AA
Project 1 2023/2024
Machine learning models for heart disease presence prediction

***Abstract*—This paper presents a comprehensive review of ML-based approaches for predicting and detecting heart disease. We discuss applying ML to cardiovascular medicine, explore data sources and algorithms, and evaluate the performance of these models.**

***Index Terms*—Machine learning, heart disease, cardiovascular medicine, early detection, predictive modeling, classification algorithms, healthcare analytics**

I. Introduction

A. Motivation

Heart disease remains one of the leading causes of mortality worldwide. With the advancement of technology and the availability of large-scale healthcare data, there is a growing interest in using machine learning techniques to improve the prediction and detection of heart disease in patients. Machine learning offers the potential to analyze complex patterns in patient data, identify risk factors and develop accurate predictive models that can assist healthcare professionals in making informed decisions.

B. Previous work

In recent years, numerous studies have explored the application of machine learning algorithms for various tasks related to heart disease, including risk prediction, early detection, subtype classification, and outcome prognosis. These studies have demonstrated promising results, showcasing the potential of machine learning models to enhance the accuracy and efficiency of cardiovascular healthcare. Some of relevant ones include: *Comparison of logistic regression and Bayesian-based algorithms to estimate posttest probability in patients with suspected coronary artery disease undergoing exercise ECG* [5], *International application of a new probability algorithm for the diagnosis of coronary artery disease* [2] and *Clinical Assessment of the Probability of Coronary Artery Disease* [1]. In out paper we aim to test simple machine learning models and compare their efficiency in detecting heart disease in patients.

II. Data Analysis

A. Data description

The dataset used for this project is processed cleveland dataset [4], most commonly used for training machine learning models for predicting the presence of heart disease in patients. The dataset provides relevant information about each patient in order to predict whether a patient is suffering from a heart disease. The data contains 303 observations with 13 numerical features and a target variable. Listed below are the features:

- **age** : age of the patient in years;
- **sex** : possible values 1 = male, 0 = female;
- **cp** : chest pain type – value 1: typical angina, – value 2: atypical angina, – value 3: non-anginal pain, – value 4: asymptomatic;
- **trestbps** : resting blood pressure (in mm Hg on admission to the hospital);
- **chol** : serum cholestoral in mg/dl;
- **fbs** : variable showing if fasting blood sugar is higher than 120 mg/dl (1 = true; 0 = false);
- **restecg** : resting electrocardiographic results – value 0: normal, – value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), – value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria;
- **thalach** : maximum heart rate achieved;
- **exang** : exercise induced angina (1 = yes; 0 = no);
- **oldpeak** : ST depression induced by exercise relative to rest;
- **slope** : the slope of the peak exercise ST segment – value 1: upsloping, – value 2: flat, – value 3: downsloping;
- **ca** : number of major vessels (0-3) colored by flouroscopy;
- **thal** : a blood disorder called thalassemia: 3 = normal, 6 = fixed defect (no blood flow in some part of the heart), 7 = reversable defect (a blood flow is observed but it is not normal);

B. Data Visualization

In order to better understand the structure, patterns and characteristics of the dataset, it is useful to visualize the data. This can show potential correlation between features and help with choosing the best methods of data preparation such as normalization, scaling, and handling outliers.

Figure 1 shows value distributions of all the features in the dataset. It is evident that there are a few binary features, but most of them have a value range. These histograms tell us that most of the patients are between 40 and 70 years old ("age") and 1/3 of them are male ("sex"). About half of the

