مجلة العلوم الهندسية
FES Journal of Engineering Sciences
http://journal.oiu.edu.sd/ojs/index.php/fjes/index
FJES
FES Journal of Engineering Sciences

K-NN Algorithm Used for Heart Attack Detection

Bah Ibrahima[1,*] and Xue Yu [1]

[1] *Nanjing University of Information Science and Technology, Nanjing, China*
* Corresponding author: Bah Ibrahima (e-mail: Ibrahimasalamatabah18@hotmail.com).

Article history: Received 23 June 2021, Received in revised form 03 July 2021, Accepted 12 September 2021
Digital Object Identifier (doi): https://doi.org/10.52981/fjes.v11i1.758

ABSTRACT Machine Learning, a branch of artificial intelligence, has become more accurate than human medical professionals in predicting the incidence of heart attack or death in patients at risk of coronary artery disease. In this paper, we attempt to employ Artificial Intelligence (AI) to predict heart attack. For this purpose, we employed the popular classification technique named the K-Nearest Neighbor (KNN) algorithm to predict the probability of having the Heart Attack (HA). The dataset used is the cardiovascular dataset available publicly on Kaggle, knowing that someone suffering from cardiovascular disease is likely to succumb to a heart attack. In this work, the research was conducted using two approaches. We use the KNN classifier for the first time, aided by using a correlation matrix to select the best features manually and faster computation, and then optimize the parameters with the K-fold cross-validation technique. This improvement led us to have an accuracy of 72.37% on the test set.

Keywords: Machine Learning (ML), Supervised Learning (SL), K-Nearest Neighbor (KNN), Correlation Matrix (CM), K-Fold Cross-Validation (K-FCV), Heart At-tack (HA)..

1. INTRODUCTION

Heart attack killed so many people around the world. Recent years of studies proved that artificial intelligence could be very helpful for the early detection of this disease to its complete prevention. Nowadays, many countries are investing a lot in the science and the development of the internet and big data, and those are key points to conduct many kinds of research- a huge amount of data are available everywhere-. A heart attack happens when there's a disruption of the blood flowing to the heart. This blockage can be affected by different types of dangerous diseases that impact human lives. According to the World Health Organization (WHO), more than 12 mil-lion deaths occur worldwide every year, and heart diseases are the reason [1]. The symptoms vary from people and sex; as with men, women's most common heart attack symptoms are chest pain (angina) or discomfort. But women are somewhat more likely than men to experience some of the other common symptoms, particularly shortness of breath, nausea/vomiting, and back or jaw pain [2]. Added to that, there are few other symptoms to watch out for carefully, such as pressure tightness, squeezing or aching sensation in the chest or arms that may spread to the neck, heartburn, abdominal pain, cold sweat, fatigue, lightheadedness, or sudden dizziness.

The risk factors of heart attack are numerous; some contribute to the unwanted buildup of fatty deposits (atherosclerosis), narrowing arteries throughout the body. It includes: the age, men of age 45 or older and women of age 55 or older are more likely to have a heart attack than is younger; Tobacco, whenever it's smoking for a long or short term; High blood pressure, over time the high blood pressure can dam-age arteries that

lead to the heart, and if the high blood pressure occurs with other conditions such as obesity, high cholesterol or diabetes, it increases the risk of HA; Metabolic syndrome, this syndrome occurs when you have obesity, high blood pressure, and high blood sugar; Family history of HA, if your family has had early heart attacks (by age 55 for males and by age 65 for females), you might be at increased risk; Lack of physical activity, being inactive contributes to high blood cholesterol levels and obesity. People who exercise regularly have better heart health, including lower blood pressure; Stress also increases the risk of Heart Attack; A history of preeclampsia causes high blood pressure during pregnancy and increases the lifetime risk of Heart Attack [2].

For the treatment of the heart attack, it should be done immediately, while waiting for the ambulance, the closest person can help by chewing and swallowing a tablet of aspirin if the person is not allergic to that, this medicine helps to thin the blood and improves blood flow to the heart. The two main treatments are using medicine and surgery.

However, our health care systems need to have more techniques and tools for extracting information from huge datasets to make a medical diagnosis. That is where computer science and its techniques such as Data Mining and Machine learning are accompanying medicine for the best of humanity. The computerized diagnostic uses medical databases to classify them to build systems capable of diagnosing and ad-dressing the emerging situations [3]. Our goal here is to minimize the number of deaths and high-risk level patients at a considerable amount. There are many techniques used for the classification task; based on the model or data, the KNN algorithm is a data-based algorithm belonging to the class of Supervised learning. Super-vised learning like Bayesian classifier and neural network is based on the training data by using the pattern of this set and do the classification on another set of data called testing set, that is not the case of KNN, this algorithm takes a bunch of labeled data points and uses them to learn how to label other points [4], the main thing is to store the training data and wait for the testing data to perform a simple comparison, that's why it's so simple that it doesn't learn at all and it is so-called lazy learning.

The paper is structured as follows: in section 2. The related works and techniques used to tackle the heart attack problem were addressed, section 3 focused on a brief introduction of supervised learning and KNN algorithm, and the detail of the pro-posed network; the 4th section is the experimentation and result, in this one, we had a look on the process of the experiments and described the results; the 5th and last section concern the conclusion of this work.

