Multiclass Classification of Coronary Artery Disease: Machine Learning Approach

1st Miguel Silva
Department of Electronics, Telecommunications and Informatics (DETI)
University of Aveiro (UA)
Aveiro, Portugal
mig.silva@ua.pt

2nd José Brito
Department of Mathematics (DMat)
University of Aveiro (UA)
Aveiro, Portugal
josencbrito@ua.pt

***Abstract*—Machine Learning techniques are becoming more often used in the field of medicine, specially in predicting heart disease classification. This work presents the utilization of 3 machine learning classifiers, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), with different feature selection methods and class balancing techniques, for heart disease classification. The models were built based on the Cleveland Heart Disease data base with the goal of obtaining a non-binary classification of coronary heart disease (from 0 to 4). The models were evaluated for each combination of feature selection method, class balancing technique and chosen classifier, based on accuracy, recall, precision, and confusion matrices. The model with best performance on accuracy and recall had a value of 0.70 for both parameters, and the one with best precision and analysis from the confusion matrix (based on a clinical point of view) had precision = 0.72.**

***Index Terms*—Coronary Artery Disease, Machine Learning, Decision Trees, Support Vector Machine, K-Nearest-Neighbors, classification.**

I. Introduction

Machine Learning (ML) and Artificial Intelligence (AI) have proven to be a handy tool in disease diagnose for physicians. This applies for diagnosis and classification of heart diseases, which are the number one cause of death in the world, where coronary artery disease (CAD) tops the list as the most deadly [1]. Therefore, multiple projects have been developed to assess this issue, where the goal was to predict the diagnosis of CAD based on clinical parameters described in Cleveland database's Heart Disease Dataset [2]. However, most of papers only had the goal of classifying CAD in a binary way (presence or absence of CAD) [1]–[3]. Therefore, in this paper we are developing several ML models to classify CAD from 0 to 4, where 0 is the absence of CAD and 1-4 indicate four different levels of severity of CAD.

In this study, we have applied three ML algorithms, Decision Trees, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), for classification of existence and severity of CAD.

The main goal was to obtain the best algorithm, applying different parameters, kernels and distance metrics, and different feature selection algorithms to analyze performance and obtain the optimal ML classification with a proper set of parameters and features, according to the accuracy of their results.

II. Related Work

Nassif et al. [1], resorting to Cleveland database, have performed a similar work as presented in this paper, however, classifying CAD on a binary basis (existence or absence of CAD). From the 13 available features, they have selected 7 according to a combination of three selection methods. These were (1) Information Gain evaluator with Ranker search, (2) Correlation evaluator with Ranker search, and (3) Classifier Subset evaluator on Naive Bayes with Best First search, having selected the following features: *cp*, *thalach*, *exang*, *oldpeak*, *slope*, *ca*, and *thal*. They have selected three classifiers: SVM, KNN, and Naive Bayes (NB). For all classifiers, they have determined the best parameters to fit each model according to the highest accuracy. Then, the models were evaluated by accuracy, recall, specificity, and precision. They have found that for all these performance metrics the NB classifier achieved higher scores, with 84% for accuracy, 88% for recall, 80% for specificity, and 83% for precision. The group highlighted that an ideal model would have had higher values for recall and precision, since in clinical context it is relevant for a model to predict correctly the highest number of positives possible. Even though the NB classifier performed better than the other models, performance was relatively good for the SVM and KNN classifiers, where recall and precision were 88% and 82% for SVM, respectively, and 84% and 80% for KNN, respectively.