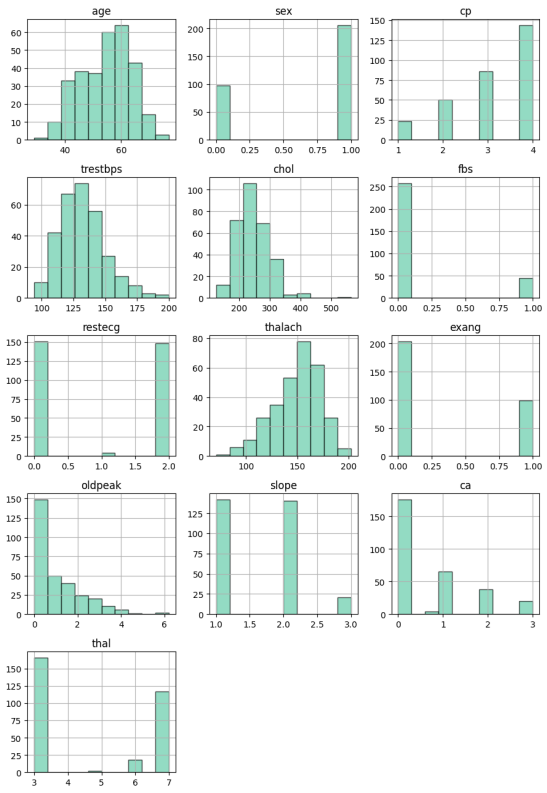

Fig. 1. Histograms of all the features

patients feel different types of chest pain, while the other half has no chest pain (”cp”). Almost all the patients have a resting blood pressure higher than normal upon admission to the hospital (”trestbps”). Half of the patients were showing probable or definite left ventricular hypertrophy (”restecg”) and achieving the maximum heart rate higher than 150 bpm (”thalach”). Interestingly, we can see that the ”thal” feature which identifies thalassemia, has some values that are around 5, but the only ”thal” values that are uniquely identified are 3, 6 and 7.

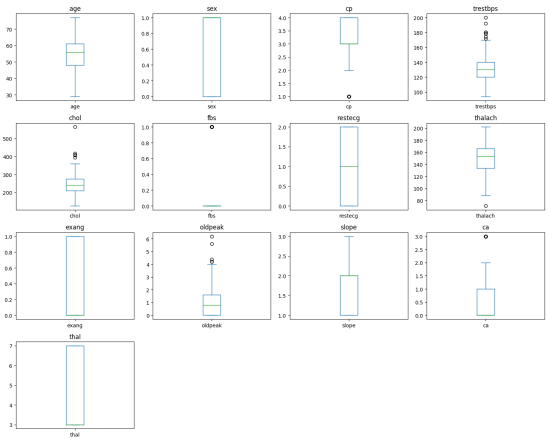

Fig. 2. Identifying outliers for each feature

Figure 2 shows box plots of all the features, which helps us determine if there are any outliers in the values. As seen in the figure, some of the features that contain outliers are: ”cp”, ”trestbps”, ”chol”, ”fbs”, ”thalach”, ”oldpeak” and ”ca”.

Fig. 3. Distribution of a target variable: 0-no heart disease, 1/2/3/4-heart disease

Figure 3 shows the distribution of a target variable. The target variable can take a value of 0, 1, 2, 3 or 4. 0 meaning no heart disease, and any other value indicating the presence of a heart disease. If seen as a binary classification problem, the dataset is very balanced. There is almost an equal amount of patients who suffer from a heart disease as well as those who have no heart disease. This means that there is no need to generate artificial data in order to balance the dataset.

In order to understand the correlation between different features, we plotted the correlation matrix. The correlation matrix shows how the two features are connected. In figure 4 we can observe that almost none of the features are positively correlated.

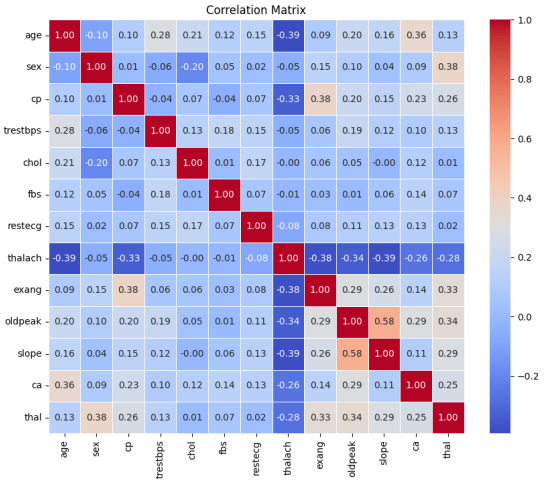

Fig. 4. Correlation matrix of all the features in the dataset. Highest positive correlation is between ”slope” and ”oldpeak” with the correlation coefficient of 0.58. Lowest negative correlation is between ”age” and ”thalach” with the correlation coefficient of -0.39.