2. Procedure for Paper Submission

In this section, we review the existing approach that deals with the heart attack problem. In [3], to deal with the heart attack problem, the authors used Pearson techniques to reduce the features and find the properties that can be considered and the characteristics that can be disregarded. They managed to use statistical techniques and Logistic Regression to make relations between attributes. In [5], based on Random Forests Bayesian classification and Logistic Regression, which provides a decision support system for medical professionals to detect and predict heart diseases and heart attacks in humans or individuals using risk factors of heart, in their work, the authors used a big dataset that has 18 features to make the comparison between these three classifiers. Dealing with data before feeding the algorithm for the classification is an important task, and this is the approach proposed in [6], the divided the attributes into ordinal attributes, discrete attributes, and binary attributes to improve classification accuracy, they also used many classifiers such as Decision Tree, Logistic Regression, Bayesian Network. Imbalanced data can be challenging when doing Machine Learning; in [7], the authors designed an algorithm by leveraging random under sampling, clustering, and oversampling techniques. This approach generates nearly balanced data which are utilized for training machine-learning models for predicting heart attack. The KNN algorithm

itself doesn't give accurate results. Hence, many authors employed optimization techniques to boost up the algorithm's efficiency; for example, [3] used Genetic Algorithm to control the basic joints of this method by determining the value of nearest neighbors. In [8], the authors proposed a novel KNN type method for classification to overcome these shortcomings. Their method is based on constructing a KNN model for the data, which replaces the data to serve as the basis of classification. Principal Component Analysis for feature selection and Support Vector Machine (SVM) can be combined to tackle cardiovascular (heart-related) disease according to [9]; in this paper, the authors tried to do an over-view of the used method for heart attack disease and a comparison between them, finally stating that SVM with PCA components give the best results.

3. PROPOSED METHOD

A. K-NN

A robust and versatile classifier, The KNN algorithm, just like Artificial Neural Networks (ANN) and Support Vector Machines (SVM), is used for most classification tasks. KNN can outperform more efficient classifiers despite its simplicity and is used in various applications such as economic analysis, data compression, and genetics [10]. KNN lies under the algorithmic family of supervised learning and finds intense application in pattern recognition, data mining.

Supervised learning is the type of machine learning where the given data is a couple of features and labels, meaning of each input correspond to a given output. Mathematically this is how supervised learning works: Given a set of N data points of the form {(x_1,y_1),…,(x_N,y_N)} such as x_i and y_i Are respectively the feature vector and the target of the i-th data point, the learning is to capture the relationship between the feature vector and the target, and the function is given as h:x→ y such that h(x) can forecast the corresponding output y with certainty when it's provided an unknown observation x. The goal is to learn a mapping from inputs x to outputs y, given a labeled input-output pair in the supervised learning approach. Supervised learning working procedure (see Fig. 1).

All the supervised learning models typically follow the figure mentioned above; they differ most from the task. According to that, we can classify the machine learning algorithms into a group of similarities; first, where the function is the same, one example is the tree-based methods and neural networks, we also have the regression algorithms, in this case, the method is to design a relationship between variables that is iteratively refined using a measure of the error in the predictions made by the model, the most used regression algorithms are linear regression, logistic regression. The third group is instance-based algorithms, and it's a kind of algorithms where the training data deemed important are used by the model to predict the new samples; the KNN algorithm is in this category.

The KNN classifier is a non-parametric, instance-based learning algorithm; KNN is preferred when all the features are continuous [11]. Non-parametric means it requires no clear assumptions regarding the functional shape of h, thus eliminating the pitfalls of modifying the underlying data distribution. For example, suppose our data is highly non-Gaussian, but the learning model we chose takes on a Gaussian type. Learning based on instance means our algorithm does not explicitly learn a model. Rather it prefers to memorize the training instances used as "information" for the prediction process. In concrete terms, the algorithm will only use the training instances to spit out the answer when a query is made to our database. The algorithm is a lazy learning method that prohibits many applications, such as dynamic web mining for a large repository [8].

The algorithm is implemented in such a way it finds the closest neighbors with a technique of choosing a value randomly, and the most important thing is to find a way of calculating the distance between two data points. Then it's assigned to the most common class of its nearest K neighbors. If K = 1, then the case is allocated to its closest neighbor's class in distance functions. There are three most commonly used formulas to calculate the distance between two points: The

Euclidian distance, the Manhattan distance, and the Minkowski distance.

Euclidian distance: the distance between two points It is the length of a line segment them, in general for points given by Cartesian coordinates in k-dimensional Euclidian space, the distance is:

$$Euclidean = \sqrt{\sum_{i=1}^{k}(Xi - Yi)^2} \quad (1)$$

Manhattan distance: The Manhattan distance between two vectors (city blocks) is equal to the one-norm of the distance between the vectors; we care about the abs value in this distance.

$$Manhattan = \sum_{i=1}^{k}|X_i - Y_i| \quad (2)$$

Minkowski distance: It's the normed vector space considered a generalization of both the Euclidian distance and the Manhattan distance.

$$Minkowski = \left(\sum_{i=1}^{k}(|X_i - Y_i|\right)^{9})^{1/9} \quad (3)$$

However, on many occasions, it can happen that the data we are dealing with doesn't only contain constant values; it can also be categorical; in that case, the method used to calculate the distance is the Hamming interval. This kind of calculation aims to find the number of bits' positions in which the two bits are different.

Using the algorithm is to find the optimal value for K, which is done by inspecting the data first. We can start by a large K value, which is generally more accurate because it reduces the overall noise, but there is no guarantee. So, to be sure to have the best value automatically, we used the k-fold Cross-validation by using an unbiased sample to test the K value. Most of the time, the optimal value of K has to be between 3-10 for most datasets.