Shadman *et al*. [3] have developed a heart disease detection using ML algorithms, further combining it with a real-time cardiovascular health monitoring system. The classification system was binary (absence of presence of CAD), and the dataset used was also the Cleveland Heart Disease Dataset, however, merged with the Statlog Heart Disease Dataset [4]. Both datasets presented the same features, with 303 and 270 samples, respectively, making a total of 566 samples (after removal of missing values). Several models were developed resorting to five algorithms: NB, Artificial Neural Networks, SVM, Random Forest, and Logistic Regression. The validation of the models was done through a 10-fold cross-validation, however, the methods for selecting the hyper-parameters of the classifiers were not mentioned, only referring that the models were developed through a Java Based Open Access Data

Mining Platform, WEKA. For performance analysis, precision, F1-score, accuracy, sensitivity (recall), and specificity were determined for all models. According to these parameters, the best model was the SVM classifier, scoring 97.53% on accuracy, 97.50% on sensitivity, 94.94% on specificity, 95.95% on precision, and 96.72% on F1-score, proving to be the best model to classify the existence of CAD. Random Forest and Logistic Regression also presented high performance scores. The group mentioned in the end that they have tested the performance of the models with a reduced number of features, but the algorithms proved to have a better performance when fitted with all 13 variables.

The above reviewed papers and other works [5] have proven well the possibility of building ML models with good performances for prediction of CAD diagnosis. However, as mentioned before, they did not contemplate the classification of CAD in multiple degrees. Therefore, this paper is picking up from there.

III. Dataset

As mentioned before, the dataset used on the project was the Cleveland Heart Disease Dataset [2]. The dataset contains 303 samples from which 6 have missing values, therefore, those samples were excluded. From the 297 patients considered, 96 were female and 201 were male, with age between 29 and 77 (mean, 54.5). There are 13 variables, as described below (detailed description from Nassif *et al.* [1]):

1) *age* (integer)
2) *trestbps* (integer) - resting blood pressure /mm Hg (on admission to the hospital)
3) *chol* (integer) - serum cholesterol in mg/dl
4) *thalach* (integer) - Maximum heart rate achieved /bpm
5) *oldpeak* (integer) - ST depression induced by exercise relative to rest
6) *ca* (integer) - Number of major vessels (0-3) colored by fluoroscopy
7) *sex* (categorical)
   a) 0: female
   b) 1: male
8) *cp* (categorical) - chest pain
   - 1: typical angina
   - 2: atypical angina
   - 3: non-anginal pain
   - 4: asymptomatic
9) *fbs* (categorical) - fasting blood sugar $>$ 120 mg/dl
   - 0: False
   - 1: True
10) *restecg* (categorical) - Resting electrocardiographic results
   - 0: Normal
   - 1: Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV)
   - 2: Showing probable or definite left ventricular hypertrophy by Estes' criteria
11) *exang* (categorical) - Exercise induced angina
   - 0: No
   - 1: Yes
12) *slope* (categorical) - The slope of the peak exercise ST segment
   - 1: Up sloping
   - 2: Flat
   - 3: Down sloping
13) *thal* (categorical)
   - 3: Normal
   - 6: Fixed defect
   - 7: Reversible defect
14) *num* (label) - The final diagnosis of heart disease (angiographic disease status)
   - 0: Absence of CAD
   - 1: First Level of Severity for CAD
   - 2: Second Level of Severity for CAD
   - 3: Third Level of Severity for CAD
   - 4: Fourth Level of Severity for CAD

The distributions of features are shown in Figure 2 and the heat-map with Spearman correlation between features is in figure 1. As we can see, *age*, *trestbps*, *chol*, *thalach*, and *oldpeak* are continuous variables, as all other variables are discrete. Also, it is perceivable that the only feature that might have a normal distribution is *age*, but other than that, distributions are very different from a normal one.

The dataset was split into a 80%-20% fraction, where the latter set was put aside as a further testing set, as to simulate real world data for performance evaluation (60 samples). This testing set was not totally randomly generated to ensure that each of the five classes had 20% of their samples in this set (random select of 20% from each class). Therefore, the dataset that was used for creating the models had 237 samples.

IV. Methodology

A. Feature selection

Feature selection finds its importance in model implementation regularly. Gokulnath *et al*. [6] explored the optimization of feature selection based on SVM classifier for heart disease classification (binary classification). They have proposed a method and compared with several existing selection algorithms, such as Information Gain. When classifying heart disease with SVM and all features, accuracy was 83.70%, but after applying the proposed method, accuracy increased to 88.34%. In this paper, we did not apply this selection method, however the group proved that reducing the number of features can be crucial for boosting performance. Therefore, in this paper we have assessed four feature selection methods:

1) All features: All 13 features from the dataset.