The only two features that are somewhat positively correlated with a correlation coefficient of 0.58 are ”slope” and ”oldpeak”. ”thalach” is a feature that has the biggest number

of negative correlations with other features. Those features are: ”age”, ”cp”, ”exang”, ”oldpeak”, ”slope”, ”ca” and ”thal”. The strongest negative correlation is that between ”age” and ”thalach” with the correlation coefficient of -0.39 as shown in figure 5. This negative correlation is easily confirmed based on the features. The older the person is, the maximum heart rate that they can reach is lower [3].

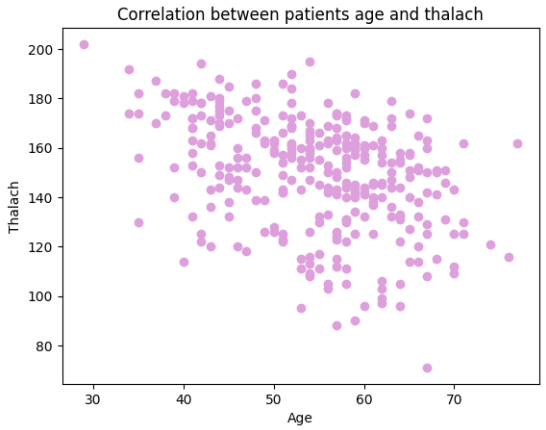

Fig. 5. Correlation between ”age” and ”thalach” with the correlation coefficient of -0.39.

C. Data Preprocessing

The original dataset was cleveland and it was cleaned and preprocessed by the authors of the dataset into a preprocessed.cleveland file used in this project. The preprocessed.cleveland dataset contained several ’?’ values which represented the missing values. The ’?’ values were replaced with nan and then replaced with the mean of the column the value belonged to (the data was interpolated).

III. Model Training and Evaluation

The problem was separated by two different solutions: binary and multiclass classification. Original dataset has target values 0,1,2,3 and 4. Zero meaning no heart disease and others presence of heart disease but in different gravity. Models are supposed to predict one of 5 classes. In the other solution, data was grouped and the problem simplified to a binary classification - either the patient has a heart disease or doesn’t (0,1). Most of the previous papers used grouped data. Expectedly, the results were much better with the binary classification.

A. Model 1 - Random Forest Classifier

The first model that was used to make predictions on the dataset was Random Forest Classifier, an estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Train test split function was used for splitting the data into train and test sets. Two versions were tested - one where the test data made 30 % of the dataset, and the other were it made 50 % of the dataset. For both of the versions, default parameters were used:

- Number of estimators = 100
- Maximum depth = None

Accuracy for the first classifier was 50.55 %. Figure 6 shows the confusion matrix for this model and the other metrics gave the following results:

- Precision: 53.68 %
- Recall: 56.04 %
- F1 Score: 52.01 %

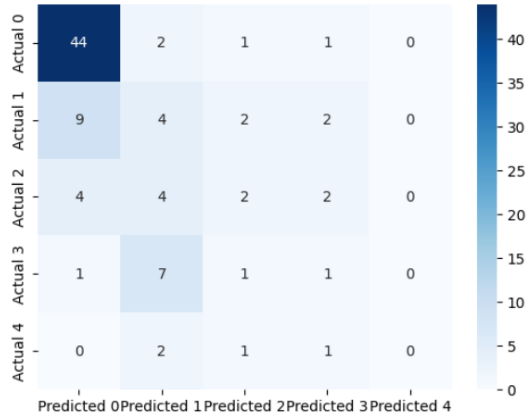

Fig. 6. Confusion matrix for Random forest classifier for multiclass classification with 30 % of dataset used for testing

