ME 276DS Statistics and Data Science for Engineers Final Project

Machine Learning for Heart Disease Analysis

Group ID: 100

Group Members: Demin Li, YuChieh Lin, Valerio He, Qianyun Lin, Yushen Feng, Yi Hong (Ian) Liu

Workload Distribution: The workload for this project is equally distributed.

Abstract

This project applies machine learning techniques to diagnose heart disease, utilizing a dataset from the Cleveland Clinic with over 300 cases and 14 attributes. Our study involves preprocessing the data, exploring different models like K-Means, KNN, Logistics Regression, Random Forest, and Naïve Bayes, and comparing their effectiveness. The goal is to identify the most accurate model for early and reliable heart disease diagnosis, contributing to improved patient care and timely medical interventions

Table of Contents

Introduction and Problem Description 1
Data Description 2
Methodology for each Model 4
3.1 Data Preprocessing 4
3.2 K-Means 5
3.3 KNN 6
3.4 Logistic Regression 7
3.5 Random Forest 7
3.6 Naive Bayes 8
3.6.1 Gaussian Naive Bayes 8
3.6.2 Bernoulli Naive Bayes 8
Results and Analysis 9
Reference 10

Introduction and Problem Description

Heart disease, or cardiovascular disease in a broader sense, is a critical health issue that affects millions of individuals. Over the past decade, cardiovascular disease has continued to be the leading cause of death worldwide [1]. Risk factors like smoking, excessive alcohol and caffeine intake, stress, lack of physical activity, obesity, hypertension, high cholesterol levels, and pre-existing heart conditions increase the risk of heart disease. Therefore, early and accurate diagnosis of heart disease plays a crucial role in implementing timely medical intervention and patient care, potentially saving lives. In this project, we aim to apply the machine learning knowledge we have obtained in our coursework to a real-world scenario, leveraging different models to assist in diagnosing heart disease.

Our dataset, 'heart disease.csv,' is sourced from the Cleveland Clinic and contains valuable information about heart disease patients. In this dataset, there are 300+ real-world examples, with 14 attributes (13 predictors and 1 class), including factors such as chest-pain type, fasting blood sugar levels, resting electrocardiographic results, and various other clinical attributes. Each data point represents an individual's health profile, and the dataset is labeled to indicate the presence or absence of heart disease. The labels provide a binary classification challenge: 0 denotes the absence of heart disease, while 1 signifies its presence. We will apply five different models: K-Means, KNN, Logistics Regression, Random Forest, and Naïve Bayes, compare their performance and analyze the results to determine the model with the maximum potential accuracy.

Data Description

In this section, data features will be introduced as well as their distribution visualizations. To help understand those attributes related to ECG signals, we provide some background about ECG. In Figure 2, it shows ECG waves are constructed by three basic waves, which are P wave, QRS wave and T wave.

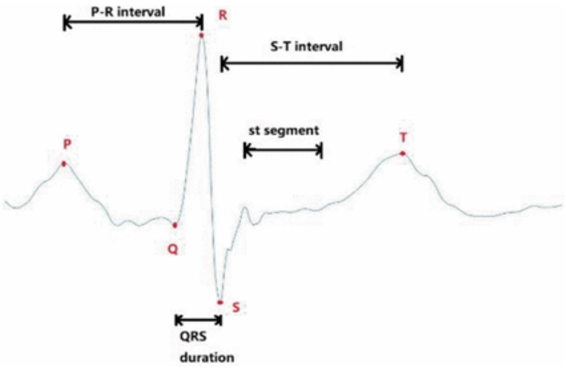

Figure 1: A general period of the signal [2]

ST-segment depression (STD) is one of the main signs indicating patients suffering from acute chest pain . In the plotted ECG data, the ST depression means the ST segment has a depressed position lower than the ST segment’s baseline. The depression can be horizontal, downsloping, or upsloping. ST segment refers to the period between ventricular depolarization and repolarization.