2) Spearman's correlation feature elimination: Spearman's correlation is applicable to compare two variables when they don't follow a normal distribution [7], which is the case in this dataset. Therefore, we proposed to select variables through elimination of features that have higher correlation with other

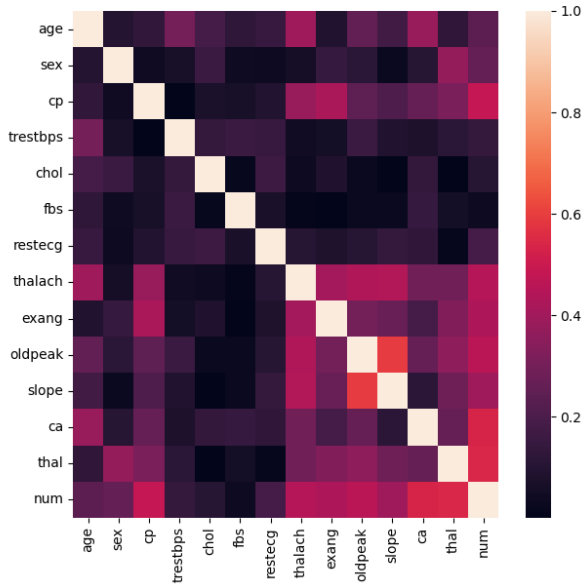

Fig. 1: Distribution of features

features. The absolute correlation threshold was set to 0.2 so that roughly half of the features (6) would be selected. As seen in figure 1, the absolute Spearman's correlation between features is generally low, therefore the threshold has been set significantly low.

3) Information Gain selection: Information Gain selection is processed through quantifying the mutual information between two variables resorting to entropy, where two dependent variables have a higher mutual information, and vice-versa [8]. After obtaining the information gain of all variables, approximately half of the features were selected.

4) 3-by-3 combinations from all features: All possible combinations of features in groups of three, so that we would have the maximum number of variables for visualization of data.

B. Class imbalance

The number of observations per class was not even: we had 128 samples from class 0, 43 from class 1, 28 from class 2, 28 from class 3, and 10 from class 4. This class imbalance could be a problem leading to bias of the model. Therefore, we assess this problem in three different ways, as presented below.

1) Cloning oversampling: For classes 1, 2, 3, and 4 we have cloned all samples by simply doubling them, however, it's worth mentioning that this method does not provide with addtional information about the data and can lead to overfitting of the model [9].

2) Synthetic Minority Oversampling Technique (SMOTE): This method creates new hypothetical samples for classes that have a lower representation in the dataset. Generally, this algorithm selects a random data point on the feature space from a minority class and then identifies the *k*-nearest neighbors of that same class, connecting these data points with virtual lines. Then, random samples are generated on those virtual lines, so that the new samples may be created between near data points of the minority class. Finally, the iterative method creates as many synthetic samples as needed, so that the number of samples from the under-represented classes is the same as majority class [10]. The limitation of this method comes with the fact that SMOTE has proven to work better with low dimensional data, and since we have an original set with data with 13 dimensions, it is likely that the algorithm will not bring many benefits when working with all features [11]. An example of data points before and after SMOTE is presented in figures 3 and 4, respectively, taking into consideration *age* and *chol* features.

3) Class weights: In this method, instead of oversampling data, we have attributed to each class a weight to be implemented as a hyper-parameter of the model. To achieve this goal, we have resorted to *Scikit-learn*'s *compute_class_weight* [12] function, which gives higher weights to classes that have lower representation in the dataset.

C. ML algorithms

1) Decision Trees: From the information in [13], Decision Trees are decision support models applying a flowchart-like model to make decisions, mainly used in multi-class classification problems. The primary objective of this decision model is to creation a structure capable of distinguish the multiple classes of different data points and assign new entries to the most adequate class.