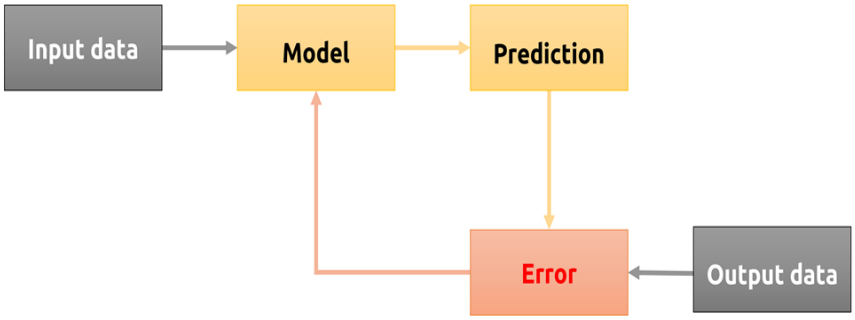

Fig. 1. Supervised Learning models

B. *KNN Algorithm Classifier*

The algorithm's training phase consists of storing the class labels and featuring train-ing sample vectors. Fig. 2. shows the sample feature space, which separates two clas-ses with the aid of the spaces Decision Surface and Voronoi [12]. The algorithm is assuming that similar things exist in close proximity. We can see the red stars together and the blue points also. That's why the idea of using this algorithm is most of the time effective. Here the K is a user-defined constant in the classification phase. The test vector is classified under the label that occurs most frequently among the K train-ing samples closest to that query point. A full description of this algorithm can be as follow:

1. Load the data
2. Preprocess the data
3. Find the best features using a correlation matrix
4. Choose the first value of K
5. For each example in the data:

5.1. Find the distance between the query data and the data in the training set

5.2. Add the obtained distance and its index to an ordered collection

6. Sort this ordered collection of distances and indices from the smallest to the largest (ascending order) by distance values.
7. Pick the top K entries from the sorted collection
8. Find the labels of the chosen K entries
9. Return the mode of the K labels.

C. *Decision Boundary*

The decision boundary is this line separating classes; it's used in several other Machine Learning models such as K-means Clustering; it's a good thing to start by first plotting the data to

see what is the distribution, and then apply the appropriate data preprocessing before using classification model to classify the data points. This line is obtained by using the algorithm that means calculating the distances and grouping similar points. In (Fig. 2), we can see the plot of the decision boundary separating two classes.

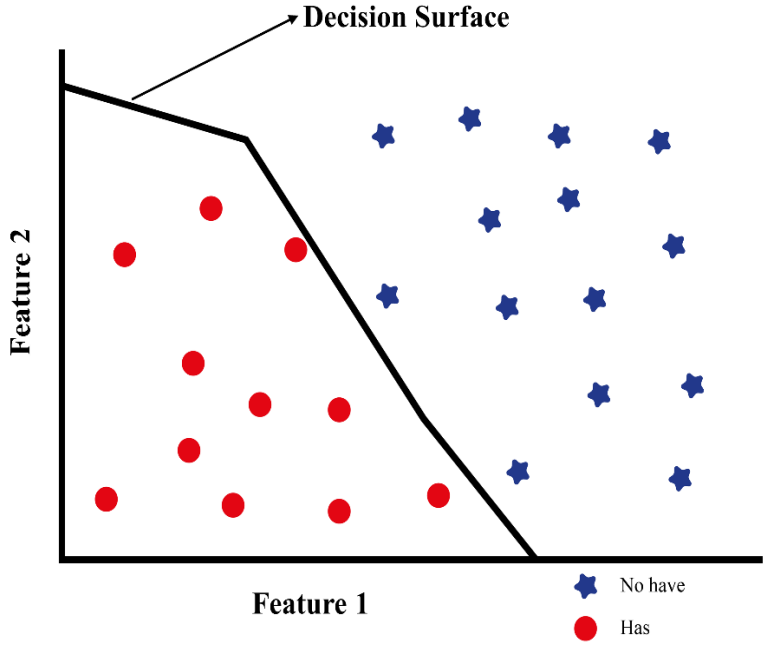

Fig. 2. Sample Features space, in this feature, we can see two classes. The first one in blue represented by stars is the class of people who don't have the disease and the ones in red is the one who has, to we used two features and the decision boundary lines fit the

D. The selection of the value of K

The best choice for selecting the K-value depends on the data. We run the KNN al-gorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before. In general, noise can be greatly reduced with the larger values of K in the classification, but it does not estab-lish clear boundaries between classes. The case where the data sample is predicted to be grouped under the nearest training sample class is the nearest neighbor algorithm, where the K value is 1. The classification process might be very tedious whenever the accuracy of the k-NN algorithm is degraded by the presence of more number of nearest points. This scenario is solved by increasing the value of K beyond the point. The classifier tries to select the top three nearest neighbors when the K value is three and decides to which majority class the data point belongs. In Fig.3, the decision boundary for two cases has been shown K=3 and K=5; In the first case, we can see that the sample data belongs to the red class because the boundary area has this class as the closest neighbor majority. The sample data is related to the blue class because we increased the number of neighbors to search for in the second case.