For the second classifier accuracy score was 54.60 %. Confusion matrix is shown in figure 7. The other metrics on the test sets were as follows:

- Precision: 49.07 %
- Recall: 55.92 %
- F1 Score: 51.82 %

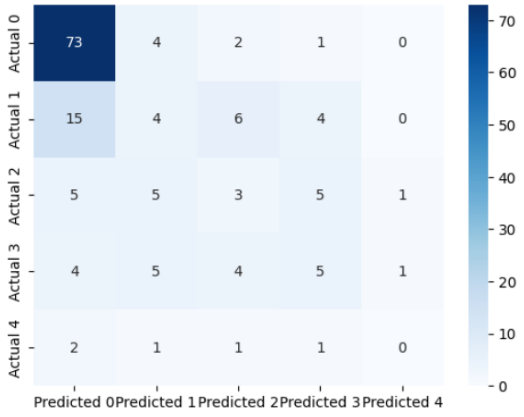

Fig. 7. Confusion matrix for Random forest classifier for multiclass classification with 50 % of dataset used for testing

The Random Forest Classifier model was then trained and tested on the data grouped to two target values 0 and 1. Random Forest Classifier model performed the binary classification. The confusion matrices are shown in figures 8 and 9. The prediction accuracy has improved to 84.6 % for the 30 % test data and the other metrics gave these results:

- Precision: 83.72 %

- Recall: 83.51 %
- F1 Score: 83.53 %

Binary classification for the for 50 % test data gave the following results:

- Precision: 82.23 %
- Recall: 82.24 %
- F1 Score: 82.23 %

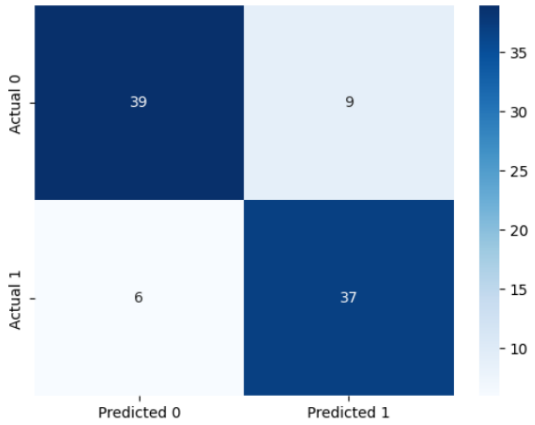

Fig. 8. Confusion matrix for Random forest classifier for binary classification with 30 % of dataset used for testing

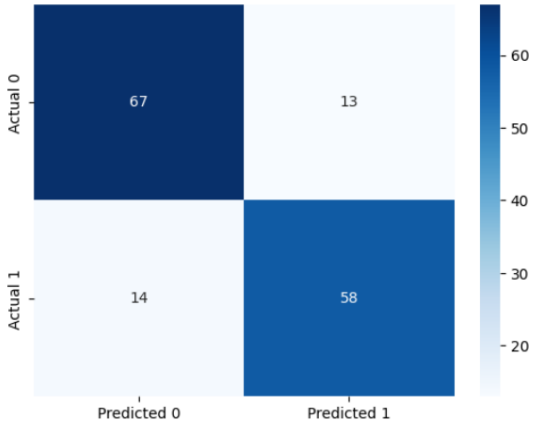

Fig. 9. Confusion matrix for Random forest classifier for binary classification with 50 % of dataset used for testing

Random Forest Clasiffier Recall		
Test %	30	50
Multiclass	53.68	55.92
Binary	83.51	82.24

B. Model 2 - SVM

The second model was SVM - support vector machine. SVM is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space. Three different kernels were tested - linear, radial basis function (rbf) and polynomial. All 3 models produced the recall of 55.92 %.

