Prediction and 10-year Risk Assessment of Major Adverse Cardiovascular Events using Machine Learning Techniques

Antonios Mavridis
Department of Digital Systems
University of Piraeus
Piraeus, Greece
antmav0@gmail.com

***Abstract*—Major Adverse Cardiovascular Events (MACE) are defined as the occurrence of any episode including death, stroke, thromboembolic event, acute myocardial infarction and stent thrombosis or target lesion revascularisation [1]. The prediction of major adverse cardiac events (MACE) could play an essential role in supporting clinical decisions that allow timely intervention for preventable and treatable complications. It could also allow management of low-risk patients without unnecessary admissions, investigations and monitoring. The risk models have played an important role in primary prevention and are often used in everyday clinical practice [2]. This paper proposes machine learning techniques for estimating a 10-year prediction of major adverse cardiovascular event. Supervised learning algorithms and classifiers such as Decisions Tree, Logistic Regression, K-Nearest Neighbors and Support Vector Machine (SVM) are used to differentiate a healthy patient from an unhealthy one. The study also shows the comparison of these methods in classification of patients.**

***Index Terms*—Machine Learning, Supervised Learning, Classification, Support Vector Machines, Decision Tree, K-Nearest Neighbors, Logistic Regression, Major Adverse Cardiovascular Events, Risk Assessment.**

I. Introduction

Machine learning is a field of study and is concerned with algorithms that learn from examples. Classification is a task that requires the use of machine learning algorithms that learn how to assign a class label to examples from the problem domain. Classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. Classification predictive modeling algorithms are evaluated based on their results. There are different types of classification methods ones which is called Binary. Binary Classification refers to those classification tasks that have two class labels. Typically, binary classification tasks involve one class that is the normal state and another class that is the abnormal state. The class for the normal state is assigned the class label 0 and the class with the abnormal state is assigned the class label 1. The most popular algorithms that can be used for binary classification are: Logistic Regression, K-Nearest Neighbors, Decision Trees, Support Vector Machine. Some algorithms are specifically designed for binary classification such as Logistic Regression and Support Vector Machine. We use this idea to address the 10-year prediction of a MACE event classification task. The study aims to train classifiers using a training set that enables the underlying algorithm to closely approximate a mathematical function that would have generated the labels of the training data. This helps classifiers to classify a test data point for which the class label is to be found. In other words, this technique finds the solution to our problem.

II. Data Set Description

The dataset contains 3315 samples from patients with acute first-ever ischemic stroke admitted between 1993 and 2016 within 24 hours after stroke onset and followed up for up to 10 years. An extended set of parameters is prospectively registered for each patient including demographics, medical history, vascular risk factors, previous treatment, stroke severity at admission, laboratory results, imaging data, in-hospital treatment and medication at discharge. Patients are followed up prospectively at the outpatient clinic at 1, 3 and 6 months after hospital discharge and yearly thereafter for up to 10 years or until death. More specifically, it is divided into 49 independent variables, mainly important factors that are associated with the occurrence of a MACE, and two dependent variables that provide the ground truth concerning the occurrence of a MACE. In our study we focused on the dependent variable named "MACE_10years". The dataset contains three different data types (String, Integer, Float). Columns that contain numerical values as string representations are transformed in the appropriate numerical data type. On the other hand, columns that contain categorical values as integer representations are transformed in the appropriate categorical data type. An overview of the dataset is shown in Table 1. In order to enhance readability, all variables are provided in rows.

TABLE I
ROWS OF CONTENT FOR THE DATASET

Variables	Description	Data Type
Independent variables		
Age	The age of the patient	String
Sex	The gender of the patient (0=female,1=male)	Integer
Hypertension	Hypertension issues detected (0= no, 1=yes)	Integer
Diabetes Melitus	Diabetes melitus detected (0= no, 1=yes)	Integer
Smoking	Patient smokes (0=no, 1=yes)	Integer
TIA	Transient Ischemic Attack	Integer
Dyslipidemia	Dyslipidemia (0 = no, 1 = yes)	Integer
Heart Failure	Heart Failure (0 = no, 1 = yes)	Integer
CAD	Coronary Artery Disease	Integer
PAD	Peripheral Artery Disease	Integer
Alcohol	Patient drinks alcohol (0= no, 1= yes)	Integer
Vascular Imaging Arterial All		Integer
Vascular Stenosis Degree All		Integer
PWML	Periventricular white matter lucencies detected (0= no, 1=yes)	String
AF	Atrial fibrillation (0= no, 1=yes)	Integer
AF Types	Atrial fibrillation types (0-2)	Integer
Obesity	Obesity detected (0 = no, 1 = yes)	String
eGFR	Estimated glomerular filtration rate (values from 0 to undefined)	String
TOAST	Trial of Org 10172 in Acute Stroke Treatment (1-2-3-7-9)	Integer
Hemorhag Transformation	Hemorhaggic transformation detected (0= no, 1 = yes)	String
Statin Prior	Administration of Statin (0 = no, 1= yes)	String
Cumadin Prior	Administration of Cumadin (0 = no, 1= yes)	String
ASA Prior	Administration of Aspirin (0 = no, 1= yes)	String
Statin Discharge	Discharge of Statin (0 = no, 1= yes)	String
Antiplatelets Discharge	Discharge of Antiplatelets (0 = no, 1= yes)	String
Cumadin Discharge	Discharge of Cumadin (0 = no, 1= yes)	String
NOACs Discharge	Discharge of Novel Oral Anticoagulants (0= no, 1= yes)	String
Diouretics Discharge	Discharge of Diouretics (0 = no, 1= yes)	String
B Block Discharge	Discharge of Beta blockers (0 = no, 1= yes)	String
CA Blocker Discharge	Discharge of Calcium Channel Blocker (0 = no, 1= yes)	String
ACE ARB Discharge	Discharge of Angiotensin converting enzyme inhibitors and Angiotensin receptor blockers (0 = no, 1= yes)	String
LeftVentricle Hypertr by ECG		Integer
ECG Ischemic Abnormal		String
MI by ECG		Integer
LeftVentricular Wall Abnormalities		String
EF	Ejection Fraction	String
Hypertrophy by ECHO		String
Left Atrial Diameter	Patient's left Atrial diameter	String
Cr Adm		String
Urea		String
Chol Adm		String
TG	Transglutaminase	String
HDL	High density Lipoprotein	String
LDL	Low density lipoprotein	String
Weight	Weight of patient	String
Height	Height of patient	String
BMI	Body Mass Index	String
Dependent Variables		
MACE 10 years	Occurrence of MACE in 10-years period (0= no, 1=yes)	Integer
MACE Time 10 years	Time in months of MACE occurrence	Float