In the search of the best model capable of creating the most suitable decision rules for the problem, two different criteria functions were considered [14]:

- Giny: $H\left(Q_m\right) = \sum_k = p_{mk}\left(1 - p_{mk}\right)$;
- Entropy: $H\left(Q_m\right) = \sum_k = p_{mk}\log\left(p_{mk}\right)$;

with Q_m the information related to node m and $p_{mk} = \frac{1}{n_m}\sum_{y \in Q_m} I\left(y = k\right)$. Also, for tuning the models, the maximum depth of the tree and the minimum number of samples per leaf were adjusted through 10-fold cross-validation.

2) Support Vector Machine: SVM is a machine learning algorithm for classification problems, usually meant for binary classification, however, it can be adapted for a multi-class problem [15], [16]. The main goal of SVM is to create boundaries for the data points, maximizing the space between those points, and distinguish the different classes through a kernel function [17].

For this study case, 3 kernel functions selected, [15], [17]:

- Linear Function: $H\left(x, x'\right) = \langle x, x' \rangle$;
- Radial Basis Function: $H\left(x, x'\right) = e^{-\gamma||x-x||^2}$;
- Sigmoid Function: $\tanh\left(\gamma\langle x, x' \rangle + r\right)$.

The different kernels were selected to obtain the function capable of maximizing the separation between the data entries or, in other words, to maximize the accuracy and precision of the results. Also, the parameters *C* and *gamma* were adjusted through 10-fold cross-validation.

For this algorithm, it was necessary to standardize the data.