The SVM model was also trained and tested on the grouped data, but this time model prediction metric *recall* changed for the different models. Linear kernel resulted in the recall of 86.81 %, rbf 65.93 % and polynomial 67.03 %. The confusion matrices for all kernels are shown in figures 10, 11, 12, 13, 14 and 15.

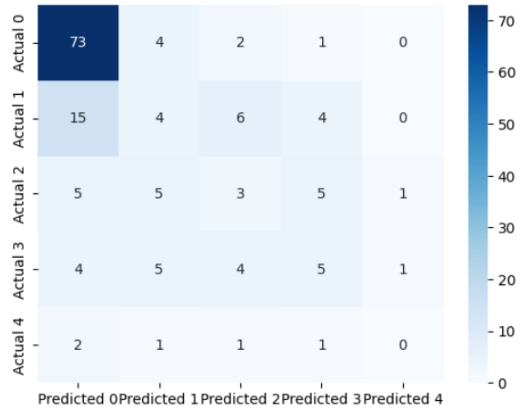

Fig. 10. Confusion matrix for SVM classifier with a linear kernel for multiclass classification

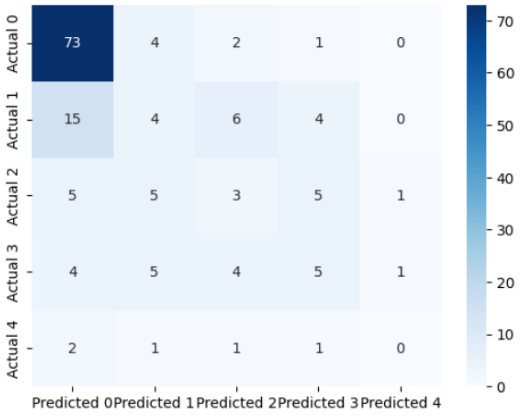

Fig. 11. Confusion matrix for SVM classifier with rbf kernel for multiclass classification

SVM recall			
Kernels	Linear	Rbf	Poly
Multiclass	55.92	55.92	55.92
Binary	86.81	65.9	67.0

C. Model 3 - Logistic regression

The third model was Logistic regression. Logistic regression model was split into train set of 70 % and test set of 30 % of the data and into train set of 80 % and test set of 20 % of the data. Originally, we used the solver *lbfgs* and a 1000 iterations. Using these parameters for the solver and number of iterations didn't get us any results because the solver failed

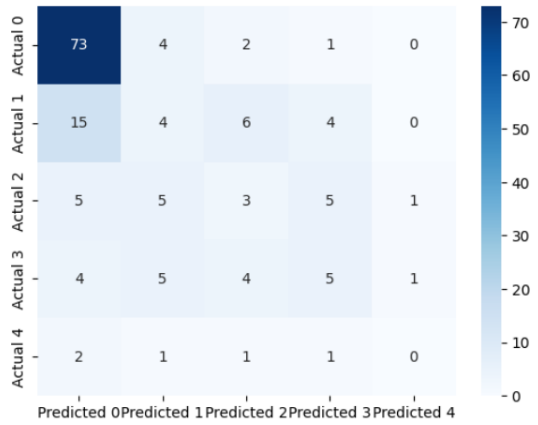

Fig. 12. Confusion matrix for SVM classifier with polynomial kernel for multiclass classification

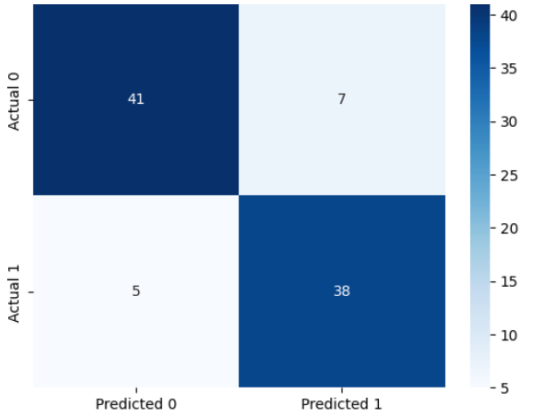

Fig. 13. Confusion matrix for SVM classifier with a linear kernel for binary classification

