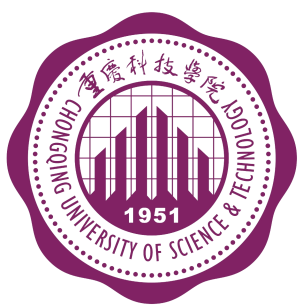
Chongqing University of Science and Technology



**Graduation Project (Dissertation)**

Title Heart Failure Prediction using Machine Learning Techniques

College School of Intelligent Technology and Engineering

Major Computer Science and Technology

Student Name HASAN NAYEM

Student Number 2018490012

Supervisor Yang Su

Reviewer

DD / MM / YYYY

**Declaration of originality of students' graduation project (dissertation)**

I declare with my reputation: the submitted graduation project (dissertation) is the design (research) work and results obtained under the guidance of the supervisor. The project (dissertation) quotes literature, data, drawings and materials from other people have been clearly marked, the conclusions and results in the dissertation are completed independently by myself, and do not include the achievements of others and the use of their materials for obtaining degrees or certificates from Chongqing University of Science and Technology or other educational institutions. The participant who work with me have clearly stated in the paper and expressed gratitude for any contributions to this design (research).

Author: Nayem

DD / MM / YYYY

Abstract

In today's world, heart failure is one of the most common diseases that can lead to a dangerous situation. Every year, nearly 26 million patients are affected by this type of disease. It is challenging to predict heart failure at the right time. According to heart consultants and surgeons, it is unclear whether this association with previous cardiac events translates into predictive value for future events or how this predictive value compares to calculations based on traditional risk factors. The diagnosis of heart failure can be tricky, even for heart failure specialists. Machine learning techniques can be potential ways to assist physicians in heart failure diagnosis. This project aims to evaluate the diagnostic accuracy of a predictive model for heart failure. This knowledge acquisition approach is to evolve the knowledge base with heart failure diagnosis.

Identifying elevated risk for HF as early as possible can help patients avoid potentially fatal events, such as heart attacks. Using a predictive data model, we can use the data in the form of prediction. And with this realization, this project, "Heart Failure Prediction using Machine Learning Techniques," comes into practical usage, benefiting both the doctors and patients.

**Keywords:** heart failure, machine learning, predictive, diagnosis

# 摘要

在当今世界，心力衰竭是可能导致危险情况的最常见疾病之一。每年有近 2600 万患者受到此类疾病的影响。在正确的时间预测心力衰竭是一项挑战。根据心脏顾问和外科医生的说法，目前尚不清楚这种与先前心脏事件的关联是否会转化为未来事件的预测值，或者这种预测值如何与基于传统风险因素的计算进行比较。即使对于心力衰竭专家来说，心力衰竭的诊断也可能很棘手。机器学习技术可以成为帮助医生进行心力衰竭诊断的潜在方法。该项目旨在评估心力衰竭预测模型的诊断准确性。这种知识获取方法是通过心力衰竭诊断来发展知识库。

尽早识别 HF 升高的风险可以帮助患者避免潜在的致命事件，例如心脏病发作。使用预测数据模型，我们可以使用预测形式的数据。有了这个认识，这个“使用机器学习技术进行心力衰竭预测”的项目投入实际使用，使医生和患者都受益。

**关键词：**心力衰竭, 机器学习, 预测, 诊断

Contents

Abstract………………………………………………………………………………………I

摘要……………………………………………………………………………………………II

1 Introduction…………………………………………………………………………………………1

1.1\*\*\*\*\*\*\*\*\*\*\*\*\*…………………………………………………………………1

1.2\*\*\*\*\*\*\*\*\*…………………………………………………………………………2

1.3\*\*\*\*\*\*\*……………………………………………………………………………3

1.3.1 \*\*\*\*\*\*\*\*\*\*\*\*\*…………………………………………………………………………3

1.3.2 \*\*\*\*\*\*\*\*\*\*\*\*…………………………………………………………………………4

1.3.3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*…………………………………………………………………6

2 Analysis……………………………………………………………………………………8

3 Design……………………………………………………………………………………15

4 Implementation……………………………………………………………………………26

5 Testing……………………………………………………………………………………38

[6 Conclusion………………………………………………………………………………40](#_Toc264821404)

……

References………………………………………………………………………………………………50

Acknowledgements……………………………………………………………………………………51

# 1 Introduction

## \*\*\*\*\*\*\*\*

……

## 1.2 \*\*\*\*\*\*

……

 （1.1）

……

Table 1.1 \*\*\*\*\*\*\*\*\*\*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AA | BB | | CC | |
| Ds(m2/s) | Q(kJ/mol) | Dc(m2/s) | Q(kJ/mol) |
| Y3+ | 8.7×10-2 | 565 | 2.1×103 | 530 |
| O2- | 2.6×10-10 | 310 | 1.1×10-3 | 360 |

……

### 1.3.3 \*\*\*\*\*\*

……

|  |
| --- |
| 常压气氛烧结 |
| Figure 1.1 \*\*\*\*\* |

References

[1] L. Robinson, J. Petrey. An Exploration of a Website Design Process and Results of a Usability Study[J]. Medical Reference Services Quarterly, 2019, 38(1):56-69.

[2] M. Richards, N. Ford. Fundamentals of Software Architecture: An Engineering Approach[M]. O’Reilly Media, 2020.

[3] V. Gruhn, R. Striemer. The Essence of Software Engineering[M]. Springer, 2018.

…….

Contents

**Abstract**………………………………………………………………………………...I

**摘要**.......................................................................................................................................II

**1 Introduction**.......................................................................................................................

1.1 Background of Study.......................................................................................................

1.2 Problem Statement..........................................................................................................

1.3 Methods..........................................................................................................................

1.4 Objectives.........................................................................................................................

1.5 Hardware and Software Requirements...........................................................................

**2 Literature Review**.............................................................................................................

2.1 Case study......................................................................................................................

Improving risk prediction in heart failure using ML..........................................................