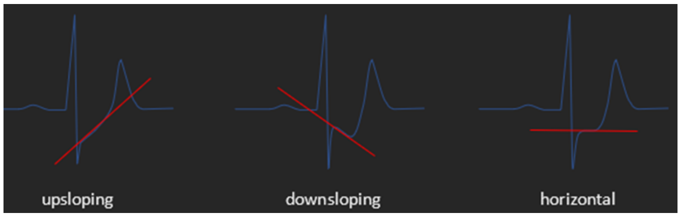

Figure 2: Three patterns of the ST segment of a STD patient

The following table shows the detailed description of each attribute and their values.

NO	Attribute	Num	Type	Description
1	Age	41	Int64	Age of Patients, in years
2	Sex	2	Int64	1 = male; 0 = female
3	Chest pain	4	Int64	4 types of chest pain (1--typical angina; 2--atypical angina; 3--non-anginal pain; 4--asymptomatic)
4	Rest blood pressure	50	Int64	Resting systolic blood pressure (in mm Hg on admission to the hospital)
5	Serum cholesterol	152	Int64	Serum cholesterol in mg/dl
6	Fasting blood sugar	2	Int64	Fasting blood sugar > 120 mg/dl (0--false; 1--true)
7	Resting ECG	3	Int64	0--normal; 1--having ST-T wave abnormality; 2--left ventricular hypertrophy
8	Max Heart Rate	91	Int64	Maximum heart rate achieved
9	Exercise-induced angina	2	Int64	Exercise-induced angina (0--no; 1--yes)
10	ST depression	40	float	ST depression induced by exercise relative to rest
11	Slope	3	Int64	the slope of the peak exercise ST segment (1--upsloping; 2--flat; 3--down sloping)
12	Num of vessels	4	object	Number of major vessels (0--3) coloured by fluoroscopy
13	Thalassemia	3	object	Defect types; 3—normal; 6—fixed defect; 7—reversible defect
14	Diagnosis heart disease	2	Int64	diagnosis of heart disease status (0--no risk; 1--potential risk)

Table 1: Description of the attributes

From this table, we can find that there are mainly three categories of these attributes: binary, slope and Gaussian. The following table categorized the attributes into these three categories.

Distribution Type	Binary	Slope	Gaussian
Attribute	Sex, Fasting Blood Sugar, Resting ECG, Exercise Induced Angina	Chest Pain Type, ST Depression Exercise, ST Segment, Number of vessels	Exercise-induced, Rest blood pressure, Serum cholesterol, Max Heart Rate

Table 2: Categories of the attributes

The following figure visualizes the distributions of each attribute after preprocessing.