E. K-fold Cross-Validation

We emphasized in the details on how the algorithm works that all depend on the value K, and the optimization of the algorithm is finding the best value of K. Doing it manually is meticulous and time-consuming, since we can use another good way to get what we want. The K-fold Cross-validation is the most used way of parameter tuning for the traditional machine learning models. The whole process of KNN is to use the test data to find the best value of K and the train data to find the nearest neighbor. Still, since when using the test data to find K, there's no more unseen data to use to be sure that the accuracy is good, we perform that K-fold Cross-validation that split the training data into K small group and performs the training on k-1 split and the evaluation on the final one. By doing that, we can ensure that the accuracy we will get in the test data is the real one, and by that, we can say the model is generalizing well. The 5-fold Cross-validation can be resumed in Table 1.

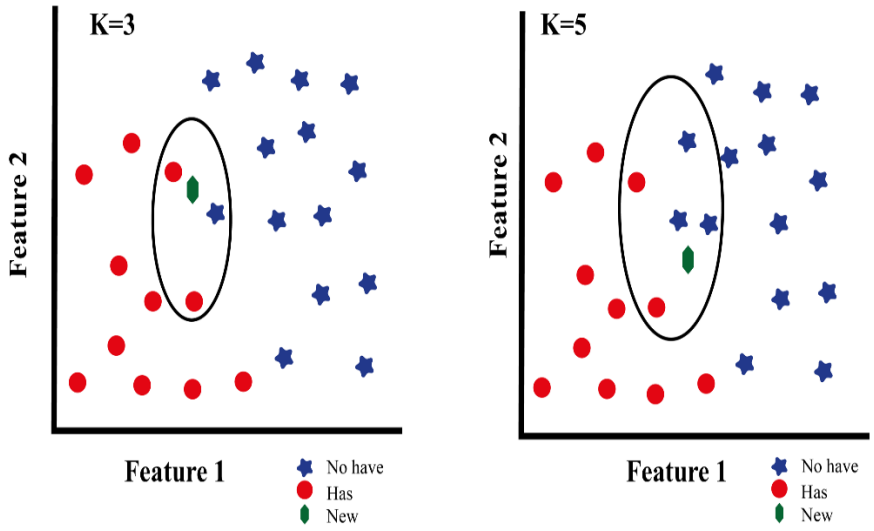

Fig. 3. Boundary Selection in Feature Space: A for k=3 and B for k=5

TABLE 1. CROSS-VALIDATION PROCESS; FOR EACH LINE, THE DATA STORING IS PERFORMED USING FOUR SPLITS, AND THE TESTING IS PERFORMED USING THE REMAINING SPLIT. AS WE HAVE FIVE SPLITS, THE PROCESS IS REPEATED FIVE TIMES BY CHANGING THE TESTING SPLIT.

Training data	Cross-validation
(D1, D2, D3, D4)	D5
(D1, D2, D3, D5)	D4
(D1, D2, D4, D5)	D3
(D1, D3, D4, D5)	D2
(D2, D3, D4, D5)	D1

F. Training Process

In HA, the cardiovascular data give a lot of historical information, and according to that, we split the dataset into the training set and testing set. A K-NN classifier is trained to store the data. Then, the classifier is used to decide if a new sample is likely to have cardiovascular disease or not. The main contribution of this paper is to optimize the KNN algorithm using the K-fold Cross-Validation. The framework is implemented as follow:

4. Experimentation and Results

This section contains the overall discussion of the algorithm used during the entire project and a detailed discussion of the expected outputs. In this work, the dataset we used is the Cardiovascular dataset publicly available in the Kaggle dataset repository; we choose to use this dataset because it contains a very important number of samples, and we can apply many techniques with this kind of dataset. It consists of three types of data: factual information, medical examination results (Examination Feature), and information given by patients (subjective features). The data can also be divided into categorical data and numerical data. The dataset contains 70000 data points and 14 features. The details are given in table 2.

A. Data Preprocessing

The dataset is a CSV file. We loaded it and did some data exploration. The first step to do is to remove the id column as it doesn't give any information. Some data visualization had to be also conducted to see the data distribution and understand more about the problem we are facing; according to that, we can find the best way to do the feature engineering. The experiment has been conducted into two parts. First, use the KNN without feature compression and then use the correlation matrix to extract the best features manually; a table of comparison is given at the end of the experiment. The programming language used for the experiments is python version 3.8, and the most used libraries are NumPy for matrix calculations, pandas for data manipulation like loading the data, matplotlib for plotting figures such as the confusion matrix, and

the most used one, scikit-learn, it has built-in functions to do the classification using KNN.

Our next figure is to tell us how some features affect the output. In this case, we plotted according to the ages, but since in our dataset we don't have the age column in years, we decided to add this column by dividing the age in days by 365 to get this output.

Let's manually go deeper into understanding this disease. We know that the rap-port between the weight and the squared height, in the technical way we call it the Body Mass Index (BMI), is more useful than using them separately. Its objective is to calculate the lifetime risk of incident CVD and subtypes of CVD and to estimate years lived with and without cardiovascular disease by weight status [13]. In our work, we decided to name it height_weight, then let's visualize the description.

Now that we know the age is a really important feature to classify whether a per-son has cardiovascular disease or not let's observe how the data are correlated and the dependency between these variables. This will help us do the data cleaning dedi-cated in the next section. In fig. 6, we observe the output. Here in this plot, we notice that we can't directly decide that the age tells us whether to have the disease or not despite the fact that's the highest correlation with the cardio column, but the value is just 0.23. A little bit far, we can see that height and gender have strong relationships when the weight is nearly the one that affects the height_weight feature the most. Fig.6 will give us more details.