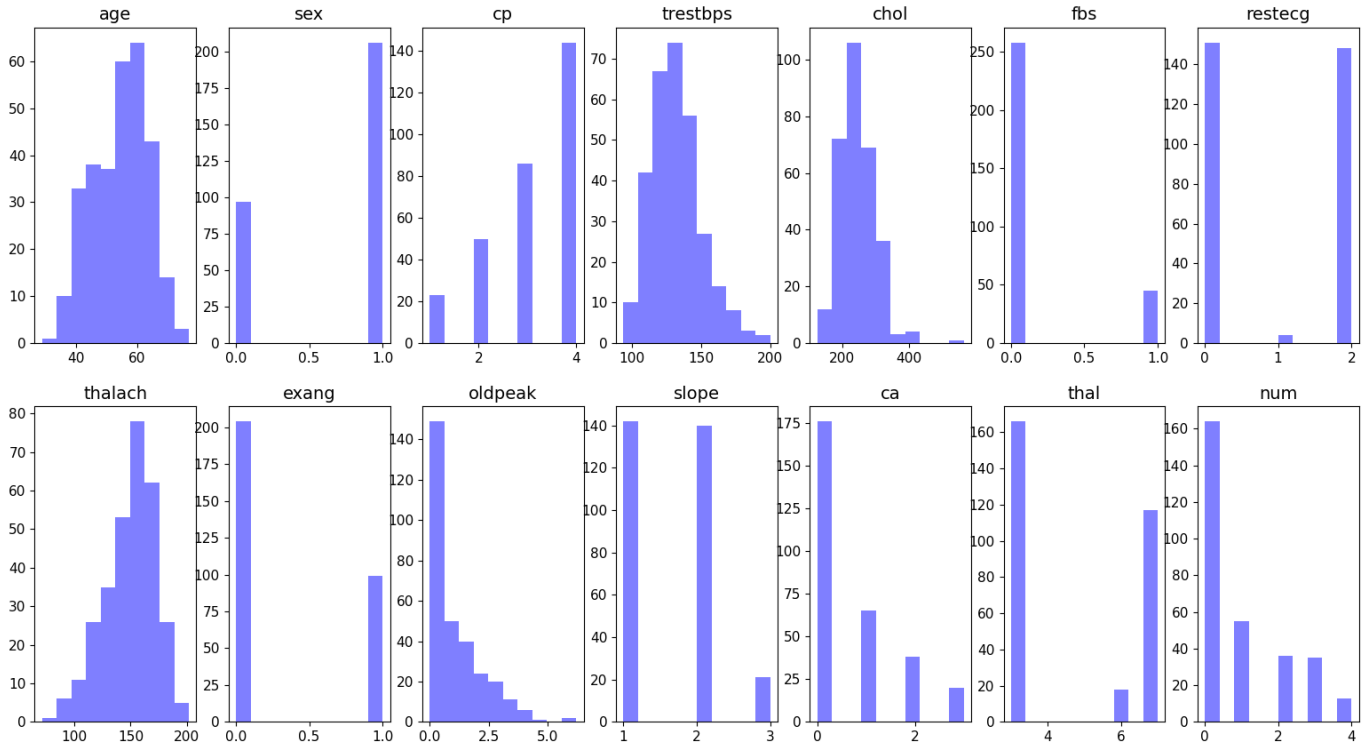

Fig. 2: Distribution of features

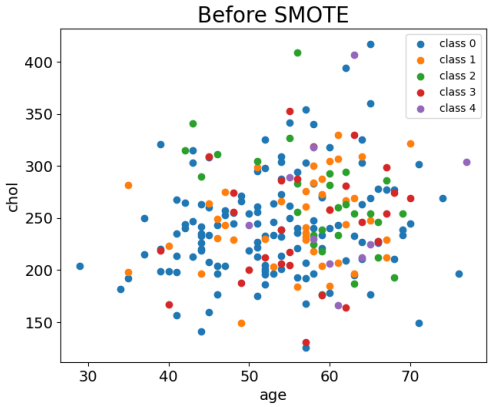

Fig. 3: Example of data before SMOTE

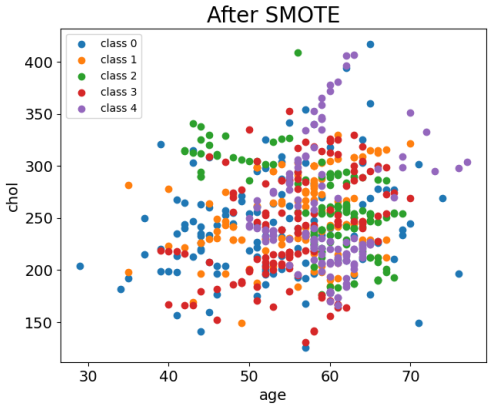

Fig. 4: Example of data after SMOTE

3) K-Nearest-Neighbors (KNN): As it is stated in [16], [18], the K-Nearest-Neighbors is a semi-supervised algorithm used for classification problems.

The algorithm consists on associating a new data point to the majority class of its K nearest data points found through a distance metric previously chosen.

As proceeded in Sections IV-C1 and IV-C2, 3 distance metrics were considered, these metrics can be consulted in [18] for more information:

- City Block Distance: $d_{st} = \sum_{j=1}^{n} |x_{sj} - y_{tj}|$;
- Euclidean Distance: $d_{st}^2 = (x_s - y_t)(x_s - y_t)'$;
- Chebychev Distance: $d_{st} = \max_j \{|x_{sj} - y_{tj}|\}$.

Afterward, the most suitable distance metric for the KNN classification problem in order to maximize performance was selected through 10-fold cross-validation, as well as for the number of neighbors.

For this algorithm, it was necessary to standardize the data.

D. Evaluation Metrics

There are various evaluation metrics that can be applied to analyze the performance of a model, [6], [19]. In this work, to evaluate the different ML algorithms, the following metrics were selected:

- Accuracy: The ratio of elements assigned to their class correctly.
- Recall: The percentage of the actual positive cases of data classified correctly.
- Precision: The percentage of the positive cases of data classified correctly.
- Confusion Matrix: A $N \times N$ matrix that describes the number of elements predicted correctly.

V. Results

To create and find the most fitting predictive model, the different ML classifiers were applied with combinations of parameters to find the best hyper-parameters for each ML algorithm used, according to their accuracy. This was done through the *GridSearchCV* function from *Scikit-learn*, with a cross validation of 10-fold cross-validation. To improve the models' results, feature selection and oversampling were applied sequentially on the training set for each predictive algorithm. It's worth mentioning that for the models that had only three features as a consequence of the feature selection from 3-by-3 combinations did not undergo grid search, since it would be computationally too demanding. In the end, a total of 867 of models per classifier were fitted, with all possible combinations of selected features and balancing techniques possible. As an example, the parameters obtained for the models with best accuracy are presented are displayed in the Table I.