**1 Introduction**

Health is one of humanity's most significant global challenges. The World Health Organization (WHO) has stated that an individual's right to health is fundamental. Thus, to maintain people's fitness and health, appropriate health care services should be provided. Heart problems account for 31% of all deaths worldwide. Heart disease diagnosis and treatment are highly complex, particularly in developing countries, due to a lack of diagnostic devices and physician and other resource shortages, which impairs accurate prediction and treatment of cardiac patients. With this in mind, in recent years, computer technology and machine learning techniques have been used to develop software that assists physicians in making early diagnoses of heart disease. Early detection of the disease and predicting a person's likelihood of developing heart disease can help reduce death rates. Medical data mining techniques are used to extract meaningful patterns and knowledge from medical data. Medical information is redundant, has multiple attributions, is incomplete, and has a solid temporal relationship. Effectively utilizing massive amounts of data has become a significant issue for the health sector. Data mining provides the methodology and technology for transforming these data mounds into actionable information for decision-making. This cardiovascular disease prediction system would enable cardiologists to make faster decisions, allowing more patients to receive treatment in a shorter period, potentially saving millions of lives.

**1.1 Background of Study**

Data mining has enormous potential for the healthcare industry, as it enables health systems to analyze and predict disease (identify) using data mining. We can analyze the data and forecast the disease, which saves money and time. According to the World Health Organization, approximately 23.6 million people will die as a result of heart fragility between now and 2030. Thus, anticipation of coronary illness should be eliminated to reduce the risk. Heart disease risk factors are classified into two broad categories. The first category includes risk factors that we cannot change, such as age, gender, and family history. We change the second group because it contains risk factors such as smoking, poor eating habits, and high cholesterol. Thus, risk factors belonging to the second class can be eliminated or controlled through lifestyle changes and medication.

**1.2 Problem Statement**

Heart disease detection and prediction are critical issues. As a result, considerable research has been conducted in this area. We classify existing research into two categories: the first analyses the approach for selecting the most relevant patient based on feature selection, and the second analyses high-accuracy learning algorithms.

**1.3 Methods**

Data mining is a methodology and technology for turning massive amounts of data into useful information for decision-making. This research compares various machine learning techniques such as Logistic Regression, Decision Tree, Support Vector Machine, and Random Forest to predict heart disease. The naive mathematician used probability to predict heart disease, the SVM technique used classification and regression, and the Random Forest technique used a variety of decision trees. These algorithms exhibit varying degrees of accuracy. We will attempt to fine-tune our processes to improve their accuracy, which will benefit more accurate predictions.

**1.4 Objectives**

The primary goal of this study is to predict whether or not a patient has heart disease using a variety of machine learning algorithms and a qualified dataset. Determine the correlations between various attributes. We are developing a solid understanding of our proposed data mining techniques, analyzing the results, and comparing the results of multiple data mining techniques. We will examine our techniques to determine if there is any way to improve our results.

The main objective of this project is to assist doctors in diagnosing patients with heart failure risk using machine learning. As doctors need to follow various factors to diagnose a patient with heart disease, they must follow every medical report to correspond to fatal events. And using machine learning and a simple user interface is effortless for both doctors and patients.

**1.5 Hardware and Software Requirements**

There are some software and tools are given below which are used for this project

**Hardware:**

Computer (i5 8400 CPU, 16gb ram, 521gb SSD, 6gb GPU)

**Software:**

* Google Chrome
* Microsoft Excel
* Vs-Code

**Technologies used:**

* Programming language & Dependencies: Python
* Sci-kit learn
* Matplotlib
* NumPy
* Pandas
* Seaborn
* Google Colaboratory
* Flask

**2 Literature Review**

This chapter will go over various machine learning classifiers and previous work on heart disease. In machine learning, we can use different algorithms, also known as classifiers, to help us make predictions for our project. Four algorithms will be used in our project to predict if someone have heart disease and do not have heart disease. We will use four algorithms because it will allow us to make more accurate and reliable predictions. Because if we use one algorithm or classifier and don't have anything else to compare it to, we can't say it's a reliable prediction because it might not be very accurate. Still, it might not be the best or most appropriate one for our scenario. Whereas if we use more than one algorithm or classifier, in our case, four of them, we can compare them and understand that something is wrong if one classifier provides accuracy that is not even in the ballpark of the other algorithm's provided accuracy. It's possible that the algorithm isn't up to the task or that we made a coding error. As a result, using more than one algorithm is critical for any prediction-based system.

The algorithms we have chosen to use in our project are as follows: 1. Logistic Regression, 2. Decision Tree, 3. SVM (support vector machine), and 4. Random Forest. Each of these algorithms will be talked about in more detail below. Finally, we'll look at previous work that has been done and show how it has changed over time and what changes we were able to use in our project. We'll go over some of the previous research on heart disease prediction.

To begin, I looked into the significance of a patient's age in determining whether or not they are at risk of developing heart failure. We can conclude that the likelihood of a death event increases as one's age increases (i.e., the older a patient is, the more likely he is to have heart failure). Many studies have been conducted on blood tests to predict heart disease. Many clues about our heart condition can be found in our blood. Our blood cholesterol level is a clear sign that we are more likely to have a heart attack.

Measurement of blood flow out of your heart's lower chambers (ventricles) is called ejection fraction (EF). This is the percentage of blood expelled from your left ventricle when your heart pumps. Your ejection fraction can help doctors determine if you have certain heart problems, particularly a specific type of heart failure. And this does not mean that your heart has stopped working but that it cannot pump enough blood for the demands placed on it by your body's systems. When choosing the best treatment for you, your ejection fraction will play an important role.

