BRAC UNIVERSITY

Inspiring Excellence

CSE422 Project

HEART FAILURE PREDICTION USING MACHINE LEARNING TECHNIQUES

Group: 04

Shah Md. Shakhawath Hossain – 18101133

Hazra Mohammed Ahnaf Faiyaz – 17241014

Mahmudul Hasan Mitul – 18101066

MD. Minhaj Rahman – 18301072

Table of Contents

1. Introduction

The project's primary objective is to forecast heart failure using a range of machine learning algorithms. The heart, a vital organ in the human body, plays a crucial role in sustaining life. It propels oxygenated blood through the circulatory system, providing essential nutrients and oxygen to various body tissues. A healthy heart is synonymous with a healthy life. Unfortunately, heart disease has emerged as a leading cause of death for both men and women globally. According to the World Health Organization, it is projected that by 2030, nearly 23.6 million individuals will succumb to heart failure.

Heart failure represents a severe stage of numerous heart conditions, often resulting in inadequate cardiac output. This condition carries a high mortality rate and places a significant financial burden on healthcare systems. Given the prevalence of heart disease, there is an urgent need for accurate and early detection methods that can potentially save countless lives. While various scanning techniques are available to diagnose heart disease, predicting its onset before it becomes symptomatic holds the promise of prevention.

Our project's core focus revolves around the utilization of some of the most proficient machine learning models, including Logistic Regression, Decision Tree, Naive Bayes, Support Vector Machines (SVMs), and K-Nearest Neighbors (KNN). Among these five models, the Naive Bayes algorithm exhibited the highest accuracy, achieving an impressive 0.8444% accuracy rate on the utilized dataset.

2. Dataset Description

Our study delved into the analysis of a dataset containing medical records from 299 patients suffering from heart failure. These records were collected at the Faisalabad Institute of Cardiology and the Allied Hospital in Faisalabad, located in Punjab, Pakistan. The data collection spanned from April to December 2015 [52, 66]. The patient cohort consisted of 105 women and 194 men, spanning an age range from 40 to 95 years old. All 299 patients exhibited left ventricular systolic dysfunction and had previously experienced heart failures, categorizing them within classes III or IV of the New York Heart Association (NYHA) classification for heart failure stages.

The dataset contains 13 features, which report clinical, body, and lifestyle information, that we briefly describe here. Some features are binary: anaemia, high blood pressure, diabetes, sex, and smoking. In the dataset not all the features are quantitative, some are categorical. Regarding the distribution of the dataset, there exists an imbalance. The patients who survived (denoted as death event = 0) total 203, whereas those who unfortunately passed away (denoted as death event = 1) number 96. In statistical terms, this results in 32.11% positive cases (deaths) and 67.89% negative cases (survivals). This dataset serves as a classification problem, given the binary nature of its outcome: predicting whether a patient is at risk of heart failure or not. This classification task holds immense clinical relevance, potentially aiding in patient care, management, and risk assessment.

3. Dataset Pre-processing

Null values or missing data can disrupt the training process and lead to biased or inaccurate models. The presence of missing data can result in incomplete patterns, reduced sample sizes and skewed features distribution. We used scikit-learn's ‘SimpleImputer’ to identify and replace missing values in the 'platelets' and 'serum_sodium' columns separately. For the 'platelets' column, it uses the median value, and for the 'serum_sodium' column, it uses the mean value to impute the missing data. This ensures that the dataset is free of missing values in these specific columns, making it suitable for further analysis or modeling.

We also used scikit-learn's LabelEncoder to encode categorical values in several columns (anaemia, diabetes, high_blood_pressure, sex, and smoking) into numerical form. Categorical variables, which represent discrete categories rather than numerical values, require special handling because most machine learning algorithms work with numerical data. If not properly encoded, categorical variables can introduce bias or misinterpretation in the model. Here's a brief explanation of how we made it work:

For each of the specified columns, the LabelEncoder is applied, which converts the categorical values into unique numerical labels. For example, in the 'anaemia' column, it assigns '0' to one category and '1' to the other. After that the transformation is performed and directly applied to the corresponding column in the DataFrame data. This replaces the original categorical values with their numerical labels. After encoding, the DataFrame data now contains these columns with numerical values, making it suitable for use with machine learning algorithms that require numerical input.

4. Dataset Splitting

In the process of training and evaluating our machine learning model, we performed a fundamental step known as dataset splitting. This step is essential to ensure the model's robustness and generalization capabilities. We partitioned our dataset into two distinct subsets: a training set and a test set, with a split ratio of 70% for training and 30% for testing. This 70-30 split was achieved using the widely adopted train_test_split function from the scikit-learn library in Python. The training set, consisting of 70% of the data, was utilized to train our model, allowing it to learn patterns and relationships within the data. Meanwhile, the test set, comprising the remaining 30% of the data, served as an independent evaluation set to assess the model's performance on unseen data. This partitioning strategy ensures that our model can be rigorously tested for its ability to generalize to new, unseen examples, providing a robust assessment of its effectiveness.