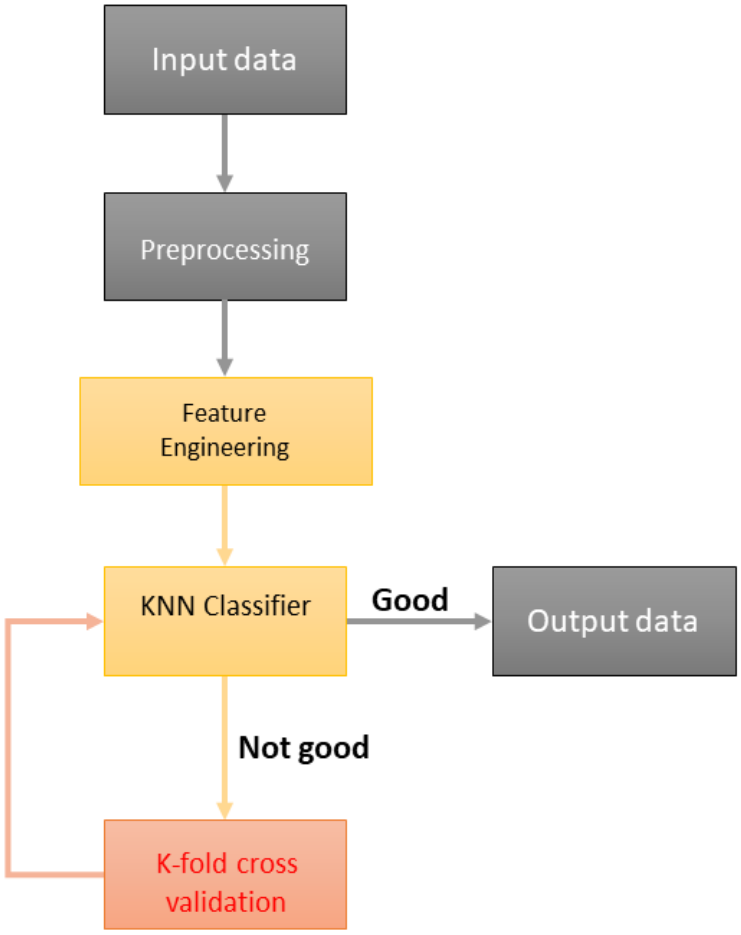

Fig. 4. The KNN algorithm

TABLE 2. DATA PRESENTATION

Feature	Description
id	Number of rows (70000)
Age	In days (70000)
Gender	1 for women and 2 for men (70000)
Height	In cm (70000)
Weight	In kg (70000)
Api_hi	Systolic blood pressure (70000)
Ap_lo	Diastolic blood pressure (70000)
Cholesterol	1: normal, 2: above, 3: well above normal (70000)
Glucose	1: normal, 2: above, 3: well above normal (70000)
Smoke	1: for smoke, 0: don't smoke (70000)
Alco	1: alcoholic, 0: not alcoholic (70000)
Active	1: does exercises, 0: doesn't do (70000)
Cardio	1: has cardio, 0: doesn't have (70000)

TABLE 3. DATA DESCRIPTION: THIS TABLE GIVES US THE STRICT DETAILS OF OUR DATASET; ALL THE COLUMNS HAVE 70000 ROWS, WHICH TELLS US NO DATA IS MISSING. WE CAN SEE ALSO THAT THERE IS A HIGH DATA DISTORTION; THE SCALING IS VERY DIFFERENT, THAT'S WHY SOME STANDARD DEVIATION IS VERY HIGH, AND SOME OTHERS ARE SO LOW; IT MEANS THAT WE HAVE TO PERFORM THE SCALING

	count	mean	std	min	25%	50%	75%	Max
age	70000	19468.8	2467.	10798	17664	19703	21327	23713
gender	70000	1.34957	0.476	1	1	1	2	2
height	70000	164.359	8.210	55	159	165	170	250
weight	70000	74.2056	14.39	10	65	72	82	200
api_hi	70000	128.817	154.0	-150	120	120	140	16020
api_lo	70000	96.6304	188.4	-70	80	80	90	11000
chol	70000	1.36687	0.680	1	1	1	2	3
gluc	70000	1.22645	0.572	1	1	1	1	3
smoke	70000	0.08812	0.283	0	0	0	0	1
alco	70000	0.05377	0.225	0	0	0	0	1
active	70000	0.80372	0.397	0	1	1	1	1
cardio	70000	0.49970	0.500	0	0	0	1	1

B. Data Cleaning and Manual Feature Selection

Looking carefully at the table, we can see something going wrong here, as the max value of the weight is 200, the one for height is 250, but the one for the height_weight is 298.666667. We had to plot the outliers (Fig.7) and determine the minimum and maximum values that those columns should take to deal with this problem. After plotting a box plot, it is a graphical interpretation of measurable information given the base, first quartile, middle, third quartile, and greatest. We understood that we had to delete all the values over 39 and under 14 for the height_weight, over 170 and under 90 for the ap_hi, and over 105 and under 65 for the ap_lo. The table shows that the value of age reduced, we changed the unit from days to year, and the correlation matrix helped us figure out that the height and weight are no more needed for the classification task.