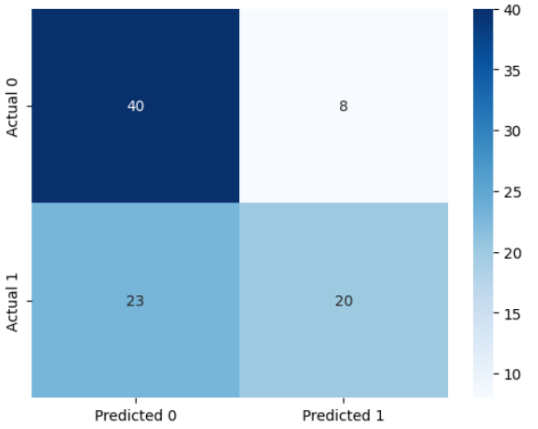

Fig. 14. Confusion matrix for SVM classifier with rbf kernel for binary classification

to converge. In order to fix this, we tried using a different solver *sag* and increasing the number of iterations to 10000. and the final parameters were: multiclass='auto', solver='sag' and maxiteration=10000. The best recall of the model was for the test set of 30 % and it amounted to 59.34 %, the worst

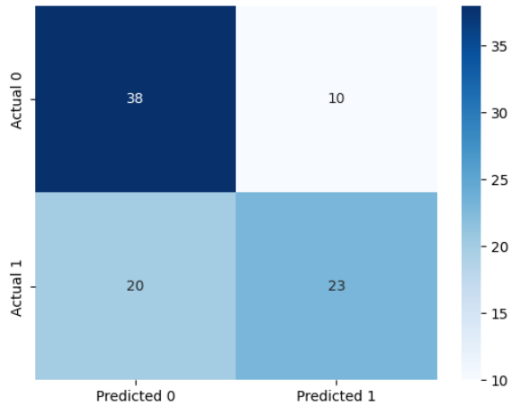

Fig. 15. Confusion matrix for SVM classifier with polynomial kernel for binary classification

out of three models.

The logistic regression model was also trained and tested on the grouped data and the results were - 86.88 % for 20 % of data used as test data and 84.61 % for 30 % of data used as test data. The confusion matrices for logistic regression models used for multiclass and binary classification are shown in figures 16, 17, 18 and 19.

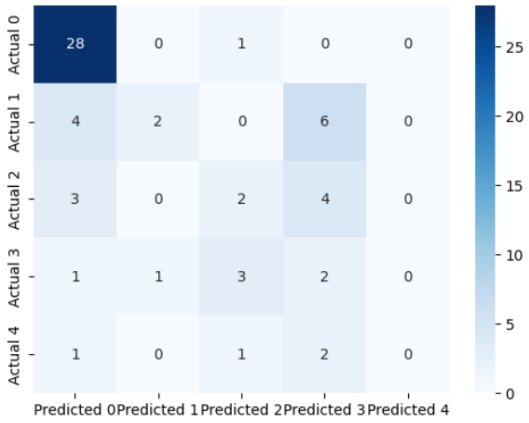

Fig. 16. Confusion matrix for Logistic regression classifier with 20 % of data used for testing for multiclass classification

Logistic Regression recall		
Test %	20	30
Multiclass	55.73	59.34
Binary	86.88	84.61

IV. Conclusions

Out of three models tested with different parameters and train/test data percentages, the best results were achieved with grouped data using logistic regression for binary classification. Overview of all the results is shown in the table below.