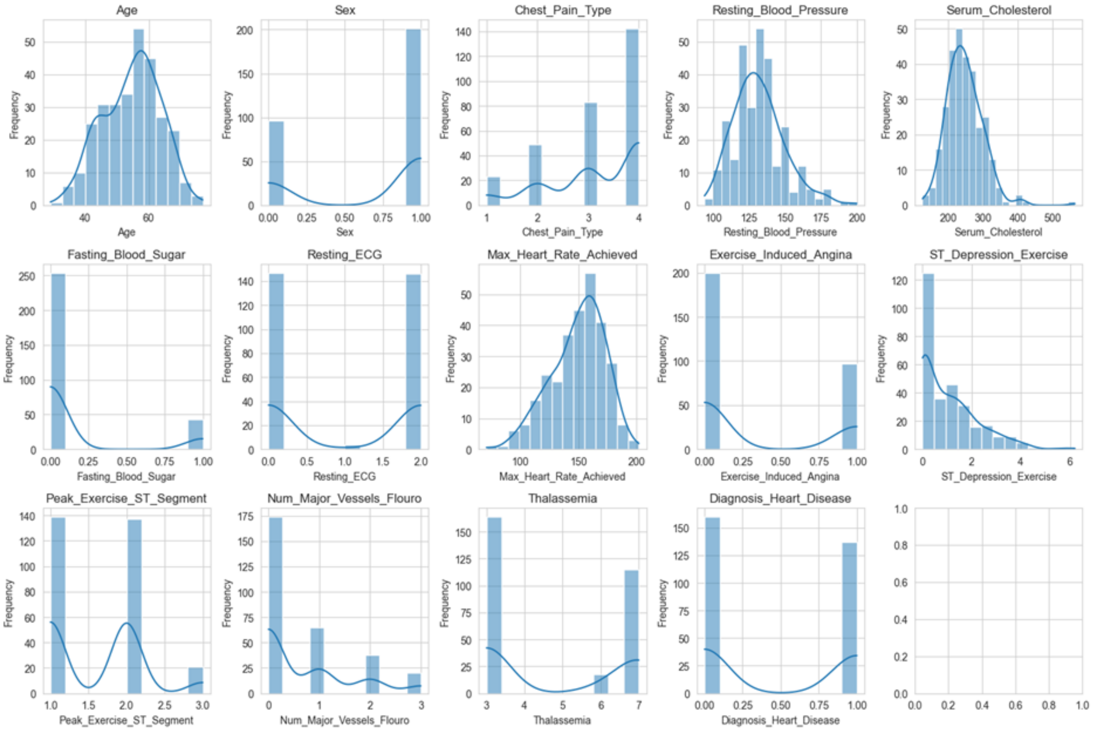

Figure 3: Distributions of the attributes

Methodology for each Model

3.1 Data Preprocessing

In clinical data analysis, some patients' data are often incomplete or missing, which is used ‘?’ to represent them in the excel sheet. Our first step is to use NAN to replace them and drop them from the dataset. After that, we plot the distribution of each feature variable to understand their individual characteristics and spreads. A correlation heatmap is generated to visualize the relationship among those feature variables. PCA is employed to reduce the dimensionality of the dataset while maintaining the most significant variance across features, which allows us to visualize how the samples cluster in a lower dimension. The cleaned dataset is subdivided into 60% training data, 20% validation data, and 20% testing data. The training data is utilized to train the models, validation data is used to fine-tune hyperparameters and validate model performance; and testing data is applied to evaluate the final model performance.

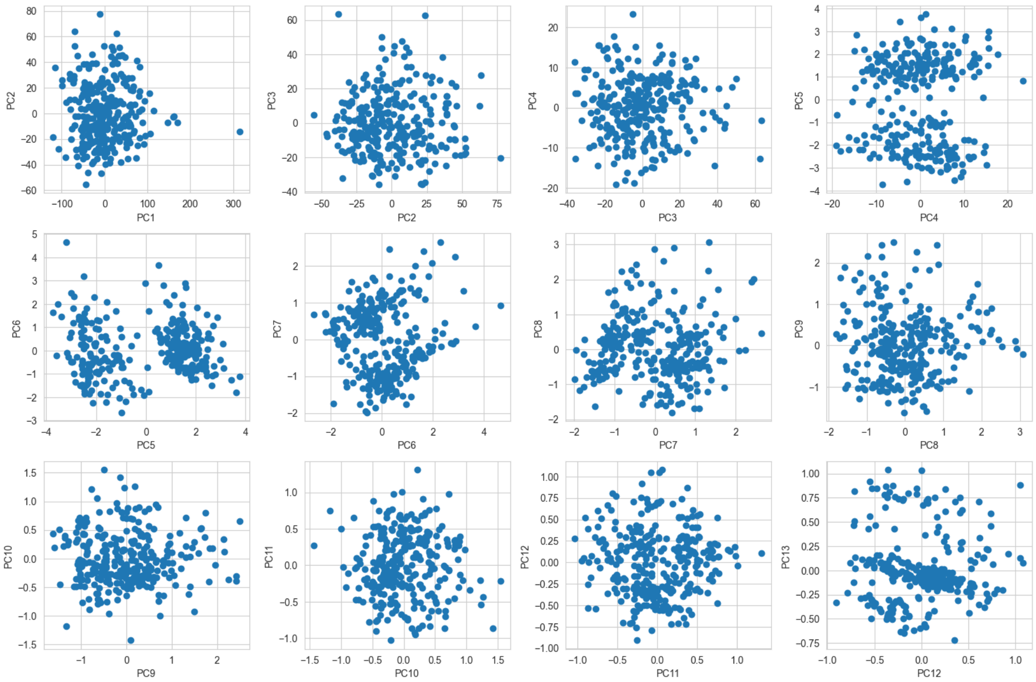

Figure 4: PCA plots of the attributes

3.2 K-Means

Description

Dimensionality Reduction: Here, after PCA, only two components are used. Different numbers are tested, but the accuracy of the model is not affected.