III. Exploratory Data Analysis

A. Data Preparation

Prior to reviewing the proposed classification and feature selection methodologies, it is deemed necessary to go through all the procedures that the raw data have been submitted in order to obtain the form, the shape and the necessary characteristics for the analysis. These procedures involve cleaning, missing value management and reshaping. In this work, cleaning of data has been performed in the columns where inconsistencies were detected due to early departure or lost track of patients. Reshaping of dataset has been performed to exclude variables where missing values exceed 30% of the total amount of samples as show in Fig. 1. Variables excluded due to extensive missing values are the following: 'BMI', 'Height', 'Weight', 'LDL', 'HDL', 'TG', 'Left Atrial diameter'. 'Hypertrophy by ECHO', 'EF', 'Left Ventricular Wall Abnormalities', 'PWML'.

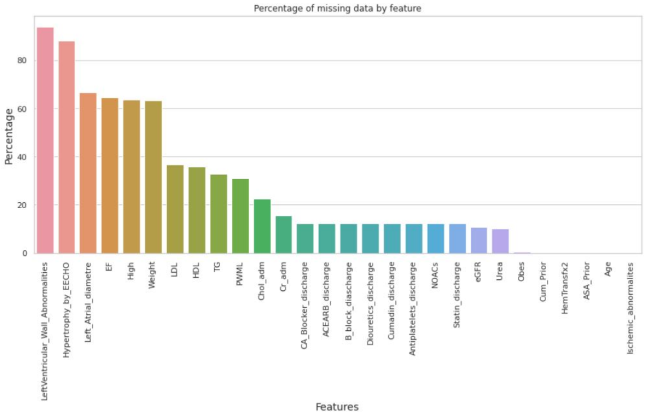


Fig. 1. Percentage of missing data by feature.

B. Data Distribution

The following are some descriptive conclusions that we have drawn from our data. Specifically, we observe that there are 1508 patients who are healthy and 572 patients who have a MACE incident, as show in Fig. 2.

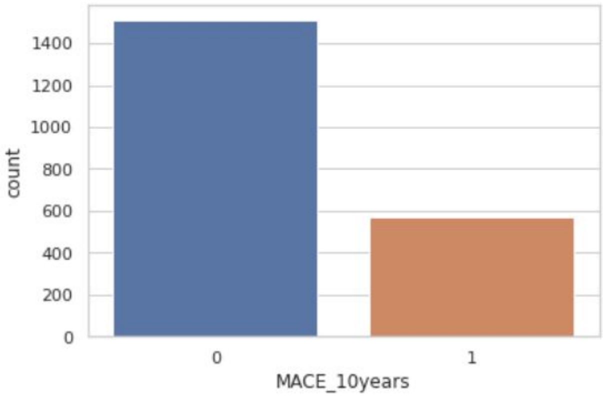


Fig. 2. Number of patients who developed MACE over a period of ten years.

Due to the imbalanced nature of the dataset it is difficult to make conclusions but based on what is observed slightly more males are suffering from MACE than females. The odds of developing MACE are higher in the diabetic patients and are almost similar between smokers and non smokers, and hypertensive and not hypertensive. The corresponding results are presented below in Fig. 3 and Fig. 4.