Other things in our blood can also help our doctor figure out if we have heart failure or if we're at risk of getting plaque deposits in our arteries, which is called atherosclerosis. It's important that a single blood test won't tell us our risk of cardiovascular disease. People with diabetes, high blood pressure, and smoking habits are more likely to develop coronary heart disease. Let's look at some of the blood tests that can be used to diagnose and manage heart disease. First and foremost, we can perform a cholesterol test. A cholesterol test, also known as a lipid panel or lipid profile, determines the levels of fats (lipids) in our blood. The measurements can predict our risk of heart attack or another type of heart disease. Typically, the test includes measurements of – (1) total cholesterol. This is a total of our blood cholesterol levels. If it is high, we are at a high risk of a heart attack. In an ideal state, total cholesterol should be less than 200 mg per deciliter (mg/dL) or 5.2 millimoles per liter (mmol/L). (2) LDL cholesterol (low-density lipoprotein cholesterol). This kind of cholesterol is referred to as "bad cholesterol." Too much of it in the blood causes fatty deposits to form in our arteries, reducing blood flow. These plaque deposits frequently rupture, causing serious heart and tube problems. To stay fit, our LDL cholesterol level should be less than 130 mg/dL. A lower level of less than 100 mg/dL is preferable, especially if we have diabetes or a history of heart attack, heart stents, heart bypass surgery, or other heart/vascular conditions. (3) HDL cholesterol (high-density lipoprotein). This is known as the 'good' cholesterin because it aids in the removal of cholesterin, keeping arteries open and blood flowing freely. A man's HDL cholesterol level should be greater than 40 mg/dL, and a woman should be greater than 50 mg/dL. (4) Triglycerides Are a different type of fat found in the blood. High lipid levels usually indicate that you consume more calories than you burn. If it becomes too high, it raises our risk of heart disease. Our triglyceride level should ideally be less than 150 mg/dL. The ideal triglyceride level is less than 100 mg/dL. (5) Low-density lipoprotein cholesterol (LDL). The distinction between total cholesterol and HDL cholesterol is non-high-density compound protein cholesterol (HDL-C). Non-HDL-C contains cholesterol in lipoprotein particles, contributing to artery hardening (atherosclerosis). Beta-lipoprotein (LDL), compound protein (a), intermediate-density compound protein, and very-low-density compound protein are examples of these. In some cases, the Non-HDL-C fraction may be a better risk predictor than LDL cholesterol. Let us now go over High-sensitivity C reactive protein in greater detail. C-reactive protein, or CRP, is produced by your liver as part of your body's response to injury or infection (inflammatory response). CRP is a useful test to check whether a patient's body is inflamed or not. However, high sensitivity CRP tests cannot determine where in the body this is occurring or why it is occurring. Inflammation is a key factor in developing atherosclerosis, a condition in which fatty deposits clog our arteries. CRP alone will not tell our doctor about our risk of heart disease. However, combining HS-CRP test results with other blood test results and risk factors for heart disease can get a better picture of our heart health. Because CRP levels vary, the test should be repeated twice, two weeks apart, to obtain a good and acceptable result. An HS-CRP level greater than 2.0 mg/L indicates an increased risk of heart disease. This test screening is not recommended for people who do not have symptoms or are at high risk of developing heart disease. Cholesterol-lowering statin medications may lower CRP levels and lower your risk of heart disease. The lipoprotein will be the subject of our next discussion (a). Lipoprotein (a), also known as Lp (a), is a subtype of LDL cholesterol. Our Lp(a) levels are determined by our genes and are not generally influenced by our lifestyle. However, a high level of Lp(a) may be a sign of an increased risk of heart disease, though the extent of the risk is unknown. If we already have atherosclerosis or heart disease but appear to have normal cholesterol levels, our doctor may order an Lp(a) test. If we have a family history of early-onset heart disease or sudden death, we may be tested for Lp(a). One thing to note here is that drugs to lower Lp(a) are being developed, but it is not yet clear what effect lowering Lp(a) will have on heart disease risk. People with high Lp(a) cholesterol levels are generally advised to maintain a low LDL cholesterol level. Following that, we will discuss plasma ceramides. This is a new type of test that measures ceramide levels in the blood. Ceramides are produced by all of our cells and play an important role in the growth, function, and death of many different tissue types. Ceramides are squarely related to coronary artery disease and are transported through the blood by lipoproteins. Three specific ceramides have been linked to artery plaque buildup and insulin resistance also causing a higher serum creatinine level. This increased level of ceramides in the blood indicates a higher risk of cardiovascular disease within one to five years. In hypertensive individuals, the elderly, and patients with myocardial infarction or stroke, elevated serum creatinine has been linked to increased mortality. All-cause mortality and significant ischemic heart disease/stroke events have been linked to a higher serum creatinine concentration in middle-aged men.

Also, when hyponatremia is diagnosed as an electrolytic abnormality, the incidence ranges from 15–30 percent for adults. Hyponatremia is a common complication of heart failure, according to the Organized Program to Initiate Life-Saving Treatment in Patients Hospitalized for Heart Failure (OPTIMIZE-HF) registry. Patients with hyponatremia had a higher mortality rate and a longer hospital stay in this registry compared to those with higher sodium levels. Hyponatremia treatment in heart failure has only been evaluated in a few studies. There are currently no recommendations for how to treat heart failure patients with low serum sodium levels; treatment typically involves fluid restriction, which has not been studied clinically. Heart failure patients with hyponatremia may benefit from the use of solute-free water excretion-enhancing vasopressin receptor antagonists.

As we can see, many studies have been conducted in response to the demand for a system that can predict heart disease.

**2.1 Case study**

**Improving risk prediction in heart failure using ML**

Machine learning is used in various healthcare applications, ranging from case management of common chronic diseases to leveraging patient health data with external influences such as pollution exposure and weather factors.

By crunching large amounts of data, machine learning technology can assist healthcare professionals in developing precise medicine solutions that are tailored to individual characteristics.

"AI will impact physicians and hospitals because it will play a key role in clinical decision support, enabling earlier disease detection and tailored treatment plans to ensure optimal outcomes". "It can also demonstrate and educate patients on potential disease pathways and outcomes given different treatment options." It has the potential to improve hospital and healthcare system efficiency while lowering costs."

Some machine learning companies are researching the ability to organize and provide doctors with patient information during a telemedicine session and capture information during the virtual visit to aid in increased efficiency and workflow.

Heart disease detection and prediction is a critical issue. As a result, much research has been conducted in this area. Existing work is divided into two categories: The first discusses selecting the most relevant patient. The second is to investigate the learning algorithms that provide high accuracy by selecting features. After thorough testing and modification with different data sets, our ml model will be more relevant in predicting Heart Failures. By providing a simple user interface, this work can be done more efficiently.

