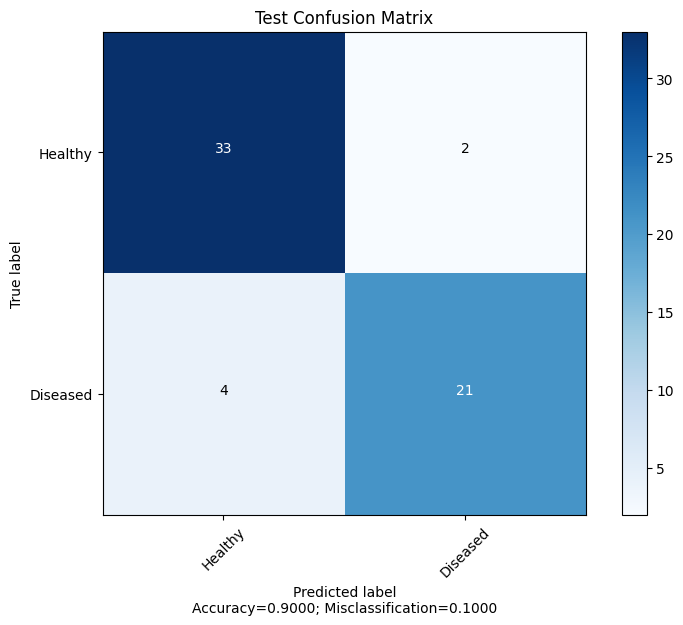
**Table 5.1:** Logistic Regression Cross-Validation Performance Metrics (Training Set)



**Figure 6.1:** Confusion Matrix for Logistic Regression (Training Set)

| **Metric** | **Value** |
| --- | --- |
| Average Accuracy | 0.9000 |
| Average Precision | 0.9130 |
| Average Recall | 0.8400 |
| Average F1 | 0.8750 |

**Table 5.2:** Logistic Regression Performance Metrics (Test Set)

  
**Figure 6.2:** Confusion Matrix for Logistic Regression (Test Set)

1. **Result**

**3.1 Performance Comparison**

In **Table 6**, Logistic Regression outperforms K-Nearest Neighbors across all metrics, demonstrating its higher precision, recall, and F1 score.

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.9000 | 0.9130 | 0.8400 | 0.8750 |
| K-Nearest Neighbors | 0.8667 | 0.8696 | 0.8000 | 0.8333 |

**Table 6:** Performance Comparison of Logistic Regression and K-Nearest Neighbors

**3.2 Feature Importance via Logistic Regression**

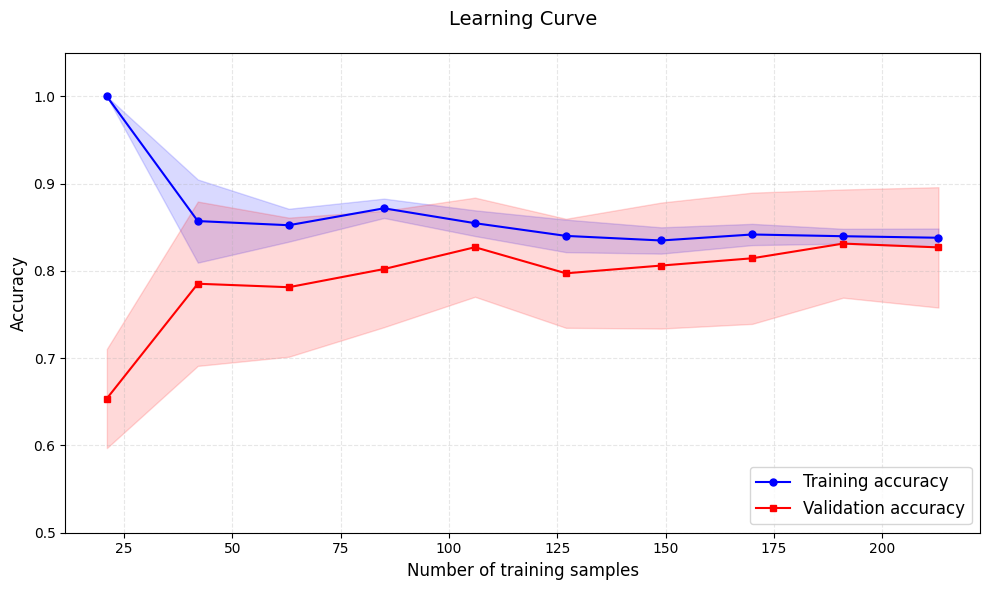
After training the model, we obtained the following feature coefficients:

* The coefficient of "age" is -0.0983
* The coefficient of "sex" is 0.5754
* The coefficient of "cp" (chest pain type) is 0.5362
* The coefficient of "trestbps" (resting blood pressure) is 0.3904
* The coefficient of "chol" (serum cholesterol) is 0.2350
* The coefficient of "fbs" (fasting blood sugar) is -0.3246
* The coefficient of "restecg" (resting electrocardiographic results) is 0.2370
* The coefficient of "thalach" (maximum heart rate achieved) is -0.4492
* The coefficient of "exang" (exercise induced angina) is 0.4213
* The coefficient of "oldpeak" (depression induced by exercise relative to rest) is 0.3000
* The coefficient of "slope" (slope of the peak exercise ST segment) is 0.3229
* The coefficient of "ca" (number of major vessels colored by fluoroscopy) is 1.1044
* The coefficient of "thal" (thalassemia) is 0.6529

The intercept is -0.07983.

**3.3 Learning Curves**

As shown in **Figure 7**, with increasing training samples, both the training accuracy and validation accuracy converge towards each other, indicating that the model is generalizing well to unseen data.

****

**Figure 7:** Learning Curve for Logistic Regression

1. **Conclusion**

This study compared the performance of K-Nearest Neighbors (KNN) and Logistic Regression in predicting heart disease using the Cleveland dataset. Key findings include:

* Logistic Regression outperformed KNN, as show in **Table 6**.
* The **number of major vessels colored by fluoroscopy (*ca*)** and **thalassemia (*thal*)** showed the strongest positive associations (coefficients: 1.1044 and 0.6529, respectively). Other significant predictors included **chest pain type (*cp*)**, **exercise-induced angina (*exang*)**, and **resting blood pressure (*trestbps*)**. Notably, **maximum heart rate (*thalach*)** and **fasting blood sugar (*fbs*)** were inversely correlated with disease risk
* Logistic Regression’s interpretability allows clinicians to prioritize high-risk patients based on key features (e.g., ca and thal), enabling targeted interventions.The model’s high precision (91.3%) reduces false positives, making it a reliable tool for initial screenings in resource-limited settings.Integration of such models into clinical workflows could support early diagnosis, reduce diagnostic costs, and improve patient outcomes through data-driven risk stratification.

These results suggest that Logistic Regression, with its balance of accuracy, efficiency, and interpretability, is a practical tool for early heart disease screening in clinical settings. Its simplicity and cost-effectiveness make it particularly suitable for resource-constrained healthcare environments.

1. **Discussion**

**Limitations and Future Works**

One major limitation of this study is that the dataset consists predominantly of patients from Western populations, which may not generalize well to other ethnic or demographic groups. For instance, if the model is to be applied in Thailand, domain adaptation techniques such as transfer learning or dataset augmentation with Thai patient data would be necessary to ensure reliability and fairness across populations.

While the current models are based on traditional machine learning algorithms such as Logistic Regression and K-Nearest Neighbors, which offer high interpretability, they may be insufficient when dealing with complex time-series data. Time-series signals—such as electrocardiograms (ECGs) and continuous blood pressure monitoring—contain rich temporal features and long-term dependencies. Traditional methods often struggle to effectively capture these sequential patterns, potentially limiting their predictive performance in real-world clinical applications.

To address these limitations, advanced deep learning architectures, particularly Transformer-based models, could be explored. Transformers have demonstrated remarkable success in modeling sequential data due to their ability to capture long-range dependencies and parallelize training efficiently. Applying such models to physiological time-series data may enhance the accuracy and robustness of heart disease prediction, especially in continuous monitoring scenarios.

1. **Acknowledgement**

We sincerely appreciate our instructors, Assoc. Prof. Dr. Suppawong Tuarob and Dr. Peter Scully, for their expert guidance and valuable contributions to this course. We are equally grateful to our learning assistants - Mr. Pisut Manorat, Mr. Krisanakorn Rugbumroong, and Mr. Ar Kar - for their patient support and insightful assistance throughout our learning journey.

1. **Reference**
2. World Health Organization. (2021). *Cardiovascular diseases (CVDs)*. <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)>
3. Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J. J., Sandhu, S., Guppy, K. H., Lee, S., & Froelicher, V. (1989). *International application of a new probability algorithm for the diagnosis of coronary artery disease.* American Journal of Cardiology, 64(5), 304–310. Retrieved from [https://archive.ics.uci.edu/ml/datasets/heart+disease](https://archive.ics.uci.edu/ml/datasets/heart%20disease)
4. Stone, M. (1974). *Cross-validatory choice and assessment of statistical predictions.* Journal of the Royal Statistical Society: Series B (Methodological), 36(2), 111–133.