TABLE I: Parameters Applied On Each ML Algorithm With The Best Accuracy

	Decision Tree	**Support Vector Machine**	**K-Nearest-Neighbors**
Criteria / Kernel / Metric	Gini	Linear	Euclidean
Maximum Depth	Until all nodes are pure	-	-
Minimum Number of Samples per Node	1	-	-
Number of Neighbors	-	-	5
Gamma	-	0.01	-
C	-	1000	-

As mentioned before, the evaluation of the models was done resorting to three performance metrics: accuracy, recall, and precision. Tables II, III, and IV present the values of these metrics, respectively. These tables also present the class balancing technique, the feature selection method and the correspondent selected features.

According to the model's evaluation by accuracy in Table II, the most important attributes were *sex*, *restecg* and *ca* for the Decision Tree model. The features *sex*, *restecg*, and *ca* were considered to be the most important for the SVM model. Finally, *sex*, *thal*, and *ca* were considered the features that have given the best accuracy. Accuracy for all models was relatively similar, however, the Decision Tree classifier proved to be the best model.

TABLE II: Best ML Models By Accuracy

Classifier	**Class Balancing**	**Feature Selection**	**Features**	**Accuracy**
Decision Tree	Doubling	3-by-3 combinations	*sex* *restecg* *ca*	0.7000
SVM	Doubling	3-by-3 combinations	*sex* *restecg* *ca*	0.6833
KNN	Doubling	3-by-3 combinations	*sex* *ca* *thal*	0.6667

TABLE III: Best ML Models By Recall

Classifier	**Class Balancing**	**Feature Selection**	**Features**	**Recall**
Decision Tree	Doubling	3-by-3 combinations	*sex* *restecg* *ca*	0.7000
SVM	Doubling	3-by-3 combination	*sex* *restecg* *ca*	0.6833
KNN	Doubling	3-by-3 combinations	*sex* *ca* *thal*	0.6667

As for recall, which is the ratio of true positive predictions out of all real positive instances, the obtained values were similar to accuracy, however the features and re-sampling methods were different, where now SMOTE re-sampling played an important role for improving the model's performance. Taking precision into consideration, the ratio of true positives out of all positive predictions, the best model was again the one with the Decision Tree classifier, with doubling of minority classes to solve class imbalance. It's worth mentioning that both recall and precision were calculated through a weighted mean of all classes' recall and precision, respectively. Finally, a relevant remark needs to be done to the feature selection methods: all optimal models underwent feature selection with 3-by-3 combinations, proving that classifications worked better under circumstances with a few number of features.

For a better evaluation of the models created, Figures

TABLE IV: Best ML Models By Precision

Classifier	**Class Balancing**	**Feature Selection**	**Features**	**Precision**
Decision Tree	SMOTE	3-by-3 combinations	*restecg* *exang* *thal*	0.7191
SVM	SMOTE	3-by-3 combinations	*sex* *oldpeak* *thal*	0.7138
KNN	Class weights	3-by-3 combinations	*cp* *exang* *ca*	0.6939

5, 6, and 7 present the confusion matrices of the models that have showed the best accuracy, recall, and precision, respectively. All of these three were trained with the Decision Tree algorithm.

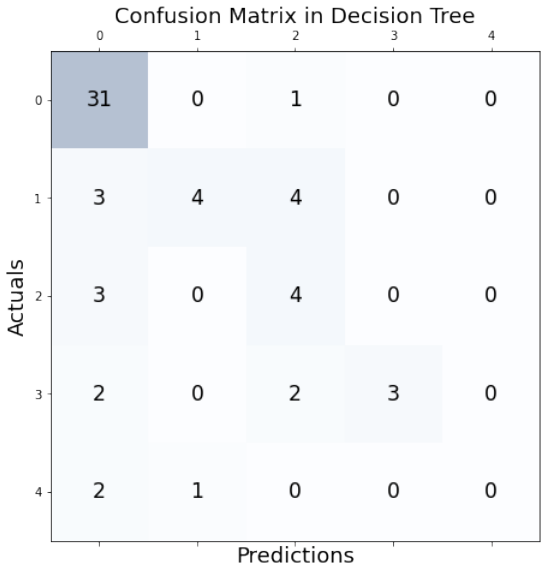

Fig. 5: Confusion Matrix of optimal model by accuracy (Decision Tree)