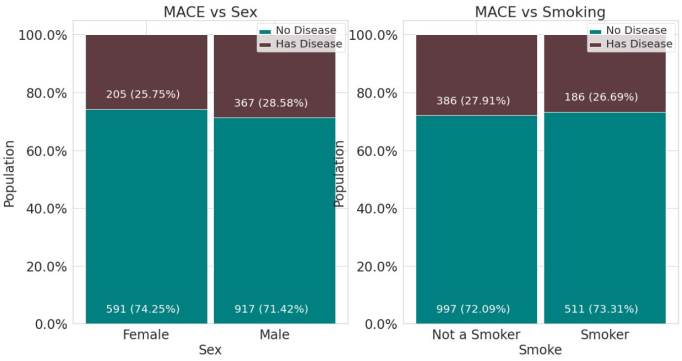


Fig. 3. Odds of developing MACE for Sex and Smoking patients.

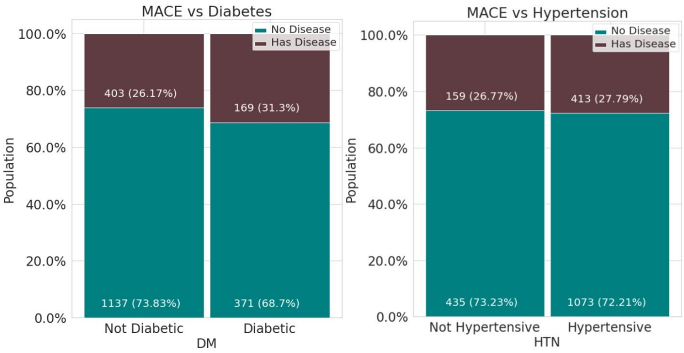


Fig. 4. Odds of developing MACE for Diabetic and Hypertension patients.

Furthermore in Fig. 5 is a breakdown of cases by age of patients. We observe that the people with the highest risk of developing MACE are between the ages of 65 and 80 i.e. the blue bars. Also in Fig. 6 shows the correlation heat map for all attributes of the data set. We observe that there are no features with more than 0.5 correlation with the ten year risk of developing MACE and this shows that the features a poor predictors. Therefore we need to carry out feature selection to pick the best features.

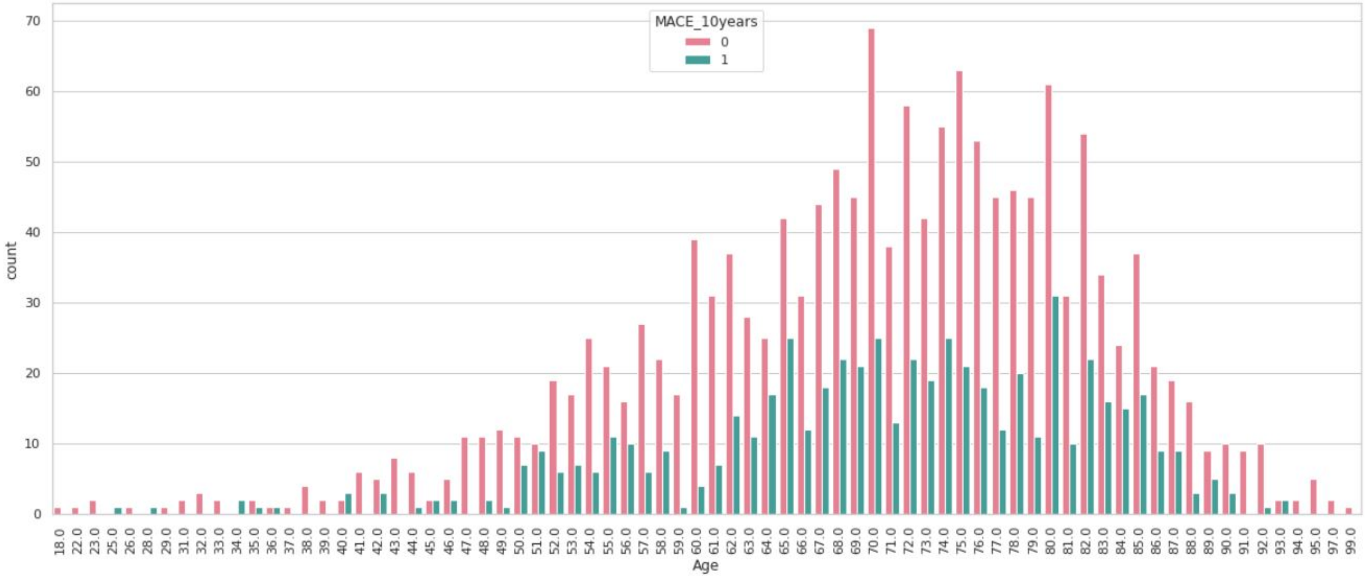


Fig. 5. Number of cases by age of patients.