**3 Algorithm Discussion**

**3.1 Algorithms**

We can use various algorithms, also known as classifiers, in machine learning to help us predict our project. Our project hopes to indicate the number of patients with heart disease and those who do not have heart disease by applying four algorithms to our data set. We're going to use four because it will allow us to make more accurate and reliable predictions. Because if we use one algorithm or classifier and don't have anything else to compare it to, we can't say it's a reliable prediction because it might be very accurate. Still, it might not be the best or most appropriate one for our scenario. Whereas if we use more than one algorithm or classifier, in our case, four of them, we can compare them and understand that something is wrong if one classifier provides accuracy that is not even in the ballpark of the other algorithm's provided accuracy. It's possible that the algorithm isn't up to the task or that we made a coding error. As a result, using multiple algorithms is critical for any prediction-based system. The algorithms we have chosen to use in our project are as follows: 1. Logistic Regression, 2. Decision Tree, 3. SVM (support vector machine), and 4. Random Forest. Each of these algorithms will be covered in detail further down

**3.1.1 Logistic Regression**

We will use Logistic Regression for our first algorithm. It is currently one of the most popular machine learning algorithms. Modeling the probability of a discrete outcome given an input variable is known as logistic regression. The most common logistic regression models produce a binary output, which can take two values: true/false, yes/no, etc. Multinomial logistic regression can model scenarios with more than two discrete outcomes.

The (binary) logistic model (or logit model) is a statistical model that models the likelihood of one event (out of two alternatives) occurring by making the event's log-odds (logarithm of the odds) a linear combination of one or more independent variables ("predictors"). The logistic model parameters are estimated using logistic regression (or logit regression) in regression analysis (the coefficients in the linear combination). An index variable encodes a single binary dependent variable in binary logistic regression. With two values ​​labeled "0" and "1", the independent variables can be either binary (two classes, encoded by an indicator variable) or continuous variables (any real value). The corresponding probability of the value labeled "1" can vary between 0 (definitely the value "0") and 1 (definitely the value "1"); thus, the labeling, the logistic function converts log-odds to probability, thus the name. The logit, or logistic unit, is the unit of measurement for the log-odds scale, hence the alternate names.

Binary variables are widely used in statistics to model the probability of a particular class or event occurring, such as the probability of a team winning, a patient being healthy, and so on (see Applications). The logistic model has been the most recent widely used binary regression model. When there are more than two possible values (for example, whether an image is of a cat, dog, lion, or another animal), binary variables can be converted to categorical variables, and binary logistic regression can be converted to multinomial logistic regression. Ordinal logistic regression can be used if the multiple categories are ordered (for example, the proportional odds ordinal logistic model). For additional extensions, see Extensions. The logistic regression model does not perform statistical classification (it is not a classifier). Still, it can be used to create one, for example, by selecting a cutoff value and classifying inputs with probability more significant than the cutoff as one class and those with chance less than the cutoff as the other; this is a common way to create a binary classifier.

Alternatives include similar linear models for binary variables that use a different sigmoid function (to convert the linear combination to a probability) instead of the logistic function. The logistic model is defined by changing one of the independent variables multiplicatively scales the odds of a given outcome constant. Each independent variable has its parameter; this generalizes the odds ratio for a binary dependent variable. In a broader sense, the logistic function is the natural parameter for the Bernoulli distribution and is thus the "simplest" way to convert an actual number to a probability. It maximizes entropy (minimizes added information) and makes the fewest assumptions possible about the modeled data; see Maximum entropy.

Logistic regression is a helpful analysis method for classification problems in which you want to know if a new sample belongs in one of several categories.

**3.1.2 Decision Tree**

We've already discussed our first machine learning algorithm, Logistic Regression. We'll talk about our second machine algorithm, the decision tree classifier, which we'll use for prediction.

They can be used to solve classification and regression problems. Now you may be wondering why we prefer the Decision tree classifier over other classifiers. There are two possibilities. One example of an 8 Algorithm Discussion is that decision trees frequently attempt to mimic how the human brain thinks, making it relatively easy to understand the data and reach sound conclusions or interpretations. The second reason is that decision trees, as opposed to a black box algorithm like SVM, KNN, and others, allow us to see the logic for the data to interpret. It is straightforward, making it a favorite among this generation's programmers. Now that we've discussed why a Decision Tree is valuable let's look at a Decision Tree Classifier. To begin, a decision tree is a tree with many nodes, each of which represents a feature (attribute), each link (branch) means a decision, also known as a rule, and each leaf of the tree represents an outcome, also known as a categorical or continuous value. The idea is to build a tree for all the data and get a result for each leaf. Let us now look at how to make a decision tree classifier.

Two different algorithms can be used to create a decision tree. CART (classification and Regression Trees) is one, and ID3 is another (iterative Dichotomies). For ID3, we start with the x value in the column and a y value that stays at the end of the queue and only has "YES" or "NO" values. For the chart above, we have (outlook, temp, humidity, windy) as our x values and play, which only has two options, "YES" or "NO," at the bottom of the column as our y value. Now we must map the x and y coordinates. As we can see, it's a binary classification problem, so let's use the ID3 algorithm to build the tree. To create a tree, we need to start with a root node, and we must choose one first. As a general rule, the root node should be the feature that has the most significant influence on the value y. Then we proceed by selecting the next most influential feature as the next node. We will use the concept of entropy, which measures the amount of uncertainty in the data set in this case. We must compute the entropy for all categorical values for the binary classification problem.

To summarize, we must first compute the entropy for the data set. Then, for each attribute/feature, we must first add entropy for all categorical values, take the average value information entropy for the current attribute, and compute how much we have gained for the present point. Then we must select the highest gain attribute and repeat until we have obtained our desired tree. That is the ID3 procedure. The decision tree classifier is based on another algorithm known as CART, which stands for classification and regression trees. We use the Gini Index as our cost function in this algorithm to evaluate splits in the dataset. Because our target variable is binary, it can only take two values (yes and no). And as we all know, there are 4 Algorithms and nine possible combinations. Now we need to calculate the Gini score, which will give us a good idea of how to divide the data. If we can achieve a Gini score of 0, we can consider it a perfect separation, whereas a split of 50/50 would be the worst-case scenario. The question now is how to calculate the Gini index value. The Gini index will remain similar if the target variable is a categorical variable with multiple levels. The Gini index for the dataset is computed as the first step in this procedure. Then, for each feature, we must add the Gini index for all categorical values and the average information entropy for the current attribute before calculating the Gini gain. After that, we can select the best Gini gain attribute and repeat until we have our desired tree. That is how the decision tree algorithm operates.