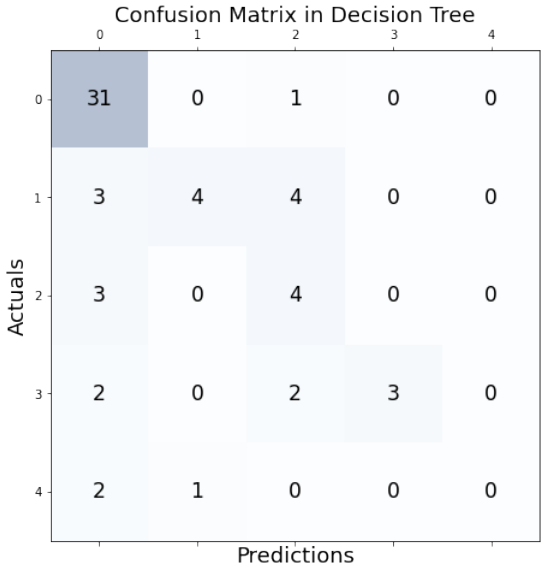

Fig. 6: Confusion Matrix of optimal model by recall (Decision Tree)

As we can see from Tables II and III, and Figures 5 and 6, the models that have achieved the highest accuracy and highest recall are the same model. Therefore, having this model achieved top-score in both metrics, we may consider it one of the best, if not the best. Also, even though it did not achieve the highest precision, its value was 0.7090, ranking third among all Decision Tree models in terms of precision.

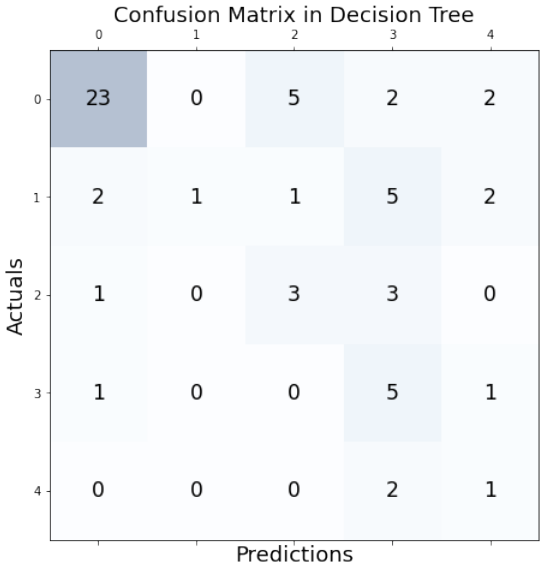

Fig. 7: Confusion Matrix of optimal model by precision (Decision Tree)

However, if we consider that having a wrong prediction that classifies the disease with a superior classification than reality is more important than the opposite situation, we can see from the confusion matrix in Figure 7 that the model that best meets this criterion is the one that scored the highest precision, although having lower accuracy and recall (0.5500 for both scores). This is visible in that most of the wrong predictions are located in the upper half of the confusion matrix relatively to the diagonal. This situation does not happen in the previously mentioned model, which can be a negative factor in the clinical context, where the model may more often classify a disease with a lower degree that it actually has. Besides, the model with higher precision was the only model to have correctly predicted CAD from class 4, which was the class with fewer samples (two samples in the test set).

Another important fact to notice is that the best performance of Decision Tree classifier occurred with discrete variables, as seen in Tables II, III and IV. This corroborates what is generally known from Decision Trees, since they tend to have better performance when features are discrete [20], mostly because these classifiers have to split continuous variables to proceed with classification.

To conclude the discussion, it is clear that the performance metrics are far from optimal. The obtained values of around 70% in all these parameters reveal relevant weaknesses of the model, being generally considered decent [21].

VI. Conclusions

In predicting the existence and severity of CAD, this research studies 3 distinctive ML algorithms.