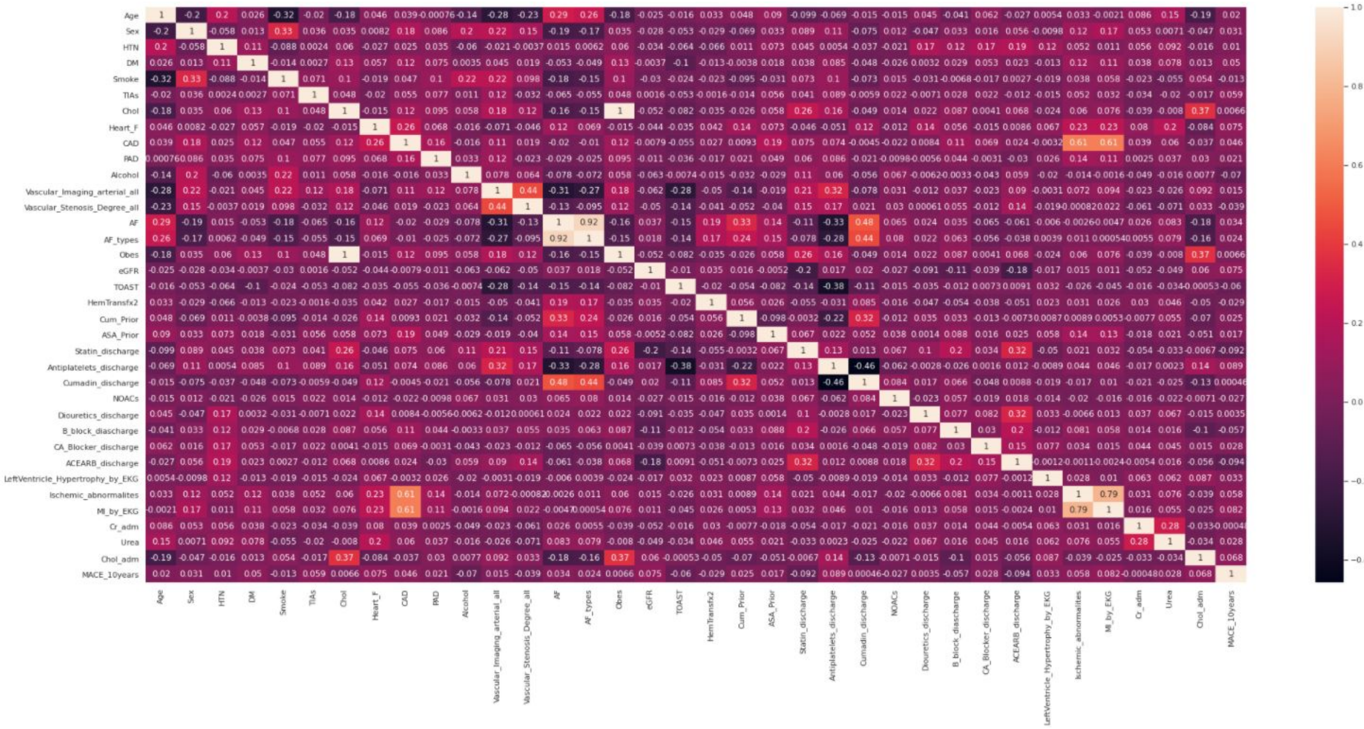


Fig. 6. Correlation heat map for all attributes of the data set.

IV. Feature Selection

Feature selection is often an important step in applications of machine learning methods and there are good reasons for this. Modern data sets are often described with far too many variables for practical model building. Usually most of these variables are irrelevant to the classification, and obviously their relevance is not known in advance. There are several disadvantages of dealing with overlarge feature sets. One is purely technical, dealing with large feature sets slows down algorithms, takes too many resources and is simply inconvenient. Another is even more important, many machine learning algorithms exhibit a decrease of accuracy when the number of variables is significantly higher than optimal. There are plenty of algorithms which were developed to reduce feature set to a manageable size. In our case we isolated the top features of the data set by using the Boruta Algorithm.

A. Boruta Algorithm

Boruta algorithm which is a wrapper built around the random forest classification algorithm. It tries to capture all the important, interesting features in a data set with respect to an outcome variable. The Boruta algorithm consists of following steps:

1) Extend the information system by adding copies of all variables (the information system is always extended by at least 5 shadow attributes, even if the number of attributes in the original set is lower than 5).
2) Shuffle the added attributes to remove their correlations with the response.
3) Run a random forest classifier on the extended information system and gather the Z scores computed.
4) Find the maximum Z score among shadow attributes (MZSA), and then assign a hit to every attribute that scored better than MZSA.
5) Find the maximum Z score among shadow attributes (MZSA), and then assign a hit to every attribute that scored better than MZSA.
6) Deem the attributes which have importance significantly lower than MZSA as ‘unimportant’ and permanently remove them from the information system.
7) Deem the attributes which have importance significantly higher than MZSA as ‘important’.
8) Remove all shadow attributes.
9) Repeat the procedure until the importance is assigned for all the attributes, or the algorithm has reached the previously set limit of the random forest runs.

After the implementation of Boruta Algorithm, we ended up with the following ten top features:

- Age
- Vascular Imaging Arterial All
- Vascular Stenosis Degree All
- Estimated Glomerular filtration Rate (eGFR)
- Trial of Org 10172 in Acute Stroke (TOAST)
- Discharge of Antiplatelets
- Discharge of ACEARB
- Cr Adm
- Urea
- Chol Adm

From the scatter plots of the top ten attributes is shown in Fig. 7, we infer that there are no features that split the data well.