K-Means Clustering: With the data transformed into a two-dimensional space, the K-Means clustering algorithm was employed. K-Means is an unsupervised machine learning algorithm used for clustering data into a predefined number of groups. It starts by initializing cluster centers (centroids) and iteratively performs two key steps: assigning each data point to the nearest centroid based on distance, and then recalculating the centroids as the mean of the assigned points. This process repeats until the centroids stabilize or no data points change clusters. Here, the cluster number should be 2. One group is for those with disease and another group is for those without disease.

Model Results

The KMeans algorithm, set to create two clusters, was trained on the PCA-transformed training data. For the training dataset, the model achieved an accuracy of approximately 59.0%, while the validation dataset saw a slightly higher accuracy of about 63.3%. These results were obtained after aligning the predicted clusters with the actual labels, a necessary step given the unsupervised nature of KMeans. The alignment process involved identifying the most common true label within each cluster and mapping the cluster labels accordingly.

The observed accuracies indicate a moderate level of effectiveness in the model's ability to discern patterns within the heart disease data. However, it's important to interpret these results with caution.

K-Means, being an unsupervised learning algorithm, doesn't inherently account for predefined labels during training. The post-hoc alignment of cluster labels to known labels can provide insights but may not fully capture the nuances of the data's underlying structure. Furthermore, the reduction of dimensionality through PCA, while beneficial for handling complex datasets, could lead to the loss of some relevant information, potentially impacting the clustering outcome. These factors, combined with the intrinsic variability and complexity of medical data, suggest that while the model offers some utility in understanding the dataset, it may require further tuning and possibly integration with other analytical approaches for more robust and clinically relevant insights.

3.3 KNN

Description

Feature extraction: After conducting data visualization, we analyzed the data type and further screened suitable machine learning models. The data is multivariate, and a correlation heatmap reveals that the different variables have low intercorrelation. Based on these characteristics, we can infer that different clinical indicators vary in importance. Therefore, before inputting data into the classification model, we use feature extraction to prioritize different clinical indicators. We employ Lasso for feature extraction because it allows for variable selection and regularization of the data.

K-Nearest Neighbors Algorithm: After data preprocessing, we randomly split the dataset (random seed: 7265) into training data (80%) which is used to train the KNN model and testing data (20%) which is applied to test model performance. We also apply cross-validation while training the model by taking 25% of the training data as the validation set. We then build a Lasso Logistic Regression Model to obtain optimized feature weights.

Model Results

According to the literature, the KNN algorithm is not well-suited for classifying high-dimensional data (multiple features) as the model is easily influenced by noise. In our case, we conducted a test where we fed raw data directly into the KNN model for training and prediction, resulting in an accuracy of only 0.633. Conversely, we first performed feature extraction using Lasso to obtain weights for each feature. Then, we multiplied the raw data by these weights before inputting it into the KNN model. This approach significantly improved our accuracy to 0.883. This indicates that feature extraction is significantly beneficial for training the KNN algorithm on high-dimensional data.

3.4 Logistic Regression

Description

After data preprocessing, we randomly split the dataset (random seed: 454) into training data (80%) and testing data (20%). By using sklearn-pipeline, we then build and train a Lasso regulated logistic regression Model to obtain optimized feature weights. Lastly, we predict the testing data and calculate the accuracy. Moreover, We apply cross-validation while training the model. We randomly divide training data into two parts: training data and validation data (25% of training data). Then we predict the testing data.

