**VIETNAM NATIONAL UNIVERSITY – HCM**

**INTERNATIONAL UNIVERSITY**

A blue and white logo

Description automatically generated**DEPARTMENT OF COMPUTER SICENCE & ENGINEER**

SEMESTER 2 (2023-2024)

**IT159IU**

**Artificial Intelligence Project**

**Topic name:**

*May 2024*

**Group 7:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full name - Student name** | **Student ID** | **Contribution (%)** |
| 1 | PHAM VU TUYET ANH | ITDSIU21073 |  |
| 2 | NGUYEN NHAT KHIEM | ITDSIU21091 |  |
| 3 | LIEU PHONG SON | ITITIU18200 |  |

**Table of Contents**

[I. INTRODUCTION 3](#_Toc166948270)

[1 Import needed libraries & data 3](#_Toc166948271)

[2 Description 3](#_Toc166948272)

[3 Visualization 3](#_Toc166948273)

[II. Preprocessing (TUYET ANH) 4](#_Toc166948274)

[1 Processing missing value 4](#_Toc166948275)

[**2** Encoding 4](#_Toc166948276)

[**3** Normalization 4](#_Toc166948277)

[III. APPLYING MODEL (KHIEM) 5](#_Toc166948278)

[1 Model 1 5](#_Toc166948279)

[2 Model 2 5](#_Toc166948280)

[3 Model 3 5](#_Toc166948281)

[4 Model 4 5](#_Toc166948282)

[5 Model 5 5](#_Toc166948283)

[IV. METRICS (KHIEM) 6](#_Toc166948284)

[1 Compare 6](#_Toc166948285)

[2 Recommendation 6](#_Toc166948286)

[V. Exited model (SON) 7](#_Toc166948287)

# INTRODUCTION

## Import needed libraries & data

## Description

## Visualization

.

# **Preprocessing (TUYET ANH)**

## Processing missing value

## Encoding

## Normalization

# APPLYING MODEL (KHIEM):

## **1. Aim**

The objective of this section is to train models for classifying heart failure patients into two categories: non-heart failure (0) and positive heart failure (1). Various machine learning models are employed to achieve this classification and conducted a thorough evaluation to determine the best model based on accuracy and other performance metrics.

## **2. Methodology**

* 1. Grid Search for Model Selection: A grid search is performed to identify the most suitable machine learning models based on their accuracy scores on the training set.

We utilized GridSearchCV to perform an exhaustive search over specified parameter values for different machine learning algorithms along with kfold train-test split. The following models and their respective parameter grids were considered: KNeighborsClassifier, GaussianNB, SVC, MLPClassifier, LogisticRegression, RandomForestClassifier.

After performing GridSearch using sklearn, the output yeilds:

* KNeighborsClassifier: 87.02%, {'leaf\_size': 1, 'metric': 'euclidean', 'n\_neighbors': 5, 'weights': 'uniform'}
* GaussianNB: 86.16%, {'var\_smoothing': 0.12328467394420659}
* SVC: 87.71%, {'C': 1, 'gamma': 0.1, 'kernel': 'poly'}
* MLPClassifier: 87.02%, {'activation': 'tanh', 'alpha': 0.05, 'early\_stopping': False, 'hidden\_layer\_sizes': (50, 100, 50), 'learning\_rate': 'constant', 'solver': 'sgd'}
* LogisticRegression: 85.64%, {'C': 5, 'penalty': 'l2', 'solver': 'lbfgs'}
* RandomForestClassifier: 87.54%, {'criterion': 'gini', 'max\_depth': 5, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 50}
  1. Model Implementation and Evaluation: Selected models are initialized, trained them on the dataset.

Each model was initialized with the optimal parameters determined from the grid search. The training process involved fitting the models on the training dataset, ensuring that they learned the underlying patterns and relationships between the features and the target variable (heart failure classification). The models were then prepared for subsequent evaluation to assess their performance on unseen test data.

The implementation process for each model is outlined below:

* KNeighborsClassifier: Initialized with the best parameters including leaf\_size, metric, n\_neighbors, and weights.
* GaussianNB: Configured with the optimal var\_smoothing parameter.
* SVC: Set up with the best values for C, gamma, and kernel.
* MLPClassifier: Defined with the optimal activation, alpha, early\_stopping, hidden\_layer\_sizes, learning\_rate, and solver.
* LogisticRegression: Implemented with the best C, penalty, and solver.
* RandomForestClassifier: Initialized with the best criterion, max\_depth, min\_samples\_leaf, min\_samples\_split, and n\_estimators.

# METRICS (KHIEM)

## Compare

A screen shot of a computer

Description automatically generated

Overall, all models performed well, each demonstrating strengths in various evaluation metrics. The RandomForestClassifier achieved the highest performance across several key metrics, with an accuracy of 91%, a recall of 94%, an F1 score of 92%, and a ROC AUC score of 90%. These metrics indicate that the RandomForestClassifier not only correctly identified the majority of positive cases but also maintained a high balance between precision and recall. Additionally, it showed robust generalization capability as evidenced by its cross-validation score of 92%.

The GaussianNB model also demonstrated strong performance, particularly in terms of training efficiency. It achieved an accuracy of 89%, a recall of 91%, an F1 score of 91%, and a ROC AUC score of 88%. Despite these metrics being slightly lower than those of the RandomForestClassifier, GaussianNB had the shortest training time at just 0.09 seconds. This makes it an attractive option for scenarios where computational resources or time are limited.

The remaining models, including KNeighborsClassifier, SVC, MLPClassifier, and LogisticRegression, also performed well, with accuracies ranging from 86% to 88% and consistent cross-validation scores around 92%. Each model has its unique advantages, such as the simplicity and interpretability of LogisticRegression or the flexibility of the MLPClassifier with its ability to capture complex patterns through neural networks.

## Recommendation

Based on the evaluation metrics and training time, the RandomForestClassifier is recommended for heart failure classification. It achieved the highest accuracy, recall, F1 score, and ROC AUC score. Although it takes longer to train than some other models, its performance makes it suitable for this classification task.

For scenarios where training time is critical, GaussianNB can be considered as it offers a good balance of performance and efficiency, with the second-highest accuracy and the shortest training time.

# Exited model (SON)