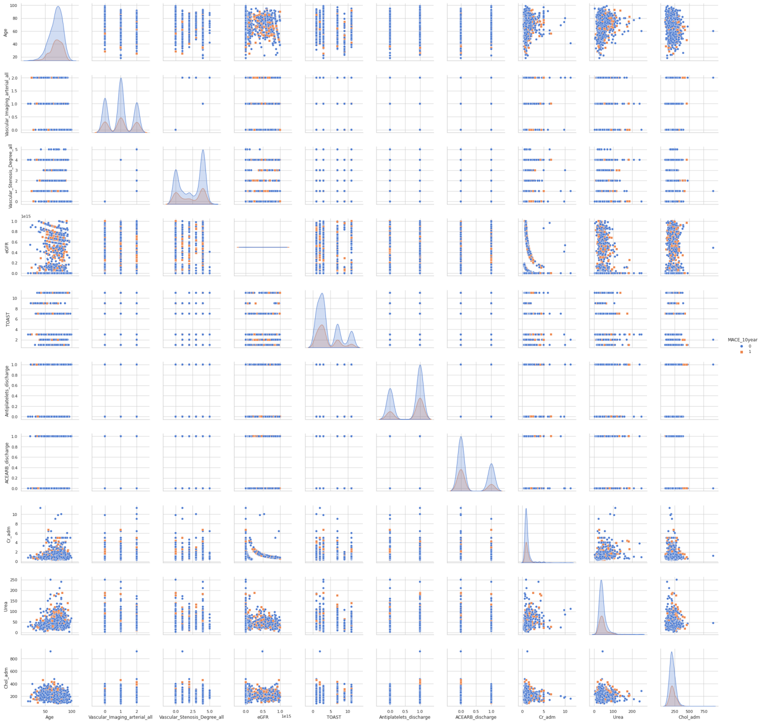


Fig. 7. A Scatter Plot between all the pairs of attributes in the dataset.

Since the dataset is imbalanced for every negative case there are about three positive cases. We may end up with a classifier that mostly predicts positive classes thus have a high accuracy but poor specificity or sensitivity. To adress this we will balance the dataset using The Synthetic Minority Oversampling Technique, or SMOTE for short.

V. Imbalanced Data Handling

A. Synthetic Minority Oversampling Technique

Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance. The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is performance on the minority class that is most important. One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don’t add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling

Technique. SMOTE first selects a minority class instance a at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b. This procedure can be used to create as many synthetic examples for the minority class as are required. It suggests first using random undersampling to trim the number of examples in the majority class, then use SMOTE to oversample the minority class to balance the class distribution. After the implementation of SMOTE technique we observe that there are 1206 patients who are healthy and 1507 patients who have MACE incident.

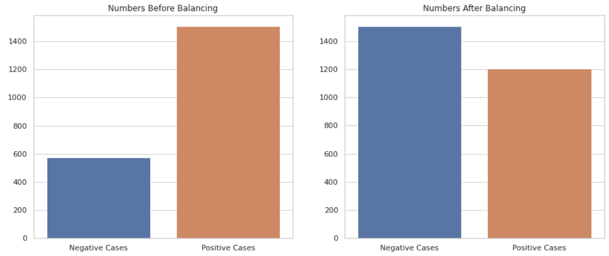


Fig. 8. A Bar Plot for the dataset before and after balancing.

VI. Classification Algorithms

Once we have defined our problem and prepared our data, we had to apply machine learning algorithms to the data to solve our problem. So, we define a test harness. The test harness is the data you will train and test an algorithm against and the performance measure you will use to assess its performance. The performance measure is the way you want to evaluate a solution to the problem. We trained the classifier on 80% of the data, and tested on the remaining 20%. To select the most appropriate method, we tested different classifiers and algorithms such as Logistic Regression, Support Vector Machines (SVM), K-Nearest Neighbor and Decision Tree.

A. *Logistic Regression*

Classification models, which directly assign class labels, without calculating probabilities, are called segregation models. One of the most well-known models in this category is called logistic regression, which directly estimates the forecast of a data instance x, using the values of its attributes [3].

Logistic regression is very similar to linear regression, but cannot be accomplished through it, due to the value calculated for a variable. In linear regression the value of the variable, which is calculated, is continuous, in contrast to the value in the logistic regression where the value is distinct. Unlike linear regression, logistic regression reduces the range of forecasting and the value it gets is always between 0 and 1.

For the analysis of logistic regression, it is initially calculated the probability ratio which called odds. If we consider p to be the probability of success of occurrence of the event and 1-p the probability of failure of occurrence of the event, then probability ratio is calculated by the formula:

$$odds = \frac{p}{1-p} \tag{1}$$

Finally, the logit function is defined, which is the physical logarithm of the probability ratio so that it can integrated into the regression model and is symbolized as:

$$logit(p) = ln(\frac{1}{p}) \tag{2}$$

The result of the logistic regression model is interpreted as the performance logarithm for the possibility of participation in the class.

B. *Support Vector Machines*

A Support Vector Machines (SVM) is a separator classification model that trains linear or non-linear decision limits in the attribute space in order to separate classes. Apart from maximizing the separability between the two classes, a support vector machine provides powerful customization capabilities, i.e. it has the ability to control the complexity of the model to ensure good generalization performance. Due to its unique ability to adapt training, SVM can train highly expressive models without overfitting problems. SVM has strong roots in statistical learning theory and is based on the principle of structural risk minimization.