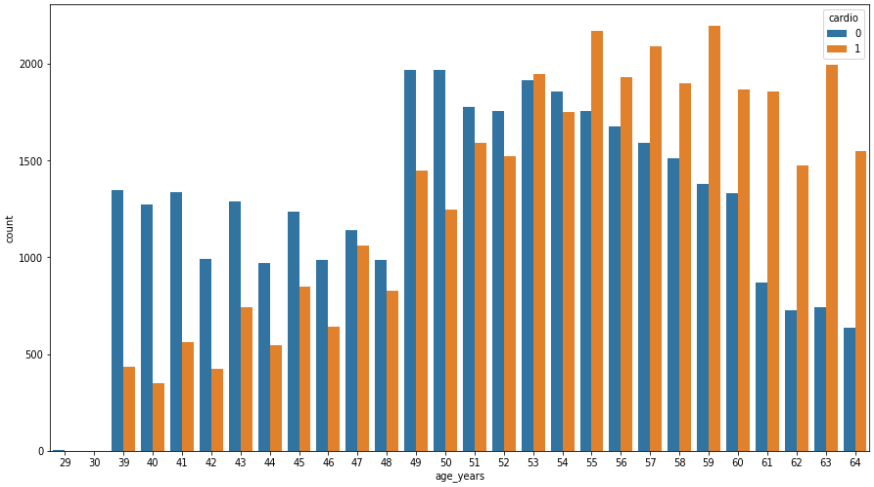

Fig. 5. Age distribution. This plot shows us that older people are more likely to have cardiovascular disease while youngsters are a bit safe.

Fig. 6. Correlation matrix

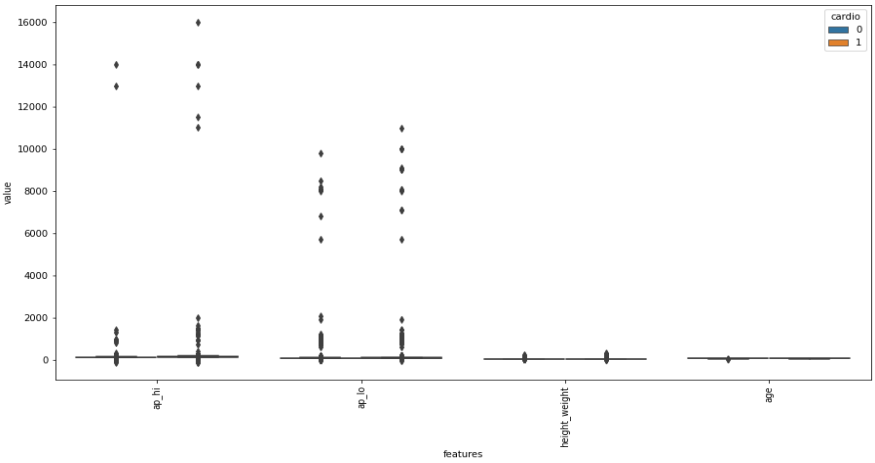

Fig. 7. Outliers' detection

TABLE 4. DATA DESCRIPTION AFTER ADDING HEIGHT_WEIGHT COLUMN AND CHANGING THE AGE

	Count	mean	std	min	25%	50%	75%	Max
age	70000	52.84	6.766	48	17664	53	58	64
gender	70000	1.34957	0.476	1	1	1	2	2
api_hi	70000	128.817	154.0	-150	120	120	140	16020
api_lo	70000	96.6304	188.4	-70	80	80	90	11000
chol	70000	1.36687	0.680	1	1	1	2	3
gluc	70000	1.22645	0.572	1	1	1	1	3
smoke	70000	0.08812	0.283	0	0	0	0	1
alco	70000	0.05377	0.225	0	0	0	0	1
active	70000	0.80372	0.397	0	1	1	1	1
cardio	70000	0.49970	0.500	0	0	0	1	1
height-weight	70000	27.5565	6.091	3.471	23.87	26.37	30.22	298.6

Moreover, the features we have to take a look at are the Diastolic blood pressure here represented as (ap_lo) and the Systolic blood pressure (ap_hi); as we can see by their name, the ap_lo must be lower than the ap_hi, and inversely, also we can't have a negative value of the blood pressure that is the difference between the systolic blood pressure, that measure the pressure in the arteries when the heartbeats and the diastolic blood pressure that measures the pressure when the heart rests between beats. But regarding the table, we can see that the difference will give a negative value when we take the min values. It's fixed by doing the same process mentioned above.

Table 5 shows the description of the cleaned data, and (fig. 8) shows how the data looks after we cleaned, and we can see that the outliers disappeared and we are ready for the classification. We finally have 60142, meaning that 9858 rows have been dropped during this data cleaning and feature engineering.

Before building our KNN classifier and feed the data, we had to do some data standardization to close this large gap between the values; in this work, the standardization we choose is the standard Scaling, in this standardization, the goal is to Standardize features by removing the mean and scaling to unit variance. An important note is to know that not all the features are concerned in this Scaling, just the follow-ing columns: age_years, ap_hi, ap_lo, height_weight. The weight and height features have been discarded from our dataset since the new height_weight issued from the feature engineering process gives us all the information needed.

B. Performance Analysis

As mentioned, we started first by giving the number of neighbors K=1, and leave all the other parameters in the default setting. After performing the split of the

data by giving 70% to the training test and 30% to the testing set, we performed the KNN classification and plotted the confusion matrix. The accuracy obtained after this first

trial is just 64%. The efficiency estimation is calculated using the confusion matrix table, the precision, the recall, the accuracy, the f1-score. Details are given below.