To handle the data applied for the case study, different methods were used to deal with class imbalance, as well as

dimension reduction techniques to decrease the complexity and over-fitting. Afterward, all models were trained, validated for hyper-parameter tuning, and tested to obtain the most accurate model.

The Decision Tree classifier achieved the best results in all performance metrics. One model achieved an evaluation of 70.00% on accuracy and recall, and another had 71.91% on precision. These results were obtained through the application of cloning oversampling and SMOTE, respectively, to deal with classes imbalance, through the application 3-by-3 combinations feature selection. However, performance is still far from ideal to classify CAD with a multi class system.

This work can be studied more intensively by learning new ways to improve the results of the Decision Tree model.

VII. Future work

As for future work, since the models with features selected through 3-by-3 combinations did not undergo grid search for finding the best hyper parameters (due to computational limitations), it could be an advantage to perform this step on the models that had the best performances.

Another relevant proposal for future work would be exploring Neural Networks as multi-class classifiers due to their greater complexity and relevance in the nowadays-ML-context.

References

[1] A. B. Nassif, O. Mahdi, Q. Nasir, M. A. Talib, and M. Azzeh, "Machine learning classifications of coronary artery disease," in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pp. 1–6, 2018.

[2] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart Disease." UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C52P4X.

[3] S. Nashif, M. R. Raihan, M. R. Islam, and M. H. Imam, "Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system," *World Journal of Engineering and Technology*, vol. 6, no. 4, pp. 854–873, 2018.

[4] "Statlog (Heart)." UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C57303.

[5] A. Caliskan and M. E. Yuksel, "Classification of coronary artery disease data sets by using a deep neural network," *The EuroBiotech Journal*, vol. 1, pp. 271–277, Oct. 2017.

[6] C. B. Gokulnath and S. Shantharajah, "An optimized feature selection based on genetic approach and support vector machine for heart disease," *Cluster Computing*, vol. 22, pp. 14777–14787, 2019.

[7] L. Myers and M. J. Sirois, "Spearman correlation coefficients, differences between," *Encyclopedia of Statistical Sciences*, 2006.

[8] P. E. Latham and Y. Roudi, "Mutual information," *Scholarpedia*, vol. 4, no. 1, p. 1658, 2009. revision #186917.

[9] J. Brownlee, "Random oversampling and undersampling for imbalanced classification," Jan 2021.

[10] J. Brownlee, "Smote for imbalanced classification with python," Mar 2017.

[11] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, Mar. 2013.

[12] scikit, " *Estimate class weights for unbalanced datasets*."

[13] X. Guan, J. Liang, Y. Qian, and J. Pang, "A multi-view ova model based on decision tree for multi-classification tasks," *Knowledge-Based Systems*, vol. 138, pp. 208–219, 2017.

[14] scikit, " *Mathematical formulation of Decision trees*."

[15] S. Abe, "Analysis of multiclass support vector machines," *Thyroid*, vol. 21, no. 3, p. 3772, 2003.

[16] D. A. Anggoro and N. D. Kurnia, "Comparison of accuracy level of support vector machine (svm) and k-nearest neighbors (knn) algorithms in predicting heart disease," *International Journal*, vol. 8, no. 5, pp. 1689–1694, 2020.

[17] T. Kavzoglu and I. Colkesen, "A kernel functions analysis for support vector machines for land cover classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 11, no. 5, pp. 352–359, 2009.

[18] K. Chomboon, P. Chujai, P. Teerarassamee, K. Kerdprasop, and N. Kerdprasop, "An empirical study of distance metrics for k-nearest neighbor algorithm," in *Proceedings of the 3rd international conference on industrial application engineering*, vol. 2, 2015.

[19] C. Boukhatem, H. Y. Youssef, and A. B. Nassif, "Heart disease prediction using machine learning," in *2022 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–6, 2022.

[20] S. R. Jiao, J. Song, and B. Liu, "A review of decision tree classification algorithms for continuous variables," *Journal of Physics: Conference Series*, vol. 1651, p. 012083, Nov. 2020.

[21] N. Parashar, "What is an accuracy score and how to check it?," Jan 2023.