Another unique aspect of SVM is that it represents the decision boundary using a subset of the training examples, known as the support vectors. Therefore, it is a divisive model, which is influenced only by the training instances that are close to the boundaries of the two classes, as opposed to learning the general distribution of each class [3].

C. *K-Nearest Neighbors*

The algorithm K-Nearest Neighbors (KNN), is quite interesting in the way it follows for the classification of data, since it works differently from the aforementioned algorithms. It is called lazy algorithm (lazy algorithm) because it memorizes the entire training set in memory. The advantage of this approach is that the classifier immediately categorizes the new training data collected.

The basic idea of the KNN algorithm for categorizing an element is the properties of each specific element given as input to the algorithm to be similar to the properties of the other points, at a certain distance from this. This distance is also called ”neighborhood”, from which its name derives algorithm. The steps that are followed for the implementation of the KNN algorithm are the following:

1) Selection of neighbors numbers and distance measure.
2) Find the neighbor where the element is to be categorized.
3) Categorize the new element.

In order to measure the similarity or the distance between points, we need to use some distance measure D (x1, x2). Such distance measures can is the Euclidean distance, the Manhattan distance, the Minkowski distance, the distance Chebyshev and the Hamming distance.

D. Decision Tree

Decision trees are the simplest and easiest way to classifies and forecasting. It is very easy to understand and interpret. In contrast with other categorization models, decision trees can be combined numerical and categorical features, but also to categorize incomplete characteristics. The process by which they end up in the categorization is done as follows:

- Each inner node of a tree is named from the attribute.
- Each branch - connection of two nodes is named with a condition or value for feature of the parent node.
- Each leaf is associated with the name of a class.

Initially, the tree receives as input a set of training data with the various features that characterize it. The branches of the tree contain the test values for each feature. The leaves of the tree correspond to the prices of the classes they have set. Input features can be discrete or continuous, as well as output value features. In case the output value is a distinct value then we have classification, while when the output value is a continuous function then we have regression. There are different implementations of this algorithm, some of which are IDE3, C4.5 and CART.

VII. Evaluation

A confusion matrix is a technique for summarizing the performance of a classification algorithm. The number of correct and incorrect predictions are summarized with count values and broken down by each class. In our two-class problem we are looking to discriminate between observations with a specific outcome, from normal observations. In this way, we can assign the event row as "positive" and the no-event row as "negative". We can then assign the event column of predictions as "true" and the no-event as "false". This gives us:

- "True Positive" (TP) for correctly predicted event values (the number of cases correctly identified as healthy patients).
- "False Positive" (FP) for incorrectly predicted event values (the number of cases incorrectly identified as healthy patients).
- "True Negative" (TN) for correctly predicted no-event values (the number of cases correctly identified as MACE patients).
- "False Negative" (FN) for incorrectly predicted no-event values (the number of cases incorrectly identified as MACE patients).

We can summarize this in the confusion matrix as follows

	Event	No-Event
Event	TP	FP
No-Event	FN	TN

Fig. 9. Confusion Matrix.

The following are the confusion matrices of the algorithms we used, which can help us to calculate more advanced evaluation metrics, which are presented below.

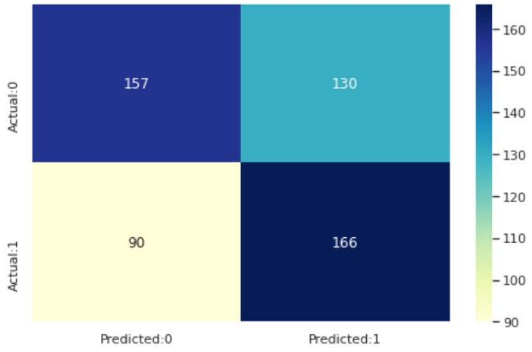


Fig. 10. Confusion Matrix for Logistic Regression.

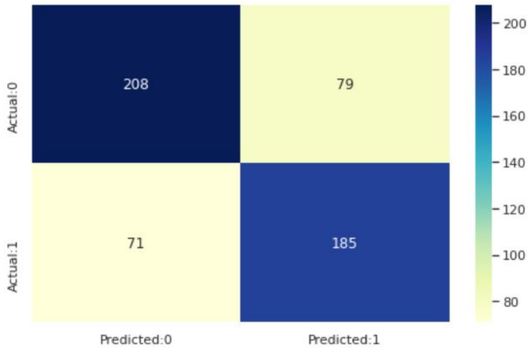


Fig. 11. Confusion Matrix for K-Nearest Neighbors.

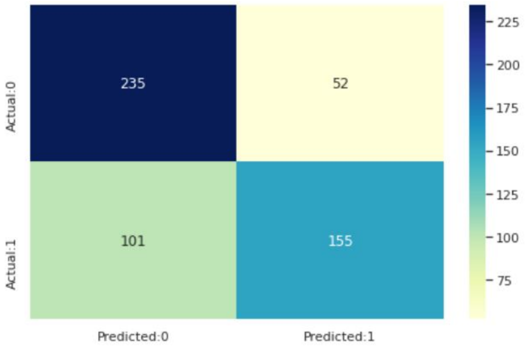


Fig. 12. Confusion Matrix for Decision Tree.