Model Results

After training the Lasso regularized logistic regression model, we determined a lambda value of 2.427 (which is the reciprocal of the parameter C in LogisticRegression), and achieved an accuracy of 0.822 in the validation tests. Subsequently, we extracted the coefficients of the logistic regression and identified the optimal coefficients to obtain the weights of the features. Finally, we made predictions on the testing set and plotted a confusion matrix.

Here, we can compare the KNN and Logistic Regression models (both using data that has undergone feature extraction and the same testing data). The accuracy of the KNN model is 0.883, while the accuracy of the Logistic Regression is 0.833. The difference in accuracy between the two is not significant. However, an interesting observation is that Logistic Regression tends to produce fewer False Negatives (FN), while KNN has fewer False Positives (FP). Based on the earlier comparison (KNN model: raw data vs. data with weights), we infer that the KNN model tends to classify testing data as Diseased. In clinical practice, the tendency of Machine Learning models towards FP and FN is particularly important for doctors and patients. In the case of FP, doctors can arrange for patients to undergo further testing or confirm with other medical data. However, if FN occurs, patients may miss the optimal treatment window. Therefore, FN is a more serious error for patients compared to FP.

3.5 Random Forest

Description

Random forest is a model composed of multiple decision trees. In this project, the random forest is a classification model which is used to predict the individual likelihood of heart disease. To be more specific, each branch indicates the training outcome, and each leaf node represents a class label. Data used for training the random forest model is preprocessed as mentioned above in the Data Preprocessing section. The RandomForestClassifier from the sklearn library was employed to construct a robust and versatile machine learning model, highly suitable for our dataset.

Model Results

To find the optimal number of trees (n_estimators), the number of trees (num_tree) values ranging from 10 to 300 with intervals of 20 are used. For each value of num_tree, the OOB score is calculated to estimate the model’s performance. The OOB scores corresponding to different numbers of trees are recorded and plotted against the num_tree to identify the best model performance, thereby providing an unbiased evaluation of the model, akin to cross-validation. The best OOB score is 0.80 corresponding to the best performance at 70 of num_tree. Furthermore, the random forest model is validated using the validation data. The model demonstrated a commendable performance, achieving an accuracy of 79.66%. This close proximity to the OOB score reinforced our confidence in the selected hyperparameters, confirming their optimality and the model's ability to generalize beyond the training data.

After fine-tuning the hyperparameters, the random forest model is retrained with the optimal parameters. By applying the **feature_importances_** function, we identified the most influential features in predicting heart disease. Features with importance greater than 0.06, such as ['Age', 'Chest_Pain_Type', 'Resting_Blood_Pressure', 'Serum_Cholesterol', 'Max_Heart_Rate_Achieved', 'ST_Depression_Exercise', 'Num_Major_Vessels_Flouro', 'Thalassemia'], are crucial in our model’s decision-making process. These features are significant as they provide insights into the factors most strongly associated with heart disease risk, guiding future research and healthcare strategies. Finally, the ultimate test of the model’s efficacy is its performance on the test data, which simulates real-world application. Our model achieved a test accuracy of 86.67%, indicating its robustness and reliability in predicting heart disease.

In conclusion, our methodology demonstrates a thorough and meticulous approach to building a Random Forest model for heart disease prediction. The model's high accuracy and ability to identify key risk factors hold great promise for aiding in early detection and informed healthcare decision-making.

3.6 Naive Bayes

Naive Bayes is a family of probabilistic algorithms that apply Bayes' theorem with a strong assumption of conditional independence between features. It was chosen as one of the models for this project since it provides a probabilistic approach to predicting heart disease, which is an inherently uncertain estimation. A comparison between the Gaussian Naive Bayes and the Bernoulli Naive Bayes models will be presented..

3.6.1 Gaussian Naive Bayes

Description

The Gaussian Naive Bayes model was chosen due to the presence of continuous features in our dataset. Our assumption, based on the nature of the medical data, was that these features could be modeled by a Gaussian distribution. This model is implemented in *scikit-learn* as *GaussianNB*, which we imported and instantiated. By referring to the code, the model is fitted to the training data using the fit method, passing *X_train_gnb* and *ytrain* as parameters. Then, the *predict* method was used to generate predictions for the validation set.