**3.1.3 Support Vector Machine (SVM)**

Before getting into SVM (support vector machine), we should be familiar with linear regression and logistic regression algorithms. If not, it is recommended that they be examined before proceeding to the support vector machine. The support vector machine is another simple algorithm that every machine learning expert should have in their arsenal. Many people prefer the support vector machine in this scenario because it produces significant accuracy with less computation power. Furthermore, support vector machines, abbreviated as SVM, can be used for regression and classification tasks. However, it is widely used in classification objectives. Now, let us go over what a support vector machine is. The use case of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N – the number of characteristics) that classifies the data points. Numerous hyperplanes could be used to separate the two classes of data points.

However, to achieve our goal, we must find a plane with the greatest margin, for example, the most significant distance between data points from both classes. Increasing the margin distance provides reinforcement, allowing us to classify future data points better. In this context, hyperplanes serve as decision boundaries, assisting in organizing data points. On each side of the hyperplane, data points will be allocated to totally distinct categories. We want to mention that the number of features determines the size of the hyperplane. If we can determine that the number of input features is two, the hyperplane is simply a line. When there are three input features, the hyperplane becomes a two-dimensional plane. We can deduce that when the number of elements exceeds three, it becomes complicated to imagine. We must never forget that support vectors are data points close to the hyperplane and influence its position and orientation. We maximize the classifier's margin using these support vectors. If we remove the support vectors, the status of the hyperplane will change. These are the points that will eventually help us build our SVM. Now we'll talk a bit about the significant margin intuition. First, we may state that in logistic regression, we use the sigmoid function to compress the linear function outputs into the range [0, 1]. The squished value is labeled one if it exceeds a threshold value (.5). It's designated 0 otherwise. In SVM, we take the linear function's output and assign it to one of two classes: one if it's more significant than one, and another if it's -1. We can obtain this reinforcement range of values ([-1, 1]) by changing the threshold value in SVM to 1 and -1.

Which acts as a margin. Next, we'll go over the cost function and gradient updates—the SVM algorithm focus on maximizing the margin between the data points and the hyperplane. Hinged loss is a loss operation that aids in margin maximization. If the predicted and actual values are of the same sign, the cost is zero. If they are not, we compute the loss value. We also add a regularization parameter to the value operation. The regularization parameter's goal is to balance margin maximization and loss. The cost functions look like this after you add the regularization parameter. With the loss function in hand, we can find the gradients by taking partial derivatives concerning the weights. We can update our weights by using the angles. Now that there is no misclassification, we can consider our options. After the model correctly predicts the class of our data point, we only need to update the gradient from the regularization parameter. And if the situation does arise in which there is a misclassification. For example, if our model makes a mistake in predicting the class of a data point, we include the loss and the regularization parameter to perform gradient updates. The following is an example of SVM implementation in Python: We will remove one of the classes from the iris dataset because it contains three categories. Now, we are left with a binary class classification issue. We'd also like to point out that we have four features at our disposal. We will only make use of two segments. For example, sepal and petal lengths. We plot these two attributes to visualize them; the graph above shows that a linear line may be used to divide the data points.

We extract the specified options and divide them into coaching and testing knowledge. The data is utilized for training 90% of the time and testing 10% of the time. So now is an excellent time to build our SVM model with the NumPy library. The learning rate is set to a (0.0001), and the regularization parameter lambda is set to 1/epochs. As a result, the regularizing value reduces the number of epochs increases. Because the test data only has 10 data points, it is time to clip the weights. We extract characteristics from test data and predict values. We collect the predictions, compare them to the specific values, and print the accuracy of our model. There is another simple way to put the SVM formula into action. We can use the sci-kit learn library to implement the SVM model and select the connected functions. When there are too few lines of code, the number of lines of code is drastically decreased. We can say that the support vector machine is an elegant and robust algorithm. We should use it wisely.