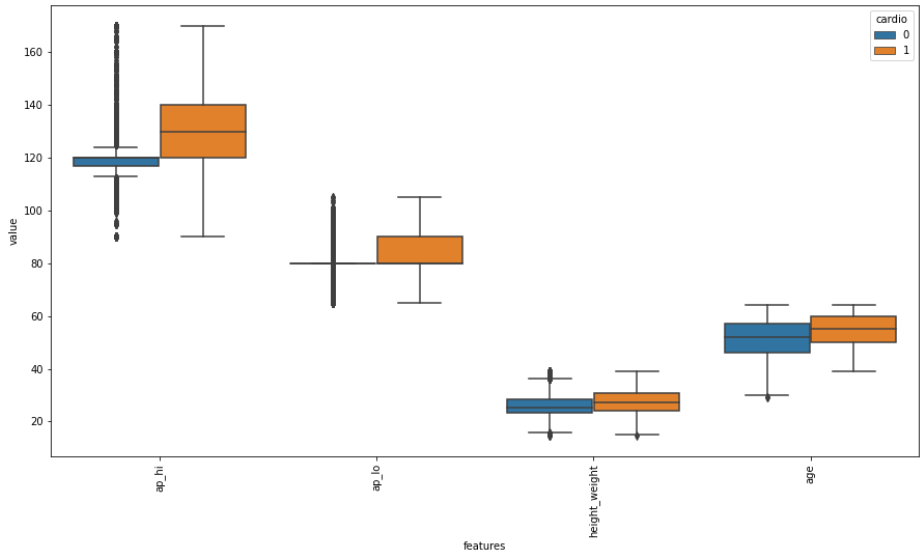

Fig. 8. Bar plot to visualize outliers after cleaning the data.

TABLE 5. DATA CLEANED

	COUNT	mean	std	min	25%	50%	75%	Max
age	61526	52.882	6.748	29	48	54	58	64
gender	61526	1.3536	0.478	1	1	1	2	2
api_hi	61526	126.28	14.24	90	120	120	140	170
api_lo	61526	81.636	7.649	65	80	80	90	105
chol	61526	1.3525	0.670	1	1	1	2	3
gluc	61526	1.2183	0.565	1	1	1	1	3
smoke	61526	0.0873	0.282	0	0	0	0	1
alco	61526	0.0523	0.226	0	0	0	0	1
active	61526	0.8046	0.396	0	1	1	1	1
cardio	61526	0.4907	0.499	0	0	0	1	1
h-w	61526	26.893	4.240	14.52	23.83	26.14	29.64	38.96

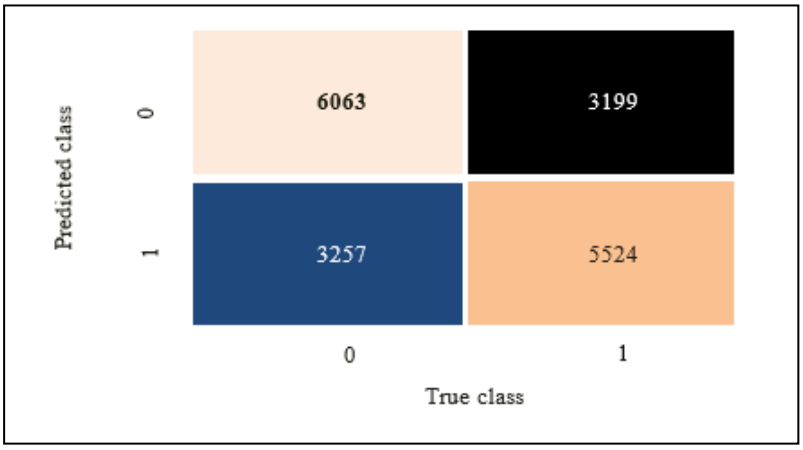

Fig. 9. Confusion matrix k=1

TABLE 6. CLASSIFICATION REPORT K=1

	precision	recall	f1-score	support
No disease	0.65	0.65	0.65	9320
Has disease	0.63	0.63	0.63	8723
Accuracy			0.64	18043
Macro accuracy	0.64	0.64	0.64	18043
Weighted accuracy	0.64	0.64	0.64	18043

TABLE 6. CLASSIFICATION REPORT AFTER PERFORMING K-FOLD CROSS-VALIDATION

	precision	recall	f1-score	support
No disease	0.71	0.78	0.75	9320
Has disease	0.74	0.66	0.70	8723
Accuracy			0.72	18043
Macro accuracy	0.73	0.72	0.72	18043
Weighted accuracy	0.73	0.72	0.72	18043

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{F1-Score} = \frac{2*(Recall \times Precision)}{Recall+Precision} \quad (7)$$

TP is True Positive: where the prediction and the real output are all true

TN is True Negative: where the prediction and the real output are all false

FN is False Negative: where the prediction is true, and the real output is false

FP is False Positive: where the prediction is false, and the real output is true

This output is not giving a good result; we can improve it by increasing the value of K; it's where the K-fold Cross-validation enters in the game, instead of looping in some values of K and evaluate the performance on the same testing data.