3.6.2 Bernoulli Naive Bayes

Description

The Bernoulli Naive Bayes model is instead suitable for datasets where features are binary. In our case, to apply this model, we had to convert our continuous features into a binary format using the Binarizer class from *scikit-learn*, which we applied to our feature matrix X. After binarizing the features, analogously as with the Gaussian model, the Bernoulli model is trained using the training set without further modifications. The fit method was used with the binarized Xtrain and ytrain. Then, predictions were obtained using the validation set.

Models Results

The performance of the two models is quantified by generating a classification report using the *classification_report* function from scikit-learn, which computes precision, recall, and f1-score for each class. Furthermore, we visualized the results using a confusion matrix, with *conf_matrix_gnb* derived from the *confusion_matrix function*. The confusion matrix was displayed as a heatmap using seaborn's heatmap function, providing a clear visual representation of the model's performance with regard to true positives, false positives, false negatives, and true negatives.

The Gaussian Naive Bayes model has an overall accuracy of 0.81. As can be seen in the classification reports, class 0 has higher precision (0.87) but lower recall (0.79) compared to class 1. The confusion matrix shows 26 TP and 22 TN for class 0, and 7 FP and 4 FN. On the other hand, the Bernoulli Naive Bayes model has a lower overall accuracy of 0.76. Class 0 has higher precision (0.85) but lower recall (0.70) compared to class 1. It can be seen that the confusion matrix shows 23 TP and 22 TN for class 0, and 10 FP and 4 FN. From the matrices, it can be seen that the Gaussian Naive Bayes model performs slightly better than the Bernoulli model in terms of overall accuracy. However, considering the specific costs associated with both overdiagnosis (false positives) and underdiagnosis (false negatives) would yield a more comprehensive evaluation.

Results and Analysis

This study focuses on predicting the likelihood of heart disease in patients. We explore a variety of data mining methods that are effective in forecasting heart disease with precision and minimal reliance on extensive attributes and tests. The research involved classifying a dataset into training and testing subsets. We employed supervised machine learning classification approaches, including K-means, KNN, Logistic Regression, Random Forest, and Naïve Bayes. The algorithms were applied to determine accuracy scores, with a comparative analysis of these scores presented in Figure 6. The results indicated that the KNN and Random forest model yielded the highest accuracy among the four algorithms tested. Future expansions of this research could involve the integration of additional data mining techniques, such as time series analysis, clustering, association rules, support vector machines, and genetic algorithms. Given the constraints of this study, there is potential for improving prediction accuracy of heart disease through the combination of more complex modeling techniques.