**3.1.4 RANDOM FOREST**

Before we start, a small brief on the random forest classifier is in order. It is a versatile, user-friendly machine learning algorithm that produces excellent results, even without hyperparameter tuning. Furthermore, because of its simplicity and the fact that it can be used for both classification and regression tasks, it is regarded as one of the most widely used algorithms. We will demonstrate how the random forest algorithm works and several other important aspects in the following sections. First, let us look at how it works. To begin with, random forest is a supervised learning algorithm. It creates a forest and makes it somewhat random, as implied by the name. The 'forest' it creates is an ensemble of decision trees, which are typically trained using the 'bagging' method. The general idea behind the textile technique is that combining different learning models will improve the outcome. To put it simply, a random forest creates multiple decision trees and merges them to produce a more accurate and stable prediction. The random forest has a significant advantage in that it is used for both classification and regression problems, which comprise the majority of current machine learning systems. Because classification is sometimes considered the building block of machine learning, I will refer to the random forest in classification. Below is an example of how a random forest with two trees might look: A random forest has nearly the same hyperparameters as decision trees and bagging classifiers. Fortunately, you do not need to combine a decision tree and a bagging classifier and can use the Random Forest classifier class. We've already mentioned that we can use the random forest to handle regression tasks using the random forest repressor. While growing the trees, Random Forest adds more randomness to the model. Instead of selecting the most important feature when creating a node, it selects the best feature from a random subset of features. This results in a good diversity, which usually results in a more robust model. As a result, the algorithm for splitting a node in a random forest considers only a random subset of the features. You can even make the trees appear more random. Furthermore, rather than searching for the best possible thresholds, use random thresholds for each feature (like a normal decision tree). Let us now look at the random forest classifier's real-world analogy. Consider a man named Andrew who wants to decide where he wants to go on a one-year vacation. He seeks advice from people he knows. First, he goes to find a friend, and then he asks Andrew where he has traveled in the past and whether he enjoyed it. Based on the responses, he can make some suggestions to St. Andrew. This is a decision tree algorithm approach in general. Using Andrew's responses, Andrew's friend devised rules to guide his decision about what to recommend. Following that, Andrew begins to ask more and more of his friends for advice, and they, in turn, ask him different questions from which they can derive some recommendations. Then he chose to go to places recommended to him the most, which is the general or alternatively known as the typical Random Forest algorithm approach. Now we'll go over some of the features of Random Forest. To begin, another outstanding feature of the random forest algorithm is its ease of measuring the relative importance of each feature on prediction. Sklearn has a good tool for this that measures the importance of an option by observing how much impurity is reduced by tree nodes that use that feature across all trees in the forest. When coaching, it computes this score for each feature and scales the results so that the sum of all importance is one. If you're unfamiliar with how a decision tree works or if you're not sure what a leaf or node is, Wikipedia has a good description. Is.A decision tree's internal nodes represent a "test" on an attribute (for example, if a coin flip results in heads or tails), each branch represents the outcome of the check, and each leaf node represents a category label (decision taken when computing all attributes). A leaf could be a node with no children. Now that we've examined the feature importance, we can decide which features we might want to remove because they don't contribute enough to the prediction process. This is significant because, in general, the more features we have, the more likely our model will suffer from overfitting, and vice versa. Below is a table and a visualization showing the significance of 13 features we used during the supervised classification project with the famous Titanic dataset that we found on Kaggle. Now we will discuss the various aspects of decision trees and random forests. Although we have already stated that a random forest is a collection of decision trees, there are some differences. For example, suppose we feed a decision tree a training dataset with features and labels. It will create a set of rules that can be used to make predictions. For example, if you want to predict whether or not someone will click on an internet advertisement, you can use it. It is possible to collect the ads that the person has clicked in the past and a few features that describe the decision he has made. When we put the features and labels into a decision tree, it generates some rules. Which then can be used to predict whether or not the advertisement will be clicked. In comparison, the Random Forest rule willy-nilly chooses observations and options to create many call trees and then averages the results. Another distinction we discovered is that "Deep" decision trees may be prone to overfitting. Random Forest often prevents overfitting by creating random subjects of the features and building smaller trees using these subsets. The subtrees are then combined. This does not always work, and it also slows down the computation depending on how many trees your random forest creates. Important Hyperparameters that we can discuss in this section are listed below. Also, it's important to understand that the hyperparameters in the random forest are used to either increase the model's predictive power or make the model faster. I'll discuss the hyperparameters of Sklearn's built-in random forest function here. Improving Predictive Power – First and foremost, the n estimator hyperparameter specifies the number of trees the algorithms construct before performing maximum voting or taking prediction averages. A larger number of trees improves performance and makes predictions more stable, but it also slows down the computation. Another important hyperparameter is the "max feature," the maximum number of features considered by random forest when splitting a node. Sklearn offers several options, which are detailed in their documentation. In terms of speed, the final important hyper-parameter to discuss is "min sample leaf." This determines whether the name already specifies the minimum number of leaves required to split an internal node. Increasing the speed of the models – The "n jobs" hyper-parameter instructs the engine on how many processors it may use. It will only use one processor if it has a value of one. A value of "-1" indicates that there is no limit. The "random state" property makes the model's output replicable. When given the same hyperparameters and training data, the model will always produce the same results when it has a definite value of a random state. Finally, the "oob score" (also known as OOB sampling) could be a random forest cross-validation methodology. In this sampling, the common fraction of the information is not used to train the model and may not be used to evaluate its performance. The samples are referred to as "out of bag samples." It is eerily similar to the leave-one-out cross-validation methodology but with no additional computational burden. As previously stated, Random Forest classifiers were thoroughly discussed. A quick look at the benefits and drawbacks of Random Forest. To begin with, as previously stated, one advantage of random forest is that it can be used for both regression and classification tasks. It is simple to see the relative importance of the input options. Random Forest is also regarded as beneficial and simple to apply the rule. The main reason is that the default hyperparameters frequently produce a good prediction result. The number of hyperparameters is also not excessive, and they are simple to grasp. Overfitting is one of the most serious problems in machine learning, but this is not as common with a random forest classifier. If the forest contains enough trees, the classifier will not overfit the model. However, the fundamental restriction of random forest, in our opinion, is that a huge number of trees can render the algorithm too slow and useless for real-time forecasts. These algorithms are generally rapid to train but sluggish to predict once learned. A second correct prediction necessitates the addition of more trees, resulting in a much slower model. In most real-world applications, the random forest rule is sufficient. However, there may be situations where run-time performance is critical, and other approaches would be preferable. Random forest may be a prognosticative modeling tool rather than a descriptive tool. However, if you're looking for a description of the relationships in our data, other approaches would be preferable

**3.2 Design of system**

In this part of our report, we will talk about how we prepared or designed the whole thing. As the book progresses, we'll discuss how we built the system and used it.

**3.2.1 Dataset**

We collected the data set for our book through Kaggle (https://www.kaggle.com/andrewmvd/heart-failure-clinical-data?select=heart failure clinical records dataset.csv). The dataset on which our thesis is based has 13 columns and 300 rows. The first 12 columns are the features that will be used to predict the final column, 'DEATH EVENT,' which will indicate whether or not the patient will be affected by heart disease. The 300 rows correspond to 300 patients whose data we extracted from the dataset.

Let me examine the role of each feature in determining whether or not a patient is at risk of developing heart failure:

**Age**: This variable contains information about the patient's age.

**Anemia**: is defined as a deficiency of red blood cells or hemoglobin.

**Creatinine** **phosphokinase**: This value indicates creatine kinase concentration in the blood. This enzyme plays a critical role in muscle function.