5. Model Training & Testing

In our comprehensive evaluation of machine learning models for the task at hand, we employed five distinct algorithms: Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), and Naive Bayes.

Logistic Regression:

Logistic Regression is a versatile and widely used model in machine learning, particularly suited for binary classification tasks. It models the relationship between input features and the probability of a binary outcome occurring. One of its strengths lies in its simplicity and interpretability, making it a valuable choice when understanding the factors that influence a particular decision is essential. Logistic Regression computes log-odds, which can be transformed into probabilities. However, it assumes a linear relationship between the features and the log-odds, which may not be accurate for all datasets.

K-Nearest Neighbors (KNN):

K-Nearest Neighbors is a non-parametric classification algorithm that classifies data points based on the majority class of their nearest neighbors in the feature space. It is an intuitive method, making it easy to grasp and implement. KNN's strength lies in its ability to capture local patterns in the data, especially when the decision boundaries are irregular. However, it can be sensitive to the choice of the 'k' parameter and becomes computationally expensive as the dataset size increases.

Decision Trees:

Decision Trees are versatile models suitable for both classification and regression tasks. They create a tree-like structure where each node represents a feature, and each branch represents a decision rule based on that feature. Decision Trees are valuable for their interpretability, as they provide a clear visual representation of the decision-making process. They can handle both numerical and categorical data and are capable of capturing non-linear relationships. However, they are prone to overfitting, especially when the tree becomes deep and complex.

Support Vector Machines (SVM):

Support Vector Machines are powerful models used for binary and multiclass classification tasks, as well as regression and outlier detection. SVMs aim to find an optimal hyperplane that maximally separates data points belonging to different classes. They are particularly effective when the data is not linearly separable, thanks to kernel tricks that allow them to work in high-dimensional spaces. SVMs are known for their ability to handle complex decision boundaries. However, they can be sensitive to the choice of the kernel and may not scale well to large datasets.

Naive Bayes:

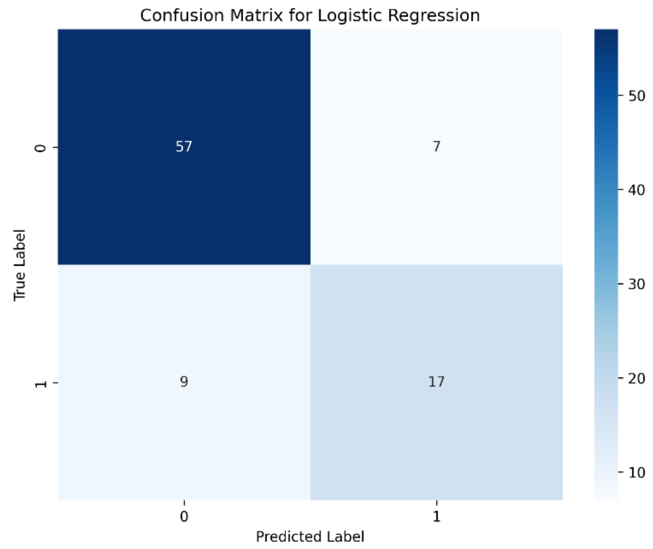
Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It is commonly used for text classification tasks, such as email spam detection, sentiment analysis, and document categorization. Naive Bayes is computationally efficient and handles high-dimensional data well, making it suitable for situations with limited data. Despite its simplifying assumption of feature independence (hence the "naive" name), it often performs surprisingly well in practice and can provide strong predictive accuracy, especially in text-related tasks.

These models were trained and tested on our dataset using a 70-30 train-test split. The results revealed varying degrees of performance across the models. Logistic Regression achieved an accuracy of 82.2%, demonstrating strong overall predictive capability, with a precision of 70.8% and recall of 65.0%. Decision Trees also exhibited promising results with an accuracy of 80%, a precision of 65.0%, and recall of 65.0%. Naive Bayes outperformed the other models with an accuracy of 84.0%, precision of 83.0%, and recall of 57.0%. However, it's worth noting that K-Nearest Neighbors and SVM exhibited lower performance in comparison, with KNN achieving an accuracy of 62.2%, a precision of 21.0%, and recall of 11.5%, while SVM demonstrated an accuracy of 71.0% but had a precision of 100.0% and a lower recall of 0.0%. These findings provide valuable insights into the suitability of different machine learning techniques for our specific task.