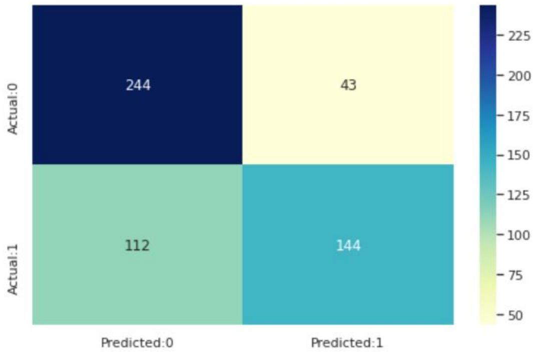


Fig. 13. Confusion Matrix for SVM.

Following measures have been used to measure the performance of the models implemented.

- Accuracy: The accuracy of the test is its ability to differentiate the patients with MACE and healthy patients correctly.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- Precision: The precision of a test is its ability to determine the number of notes that classifier labeled as MACE is actually MACE.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

- Recall: The recall of the test is its ability to determine how we predicted correctly out of all the positive classes.

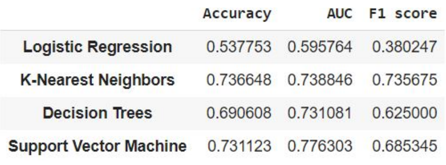
$$Recall = \frac{TP}{TP + FN} \tag{5}$$

- F1–Measure: F1-score helps to measure Recall and Precision at the same time.

$$F1 - Measure = 2\frac{Precision*Recall}{Precision+Recall} \tag{6}$$

Furthermore, another performance measurement for classification problem is AUC - ROC curve. A Receiver Operator Characteristic (ROC) curve is a graphical plot used to show the diagnostic ability of binary classifiers. A ROC curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR). The true positive rate is the proportion of observations that were correctly predicted to be positive out of all positive observations (TP / (TP + FN)). Similarly, the false positive rate is the proportion of observations that are incorrectly predicted to be positive out of all negative observations (FP / (TN + FP)). The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance [4].
To compare different classifiers, it can be useful to summarize the performance of each classifier into a single measure. AUC represent degree or measure of separability. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between genuine and fictious banknotes. An excellent model has AUC near to the 1 which means it has good measure of separability. An excellent model has AUC near to the 1 which means it has good measure of separability [4].

The following are the evaluation metrics of the algorithms we used:



	Accuracy	AUC	F1 score
Logistic Regression	0.537753	0.595764	0.380247
K-Nearest Neighbors	0.736648	0.738846	0.735675
Decision Trees	0.690608	0.731081	0.625000
Support Vector Machine	0.731123	0.776303	0.685345

Fig. 14. Evaluation Metrics.

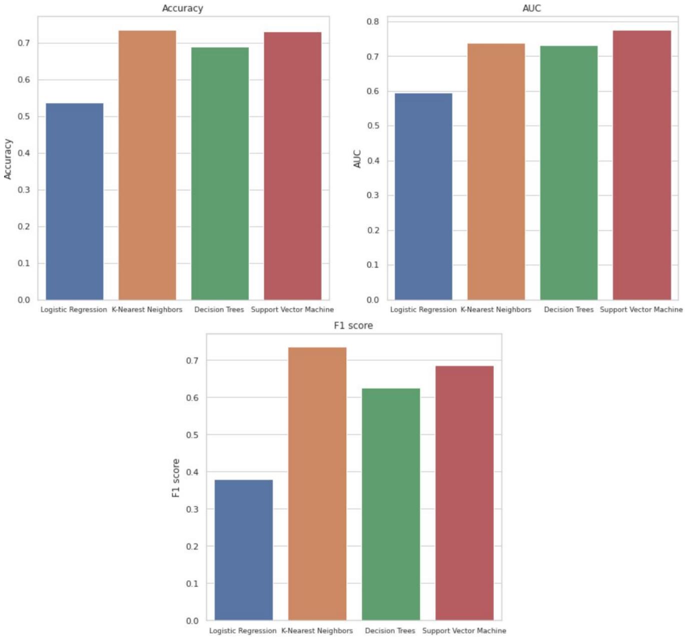


Fig. 15. A Bar Chart for Evaluation Metrics.

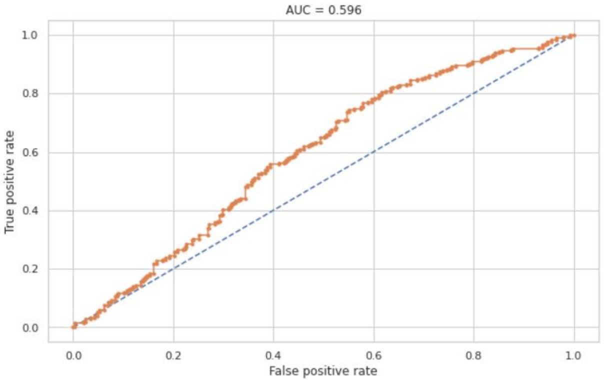


Fig. 16. ROC curve for Logistic Regression Classifier.