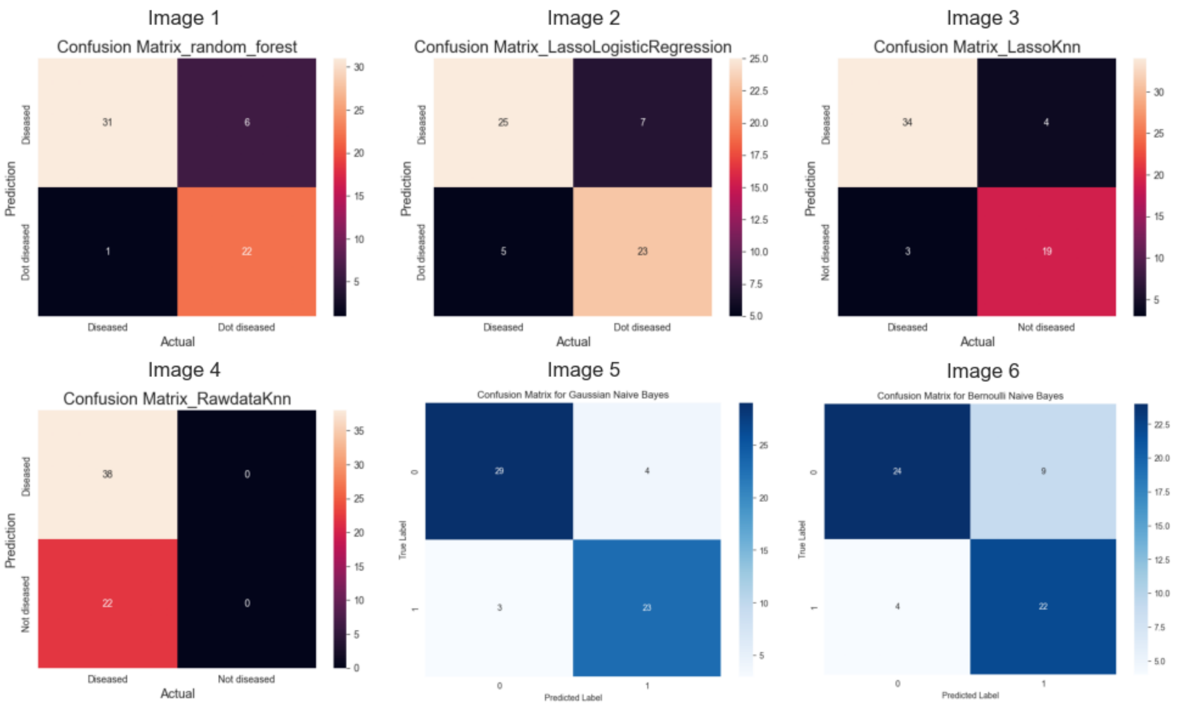

Figure 5: Confusion Matrix

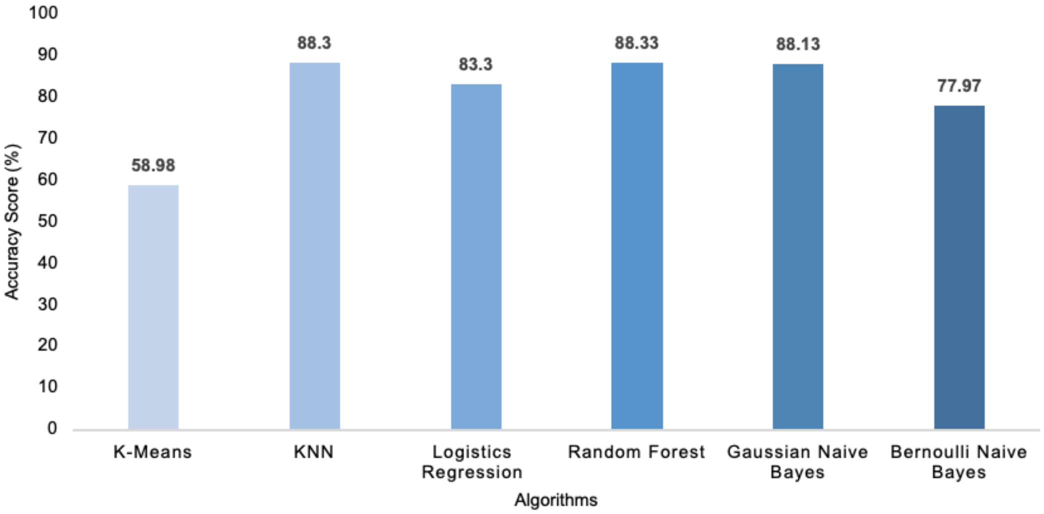

Figure 6: Comparative result of classification techniques

Reference

[1] Shah, D., Patel, S., & Bharti, S. K. (2020). Heart disease prediction using machine learning techniques. *SN Computer Science*, *1*, 1-6.

[2]] Z. Liu, X. Meng, J. Cui, Z. Huang and J. Wu, "Automatic Identification of Abnormalities in 12-Lead ECGs Using Expert Features and Convolutional Neural Networks," 2018 International Conference on Sensor Networks and Signal Processing (SNSP), 2018, pp. 163-167, doi: 10.1109/SNSP.2018.00038.