**Diabetes**: is a chronic disease that results in elevated blood sugar levels.

**Ejection** **fraction**: indicates the proportion of blood that leaves the heart during each contraction.

**High blood pressure**: that is higher than average is referred to as hypertension.

**Platelets**: are tiny blood cells that aid in the formation of clots to stop bleeding.

**Serum** **creatinine**: level is the amount of creatinine in the blood.

**Serum** **sodium**: level is the amount of sodium in the blood.

**Sex**: the patient's gender

**Time**: This accurately depicts the time of the event.

**Death** **event**: which is the predictor variable.

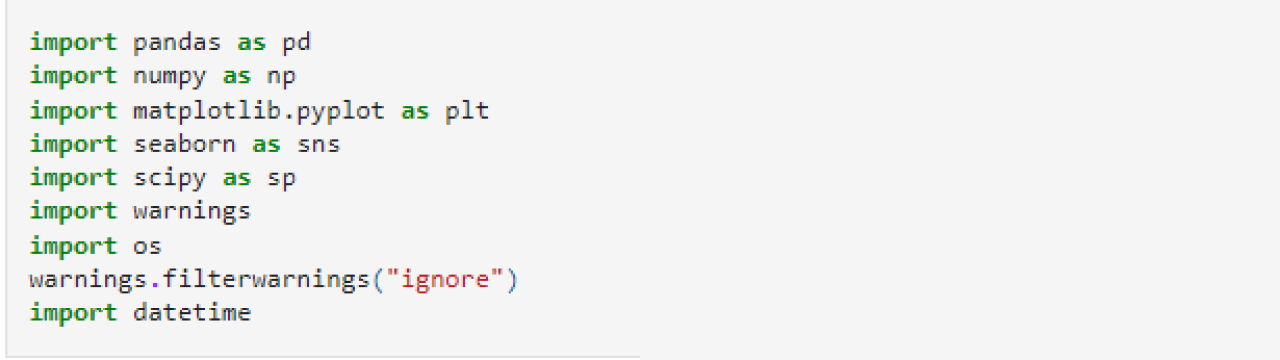
Now that we understand the purpose of each feature let us begin.

**3.2.2 Data Preprocessing**

Before we begin, let's go through some basics about processing data. Data preprocessing is a type of data processing that entails reorganizing data into a more understandable format. Real-world data is frequently insufficient, inconsistent, and deficient in certain behaviors or trends and is therefore prone to contain numerous errors. Preprocessing data may be a tried-and-true technique for partitioning such issues. Preprocessing data prepares it for subsequent processing.

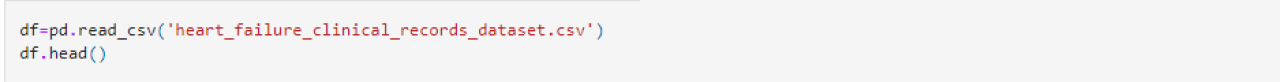
Preprocessing data is necessary for database-driven applications. We are preprocessing our data using the standard scaler from the sklearn library for our thesis. We chose this one over the numerous others because it fits our system perfectly.

Importing Libraries:



Import the Dataset:

The dataset used to build this model was downloaded from Kaggle as a CSV file to my PC.

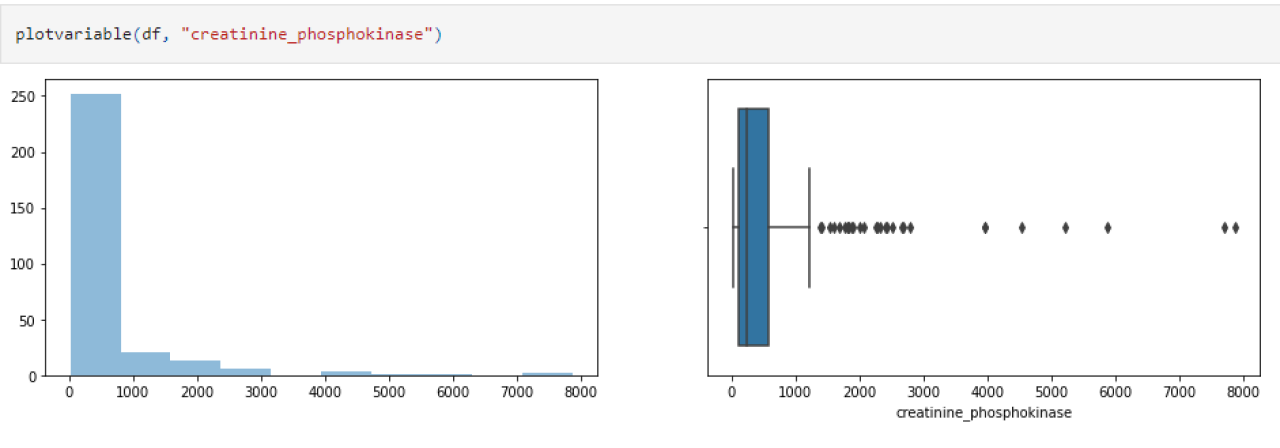


Data Cleaning:

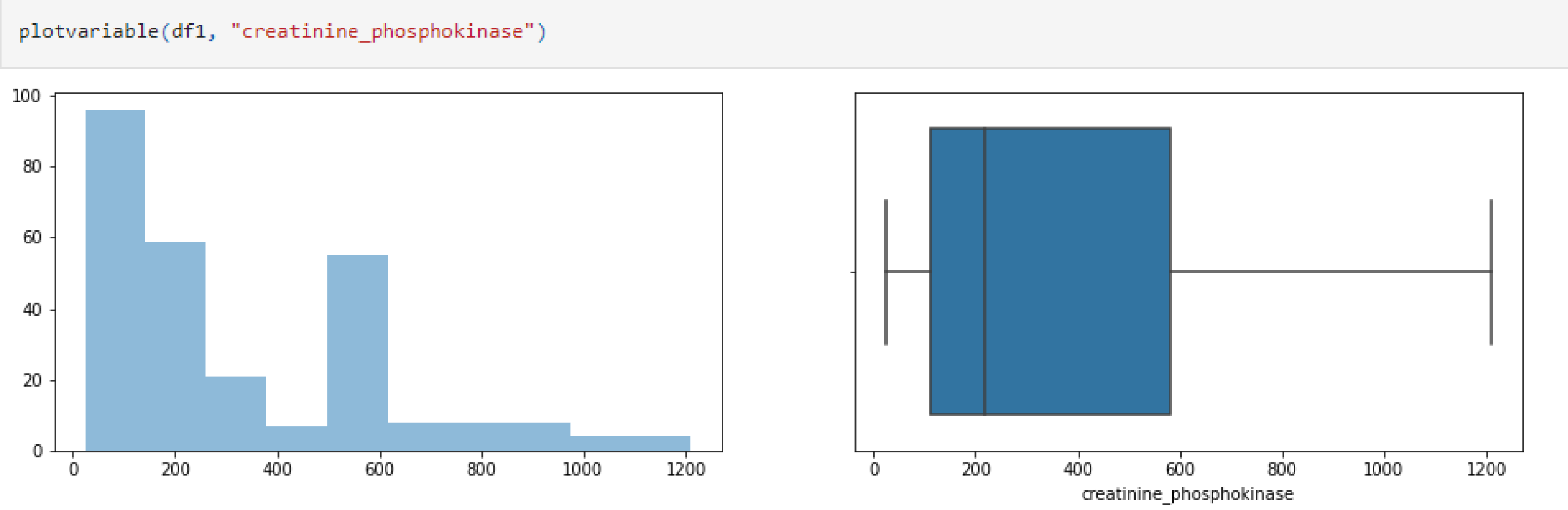
I didn't have to do all the additional cleanup on this data. However, there are still important details that can be discovered like, excluding outliers can cause results to become statistically significant.