C. Performance of the K-fold Cross-validation

When performing the K-fold Cross-validation, we will not just only look at the value of K, we will also try to check other parameters such as the algorithm (ball_tree, kd_tree, brute, auto); or the leaf size that take an integer as a parameter; or the used metric distance (Minkowski, Euclidian, Manhattan), p is a parameter used when the metric is set to default or Minkowski (1 for Euclidian and 2 for Manhattan).The best parameters are: {n_neighbors=37, algorithm='ball_tree', p=2, weights='uniform'}

Although this k-fold Cross-validation ran for a longer time, 40 min, this time took that long because our dataset is a bit big. We were looking through a lot of parameters to be sure to have an accurate result. Still, it is worth it, and we can see a big difference in the classification report table. We finally got a higher accuracy that is 72,37% precisely. In this work, we focused on the number of neighbors; we considered the algorithm used to compare the nearest neighbors; here, the Cross-validation tells us the ball_tree one is performing better. The ball_tree algorithm is used to organ-ize the points in a multi-dimensional space; in the KNN algorithm, the goal is to group all the similar data points, and the ball_tree proved to be pretty good.

As we already know, using the brute force approach, each time a prediction has to be made for a new data point, the algorithm will calculate the distance between the data points and the stored data. Only then will it use the voting according to the value of K and give the class, so to avoid measuring the distance from all the data points and reduce time-consuming the ball_tree has been developed and is efficient, especially when there is a large number of features.

Another important parameter is weight. In this case, the Cross-validation decided the best value of the weights to be uniform; the weight is used to penalize the KNN algorithm when classifying the new data points so that it doesn't include many points from the other class or be more sensitive to outliers depending on the value of neigh-bors is large or small. But sometimes it happens that the distance between data points from different classes are very large in that case, even having a relatively good value of K will not be helpful, and

it's when weighted KNN comes to the rescue giving the nearest K points some weights by using a kernel function. This kernel will give big weight to the nearby and lower weight to the far away points.

The third parameter p here tells us which distance measurement to use, in this case, the selected is the Manhattan distance with the value corresponding to 2. According to all the parameters cited above, the best value for K is 37; usually, it's always an odd value that gives the best performance. The output is summarized in the classification report showed in table 6.

5. CONCLUSION

Our work mainly focused on optimizing KNN algorithm parameters using k-fold Cross-validation; here, we implemented an efficient way to detect if an individual is likely or not Heart Attack (HA). The Cross-validation helped us find the best parameters at the same time instead of using an analysis based on error rate or accuracy to tune the value of the nearest neighbors, using it also allowed us to try many other parameters such as the ball_tree algorithm, the weights sometimes neglected but have an impact on the generalization. In this work, we put all the pieces together to con-duct a machine learning project combining all the techniques available from feature engineering and data cleaning using the correlation matrix. The building of the classifier and its generalizing can be fast and less the computation time. This procedure can also be applied to many other real-world problems. In future work, we will try to improve the KNN algorithm using feature selection techniques.

REFERENCES

[1] Jabbar MA (2017) Prediction of heart disease using k-nearest neighbor and particle swarm optimization. Biomed Res 28:4154–4158

[2] Heart attack - Symptoms and causes - Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/heart-attack/symptoms-causes/syc-20373106

[3] Kadhum N (2018) Improving The Accuracy Of KNN Classifier For Heart Attack Using Genetic Algorithm. 116–125

[4] Intro to types of classification algorithms in Machine https://medium.com/sifium/machine-learning-types-of-classification-9497bd4f2e14

[5] T. Obasi and M. Omair Shafiq, "Towards comparing and using Machine Learning techniques for detecting and predicting Heart Attack and Diseases," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 2393-2402, doi: 10.1109/BigData47090.2019.9005488.

[6] Salman I (2019) Heart attack mortality prediction: An application of machine learning methods. Turkish J Electr Eng Comput Sci 27:4378–4389. https://doi.org/10.3906/ELK-1811-4

[7] M. Wang, X. Yao and Y. Chen, "An Imbalanced-Data Processing Algorithm for the Prediction of Heart Attack in Stroke Patients," in IEEE Access, vol. 9, pp. 25394-25404, 2021, doi: 10.1109/ACCESS.2021.3057693.

[8] Guo G, Wang H, Bell D, et al (2003) KNN model-based approach in classification. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2888:986–996. https://doi.org/10.1007/978-3-540-39964-3_62

[9] Perumal, P., and P. T. Priyanka. "SUPERVISED HEART ATTACK PREDICTION USING SVM WITH PCA." Journal of Critical Reviews 7.19 (2020): 8089-8095.

[10] Kevin Zakka, A Complete Guide to K-Nearest-Neighbors with Applications in Python and R, https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/, last accessed 2020/06/02.

[11] Warning Signs of a Heart Attack | American Heart Association. https://www.heart.org/en/health-topics/heart-attack/warning-signs-of-a-heart-attack

[12] Joonghyun, Ryu, et al. "Computation of molecular surface using Euclidean Voronoi Diagram." Computer-Aided Design and Applications 2.1-4 (2005): 439-448.

[13] Khan SS, Ning H, Wilkins JT, Allen N, Carnethon M, Berry JD, Sweis RN, Lloyd-Jones DM. Association of Body Mass Index with Lifetime Risk of Cardiovascular Disease and Compression of Morbidity. JAMA Cardiol. 2018 Apr 1;3(4):280-287. doi: 10.1001/jamacardio.2018.0022. PMID: 29490333; PMCID: PMC5875319.

[14] C. Thirumalai, A. Duba and R. Reddy, "Decision making system using machine learning and Pearson for heart attack," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017, pp. 206-210, doi: 10.1109/ICECA.2017.8212797.