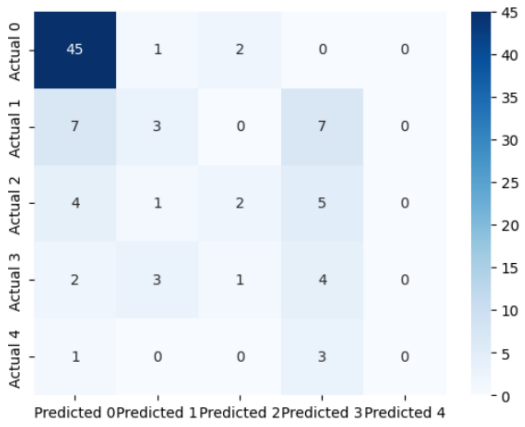

Fig. 17. Confusion matrix for Logistic regression classifier with 30 % of data used for testing for multiclass classification

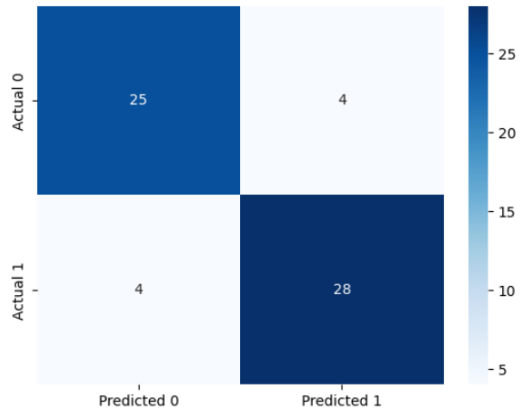

Fig. 18. Confusion matrix for Logistic regression classifier with 20 % of data used for testing for binary classification

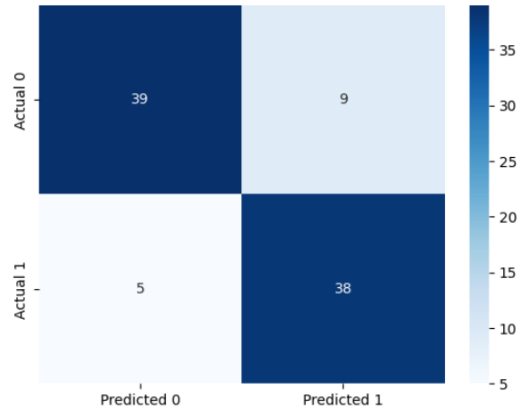

Fig. 19. Confusion matrix for Logistic regression classifier with 30 % of data used for testing for binary classification

Model recall			
Type of classification	RFC	SVM	LR
Multiclass	55.92 %	55.92 %	59.34 %
Binary	83.51 %	86.81 %	86.88 %

Overall, the binary classification showed much better results thanks to the balanced dataset. When trying multiclass classification, some classes were less represented than the other ones. This makes it difficult for the models to train for proper predicting. This problem could be tackled by acquiring more data.

For the further research we recommend testing different models for classification and trying different parameters for the models shown in this paper. Research in cardiovascular area is always important because it could assist in lowering the percentage of mortality when used as early detection system.

References

[1] Marco Bobbio, Robert C. Detrano, Adrian H. Shandling, Myrvin H. Ellestad, J. Clark, Oleh S. Brezden, Ana Abecia, and Diego Martínez-Caro. Clinical assessment of the probability of coronary artery disease. *Medical Decision Making*, 12:197 – 203, 1992.

[2] Robert C. Detrano, András Jánosi, Walter Steinbrunn, Matthias Emil Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64 5:304–10, 1989.

[3] Centers for Disease Control, Prevention, et al. National center for chronic disease prevention and health promotion-nccdphp. 2002.

[4] Steinbrunn William Pfisterer Matthias Janosi, Andras and Robert Detrano. Heart Disease. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C52P4X.

[5] Anthony P Morise, Robert Duval, Robert C. Detrano, Marco Bobbio, George A. Diamond, and George A. Diamond. Comparison of logistic regression and bayesian-based algorithms to estimate posttest probability in patients with suspected coronary artery disease undergoing exercise ecg. *Journal of electrocardiology*, 1992.