\*After searching for outliers, we can see there are some outliers in 5 features. "creatinine\_phosphokinase","ejection\_fraction", "platelets","serum\_creatinine","serum\_sodium"

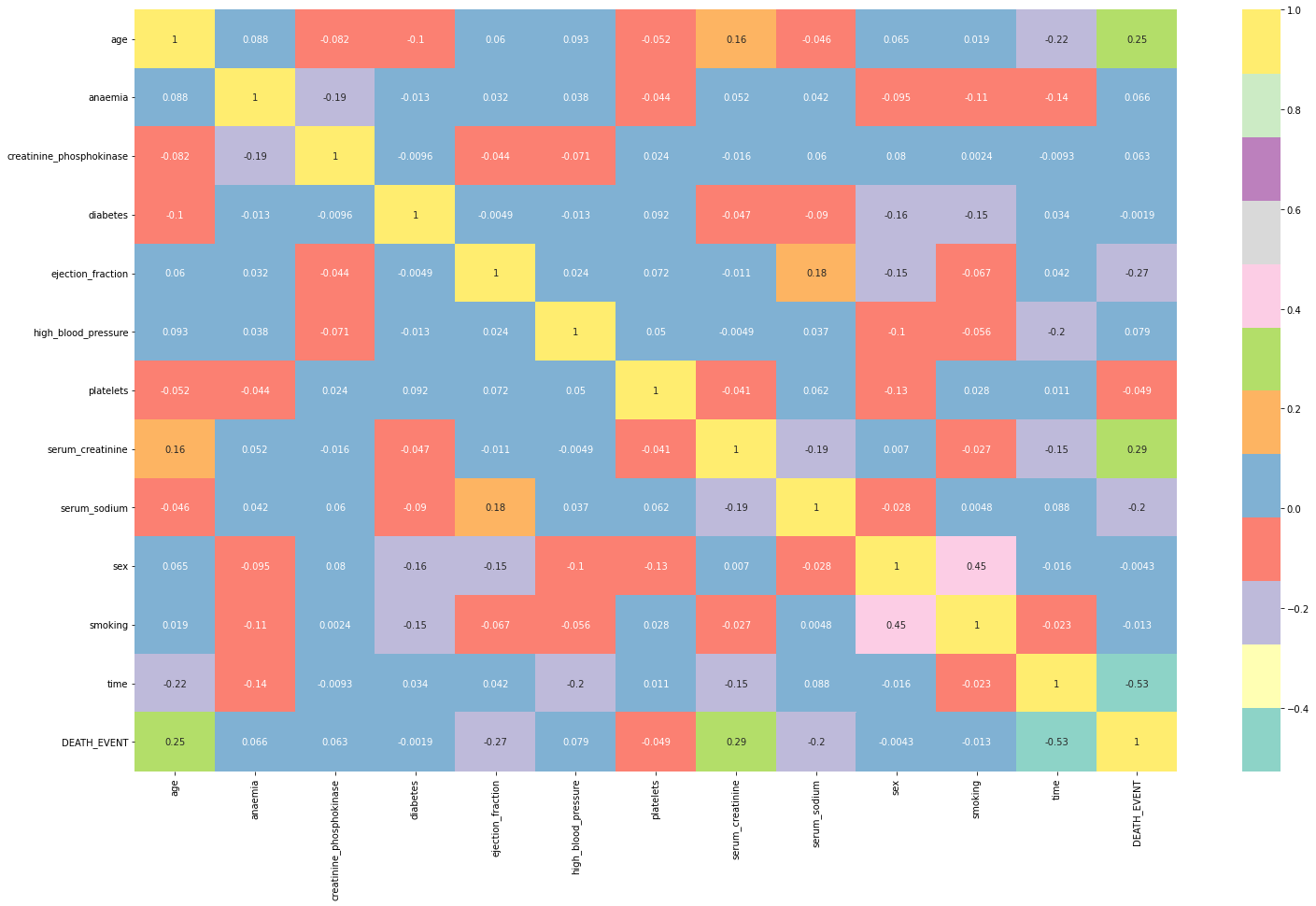
Before:



After:



I use a heatmap to visually visualize the relationship between the various input variables and the result I'm trying to reach visually.



**3.2.3 Analyze features**

This section will help us choose appropriate accuracy metrics and thus properly assess different machine learning models. First, we will count the values of the explained variable, also known as the determining variable, which will tell us whether a patient has heart disease or not.

To begin, I discussed the significance of the Age feature in determining whether a patient is at risk of developing heart failure or not. The older a patient is, the more likely he will have heart failure. We can also figure out that these two variables are linked together because when one variable rises, the other rises. However, a correlation matrix will still be drawn up to make sure.

In general, creatinine levels should be between 0.9 and 1.3. We can see from the plot of serum creatinine against Death event above that the chances of survival are better in this range.

Also, there is a range for serum sodium that is between 135–147 mmol/L. From the plot above, the survival rate only starts to rise at this point. This feature also has a lot to do with Death Event.

**3.2.4 Modeling and predicting with machine learning**

The project's primary goal is to accurately predict heart disease occurrence. We will test several classification algorithms to achieve this. This section summarizes the study's findings and highlights the top performer by accuracy metric. I chose supervised learning algorithms for classification problems. First, let us create a handy tool that uses the SciKit Learn library to form a general function for training our models. The reason for showing accuracy on both train and test sets is to assess the model's fit to the data (the bias/variance tradeoff).

Then we'll build a model to run all our algorithms.