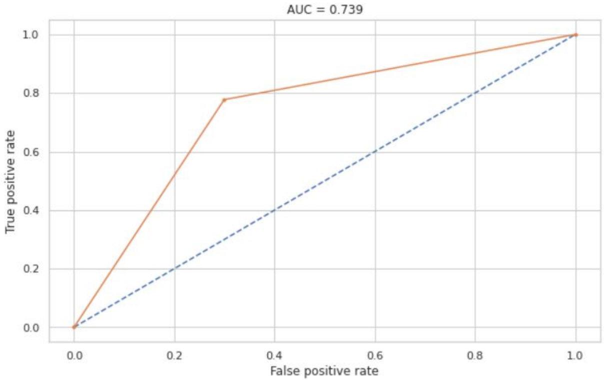


Fig. 17. ROC curve for K-NN Classifier.

Above in Fig. 16, Fig. 17 and below in Fig. 18 and in Fig. 19 are the ROC curves and AUC measurements for the classifiers we used to solve our problem.

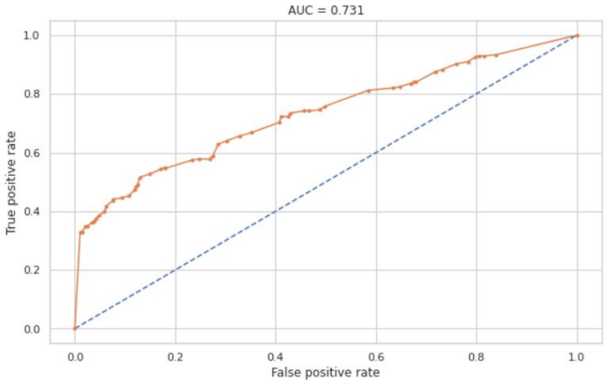


Fig. 18. ROC curve for Decision Tree Classifier.

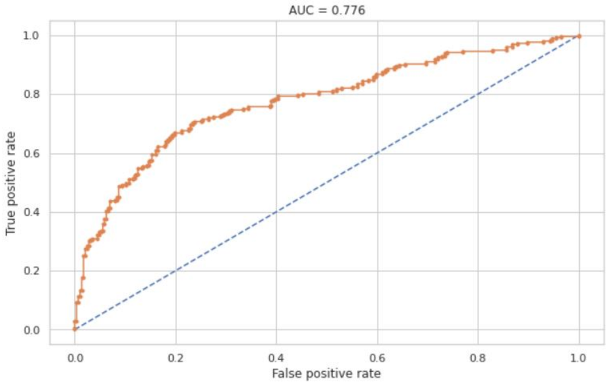


Fig. 19. ROC curve for SVM Classifier.

VIII. Results and Discussion

In the learning phase, the models are trained with 80% data i.e 2.713 samples out of which 985 samples were of healthy patients and 1205 were of patients who have MACE. For testing the models, remaining 523 samples have been used, where 232 were of healthy patients and 291 were of patients who have MACE. In our experiment, we evaluated the performance of four models, Logistic Regression and Support Vector Machines (SVM), K-Nearest Neighbors and Decision Tree. The results after the application of Feature Selection with the implementation of the Boruta Algorithm and after the Data Balancing with the SMOTE technique shows that K-Nearest Neighbors and SVM models predict our data more accurately. Then was followed by further exploratory evaluation of the models that we used. The performance of the models are presented in Fig. 14. Furthermore, the metric AUC for SVM is higher than that of K-Nearest Neighbors. Based on our results, we conclude that the Support Vector Machine achieves a very good classification and better than K-Nearest Neighbors.

IX. Conclusion

After analyzing various techniques used to detect major adverse cardiovascular events over a period of 10 years, this paper presents four supervised learning techniques that are used to predict the early diagnosis of an event. Extensive experiments have been performed on the data set using the four models to find the best model suitable for the classification of the patients. ROC and other metrics have been calculated to compare the performance of all the techniques. The results show that Support Vector Machines outperforms the other methods and give a higher success rate.

References

[1] Fauchier L, Lecoq C, Ancedy Y, Stamboul K, Saint Etienne C, Ivanes F, Angoulvant D, Babuty D, Cottin Y, Lip GY. ”Evaluation of 5 Prognostic Scores for Prediction of Stroke, Thromboembolic and Coronary Events, All-Cause Mortality, and Major Adverse Cardiac Events in Patients with Atrial Fibrillation and Coronary Stenting.”. Am J Cardiol. 2016 Sep 1;118(5):700-7. doi: 10.1016/j.amjcard.2016.06.018.

[2] N. Liu, X. Z. Koh, J. Goh, Z. Lin, B. Haaland, B. P. Ting and M. E. H. Ong, ”Prediction of adverse cardiac events in emergency department patients with chest pain using machine learning for variable selection”

[3] P. N. Tan, M. Steinbach, A. Karpatne and V. Kumar, 2005 “Introduction to Data Mining” J. Name Stand. Abbrev., in press.

[4] https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/ interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].