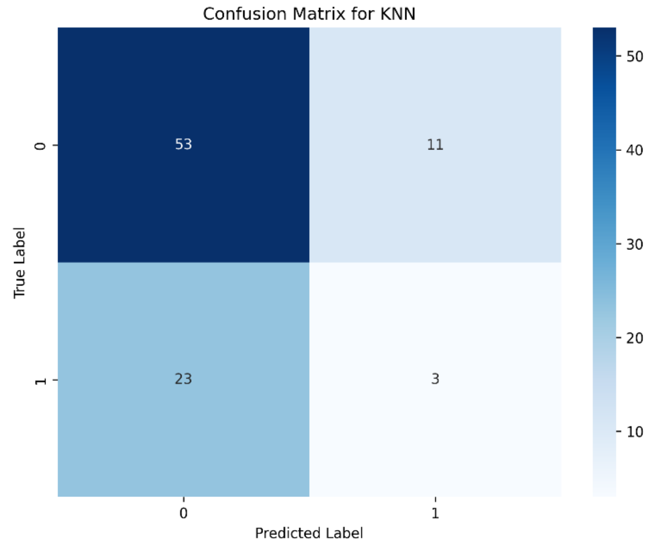
6. Model Comparison Analysis

Confusion matrix for the used models,

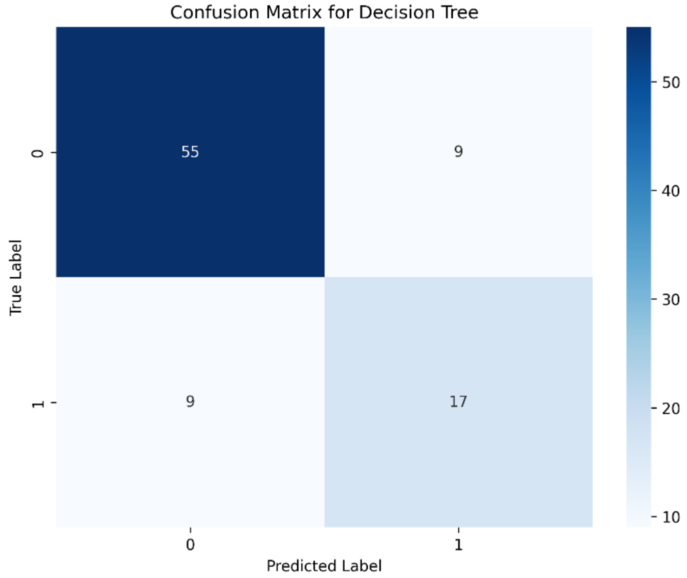
Logistic Regression:



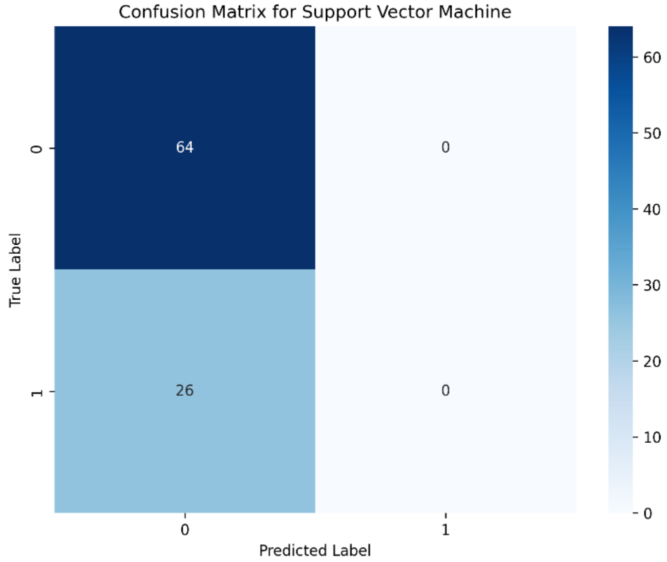
K-Nearest Neighbors (KNN):



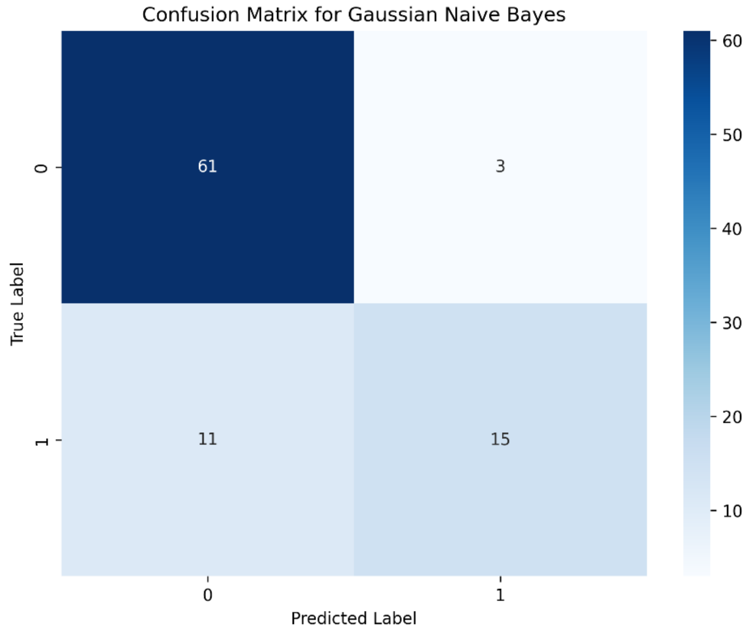
Decision Trees:



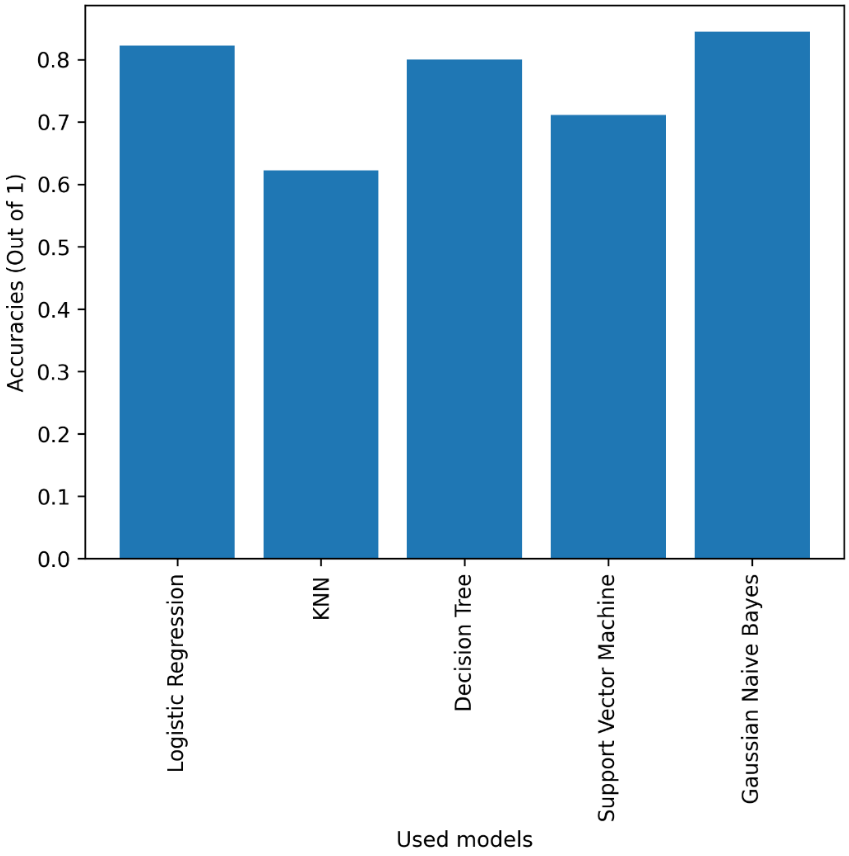
Support Vector Machines (SVM):



Naive Bayes:



Bar chart showcasing prediction accuracy of all models,



The model selection and comparison analysis conducted in this study shed light on the relative strengths and weaknesses of various machine learning algorithms in the context of our specific task. The insights garnered from our evaluation reveal that not all models perform equally well on the given dataset. Logistic Regression and Decision Trees showcased competitive results, with accuracies of 82.2% and 80.0%, respectively, and demonstrated balanced trade-offs between precision and recall. Naive Bayes, however, emerged as the top-performing model, achieving the highest accuracy of 84.0% along with a commendable precision of 83.0%, albeit with a slightly lower recall of 57.0%. On the other hand, K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) exhibited suboptimal performance for this specific task. KNN struggled with a low recall of 11.5%, while SVM exhibited an unusual precision-recall trade-off, with perfect precision but no recall. These findings underline the importance of a careful model selection process, taking into consideration the specific characteristics of the dataset and the desired balance between precision and recall. In our case, Naive Bayes stands out as the most promising model for further refinement and deployment in real-world applications.

7. Conclusion

Our study embarked on a comprehensive exploration of machine learning models to address the task at hand. Through rigorous evaluation and comparison, we gained valuable insights into the performance of five prominent algorithms: Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), and Naive Bayes. The results highlighted the diversity of these models in terms of predictive accuracy, precision, and recall. While Logistic Regression and Decision Trees exhibited commendable overall performance, Naive Bayes emerged as the frontrunner, showcasing the highest accuracy and a balanced precision-recall trade-off. Our findings underscore the significance of judicious model selection, considering the specific requirements of the task and dataset. Additionally, they emphasize the potential of Naive Bayes as a robust choice for further refinement and application. In future work, we plan to delve deeper into feature engineering and hyperparameter tuning to enhance model performance even further. Ultimately, this research contributes valuable insights to the field of machine learning and aids in the selection of suitable algorithms for similar classification tasks.